IMPROVING PROXY TRANSFER VIA INTERMEDIATE PROXY TUNING

Kevin Kuo, Ayush Sehgal, Robert Pare, Virginia Smith
Carnegie Mellon University

ABSTRACT

While large language models (LLMs) have been shown to benefit from fine-tuning on downstream data, on-device settings face compute and data privacy constraints which make directly finetuning these models infeasible. In settings where a large black-box (target) model cannot run on-device but can be remotely queried at deployment time, proxy tuning (PT) is a natural solution that fine-tunes a small white-box (proxy) model and combines its predictions with those of the target. However, when the pretrained knowledge of the two models differs significantly, PT can surprisingly perform worse than using either the target or proxy alone. To improve transferability of the proxy model without assuming training-time access to the target model, we propose IPT (Intermediate Proxy Tuning), which guides proxy tuning with an intermediate model. On three NLP tasks of next token predictions, topic classification, and question answering, IPT improves post-transfer accuracy by up to 5% and lowers PPL by up to 6 points over naive PT.

1 INTRODUCTION

New public data for large language model (LLM) training is becoming scarce. Within a decade, LLMs are projected to be trained on datasets the size of the total stock of public human data (Villalobos et al., 2024), while individuals and organizations are increasingly restricting access to their data due to economic and privacy concerns (Longpre et al., 2024). Therefore, it is necessary to develop methods to enable machine learning on restricted sources of data in otherwise data-scarce domains.

In this work, we consider a two-party setting involving a low-resource **client** with private data and a high-resource **server** which serves an LLM. Due to these constraints, it is infeasible for the client to run inference on an LLM, much less personalize it via fine-tuning. Furthermore, the client cannot simply send their data to the server, because this data may be their key intellectual property or protected under privacy law.

We study **Proxy Tuning (PT)** as a natural solution to these challenges. PT combines the logits of a black-box LLM (the "target") and a smaller white-box LM (the "proxy") fine-tuned on downstream data (Liu et al., 2024). More precisely, the difference in the fine-tuned and untuned proxy LM logits is added to the logits of the target LLM. A key feature of PT is that the proxy is fine-tuned in isolation, which makes training costs cheap and keeps private data at the client. However, this also means the proxy is trained without knowledge of future transfer, which can cause PT to perform worse than simply using the proxy or target alone. A solution to this is **Consistent Proxy Tuning (CPT)**, which uses the target LLM during training to construct outputs consistent with those expected during evaluation (He et al., 2024). While this method achieves superior performance due to its use of "ground-truth" target predictions, it requires either running the target LLM on-device (computationally infeasible) or sending training data to the LLM provider (violating privacy constraints). Thus, a natural question is:

"How can we improve the transferability of the proxy LM to the target LLM without having direct access to the target LLM during training?"

Correspondence to: kkuo2@andrew.cmu.edu

To address this challenge, we propose a novel proxy tuning method called **Intermediate Proxy Tuning** (IPT). IPT bridges the gap between PT and CPT by achieving similar performance to CPT without requiring training-time access to the target model. In summary, our contributions are:

1. We propose IPT, a novel method which uses an intermediate model to improve transferability of models learned via proxy tuning. Despite its simplicity, IPT has been overlooked as an effective and efficient improvement over naive PT.
2. We validate our approach on three NLP datasets and two Transformer-based model families. IPT empirically improves performance over PT by up to 5% accuracy or 6 PPL, while its design retains the key advantages of PT: efficient proxy training and only test-time access to the target model.
3. IPT does not rely on server-side computation or the target LLM at all during training. This makes IPT a more practical method to use compared to works in transfer learning which use the target LLM to generate side information at the server, such as Private Evolution (Hou et al., 2024) or POST (Wang et al., 2025)).

2 Related Work

Transfer learning is a class of methods that "transfer" knowledge learned from one task to another. In most cases, this is realized in the form of large-scale pretraining followed by task-specific fine-tuning (Deng et al., 2009; Bengio, 2012; Huh et al., 2016; Radford et al., 2019; Touvron et al., 2023). While parameter compression (Houlsby et al., 2019; Frantar & Alistarh, 2023; Dettmers et al., 2023) methods can reduce fine-tuning costs, this "large-to-small" paradigm of transfer learning is limited by the size of the small fine-tuned model. In this work, we study an emerging "small-to-large" paradigm of transfer learning which aims to do the opposite: we first learn task-specific knowledge using a small model and then transfer this knowledge to a larger model outside our training budget, as to push performance beyond what is possible with the small model alone (Kang et al., 2025; Liu et al., 2025).

Black-box adaptation methods are a natural approach to small-to-large transfer learning, as they do not depend on a target model's internal parameters. Examples of these techniques are knowledge distillation of soft labels (Hinton et al., 2015; Lin et al., 2020; Cho et al., 2022), synthetic data generation (Hou et al., 2024; Zhang et al., 2025), prompt transfer (Zhang et al., 2023; Hong et al., 2023; Wang et al., 2025), and proxy tuning with logit arithmetic (Liu et al., 2024).

Proxy Tuning (PT) methods enable black-box adaptation by aggregating (scaled) logit offsets of small tuned "proxy" models with the logits of a large black-box "target" model (Liu et al., 2021; 2024; He et al., 2024; Ronaghi et al., 2026). While PT is efficient and avoids sharing the client's training data, prior PT methods typically consider fixed proxy and target models. In contrast, our work studies an underexplored setting of using intermediate models for "target-aware" proxy tuning.

3 Method

3.1 Baselines

Consider a large black-box **target** model L with inaccessible pretrained parameters θ_L and a small white-box **proxy** model S with accessible pretrained parameters θ_S. The goal of **Proxy Tuning (PT)** (Liu et al., 2024) is to tune S on a downstream dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ to produce a tuned model S'. At evaluation time, we make predictions which combine the task-specific knowledge of S' and the pretrained knowledge of L. Specifically, the difference in the outputs of the tuned (S') vs. untuned (S) small model is applied as an offset to the logits of L:

$$z_{L;S'} = z_L + (z_{S'} - z_S), \tag{1}$$

We use z_M as shorthand for $z(x; \theta_M)$, the output logits of model M with parameters θ_M on input x.

In vanilla PT, S is tuned independently without any target offset:

$$\theta_{S'} = \arg\min_{\theta_{S'}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(z_{S'}, y\right) \right]. \tag{2}$$

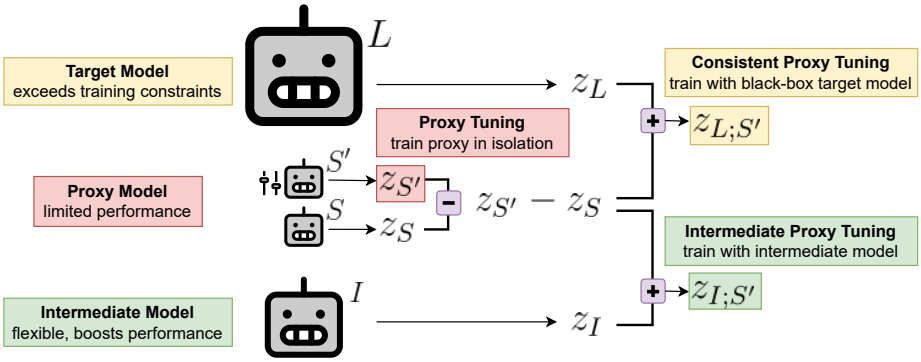

Figure 1: Our Intermediate Proxy Tuning method provides a flexible tradeoff between naive Proxy Tuning and Consistent Proxy Tuning by leveraging an intermediate model chosen to be larger than the proxy but smaller than the target. Highlighted boxes indicate the output logits which are used to compute the training loss, while evaluation always uses the adjusted target logits $z_{L;S'}$.

This creates an inconsistency: training optimizes the proxy model alone, while evaluation applies the learned offset to the target model's logits. **Consistent Proxy Tuning (CPT)** (He et al., 2024) addresses this by incorporating the target L during training:

$$\theta_{S'} = \arg\min_{\theta_{S'}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(z_L + \alpha_{\text{train}}\left(z_{S'} - z_S\right), y\right)\right], \tag{3}$$

where α_{train} scales the proxy offset during optimization. When $\alpha_{\text{train}} = 1$, this is equivalent to training with the adjusted target logits $z_{L;S'}$. By using the same target model (L) during both training and inference, CPT ensures consistency in the optimization objective. However, obtaining logits from the large target model L can be too computationally expensive to run on-device, and data owners may face restrictions against sending their training data to a model provider.

3.2 Intermediate Proxy Tuning (IPT)

We propose a middle ground that balances consistency with accessibility by using an intermediate-sized model I with parameters θ_I. Surprisingly, we find that existing pretrained models are an effective choice for I, offering better transfer performance (vs. PT) and efficiency (vs. CPT).

During training, we incorporate an offset from the intermediate model logits z_I:

$$\theta_{S'} = \arg\min_{\theta_{S'}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(z_I + \left(z_{S'} - z_S\right), y\right)\right], \tag{4}$$

In other words, we train on the adjusted intermediate logits $z_{I;S'}$. At evaluation time, we apply the learned proxy offset to the large target model using Equation 1. IPT is equivalent to CPT when the intermediate and target models are the same ($I = L$), and is equivalent to PT when the proxy and intermediate models are the same ($I = S$).

4 Results

Datasets. We experiment with three datasets: Penn Treebank (PTB), Reddit, and SQuAD 2.0. PTB is a dataset of Wall Street Journal articles on which we train models for general language modeling. For Reddit, we construct a subreddit classification task by filtering comments from December 2012 to retain the 100 most prolific users, then selecting comments from the 5 most popular subreddits. SQuAD 2.0 is a reading comprehension benchmark with questions posed by crowdworkers on Wikipedia articles, where answers are extracted spans from the passage or marked unanswerable.

PT can fail to improve over naive baselines. Surprisingly, we find that PT can result in *negative transfer*, where applying a transfer learning algorithm makes performance *worse* than before (Pan & Yang, 2009). In Table 1, we run a grid of experiments varying intermediate and target model sizes while keeping the proxy model fixed. On Reddit, we find that PT degrades accuracy compared to the

Intermediate Model	Target Model				
	flan-t5-small	base	large	xl	xxl
Pretrained	0.323	0.574	0.669	0.786	0.785
flan-t5-small	0.742	0.498	0.638	0.685	0.675
flan-t5-base	—	0.757	0.771	0.778	0.806
flan-t5-large	—	—	0.784	0.788	0.799
flan-t5-xl	—	—	—	0.793	0.791
flan-t5-xxl	—	—	—	—	0.806

Table 1: Reddit accuracy (↑) after transferring a flan-t5-small proxy model to a target model. "Pretrained" is the performance of the target model before proxy tuning. The **intermediate (row)** model is used to guide training of the proxy model. We then evaluate how much the proxy improves the **target (column)** model. Orange numbers indicate performance of CPT (same intermediate and target), while Red indicates PT (same proxy and intermediate). Green numbers are our method (IPT), which outperforms PT without training-time access to the target model unlike CPT.

Intermediate Model	Target Model				
	distilgpt2	gpt2	gpt2-medium	gpt2-large	gpt2-xl
Pretrained	89.91	63.85	47.25	40.28	36.99
distilgpt2	29.63	36.15	37.60	33.17	33.75
gpt2	—	29.94	27.94	26.37	25.02
gpt2-medium	—	—	26.21	25.71	24.04
gpt2-large	—	—	—	24.50	23.60
gpt2-xl	—	—	—	—	22.52

Table 2: PTB perplexity (↓) after transferring a distilgpt2 proxy model to another target model.

"Pretrained" target model and also performs worse than the tuned proxy alone (flan-t5-small with 0.742 accuracy). In Table 2 (PTB), we find that PT performs better (lower perplexity) than "Pretrained", but still underperforms relative to using the tuned distilgpt2 model alone.

IPT outperforms PT. In Tables 1 and 2, the rows below the smallest model show performance when a larger intermediate model I is used to generate the IPT training logits $z_{I;S'}$. Surprisingly, a slight increase in the size of the intermediate model significantly improves performance. For example, while distilgpt2 and gpt2 achieve similar PPL when finetuned in isolation (29.63 vs. 29.94 when the intermediate and target are both either distilgpt2 or gpt2), gpt2 performs much better as an intermediate model after proxy transfer, from comparing the two distilgpt2 and gpt2 rows within the same target model column.

IPT improves the memory-utility tradeoff. A natural concern is that IPT is more expensive than PT since it requires inference on an additional intermediate model I during training. However, this cost can be controlled by choosing a smaller I. The IPT curves in Figure 2 are obtained by varying the intermediate model size while keeping the proxy and target fixed as the smallest (distilgpt2) and largest (gpt2-xl) models within the family, respectively. In the left plot of Figure 2, peak memory is measured as the maximum of two costs: (1) running inference on the intermediate (target for CPT) model and (2) training the proxy model. This is because (1) and (2) can be run sequentially; after (1), we can save the intermediate logits to CPU and swap from the intermediate to proxy model on the GPU. Using gpt2 as the intermediate has similar peak memory (1 GB) as PT (finetuning distilgpt2 alone) but improves perplexity by 8 points.

In the right plot of Figure 2, we analyze the KL divergence between varying sizes of the intermediate model and the target (gpt2-xl) model on PTB. As KL divergence increases, perplexity tends to also increase, suggesting that intermediate models more similar to the target yield better performance.

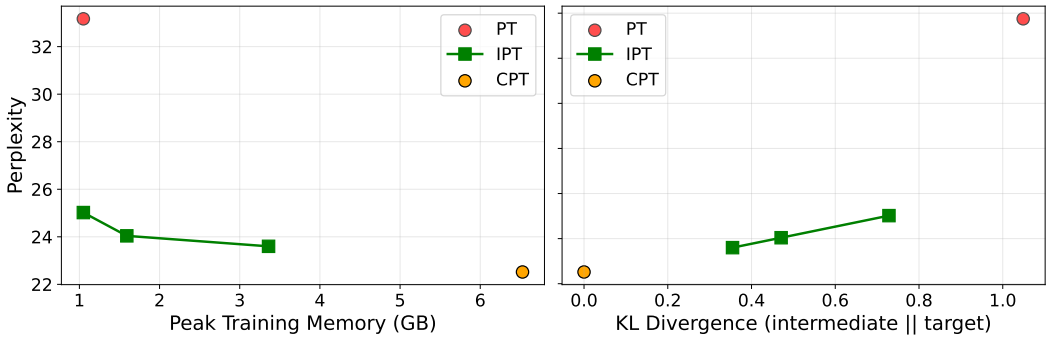

Figure 2: PTB perplexity (↓) after transferring a `distilgpt2` proxy model to a `gpt2-xl` target model. Left: We measure peak GPU memory with a batch size of 1 and sequence length of 512. Right: We measure KL divergence between the intermediate and target models.

5 Conclusion

In this work, we propose Intermediate Proxy Tuning (IPT), a method which can be viewed as an interpolation between two extremes of Proxy Tuning (PT) and Consistent Proxy Tuning (CPT). IPT not only offers a flexible tradeoff between utility and efficiency, but also improves upon both methods when constrained to the same training budget.

Our work is an important step towards unlocking the potential of restricted data in low-resource and private settings, as it allows data owners to efficiently and effectively personalize LLMs to their data without needing to disclose training data to the LLM provider. In future work, we aim to investigate other ways to close the gap between intermediate and target models as well as provide deeper insight ino the failure modes of proxy tuning.

REFERENCES

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *arXiv preprint arXiv:2204.12703*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Yuanyang He, Zitong Huang, Xinxing Xu, Rick Siow Mong Goh, Salman Khan, Wangmeng Zuo, Yong Liu, and Chun-Mei Feng. Cpt: Consistent proxy tuning for black-box optimization. *arXiv preprint arXiv:2407.01155*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Junyuan Hong, Jiachen T Wang, Chenhui Zhang, Zhangheng Li, Bo Li, and Zhangyang Wang. Dp-opt: Make large language model your privacy-preserving prompt engineer. *arXiv preprint arXiv:2312.03724*, 2023.

Charlie Hou, Akshat Shrivastava, Hongyuan Zhan, Rylan Conway, Trang Le, Adithya Sagar, Giulia Fanti, and Daniel Lazar. Pre-text: Training language models on private federated data in the age of llms. *arXiv preprint arXiv:2406.02958*, 2024.

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*, pp. 2790–2799. PMLR, 2019.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Yan Kang, Tao Fan, Hanlin Gu, Xiaojin Zhang, Lixin Fan, and Qiang Yang. Grounding foundation models through federated transfer learning: A general framework. *ACM Transactions on Intelligent Systems and Technology*, 16(4):1–54, 2025.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33:2351–2363, 2020.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*, 2021.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. Tuning language models by proxy. In *First Conference on Language Modeling*, 2024.

Yang Liu, Bingjie Yan, Tianyuan Zou, Jianqing Zhang, Zixuan Gu, Jianbing Ding, Xidong Wang, Jingyi Li, Xiaozhou Ye, Ye Ouyang, et al. Towards harnessing the collaborative power of large and small models for domain tasks. *arXiv preprint arXiv:2504.17421*, 2025.

Shayne Longpre, Robert Mahari, Ariel Lee, Campbell Lund, Hamidah Oderinwale, William Brannon, Nayan Saxena, Naana Obeng-Marnu, Tobin South, Cole Hunter, et al. Consent in crisis: The rapid decline of the ai data commons. *Advances in Neural Information Processing Systems*, 37: 108042–108087, 2024.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

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

Sasha Ronaghi, Chloe Stanwyck, Asad Aali, Amir Ronaghi, Miguel Fuentes, Tina Hernandez-Boussard, and Emily Alsentzer. Training-free adaptation of new-generation llms using legacy clinical models. *arXiv preprint arXiv:2601.03423*, 2026.

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.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

Xun Wang, Jing Xu, Franziska Boenisch, Michael Backes, Christopher A Choquette-Choo, and Adam Dziedzic. Efficient and privacy-preserving soft prompt transfer for llms. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. Vpgtrans: Transfer visual prompt generator across llms. *Advances in Neural Information Processing Systems*, 36: 20299–20319, 2023.

Tuo Zhang, Tiantian Feng, Samiul Alam, Dimitrios Dimitriadis, Sunwoo Lee, Mi Zhang, Shrikanth S Narayanan, and Salman Avestimehr. Gpt-fl: Generative pre-trained model-assisted federated learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1761–1770, 2025.

A Appendix

A.1 Experiment: SQuAD 2.0

In Table 3, we run the same experiment setup as Tables 1 and 2 on the SQuAD 2.0 dataset.

Intermediate Model	Target Model				
	`flan-t5-small`	`base`	`large`	`xl`	`xxl`
Pretrained	0.383	0.516	0.650	0.675	0.674
`flan-t5-small`	0.501	0.655	0.732	0.693	0.733
`flan-t5-base`	—	0.667	0.757	0.684	0.745
`flan-t5-large`	—	—	0.754	0.716	0.742
`flan-t5-xl`	—	—	—	0.768	0.741
`flan-t5-xxl`	—	—	—	—	0.749

Table 3: SQuAD 2.0 accuracy (↑) after transfer learning using a `flan-t5-small` proxy model.