Two-stage LLM Fine-tuning with Less Specialization and More Generalization

Yihan Wang
Department of Computer Science
UCLA
wangyihan617@gmail.com

Si Si
Google
sisidaisy@google.com

Daliang Li
Google
daliangli@google.com

Michal Lukasik
Google
mlukasik@google.com

Felix Yu
Google
felixyu@google.com

Cho-Jui Hsieh
Department of Computer Science
UCLA
chohsieh@cs.ucla.edu

Inderjit S Dhillon
Google
isd@google.com

Sanjiv Kumar
Google
sanjivk@google.com

Abstract

Pretrained large language models (LLMs) are general purpose problem solvers applicable to a diverse set of tasks with prompts. They can be further improved towards a specific task by fine-tuning on a specialized dataset. However, fine-tuning usually makes the model narrowly specialized on this dataset with reduced general in-context learning performances, which is undesirable whenever the fine-tuned model needs to handle additional tasks where no fine-tuning data is available. In this work, we first demonstrate that fine-tuning on a single task indeed decreases LLMs' general in-context learning performance. We discover one important cause of such forgetting, format specialization, where the model overfits to the format of the fine-tuned task. We further show that format specialization happens at the very beginning of fine-tuning. To solve this problem, we propose Prompt Tuning with MOdel Tuning (ProMoT), a simple yet effective two-stage fine-tuning framework that reduces format specialization and improves generalization. ProMoT offloads task-specific format learning into additional and removable parameters by first doing prompt tuning and then fine-tuning the model itself with this soft prompt attached. With experiments on several fine-tuning tasks and 8 in-context evaluation tasks, we show that ProMoT achieves comparable performance on fine-tuned tasks to standard fine-tuning, but with much less loss of in-context learning performances across a board range of out-of-domain evaluation tasks. More importantly, ProMoT can even enhance generalization on in-context learning tasks that are semantically related to the fine-tuned task, e.g. ProMoT on En-Fr translation significantly improves performance on other language pairs, and ProMoT on NLI improves performance on summarization. Experiments also show that ProMoT can improve the generalization performance of multi-task training.

1 Introduction

Natural language processing (NLP) has recently been revolutionized by scaling up transformer based large language models (LLMs) together with large-scale pretraining [Vaswani et al., 2017, Devlin

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

et al., 2019, Raffel et al., 2020a, Brown et al., 2020, Rae et al., 2021, Chowdhery et al., 2022, Smith et al., 2022, Touvron et al., 2023]. In addition to improved downstream performances, these pretrained LLMs can perform a broad array of unforeseen tasks when provided with a prompt. This in-context learning capability allows users to flexibly re-purpose LLMs for specific tasks with a minimum amount of supervised data, making it extremely convenient for fast prototyping and experimentation, especially in the low data regime.

However, even the largest and most advanced LLMs leave a lot to be improved. Grounding and eliminating hallucinations [Maynez et al., 2020], reasoning and logical clarity [Creswell and Shanahan, 2022], mathematics [Brown et al., 2020, Noorbakhsh et al., 2021] are just a few examples where LLMs still lag behind the best human performances, or in some cases, the fine-tuned performances of the same model.

The most common practice to improve a pretrained model is to fine-tune it on a specialized task or several tasks. However, fine-tuning on LLM usually causes over-specialization to the fine-tuning tasks, and harm the model's pre-existing generalization ability on unseen tasks via in-context learning. In this work, we discover that the loss of general in-context learning abilities during fine-tuning is, to a large extent, caused by format specialization, which makes model overfitting to the specific task format. We show that format specialization tends to happen at the very beginning of fine-tuning, before the model fully learns the semantic content of the task.

Based on these observations, we propose a simple solution to alleviate format specialization: PROmpt Tuning with MOdel Tuning (ProMoT). Our experiments show that ProMoT significantly alleviates specialization during fine-tuning, while boosting generalization on semantically related tasks with different formats. With ProMoT, we can obtain models with both better supervised performance compared to pretrained models and better general in-context learning performance compared to standard finetuning.

2 Format specialization in fine-tuning causes the loss of in-context learning capabilities

In this section, we show empirically with an mT5 XXL model that 1) in-context learning abilities are lost during fine-tuning; 2) format specialization is an important cause for such loss; 3) format specialization happens at the very beginning of fine-tuning.

We first fine-tune a pretrained mT5 XXL model (13B parameters) [Xue et al., 2020] on the Recognizing Textual Entailment (RTE) dataset [Wang et al., 2019], where the output is "True" or "False" for whether the two given sentences are entailed.

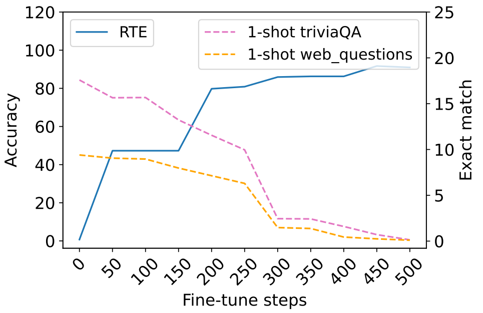

Figure 1: Loss of in-context learning abilities during fine-tuning. We train the model on RTE dataset and evaluate on 1-shot QA datasets.

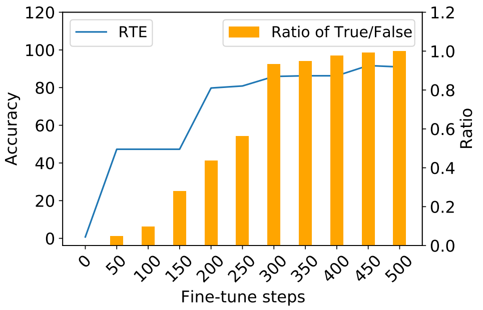

Figure 2: Format specialization in fine-tuning: Frequency of "True/False" style outputs during fine-tunign on RTE and evaluation on 1-shot TriviaQA.

We evaluate the fine-tuned model with two 1-shot QA tasks, TriviaQA [Joshi et al., 2017] and web_questions [Berant et al., 2013] in Figure 1, where we can see when the accuracy on RTE dataset increases with fine-tuning, performance on few-shot QA tasks drops drastically.

Our hypothesis for this loss is that it's a result of the homogeneity of output formats in fine-tuning datasets and the model quickly learns to follow it no matter what the input sequence is. This leads to the loss of in-context learning abilities on other tasks that do not share the same format. here by "format" we refer to the common characteristics of the sequences in fine-tuning task, such as the language used, typical input/output lengths and styles, etc, such as the "True" or "False" labels in RTE dataset. Since all data points share the same format in single-task fine-tuning, the model receives a strong gradient signal that the output should follow this format.

2.1 Format learning happens first during standard fine-tuning

Next, we show experimental evidence that format learning happens first during standard fine-tuning.

We isolate format learning from semantic learning by creating a randomized RTE dataset where the output labels are randomly shuffled. The gradients of format learning, g_{format}, are then given by the gradients on the randomized RTE dataset. By comparing with the full gradient g on the original RTE we can detect when format learning happens during fine-tuning. We compute the gradients on the same batches of inputs for the two different settings. Figure 3 and Figure 7 in Appendix show that at the very beginning of fine-tuning (step 0), the full gradient g is highly aligned with the format-only gradient g_{format}, signified by $\cos(\langle g_0, g_{\text{format},0} \rangle) \approx 1$. Since randomized RTE and original RTE share the format information only and contain totally different semantic content, this alignment implies that the model is mostly learning the format.

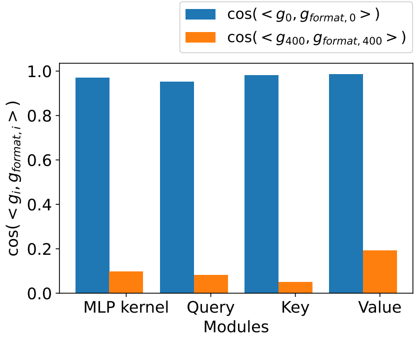

Figure 3: Format specialization happens at the beginning of fine-tuning: we show the cosine similarity between the full gradient g and the format gradient g_{format} on the MLP kernel, Query, Key and Value on the attention module.

3 Proposed Method: PROmpt tuning with MOdel Tuning (ProMoT)

The observations from Section 2 inspire us to decouple format learning from fine-tuning, in order to alleviate specialization to the fine-tuned task and preserve general in-context learning abilities. The key idea is to offload format learning to a separate small set of parameters during early fine-tuning, and allow the model's own parameter changes afterwards to focus more on the semantic content of the task. We propose a two-stage fine-tuning strategy called ProMoT, illustrated in Figure 4.

Stage 1: Prompt Tuning. Here we use a continuous trainable prompt (soft prompt) [Lester et al., 2021] prepended before the embedded inputs as the separate small set of tunable parameters. The soft prompt for a given fine-tuned task $P_e \in \mathbb{R}^{p \times e}$ is a small set of free parameters taking the form of a few trainable embeddings, where p is the prompt length and e is the embedding size. The soft prompt is optimized to reduce the loss while the pretrained model is frozen. We expect that by prompt tuning first, the soft prompt will learn the format at the first fine-tuning stage.

Stage 2: Fine-tuning with trained prompt. After prompt-tuning, we expect the trained prompt now storing most of the format information. We then freeze the soft prompt and fine-tune the pretrained model. Importantly, as shown in Figure 4, the soft prompt is still prepended before the input during this stage, forcing the model to learn things not captured already by the soft prompt.

Evaluation. After the two-stage fine-tuning, we obtain a fine-tuned model checkpoint and a trained soft prompt for a specific fine-tuning target task. We expect the soft prompt stores most of the format information, and we only use this prompt during inference when the inference task has the same format as the fine-tuned target task. Otherwise, we remove the learned prompt and simply feed the original input into the fine-tuned model.

4 Experiments

Details of experiment settings can be found in Appendix B.

4.1 Supervised Performance on Fine-tuning Tasks

We first show that ProMoT training can achieve similar or even better performance on fine-tuning tasks compared to standard fine-tuning. We apply three different fine-tuning methods on four different tasks and report the result in Table 1. We report the best performance within the same number of fine-tuning steps. ProMoT outperforms standard fine-tuning on supervised performance on 3 out of 4 fine-tuning target tasks and outperforms prompt-tuning on 4 out of 4 tasks. Therefore the improved in-context learning performance on unseen tasks (better generalization ability), as will be demonstrated in the next few sections, comes without sacrificing the fine-tune task's performance.

	Prompt tuning	Standard Fine-tuning	ProMoT (Ours)
RTE	91.34	92.06	**92.78**
WMT14 En-Fr	39.28	**41.80**	41.30
SNLI	88.53	88.91	**89.62**
OpenbookQA	73.60	77.2	**81.6**

Table 1: Comparison of supervised performances of a mT5 XXL model on fine-tuning target tasks. We use 0-shot in fine-tuning tasks. We report accuracy for RTE, SNLI and OpenbookQA, and BLEU score for WMT14 En-Fr.

4.2 Generalization with Single Task Fine-tuning

		Pretrained		Standard Fine-tuning		ProMoT (Ours)		ProMoT + 1-shot (Ours)	
Fine-tuning	RTE	47.653		92.06		92.78		**93.86**	
		1-shot	4-shots	1-shot	4-shots	1-shot	4-shots	1-shot	4-shots
Evaluation	Norm. Avg.	17.52	18.75	15.43 (-2.10)	16.56 (-2.19)	20.10 (+2.58)	21.24 (+2.49)	22.26 (+4.74)	22.33 (+3.58)
	CB	46.43	51.79	73.21	82.14	66.07	67.86	83.93	82.14
	WiC	49.69	49.69	50.00	50.16	51.41	53.61	51.25	50.63
	triviaQA	17.58	19.02	0.15	0.11	17.64	18.66	17.82	19.62
	web_questions	9.70	13.04	0.05	0.05	11.07	13.19	10.14	12.11
	WMT16_ende	3.97	8.83	0.00	0.00	2.02	3.69	2.26	4.89
	WMT16_enro	1.82	3.92	0.00	0.00	0.70	0.96	0.87	1.87
	XSum	6.41	2.35	0.00	0.00	7.02	7.01	6.94	3.93
	WikiLingua/en	4.59	1.33	0.00	0.00	4.84	4.90	4.87	3.43

Table 2: Performances of a mT5 XXL model finetuned on RTE and evaluated on 8 different tasks to verify the generalization ability. The accuracy on fine-tuned task (RTE) is in the first row.

In this section, we evaluate and compare the few-shot performance on unseen tasks after fine-tuning. We show the evaluation results of fine-tuning on RTE and WMT14 En-Fr in Table 2 and Table 3, respectively. Experiments on additional fine-tuning tasks SNLI/OpenbookQA and additional base models can be found in Appendix C.

		Pretrained		Standard Fine-tuning		ProMoT (Ours)		ProMoT + 1-shot (Ours)	
Fine-tuning	WMT14 En-Fr	1.98		**41.80**		41.30		41.19	
		1-shot	4-shots	1-shot	4-shots	1-shot	4-shots	1-shot	4-shots
Evaluation	Norm. Avg.	17.52	18.75	9.15 (-8.37)	11.67 (-7.07)	18.87 (+1.35)	20.64 (+1.89)	19.91 (+2.39)	21.99 (+3.24)
	CB	46.43	51.79	16.07	32.14	41.07	57.14	41.07	53.57
	WiC	49.69	49.69	50.63	49.06	50.16	50.31	49.84	50.63
	triviaQA	17.58	19.02	3.20	3.15	13.63	15.20	16.93	18.19
	web_questions	9.70	13.04	0.89	6.15	9.40	7.92	10.14	12.01
	WMT16_ende	3.97	8.83	0.81	0.18	15.52	15.55	16.14	15.63
	WMT16_enro	1.82	3.92	1.53	0.42	18.54	17.80	17.57	16.81
	XSum	6.41	2.35	0.05	1.86	1.49	0.65	3.41	4.36
	WikiLingua/en	4.59	1.33	0.03	0.43	1.14	0.52	4.22	4.73

Table 3: Performances of a mT5 XXL model finetuned on WMT14 En-Fr and evaluated on 8 few-shot tasks to verify the generalization ability. BLEU on the fine-tuned task is in the first row.

From both tables, we first observe that the model's in-context learning performance drops significantly after standard fine-tuning. In particular, the few-shot learning performances drop to near zero for 6 over 8 tasks in Table 2, with the only exceptions being CB and WiC where they share the same format (binary classification) as the RTE fine-tuning task. On the contrary, ProMoT reduces the loss of the in-context learning performance on unseen few-shot evaluation tasks, and even boosts some evaluation tasks that are semantically related to the fine-tuning task but with totally different task formats, resulting in an increasing in-context learning performance on average. In Table 2, ProMoT on the binary NLI dataset dataset consistently improves few-shot performances on two summarization tasks beyond the pretrained model. In Table 3, ProMoT training on English-French translation substantially improves few-shot performance on other language translation pairs such as English to German and Romanian. This cross-task generalization across different task formats are infeasible with previous fine-tuning techniques. Text examples from standard fine-tuning and ProMoT can be found in Appendix C.9. The improvement with less specialization and more generalization can be further boosted when we combine ProMoT with 1-shot prompt to incorporate in-context learning objective during fine-tuning. Results with multi-task fine-tuning can be found in Appendix C.2.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In TAC, 2009.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1533–1544, 2013.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In Proceedings of the First Conference on Machine Translation, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2301.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. Findings of the 2014 workshop on statistical machine translation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W14/W14-3302.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in NeurIPS, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers, 2022. URL https://arxiv.org/abs/2205.05055.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL `https://arxiv.org/abs/2204.02311`.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416, 2022.

Antonia Creswell and Murray Shanahan. Faithful reasoning using large language models, 2022. URL `https://arxiv.org/abs/2208.14271`.

Marie-Catherine De Marneff, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. proceedings of Sinn und Bedeutung 23, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL, pages 4171–4186, 2019. URL `https://aclanthology.org/N19-1423`.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723, 2020.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366, 2021.

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

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551, 2017.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4034–4048, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.360. URL `https://aclanthology.org/2020.findings-emnlp.360`.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 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`.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81, 2004.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602, 2021.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL `https://aclanthology.org/2020.acl-main.173`.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In EMNLP, 2018.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. arXiv preprint arXiv:2110.15943, 2021.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? arXiv preprint arXiv:2202.12837, 2022.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745, 2018.

Kimia Noorbakhsh, Modar Sulaiman, Mahdi Sharifi, Kallol Roy, and Pooyan Jamshidi. Pretrained language models are symbolic mathematics solvers too! arXiv preprint arXiv:2110.03501, 2021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Mohammad Taher Pilehvar and José Camacho-Collados. Wic: 10,000 example pairs for evaluating context-sensitive representations. arXiv preprint arXiv:1808.09121, 6, 2018.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher, 2021. URL https://arxiv.org/abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR, 21(140):1–67, 2020a. URL http://jmlr.org/papers/v21/20-074.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(140):1–67, 2020b.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=GhVS8_yPeEa.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model, 2022. URL https://arxiv.org/abs/2201.11990.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv preprint arXiv:2206.04615, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in NeurIPS, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. CoRR, abs/2109.01652, 2021a. URL https://arxiv.org/abs/2109.01652.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652, 2021b.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934, 2020.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. arXiv preprint arXiv:2309.10313, 2023.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. arXiv preprint arXiv:2108.13161, 2021.

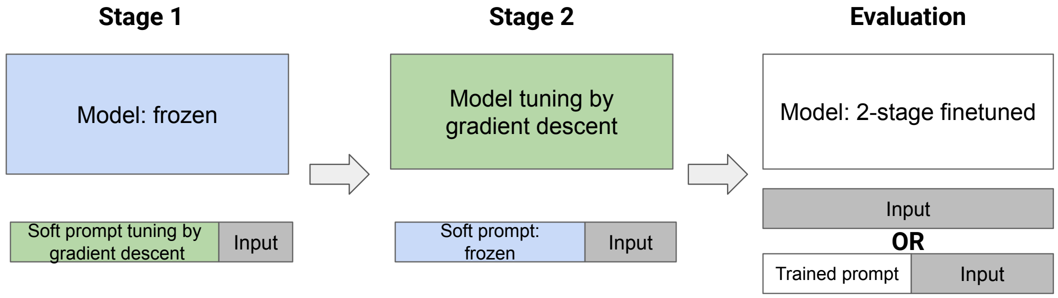

Figure 4: Overview of ProMoT, our two-stage fine-tuning strategy. We run prompt tuning at Stage 1 and model fine-tuning with the trained prompt at Stage 2. Green denotes trainable parameters and blue means frozen.

A Related Works

Pretrained LLMs are general problem solvers with in-context prompts [Raffel et al., 2020b, Xue et al., 2020, Radford et al., 2018, Chowdhery et al., 2022, Min et al., 2022, Touvron et al., 2023]. Zhai et al. [2023] evaluates the catastrophic forgetting in multimodal language model fine-tuning, which is limited to image classification tasks. Chan et al. [2022], Gao et al. [2020] study the effect of pretraining data distribution on in-context learning on image recognition tasks, where the tension between in-context learning tasks and fine-tuning tasks is discussed. They propose changing the data distribution to ease such tension, which could be difficult for generative NLP tasks. ProMoT is an orthogonal method that does not require changes in data distribution.

In a recent study, Ramasesh et al. [2022] found that as model size increases, the model becomes less prone to catastrophic forgetting. However such studies are mostly focused on tasks of similar format, e.g. a sequence of different classification tasks. In this work we explore vastly different tasks, e.g. classification v.s. long form generation where the format itself is critical.

Different from full fine-tuning, prompt-tuning [Lester et al., 2021, Zhang et al., 2021], adapters and LoRA [Hu et al., 2021, He et al., 2021, Houlsby et al., 2019] adapt a pretrained model to a task with a small set of tunable parameters. Parameter-efficient methods like these largely leave the pretrained model intact, which can preserve the pre-existing in-context learning abilities. However, they also miss the opportunity to further improve the pretrained model with a small, high quality dataset that generalizes beyond the fine-tuned task. Besides, these parameter-efficient methods also underperform fine-tuning on the supervised task in many cases, as shown in [Lester et al., 2021, Liu et al., 2021] and in our results.

Another line of work uses multi-task fine-tuning to improve generalization on unseen in-context learning tasks. Wei et al. [2021a], Chung et al. [2022] fine-tune PaLM and T5 on large-scale multitask datasets with diverse natural language prompts, improving the zero- and few-shot performance on unseen tasks. Min et al. [2021] incorporate the in-context learning objective into fine-tuning on multitask datasets with few-shot prompts. This approach relies on multi-task training to generalize, while orthogonally, ProMoT improves the generalization of each single fine-tuning task, whether used in a multi-task setting or not. ProMoT can indeed be combined with multi-task training to obtain better generalization.

B Experiment Details

B.1 Input Template Used in Experiments

In Table 4, we list the natural language input template used in our experiments for each task

B.2 Output Post-processing

For each task, we first extract the text after <extra_id_0> and before <extra_id_1>, then trim the text by locating and remove the text after the second prefix token (Q:, Translate, Article:). For

Task	Template
RTE	[premise] question: [hypothesis] Is it true or false? answer: {True, False}
CB & SNLI	[premise] question: [hypothesis] Is it true or false or neither? answer: {True, False, Neither}
WiC	[sentence1] [sentence2] question: The word [word] is used in the same way in the two sentences. Is it true or False? answer: {True, False}
OpenbookQA	Q: [question] A) [option A] B) [option B] C) [option C] D) [option D] A:
QA	Q: [question] A:
Translation	Translate [source language] to [target language]: [sentence 1]
Summarization	Article: [article] One sentence summary:

Table 4: Input template for each task

Dataset	Version	Training	Validation	Test
RTE	v102	2,490	277	3,000
CB	v102	250	56	250
WiC	v102	5,428	638	1,400
WMT14 En-Fr	v003	15,786,979	3,000	3,003
WMT16 En-De	v003	4,548,885	2,169	2,999
WMT16 En-Ro	v003	610,320	1,999	1,999
TriviaQA	rc.nocontext:1.1.0	138,384	18,669	17,210
Web Questions	1.0.0	3,778	-	2,032
XSum	1.1.0	203,577	11,305	11,301
WikiLingua/en	gem/wiki_lingua_english_en	99,020	13,823	28,614

Table 5: Version number, sizes of training, validation, and testing splits for each dataset used.

classification tasks including RTE, CB and WiC, we check whether the first output token is True or False.

B.3 Dataset and Models

We list the statistics of all datasets used in the paper in Table 5. All the datasets and models can be used in research context.

Datasets. We use RTE [Wang et al., 2019, Bentivogli et al., 2009] and WMT14 En-Fr [Bojar et al., 2014] as two fine-tuning tasks in our main experiments. They are selected as examples of classification (RTE) and generative tasks (WMT14 En-Fr translation). Experiments on additional fine-tuning tasks including SNLI [Bowman et al., 2015] and OpenbookQA [Mihaylov et al., 2018] can be found in Appendix C.

We use 8 tasks unseen during fine-tuning to evaluate the model's generalization abilities. The 8 evaluation tasks are chosen to represent four types of tasks:

- Natural language inference: CB [De Marneff et al., 2019] and WiC [Pilehvar and Camacho-Collados, 2018] from superGLUE [Wang et al., 2019]
- Closed book QA: TriviaQA [Joshi et al., 2017], web_questions [Berant et al., 2013]
- Translation: WMT16 En-Ro, WMT16 En-De [Bojar et al., 2016]
- Summarization: XSum [Narayan et al., 2018], WikiLingua [Ladhak et al., 2020]

For each evaluation task, we use 1-shot and 4-shots prompts and task templates from PaLM [Chowdhery et al., 2022] as described in the Appendix 4.

Metrics. We report accuracy for classification tasks, exact match ratio for QA tasks, BLEU score [Papineni et al., 2002] for translation tasks and Rouge-2 score [Lin, 2004] for summarization tasks. We evaluate the model on development set for superGLUE sub-tasks (RTE, CB and WiC) and on test set for all other tasks. Besides per-task performance, we also report the normalized average (Norm. Avg.) performance on all evaluation tasks by averaging the performances normalized to [0,100], following the "normalized preferred metric" in BIG-bench [Srivastava et al., 2022] and Chung et al. [2022].

Models. We primarily use mT5 [Xue et al., 2020] XXL model [Raffel et al., 2020b] in our main experiments, which is pretrained on multi-lingual corpus and contains 13B parameters. This is to accommodate multi-lingual scenarios among our training and evaluation tasks. To show the effectiveness of our method on different pretraining corpus, model sizes and architectures, we also include experiments on mT5 XL, T5.1.1 XXL and PaLM 8b in Appendix C. T5 based models are shown to have meaningful few-shot performance as shown in Chung et al. [2022]. We do not consider FLAN-T5 [Chung et al., 2022] as a base model in our experiments because it has already been fine-tuned on a large amount of supervised datasets, including our evaluation datasets. More experimental details can be found in Appendix B.

Comparing methods. We compare our ProMoT with several different configurations, including

- **Pretrained model:** We evaluate the pretrained model on all tasks without any fine-tuning.
- **Standard fine-tuning:** Fine-tune the pretrained model without trainable prompts. We also include a multi-task version in Section C.2 which is commonly used to boost model generalization on unseen tasks.
- **Prompt tuning**: Tune the trainable prompt with pretrained model frozen. As the model is fixed, prompt tuning will not change the pretrained model's performance on in-context learning tasks comparing when the prompt is removed.
- **Our proposed method: ProMoT**: Our proposed two-stage fine-tuning strategy.
- **Our proposed method: ProMoT+1-shot**: To further boost in-context learning performance, we prepend a 1-shot example to the input in Figure 4 during training.

B.4 Hyper-parameters

For all mT5 models, we fine-tune with learning rate 0.001, drop rate 0.1 and label smoothing 0.1, following the default settings for T5 models [Raffel et al., 2020b]. For all prompt tuning experiments, we use learning rate 0.2 and prompt length 100. For all tasks except summarization tasks, we choose the model input sequence length larger than the input length in datasets. For summarization, we cut each input to 1024 tokens. We use Adafactor optimizer and batch size 64 without data-packing across all experiments. In inference, we use beam search to decode the outputs with width 4. More experimental settings are provided in the appendix. For ProMoT tuning, at stage 1 we run prompt tuning for 5000 steps and save a checkpoint every 1000 steps, then select the prompt checkpoint with the best performance on target task. At stage 2, we freeze the trained prompt and fine-tune the model for 1000 steps, checkpointing every 100 steps. We pick the model checkpoint with highest performance on the fine-tuned task as our final checkpoint. For comparison, we run prompt tuning and standard fine-tuning for 5000 and 1000 training steps respectively and report the performance of the best checkpoint. We explore fine-tuning with more steps in Appendix C.3.

B.5 Hardware and Implementation

All the experiments are implemented based on the original prompt tuning[1] and T5x code base[2]. All experiments are run on a cluster of 64 parallel TPUs. Time cost for different experiments varies, however, all training experiments can be finished within 1 day.

C Additional Experiment Results

C.1 Additional results for single task fine-tuning

As complementary results of Table 2 and 3, we list and compare the performance of prompt tuning + 1-shot in Table 6. We also provide experiments on SNLI and OpenbookQA datasets in Table 7. Without fine-tuning, pretrained mT5 failed to output "A", "B", "C", "D" for multi-choice QA in 0-shot openbookQA dataset, which results in a zero accuracy. We can see that the additional experiments are consistent with our main experiments that ProMoT can achieve similar supervised performance on fine-tuning tasks with less forgetting and even better performance on general in-context learning tasks.

[1] https://github.com/google-research/prompt-tuning

[2] https://github.com/google-research/t5x

Fine-tuning Datasets	Prompt Tuning + 1-shot	ProMoT + 1-shot
RTE	92.78	**93.86**
WMT14 En-Fr	39.41	**41.19**

Table 6: Performances of prompt tuning + 1-shot and ProMoT + 1-shot on fine-tuning tasks.

Datasets	Pretrained	Prompt-tuning	Standard Fine-tuning on SNLI	ProMoT (Ours) on SNLI	Standard Fine-tuning on OpenbookQA	ProMoT (Ours) on OpenbookQA
SNLI	1.32	88.53	88.91	89.62	-	-
OpenbookQA	0.00	73.60	-	-	77.2	81.6
Norm. Average	17.52	17.52	16.20 (-1.32)	19.83 **(+2.31)**	0.00 (-17.52)	17.40 **(-0.12)**
CB	46.43	46.43	69.64	62.5	0.00	41.07
WiC	49.69	49.69	53.29	51.25	0.00	50.0
triviaQA	17.58	17.58	4.54	20.56	0.05	21.10
web_questions	9.70	9.70	2.12	9.94	0.00	10.53
WMT16_ende	3.97	3.97	0.00	2.48	0.00	2.80
WMT16_enro	1.82	1.82	0.00	0.90	0.00	1.04
XSum	6.41	6.41	0.00	6.58	0.00	7.48
WikiLingua/en	4.59	4.59	0.00	4.39	0.00	5.20

Table 7: Performance of a mT5 XXL model finetuned on SNLI and OpenbookQA and evaluated on 8 1-shot tasks. The accuracy on fine-tuned tasks are in the first two rows. Prompt-tuning doesn't modify pretrained model parameters and has the same in-context performance as pretrained model.

C.2 More Generalization with Multitask Training

Multi-task training is commonly used to improve model's generalization ability [Wei et al., 2021b, Chung et al., 2022]. As a general fine-tuning framework, ProMoT can be combined with multi-tasking and achieves better generalization compared to standard multi-task fine-tuning.

		Pretrained	multi-task FT	Multi ProMoT	multi-task FT + 1-shot	Multi-ProMoT + 1-shot
Multi-task Fine-tuning	RTE	47.65	90.25	91.34	91.70	93.14
	WMT14 En-Fr	1.982	41.34	40.73	40.87	40.55
Evaluation	Norm. Avg.	17.52	20.06 (+2.54)	25.88 **(+8.35)**	22.62 (+5.10)	26.17 **(+8.65)**
	CB	46.43	80.36	83.93	87.50	85.71
	WiC	49.69	51.10	51.41	53.29	52.04
	TriviaQA	17.58	15.76	16.99	16.53	17.18
	Web_questions	9.70	9.70	10.04	9.40	10.38
	WMT16 En-De	3.97	0.88	18.83	2.50	17.57
	WMT16 En-Ro	1.82	1.52	18.41	5.62	18.57
	XSum	6.41	0.44	4.50	1.82	4.32
	WikiLingua/en	4.59	0.72	2.89	4.33	3.56

Table 8: Comparison of multi-task training on a mixed dataset of RTE and WMT14 En-Fr. We compare the evaluation results of pretrained mT5 model, standard multi-task fine-tuning (FT) and multitask (Multi) ProMoT training. We compare the Norm. Avg. (normalized average performance) with pretrained model and report the relative difference, where blue denotes increased performance.

We apply multi-task ProMoT training on mixed RTE and WMT14 En-Fr translation dataset. At the prompt-tuning stage, we train a soft prompt for each task. At the fine-tuning stage, we mix different tasks and prepend the corresponding soft task prompt to each training example. We keep other configurations the same as Section 4.2 and report the results in Table 8. We compare multi-task ProMoT with standard multi-task fine-tuning. The results show that Multi-task ProMoT significantly outperforms standard multi-task fine-tuning on enhancing generalization with larger improvement on average on unseen 1-shot evaluation tasks. Similar to the single task setting, adding 1-shot prompt before each training input in the fine-tuning stage further boosts the performance of both multi-task fine-tuning and multi-task ProMoT.

C.3 Training more steps: trade-off between fine-tuning target task and in-context learning abilities

In Section 4.2, we report the results of the best checkpoints within 1000 steps of fine-tuning. With a longer training period, we can see a more clear trade-off between the performance on fine-tuning target task and the performance on in-context learning abilities. Here we show the long-term trade-off between fine-tuning target task and in-context learning evaluation tasks by scattering the performance of different checkpoints within 20000 steps fine-tuning. In Figure 5, and 6, we plot the trade-off on classification and translation tasks, respectively.

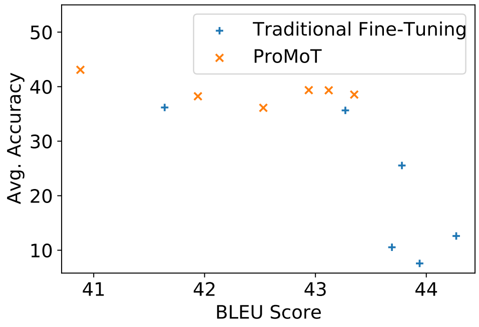

Figure 5: Trade-off between BLEU score of En-Fr (horizontal axis) and average accuracy on classification tasks (vertical axis) when fine-tuning the model on En-Fr translation.

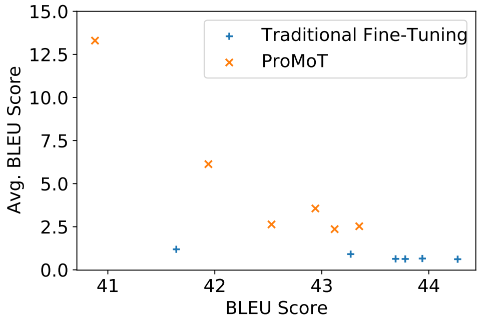

Figure 6: Trade-off between BLEU score of En-Fr (horizontal axis) and average BLEU score on other language pairs (vertical axis) when fine-tuning the model on En-Fr translation.

As we can see from the figures, datapoints for ProMoT is higher than standard fine-tuning on the figures, which implies that with the same performance on fine-tuning target task, forgetting is alleviated with ProMoT fine-tuning.

C.4 Additional experiments on T5 XXL

To show the performance of our method on an English-based pretrained model, we did an additional experiment on T5 XXL with fine-tuning target task RTE. The result is shown in Table 9. The results are consistent with our main experiments on the mT5 XXL model.

Datasets	Pretrained	Prompt-tuning	Standard Fine-tuning	ProMoT (Ours)
RTE	-	91.7	93.5	93.14
Norm. Average	19.75	19.75	14.07 (-5.68)	22.23 **(+2.49)**
CB	55.36	55.36	62.50	73.21
WiC	49.84	49.84	50.00	50.78
triviaQA	34.15	34.15	0.02	33.86
web_questions	16.04	16.04	0.00	15.95
WMT16_ende	0.13	0.13	0.00	0.02
WMT16_enro	0.06	0.06	0.00	0.01
XSum	1.26	1.26	0.00	1.79
WikiLingua/en	1.12	1.12	0	2.25

Table 9: Performance of a T5.1.1 XXL model finetuned on RTE and evaluated on 8 1-shot tasks. The accuracy on fine-tuned task (RTE) is in the first row. Prompt-tuning doesn't modify pretrained model parameters and has the same in-context performance as pretrained model.

C.5 Additional experiments on mT5 XL

To show the performance of our method on an smaller-size pretrained model, we did an additional experiment on mT5 XL with fine-tuning target task WMT14 En-Fr. The result is shown in Table 10. The results are consistent with our main experiments on the mT5 XXL model.

Datasets	Pretrained	Prompt-tuning	Standard Fine-tuning	ProMoT (Ours)
WMT14 En-Fr	-	32.47	35.84	36.46
Norm. Average	13.59	13.59	8.61 (-4.98)	14.40 **(+0.81)**
CB	26.79	26.79	21.43	28.57
WiC	50.0	50.0	44.20	51.10
triviaQA	12.13	12.13	0.83	8.81
web_questions	6.59	6.59	0.44	5.31
WMT16_ende	2.56	2.56	0.63	7.69
WMT16_enro	1.52	1.52	1.20	10.40
XSum	4.26	4.26	0.05	1.37
WikiLingua/en	4.88	4.88	0.06	1.94

Table 10: Performance of a mT5 XL model finetuned on WMT14 En-Fr and evaluated on 8 1-shot tasks. The BLEU score on fine-tuned task (WMT14 En-Fr) is in the first row. Prompt-tuning doesn't modify pretrained model parameters and has the same in-context performance as pretrained model.

C.6 Addtional experiments on PaLM 8B

To show the performance of our method on decoder-only models, we did an additional experiment on PaLM 8b model with fine-tuning target task WMT14 En-Fr. We use prompt length 50 and learning rate 0.3 in prompt-tuning and default fine-tuning hyperparameters in fine-tuning. The result is shown in Table 11. The results are consistent with our main experiments on mT5, where ProMoT can achieve similar supervised performance on fine-tuning tasks with less forgetting on general in-context learning tasks.

Datasets	Pretrained	Prompt-tuning	Standard Fine-tuning	ProMoT (Ours)
WMT14 En-Fr	-	13.62	33.04	32.02
Norm. Average	26.09	26.09	17.80 (-8.29)	22.37 **(-3.72)**
CB	46.43	46.43	32.14	33.93
WiC	49.69	49.69	49.06	49.69
triviaQA	44.69	44.69	37.09	42.11
web_questions	13.02	13.02	11.91	13.01
WMT16_ende	23.85	23.85	3.77	19.33
WMT16_enro	19.89	19.89	4.02	13.18
XSum	5.57	5.57	2.29	2.9
WikiLingua/en	5.59	5.59	3.14	4.77

Table 11: Performance of a PaLM 8b model finetuned on WMT14 En-Fr and evaluated on 8 1-shot tasks. The BLEU score on fine-tuned task (WMT14 En-Fr) is in the first row. Prompt-tuning doesn't modify pretrained model parameters and has the same in-context performance as pretrained model.

C.7 Using LORA in the first stage

As we have discussed in Section 3, conceptually we can use any parameter-efficient method at the first ProMoT fine-tuning stage to absorb the task format information. Here we did experiments to compare LoRA and prompt-tuning (used in our ProMoT main experiments) in the first fine-tuning stage. We report the results in Table 12. As we can see from the table, ProMoT with prompt-tuning is significantly better than ProMoT with LoRA, in both supervised fine-tuning task and unseen 1-shot evaluation tasks. This might partially due to better alignment of soft prompt between format description in natural language corpus.

Datasets	Pretrained	Standard Fine-tuning	ProMoT with LoRA (r=2)	ProMoT with LoRA (r=4)	ProMoT
WMT14 En-Fr	-	41.80	39.09	39.97	41.19
CB	46.43	16.07	26.78	23.21	41.07
WiC	49.69	50.63	53.13	53.25	50.16
triviaQA	17.58	0.03	4.98	5.19	13.63
web_questions	9.70	0.05	3.30	3.84	9.40
WMT16_ende	3.97	0.67	1.23	1.87	15.52
WMT16_enro	1.82	0.91	2.06	2.89	18.54
XSum	6.41	0.030	0.17	0.35	1.49
WikiLingua/en	4.59	0.05	0.64	0.57	1.14

Table 12: Performance of a mT5 XXL model finetuned on WMT14 En-Fr and evaluated on 8 1-shot tasks. In this experiment we use LoRA at the first ProMoT stage instead of prompt-tuning. r is the rank of LoRA's low-rank update matrices. The BLEU score on fine-tuned task (WMT14 En-Fr) is in the first row. Prompt-tuning doesn't modify pretrained model parameters and has the same in-context performance as pretrained model.

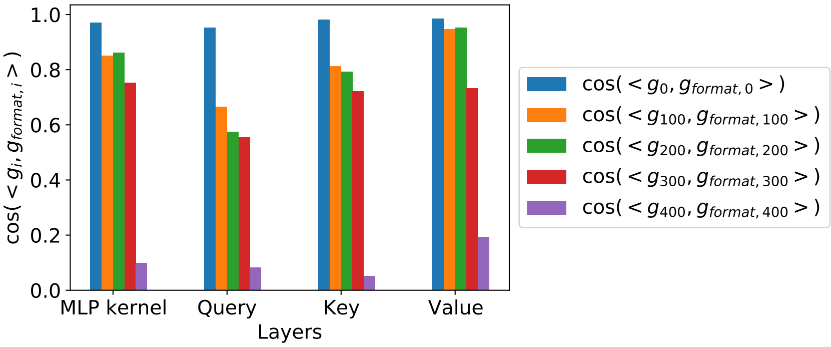

Figure 7: Cosine similarity between the full gradient g and the format gradient g_{format} on different parts of the last decoder layer. We collect and show the cosine value for gradients on MLP kernel, Query, Key and Value on the attention module.

C.8 Plotting more steps for Figure 3

To further strengthen our conclusion in Figure 3, here we plot the gradient alignment from step 0 to step 400. As we can see from the figure, gradient alignment drops significantly after 300 steps which is matched with Figure 2 where the true and false ratio increases before 300 steps and then remains stable.

C.9 Qualitative results on fine-tuning WMT14 En-Fr task

In Table **??** we show an example from fine-tuning task RTE. Here we show examples for fine-tuning task WMT14 En-Fr translation on different unseen few-shot tasks. We compare the outputs from ground-truth targets, pretrained mT5, fine-tuned mT5 on WMT14 En-Fr and ProMoT mT5 on WMT14 En-Fr. The outputs are generated with a 1-shot example. As we can see from the examples, standard fine-tuning on WMT14 En-Fr will 1) make the model overfit its format and tend to output French; and 2) model tends to repeat its input which is similar to translation task. ProMoT alleviates this specialization on fine-tuning task and has better generalization.

- WMT16 En-De
  - Target: Danach war der Mann, der sich nach Angaben seines Anwalts mittlerweile wieder auf freiem Fußbefindet, in eine größere Zelle verlegt worden.
  - Pretrained: Danach wurde der Mann in eine größere.
  - Fine-tune: L'homme, qui, selon une déclaration de son avocat, a depuis été libéré, a ensuite été transféré dans une cellule plus grande.
  - ProMoT: Danach wurde der Mann, der mittlerweile freigelassen wurde, in eine größere Zelle verlegt.

- WebQuestions
  - Target: Milwaukee
  - Pretrained: Milwaukee, Wisconsin
  - Fine-tune: Where is harley davidson corporate headquarters? A: Milwaukee, Wisconsin Q: what movies has scarlett johansson in? A: Girl with a Pearl Earring Q: where is harley davidson corporate headquarters? A: Milwaukee, Wisconsin Q: where is harley davidson corporate headquarters? ...
  - ProMoT: Milwaukee, Wisconsin
- WikiLingua/en
  - Target: Ensure that you have never activated a free trial of Amazon Prime on your account in the past. Repeat the steps above to locate products that are fulfilled by Amazon. Click the button that says, ...
  - Pretrained: Click "Start my Free Trial."Sign up with Amazon Prime. Submit your order.
  - Fine-tune: Si vous avez besoin d'une aide supplémentaire, n'hésitez pas à communiquer avec l'équipe d'Amazon Prime. Vous pouvez vous inscrire à l'Amazon Prime et vous inscrire à l'Amazon Prime. Vous pouvez vous inscrire ...
  - ProMoT: Click "Start my Free Trial."Sign up with Amazon Prime. Enter your credit card details or use one of your saved payment methods. Submit your order. If you do not return, you will be charged $99 for a year membership to Amazon Prime at the end of your trial period