POST ✉: A Framework for Privacy of Soft-prompt Transfer

Xun Wang [1] **Jing Xu** [1] **Franziska Boenisch** [1] **Michael Backes** [1] **Adam Dziedzic** [1]

Abstract

Prompting has emerged as a dominant learning paradigm for adapting large language models (LLMs). While discrete (textual) prompts prepend tokens to the input for optimized outputs, soft (parameter) prompts are tuned in the embedding space via backpropagation, requiring less engineering effort. However, unlike semantically meaningful discrete prompts, soft prompts are tightly coupled to the LLM they were tuned on, hindering their generalization to other LLMs. This limitation is particularly problematic when efficiency and privacy are concerns, since (1) it requires tuning new prompts for each LLM which, due to the backpropagation, becomes increasingly computationally expensive as LLMs grow in size, and (2) when the LLM is centrally hosted, it requires sharing private data for soft prompt tuning with the LLM provider. To address these concerns, we propose a framework for Privacy Of Soft-prompt Transfer (POST), a novel method that enables private soft prompt tuning on a small language model and then transfers the prompt to the large LLM. Using knowledge distillation, we first derive the small language model directly from the LLM to facilitate prompt transferability. Then, we tune the soft prompt locally, if required with privacy guarantees, *e.g.,* according to differential privacy. Finally, we use a small set of public data to transfer the prompt from the small model to the large LLM without additional privacy leakage. Our experimental results demonstrate that our method effectively transfers soft prompts while protecting local data privacy and reducing the computational complexity over soft prompt tuning on the large model.

[1]CISPA Helmholtz Center for Information Security, Germany. Correspondence to: Adam Dziedzic <adam.dziedzic@cispa.de>.

Published at ICML 2024 Workshop on the Next Generation of AI Safety. Copyright 2024 by the author(s).

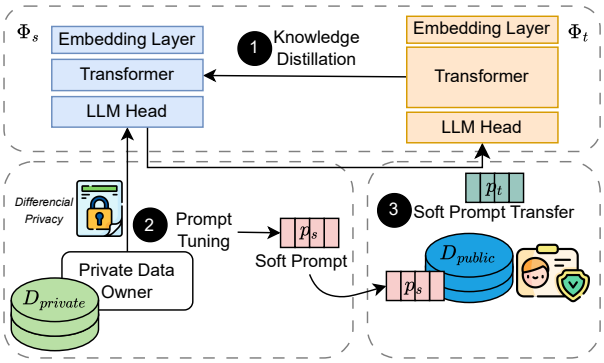


Figure 1. **POST ✉ Framework.** ❶ An LLM provider compresses Φ_t into a smaller LLM Φ_s by using knowledge distillation. ❷ The private data owner learns a specific soft prompt p_s on the Φ_s using their private dataset (optionally with differential privacy guarantees). ❸ The LLM provider obtains the soft prompt p_t for solving the user's task by transferring p_s to the target LLM Φ_t—solely relying on a small public dataset and no access to the private data for transfer.

1. Introduction

Large Language Models (LLMs) are strong general-purpose language generators that can be adapted to solve various private downstream tasks [4; 20]. One prominent paradigm for adapting LLMs to private tasks is prompting [4; 20]. *Soft prompts*, which prepend trainable vectors to the input and can be tuned automatically on the private data using gradient-based approaches, are generally known to yield higher performance at lower computational costs [16].

Yet, soft prompt tuning has two major limitations. 1) As LLMs grow in size [10; 2; 1], it becomes significantly more expensive in terms of compute. 2) At the same time, when the LLM is centrally hosted, it requires users to share their private data with the LLM provider, which causes privacy leakage. An alternative solution to address the privacy concern would be for the LLM provider to share their LLM with the user. However, this would put the intellectual property of the LLM provider at risk. From the user's perspective, it would also be impractical as they lack the computational resources to deploy and backpropagate through large models.

A potential solution to both problems is to tune the soft prompt locally on a smaller model and then transfer it to the

large LLM, which is commonly known as "prompt transfer" [26; 29; 30]. However, soft prompts are highly coupled to the LLM they were tuned on, making them difficult to transfer. Existing approaches for transferring soft prompts between two LLMs either require both the local small and the central large model to access the private data [26], leading to privacy leakage, or are ineffective, as the transferred prompt's utility on the large central LLM often underperforms compared to the prompted small model [29], disincentivizing the use of the large model altogether.

To address these challenges, we propose **POST**, a framework for **P**rivacy **O**f **S**oft-prompt **T**ransfer. POST consists of three key steps. (1) First, the LLM provider performs a *knowledge distillation* [12] to compress their large LLM into a smaller model. (2) Next, the user performs *local prompt tuning* using their private data on this smaller model, potentially incorporating formal privacy guarantees through differential privacy [8]. The user then provides this prompt to the LLM provider, who finally (3) *transfers the prompt* to achieve strong performance on the large LLM. To prevent any additional privacy leakage from the user's private data, we equip POST with a novel prompt transfer method that relies purely on access to a small public dataset rather than the user's private data for transfer. We provide an overview of POST in Figure 1.

Our thorough experimental evaluation on both masked language models and auto-regressive language models demonstrates that our method can efficiently and privately transfer soft prompts at high utility. In summary, we make the following contributions:

- We propose POST, a framework for privacy of soft prompt transfer. POST preserves the confidentiality of the users' private data and can be additionally equipped with strong privacy guarantees according to differential privacy.
- We design a novel method to transfer private prompts between LLMs by purely relying on public data which we integrate into POST.
- We provide detailed experimental analysis on four datasets and two different types of LLMs to show the effectiveness and efficiency of our method.

2. Background

Prompt Tuning. Prompt tuning (PT) aims to adapt a pre-trained LLM to various natural language downstream tasks. There are two major types of prompts, 1) *hard or discrete prompts* [24; 25; 9], which are discrete textual tokens prepended to the input text of the LLM, and 2) *soft prompts* [11; 19; 33] which are tunable embedding vectors provided to the LLM's input. While discrete prompts require thorough engineering to yield good performance on downstream tasks, soft prompts can be tuned through standard gradient-based training approaches [14].

Soft Prompt Transfer. Tuning soft prompts via backpropagation can be computationally expensive as LLMs grow in size. This motivates the emergence of attempts to transfer, *i.e.,* to reuse, existing soft prompts. There are two broad scenarios for prompt transfer, *cross-task transfer* [27; 26; 32] and more difficult *cross-model transfer*. Su et al. [26] address the latter, *i.e.,* transferring the soft prompt between different LLMs by using guidance of the private data. However, this exposes the private data directly to the second LLM. Wu et al. [29] present a zero-shot prompt transfer method, where source prompts tuned on a given LLM are encoded into a relative space and used as a form of support vector when finding target prompts on the second, *i.e.,* target model. Unfortunately, in their approach, the target model with the transferred prompt performs worse than the prompted source model, leaving no incentive to use the target model rather than the source model with the prompt.

Differential Privacy for Soft Prompts. Differential privacy (DP) [7] is a mathematical framework that provides privacy guarantees for ML by implementing the intuition that a model $\mathcal{M} : I \rightarrow S$, trained on two neighboring datasets D, D' that differ in only one data point, will yield roughly the same output, *i.e.,* $\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in S] + \delta$. The privacy parameter ε specifies by how much the output is allowed to differ and δ is the probability of failure to meet that guarantee. To adapt soft prompts with DP guarantees, Duan et al. [6] proposed the PromptDPSGD algorithm.

3. Setup and Problem Formulation

The Setup. We consider a setup with two parties, an LLM provider and a user, as shown in Figure 2. The LLM provider deploys a general-purpose LLM and offers paid query access to it. The user holds private data and wants to adapt the LLM on this data to solve their downstream tasks while ensuring the confidentiality and privacy of their data towards the LLM provider.

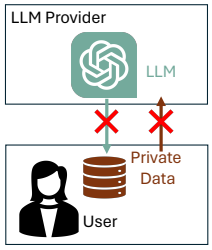


Figure 2. **The Setup.**

The Problem. Unfortunately, since soft prompt tuning relies on computing the gradients of the data with respect to the model, both data and LLM are required to "interact" directly. The problem is that the user cannot share their data with the LLM provider due to privacy concerns while the LLM provider cannot share their LLM because of 1) intellectual property concerns and since 2) this would disrupt their business model, as users would no longer be required to pay for accessing model queries. Additionally, most users would lack the necessary computational resources to tune the soft prompt on the large LLM locally, as this requires

calculating gradients over the entire model. Consequently, due to these limitations, the powerful LLM could not be used for private tasks.

4. Our Private Transfer of Soft Prompts Framework

We solve the above-mentioned problem by proposing **P**rivacy **O**f **S**oft-prompt **T**ransfer (POST). POST consists of three main building blocks, (1) a knowledge distillation from the LLM to a small model, (2) private prompt tuning, and (3) a privacy-preserving prompt transfer using public data. We detail those building blocks in the following.

4.1. Knowledge Distillation

We denote the large LLM (teacher) model as Φ_t, the small student models as Φ_s, the input sequence to an LLM as x. We leverage KD in [23] to derive Φ_s from Φ_t. Different from previous work in LLM distillation [23; 30] that moderately compresses the LLM and tunes the whole model to recover performance as much as possible, we perform a more aggressive KD without emphasis on the student model's performance.

The objective used in the knowledge and the way we construct the student model is show in Appendix C.1

4.2. Private Prompt Tuning

The goal is to tune a local prompt p_s on the small source model Φ_s using the private data D_{pri} such that p_s minimizes the loss $\mathcal{L}$ on the private downstream task as

$$\arg\min_{p_s} \sum_{x \in D_{pri}} \mathcal{L}(\Phi_s, p_s + x). \tag{1}$$

This approach can be performed with standard PT. However, this only provides confidentiality for the private data since the data is not directly sent to the LLM provider. Recent work [5], however, highlights that private information can leak from tuned prompts.

To formally bound privacy leakage, p_s can also be tuned with DP guarantees, for example, using the PromptDPSGD algorithm [6]. During optimization, PromptDPSGD clips the per-sample gradients of the loss to a clip norm c and adds Gaussian noise drawn from $\mathcal{N}(0, \sigma^2, c^2)$ to provide (ε, δ)-DP guarantees.

4.3. Privacy-Preserving Prompt Transfer through Public Data

The prompt p_s, tuned on the small source model Φ_s, could, in principle, be directly applied to the large target LLM Φ_t. However, as described above, they do not initially perform very well on other LLMs. A naive solution is to fine-tune the target prompt p_t on the private data D_{pri}. However, this would disclose the private data to the LLM provider and is, hence, not acceptable. As an alternative, we propose a privacy-preserving prompt transfer that leverages a small public dataset D_{pub} in an efficient transfer step to derive a high-utility prompt p_t from p_s.

We start by initializing the target prompt p_t with the same initialization of p_s, then iteratively update p_t. For the iterative update, we use the loss function

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_1 + \alpha\mathcal{L}_2, \tag{2}$$

that consists of two different loss terms. The first loss term is defined as

$$\mathcal{L}_1 = \sum_{\hat{x} \in D_{pub}} \text{KLDiv}(\Phi_t(p_t + \hat{x})), \Phi_s(p_s + \hat{x})), \tag{3}$$

where KLDiv denotes the Kullback–Leibler divergence. It aims for aligning the predictions of the prompted source and target model on the public data. The second loss term is defined by

$$\mathcal{L}_2 = \sum_{\hat{x} \in D_{pub}} \text{KLDiv}((\Phi_t(p_t + \hat{x})) - \Phi_t(\hat{x})), (\Phi_s(p_s + \hat{x}) - \Phi_s(\hat{x})), \tag{4}$$

and optimizes to align the direction change induced by the private prompt between Φ_t and Φ_s, again on the public data.

The hyperparameter α in Equation (2) controls the balance between the two loss terms. We observe that a good choice of α depends largely on the model's zero-shot performance.

5. Empirical Evaluation

5.1. Experimental Setup

Models and Datasets. To obtain the compressed model, we follow Sanh et al. [23] to aggressively distill a 12-layer Roberta-base [17] into a 2-layer model and a 48-layer GPT2-XL [21] into a 4-layer small model. We evaluate the performance of our proposed method on four classification datasets: sst2 from the GLUE benchmark [28], imdb [18], tweet [22] and arisetv [3]. We use these four datasets along with agnew [31] as public datasets to facilitate the prompt transfer. We discuss the choice of the public datasets for transfer in detail in Appendix C.4. We follow Li et al. [15] to formulate the classification task as a text-infilling task.

KD, Prompt Tuning, and Prompt Transfer. We follow [23] to set the hyperparameters of knowledge distillation (see Appendix C.1 for details). To train soft prompt, we follow settings in Su et al. [26]. When applying DP, we use PromptDPSGD proposed by Duan et al. [6]. Our prompt tuning settings are presented in Appendix C.3. During the prompt transfer, the model provider has no access to the private dataset to find the right moment to stop the transfer,

Table 1. **Runtime of POST vs. Full PT.** We present the runtime for our method, split by its individual components and compare against full prompt tuning on the large LLM. We use arisetv and sst2 as private data. We execute 5000 steps of transfer. PT on Φ_t, Φ_s takes 20 epochs until convergence. All experiments are executed on a single A100 GPU.

Method	arisetv (min)	sst2 (min)
PT on Φ_t	184	2660
(1) PT on Φ_s	23	310
(2) Transfer	99	99
Ours total (1)+(2)	**122**	**409**

so we report the transferred accuracy at fixed steps. We use 5000 steps for Roberta-base and 8000 steps for GPT2-XL. For each private dataset, we report the transfer performance obtained using two different public datasets.

Metrics and Baselines. To evaluate the success of our method, we report the accuracy of the test data split of our private datasets for the large LLM with the transferred prompt (**Private Transfer**). As baselines for comparison, we include the zero-shot performance of the large LLM on the private tasks' test sets (**Full ZS**), representing the lower bound our method should improve upon. Additionally, we provide the performance of tuning the prompt for the large LLM on the private training data, which, due to privacy concerns, is not feasible in practice (**Full PT**). This serves as the theoretical upper bound for potential performance. We also report the accuracy of the prompted compressed model after tuning the prompt on it (**Compressed PT**), as our private transfer must improve over this metric to justify using the large LLM instead of the small prompted one. Finally, we report the direct transfer accuracy (**Direct Transfer**), which is the accuracy achieved when the prompt tuned on the small model is directly applied to the large one, highlighting the effectiveness of our prompt transfer step.

5.2. Private Prompt Transfer with POST

Confidential Transfer. In Table 2, we evaluate the performance of our method in a scenario where only the confidentiality of the private data is protected. Therefore, the user locally tunes a soft prompt without DP guarantees. For each private dataset, we experiment with two different public datasets for prompt transfer and report the respective transferred accuracy on the private dataset. We first observe that the transferred performance is significantly higher than the zero-shot performance. Additionally, after the prompt transfer with POST, we outperform the small compressed prompted model, giving users a strong incentive to transfer their prompt back to the large LLM. Additionally, we show that our prompt transfer described in Section 4.3 is highly effective as it improves over the direct transfer performance by a large margin. For the arisetv dataset, the transferred soft prompt even outperforms the soft prompt directly tuned on the large model. In contrast to the soft prompt transfer method by Wu et al. [29] which showed a *decrease* in accuracy after transfer, our results highlight the practical applicability and the benefits of using our method.

Differntially Private and Confidential Transfer. In addition to just transferring the locally tuned prompt—which provides confidentiality of the private data towards the LLM provider—we also perform experiments where we tune the local prompt with DP. This yields provable upper bounds for the privacy leakage towards the LLM provider and towards third parties that might interact with the prompted LLMs eventually. Since the prompt transfer is executed using a few *public* data points, no additional privacy leakage is incurred in that step. We show the results of our experiments with privacy guarantees for $\varepsilon = 8$ in Table 3. The trends observed for the confidential prompt transfer also hold under local soft prompt tuning with DP. In particular, we observe that the improvement of the transfer performance to the large LLM over the performance on the prompted compressed model is even more significant than in the non-DP setup.

5.3. Number of Public Samples, Transfer Steps, and Runtime

We also investigate the influence of the size of the public dataset required to complete the transfer and how many transfer steps are required to obtain good performance. The results in Appendices D.2 and D.3 show that our method only needs less than 100 samples and executes about 1000 steps for Roberta-base and about 2000 steps for GPT2 to achieve comparable performance.

We also compare the runtime of our method with prompt tuning on the full model in Table 1. We show that our method can achieve 1.5× speedup on the smaller arisetv dataset, and about 6× speedup on sst2 dataset when transferring with 5000 steps. See Appendix D.4 for detail. These results highlight that our POST also yields substantial improvements in computational efficiency

5.4. Comparing against State-of-the-Art Prompt Transfer Approaches

We compare against two baselines, namely the Zero-Shot transfer by Wu et al. [29] and DP-OPT by Hong et al. [13]. **Zero-Shot transfer** operates in the same setup as we do and also relies on soft prompts. They perform prompt transfer by using the embeddings of some tokens from the vocabulary as a form of support vector to transform the source prompts into a relative space, and then search for the corresponding target prompt embeddings for the target model. To provide the optimal source model for their approach, we use a compressed model that we obtained by keeping the embedding layer frozen during KD (see row 3 in Table 11). **DP-OPT**, in contrast to ours, is designed for discrete prompts. They first tune a discrete prompt locally and then directly use it

Table 2. **Confidential prompt transfer performance.** We compress Roberta-base and GPT2-XL, tune prompts for different private dataset on the compressed models, and transfer them back using different public datasets (POST). Our POST significantly improves performance over the small prompted model and our prompt transfer yields a strong improvement over the direct transfer.

						POST (ours)		
Private	Full ZS	Full PT	Compressed PT	Direct Transfer	Public	Test acc	Public	Test acc
sst2	72.25	91.74	79.10	76.49	tweet	**87.73**	imdb	85.21
imdb	72.19	89.88	78.85	76.92	tweet	**83.96**	sst2	80.27
tweet	36.53	68.68	56.65	43.10	imdb	54.55	sst2	**58.25**
arisetv	38.80	78.55	70.98	47.82	agnews	**82.73**	tweet	68.48

(a) Roberta-base.

						POST (ours)		
Private	Full ZS	Full PT	Compressed PT	Direct Transfer	Public	Test acc	Public	Test acc
sst2	60.78	94.84	80.94	59.06	tweet	**85.89**	imdb	83.49
imdb	60.27	93.28	81.32	60.34	tweet	**83.93**	sst2	82.15
tweet	34.71	68.60	**63.13**	41.50	imdb	61.75	sst2	57.70
arisetv	52.98	87.22	77.10	55.43	agnews	**87.56**	tweet	82.12

(b) GPT2-XL.

Table 3. **Differentially Private and Confidential prompt transfer performance.** We compress Roberta-base and GPT2-XL, tune prompts for different private dataset on the compressed models with Differential Privacy guarantees ($\varepsilon = 8$), and transfer them back using different public datasets (POST). Our POST significantly improves performance over the small prompted model and our prompt transfer yields a strong improvement over the direct transfer.

						POST (ours)		
Private	Full ZS	Full PT	Compressed PT	Direct Transfer	Public	Test acc	Public	Test acc
sst2	72.25	90.14	67.54	77.06	tweet	**84.40**	imdb	81.42
imdb	72.19	88.55	72.22	74.35	tweet	79.64	sst2	**80.64**
tweet	36.53	62.05	40.87	43.15	imdb	55.65	sst2	**59.25**
arisetv	38.80	72.53	64.25	47.34	agnews	**79.11**	tweet	71.98

(a) Roberta-base.

						POST (ours)		
Private	Full ZS	Full PT	Compressed PT	Direct Transfer	Public	Test acc	Public	Test acc
sst2	60.78	91.28	74.31	57.80	tweet	79.93	imdb	**84.06**
imdb	60.27	89.59	74.81	63.66	tweet	**78.03**	sst2	75.16
tweet	34.71	61.47	48.60	41.50	imdb	**58.05**	sst2	54.75
arisetv	52.98	79.03	67.16	57.25	agnews	**82.12**	tweet	80.55

(b) GPT2-XL.

Table 4. **Baseline comparison.** We present the performance of our method against state-of-the-art baselines. We report test accuracies over different private datasets D_{pri}. For our POST, we report the accuracies under the best public dataset (see Table 2 and Table 3).

Method	Φ_t	Φ_s	sst2	imdb	tweet	arisetv
OPT [13]	GPT2-XL	our compressed	60.67	61.70	30.70	42.87
OPT [13]	GPT2-XL	GPT2	62.16	63.18	35.20	46.38
Zero-Shot Transfer [29]	GPT2-XL	our compressed	63.65	61.27	41.60	56.64
Zero-Shot Transfer [29] with DP	GPT2-XL	our compressed	63.42	61.71	41.35	57.25
POST (ours)	GPT2-XL	our compressed	**85.89**	**83.93**	**61.75**	**87.56**
DP-POST (ours)	GPT2-XL	our compressed	84.06	78.03	58.05	82.12

on the large model. Since their method relies on the small model having good performance, we execute their method in two setups for a fair comparison. 1) We tune their source prompt using our compressed model as the small model, and 2) we use GPT2 as the small model. The latter is expected to have significantly higher performance and yield better prompts. To avoid the massive hyperparameter tuning required for the private tuning in DP-OPT, we resolve to the standard OPT without DP guarantees following their implementation [13]. The obtained results represented an upper bound of DP-OPT, as introducing DP usually degrades performance. Our results in Table 4 highlight that our POST significantly outperforms all baselines.

6. Conclusions

We present POST, a framework for the private transfer of soft prompts that enables adapting LLMs of an LLM provider with users' private data while protecting both the user's privacy and the LLM provider's intellectual property. POST achieves significant improvements on the private tasks through the prompt transfer, improves computational efficiency of prompt tuning, and outperforms all private prompt transfer baselines. Thereby, our work paves the way for a wider and more trustworthy application of LLMs.

References

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[3] chimaobi Samuel, O. news-data. Huggingface, 2022.

[4] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Duan, H., Dziedzic, A., Yaghini, M., Papernot, N., and Boenisch, F. On the privacy risk of in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[6] Duan, H., Dziedzic, A., Papernot, N., and Boenisch, F. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[7] Dwork, C. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

[8] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

[9] Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. In *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. 3816–3830, 2021.

[10] Geng, X. and Liu, H. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

[11] Hambardzumyan, K., Khachatrian, H., and May, J. Warp: Word-level adversarial reprogramming. In *ACL-IJCNLP 2021-59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 4921–4933, 2021.

[12] Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[13] Hong, J., Wang, J. T., Zhang, C., Li, Z., Li, B., and Wang, Z. Dp-opt: Make large language model your privacy-preserving prompt engineer. *ArXiv*, abs/2312.03724, 2023. URL https://api.semanticscholar.org/CorpusID:266051675.

[14] Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[15] Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=bVuP3ltATMz.

[16] Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[17] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

[18] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

[19] Qin, G. and Eisner, J. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021.

[20] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

[21] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

[22] Rosenthal, S., Farra, N., and Nakov, P. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 502–518, 2017.

[23] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[24] Schick, T. and Schütze, H. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269, 2021.

[25] Schick, T. and Schütze, H. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2339–2352, 2021.

[26] Su, Y., Wang, X., Qin, Y., Chan, C.-M., Lin, Y., Wang, H., Wen, K., Liu, Z., Li, P., Li, J., et al. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3949–3969, 2022.

[27] Vu, T., Lester, B., Constant, N., Al-Rfou, R., and Cer, D. Spot: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5039–5059, 2022.

[28] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

[29] Wu, Z., Wu, Y., and Mou, L. Zero-shot continuous prompt transfer: Generalizing task semantics across language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[30] Xiao, G., Lin, J., and Han, S. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 2023.

[31] Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[32] Zhong, Q., Ding, L., Liu, J., Du, B., and Tao, D. Panda: Prompt transfer meets knowledge distillation for efficient model adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[33] Zhong, Z., Friedman, D., and Chen, D. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5017–5033, 2021.

[34] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

A. Limitations

Our work proposes a method to protect the privacy and confidentiality of private data during the prompt tuning phase, however, we didn't address the privacy leakage risk during the inference phase. Also, compression of the LLMs through knowledge distillation techniques may be computationally expensive for LLM providers. Additionally, in our method, the selection of a public dataset will affect the transfer performance of soft prompts. While we observe, in general, that public datasets that have a similar structure to the private data work best for transfer, there is no ideal strategy for selecting the optimal public dataset

B. Broader Impacts

Regarding the broader impacts of our work, we propose a private transfer of soft prompts from a small language model to a large LLM. The primary positive societal impact of our work is that our method can protect local data privacy and also the intelligent property of the large model provider, which encourages wider and more trustworthy applications of LLMs. Additionally, since our transfer enables more compute efficient prompt tuning and enables to re-use existing prompts, it can have a positive environmental impact.

C. Experimental Setup

C.1. Knowledge Distillation

We follow the procedure of [23] to initialize and distill our compressed model. In detail, we rely on the following loss from [23] to distill Φ_s from Φ_t:

$$\mathcal{L}_{distil} = \alpha_{ce}\mathcal{L}_{ce} + \alpha_{lm}\mathcal{L}_{lm} + \alpha_{cos}\mathcal{L}_{cos}. \tag{5}$$

The objective is a linear combination of distillation loss $\mathcal{L}_{ce}$, language modeling loss $\mathcal{L}_{lm}$ and embedding cosine loss $\mathcal{L}_{cos}$. Where $\mathcal{L}_{ce}$ is the Kullback–Leibler divergence loss between the logits of Φ_s and Φ_t, $\mathcal{L}_{lm}$ is the standard objective used in pre-train a language model, *i.e.,* the cross entropy loss for predicting the masked/next tokens, and $\mathcal{L}_{cos}$ is the cosine distance of the embedding of Φ_s and Φ_t with α_{ce}, α_{lm} and α_{cos} weighting the respective losses.

We use the first and last layers of Roberta-base and the first two and last two layers of GPT2-XL to initialize our compressed Roberta-base and GPT2-XL before knowledge distillation. We also initialize the small student model's word embedding and language modeling head the same as their teacher model. We conduct experiments on whether to freeze the language modeling head and/or word embedding during knowledge distillation. The model's structure and size are listed in Table 5.

Table 5. **Model size before and after distillation.**

model	layer number	hidden dimension	head number	parameter num (M)
Roberta-base	12	768	12	125
our distilled Roberta-base	2	768	12	53
GPT2-XL	48	1600	25	1560
our distilled GPT2-XL	4	1600	25	205

During knowledge distillation, we use the BookCorpus [34] dataset, and we took the checkpoint model that distilled for 50,0000 steps. The hyperparameters used in knowledge distillation are shown in Table 6.

Table 6. **Hyperparameters in knowledge distillation.**

α_{ce}	α_{lm}	α_{cos}	lr	batch size
5.0	2.0	1.0	0.00025	5

C.2. Text-infilling tasks

We use the text-infilling setting for the classification task. The setting is to let the model predict the ground truth text instead of using a classification head to output the class probability. To increase the robustness of this method, we use multiple ground truth text labels, and compare the average probability of outputting those text labels. See Table 7 for task templates and the ground truth labels used in our experiment.

Table 7. **Task template and ground truth labels used in text-infilling.** <s>means the sentence used in the dataset.

Dataset	Task Template Roberta	Task Template GPT2	Ground Truth Text Label
sst2	<s>, it was <mask>	<s>, it was	0: [" terrible"," negative"," bad"," poor"," awful"] 1: [" positive"," good"," great"," awesome"," brilliant"," amazing"]
imdb	<s>, it was <mask>	<s>, it was	0: [" terrible"," negative"," bad"," poor"," awful"] 1: [" positive"," good"," great"," awesome"," brilliant"," amazing"]
tweet	<s>, it was <mask>	<s>, it was	0: [" terrible"," negative"," bad"," poor"," awful"] 1: [" moderate"," neutral"," balanced"]] 2: [" positive"," good"," great"," awesome"," brilliant"," amazing"]
arisetv	<s>, it was about <mask>	<s>, it was about	0: [" business"], 1: [" sports"], 2: [" politics"] 3: [" health"],4: [" entertainment"],5: [" technology"," science"]

C.3. Prompt tuning

Following [26]'s setting, we use the soft prompt with a length of 100 tokens in all our experiments. We follow [6]'s setting to obtain DP private prompt with PromptDPSGD. Table 8 shows the hyperparameters used in this experiment.

Table 8. **Hyperparameters used during promptDPSGD.**

dataset	δ	epochs	lr
sst2	1.5×10^{-5}	20	0.1
imdb	4×10^{-5}	20	0.1
tweet	2×10^{-5}	20	0.1
arisetv	2×10^{-4}	20	0.1

C.4. Public Datasets for Prompt Transfer

We rely on small public datasets to perform our prompt transfer. A question is the right choice of the public dataset. We normally choose the public dataset that performs a similar task as the private dataset, such as choosing imdb or tweet as the public dataset of sst2 as they are all sentiment classification tasks. Transferring with a public dataset that performs a different task from the private dataset may lead to suboptimal performance, we tested this setting to transfer soft prompt trained on arisetv, a topic prediction dataset. The transfer performance of using tweet as public dataset is acceptable but generally worse than using agnews, another topic prediction dataset, as a public dataset. In general, we found that the public and private dataset do not need to have the same structure, such as class number. For example, using tweet (3 classes) as a public dataset leads to better transfer performance than imdb (2 classes) on sst2 (also 2 classes). This highlights the robustness of our method and the broad selection of public datasets for the transfer.

We report the hyperparameters used in the transfer experiments as Tables 9 and 10.

Table 9. **Hyperparameters used during prompt transfer.**

model	batch size	optimizer	lr
Roberta-base	32	Adam	0.001
GPT-XL	8	Adam	0.001

Table 10. **Setting of α for different datasets and models during prompt trasnfer.**

	dataset			
model	sst2	imdb	tweet	arisetv
Roberta-base	0.8	0.8	0.5	0.5
GPT2-XL	0.7	0.7	0.2	0.6

D. Additional Experiments

D.1. Ablations

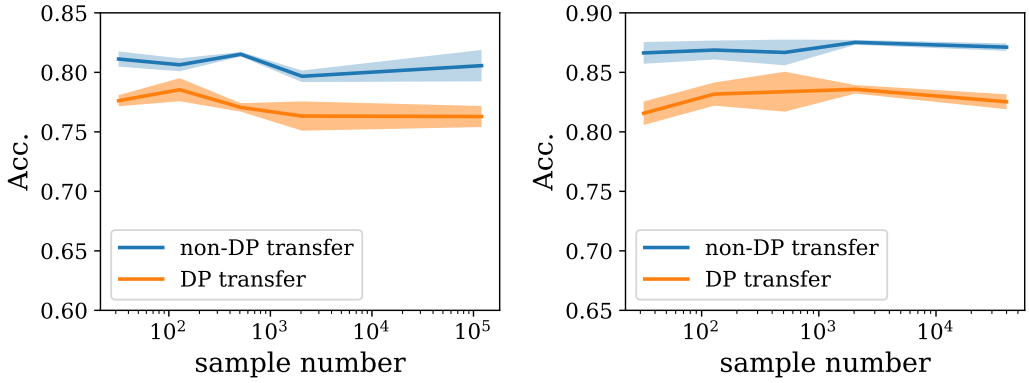
We also investigated the best way of performing KD to improve prompt transferability. In particular, we analyzed the impact of keeping the word embedding or(and) language modeling heads frozen during KD on the prompt transfer performance. Our results in Table 11 highlight that keeping the language modeling head fixed performs slightly better than the alternative which mainly perform on-par. These results indicate that the successful transfer of our method is robust to the KD and independent of any specific KD setting.

Table 11. **Analyzing the KD setup.** We perform an ablation on different designs of the KD and present their impact on the prompt transfer for the private arisetv dataset, using agnews as public data. We analze different combinations of freezing the embedding (Fix emb) and freezing the language modeling head (Fix head).

model	Fix emb	Fix head	Acc.	model	Fix emb	Fix head	Acc.
Roberta-base	×	✓	81.68 ±0.764	GPT2-XL	×	✓	87.52 ±0.505
	✓	✓	80.79 ±0.885		✓	✓	86.51 ±0.726
	✓	×	80.84 ±0.360		✓	×	86.81 ±0.732
	×	×	80.11 ±0.738		×	×	87.48 ±0.170

D.2. Effect of Number of Public Samples used for Transfer

We also investigate the influence of the size of the public dataset required to complete the transfer. Our results in Figure 3 show that we can already yield high transfer performance with less than 100 public data points. This small size of public datasets needed makes our method highly practical.



(a) Roberta-base

(b) GPT2-XL

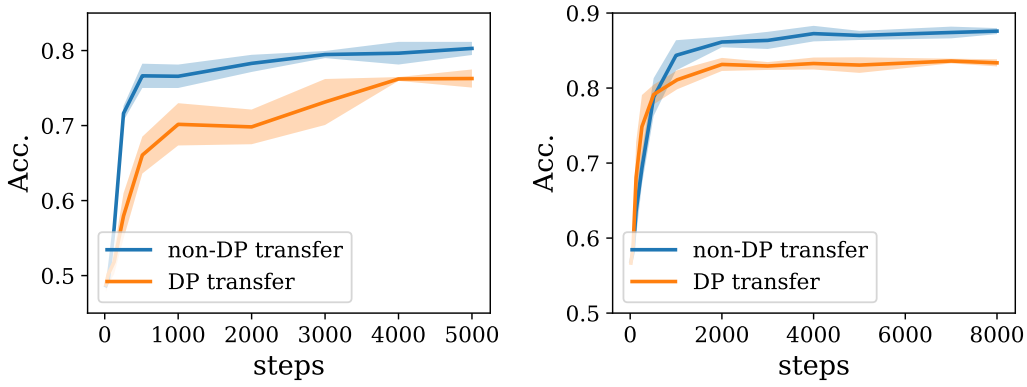
Figure 3. **Effect of number of public samples.** We depict the number of samples from the public dataset used to perform our prompt transfer. We plot results for arisetv as the private dataset with data subsampled from agnews as public data. Our results highlight that with even less than 100 public data samples, our transfer yields high performance.

D.3. Effect of Number of Transfer Steps

We additionally investigate how many transfer steps are required to obtain good performance. Based on the insights from the previous section, we randomly subsample 128 samples from the agnews dataset as public data and report the achieved accuracy on arisetv as private data over different numbers of transfer steps. Our results in Figure 4 highlight that only a small number of transfer steps is enough for convergence and high accuracy on the private task. In particular, while for GPT2-XL, performance converges at around 2,000 steps for Roberta-base, we already observe convergence starting at 1,000 steps.

D.4. Runtime of our Method vs. Full Prompt Tuning on the Large Model

While, in practice, tuning the large LLM with the private data exhibits severe privacy risks and is, hence, not applicable, we compare runtimes to get an insight on the computational gains introduced by tuning the prompt on a small model and



(a) Roberta-base

(b) GPT2-XL

Figure 4. **Effect of number of transfer steps.** We vary the number of steps during our private prompt transfer. We plot results for arisetv as the private dataset and agnews as public data. We observe that already a small number of transfer steps yields high performance.

then transferring it. Since the PT time is determined by the size of the dataset if we want to backpropagate over all private training examples, we present the runtimes of our approach vs. prompt tuning on the large LLM for two different-sized datasets in Table 1. While on the small arisetv dataset, PT on the large model takes 150% of the time of executing our POST, for the larger sst2 datasets, our method improves the runtime roughly by a factor of six (409 instead of 2660 minutes on an A100). These results highlight that beyond the privacy protection, our POST also yields substantial improvements in computational efficiency.