Base-Change at Prediction: Inference-Time Update of Fine-Tuned Models

Daiki Chijiwa [*1] **Taku Hasegawa** [*2] **Kyosuke Nishida** [2] **Kuniko Saito** [2] **Susumu Takeuchi** [1]

Abstract

Foundation models play a central role in recent developments of artificial intelligence on both vision and language domains. However, even if a foundation model is powerful enough at the time to be fine-tuned for various tasks, it will be eventually outdated due to its old knowledge or inadequate capability for new tasks, and then a new foundation model will be prepared by re-training the outdated model with updated data. As a result, the various fine-tuned models based on the outdated model also have to keep up with the new foundation model, typically by fine-tuning again the new foundation model for each task, which should be costly if the number of fine-tuned models or the frequency of updates increases. In this paper, with our simplified theoretical framework, we first derive a probabilistic formula for the fine-tuned model of the new foundation model. Then, based on the formula, we propose a method to avoid the fine-tuning of new foundation models, by editing the predictions of the fine-tuned model in direction to the new foundation model. Compared to previous methods, which edit the predictions of the new foundation model instead, our method consistently keeps or improves accuracy of fine-tuned model for various tasks.

1. Introduction

Foundation models have been the basis of recent advances of artificial intelligence in both vision (Radford et al., 2021) and language domains (Devlin et al., 2018; Brown et al., 2020). They are pre-trained on a large amount of data, and enable us to obtain various task-specific models by fine-tuning with relatively small data for the tasks. However, in spite of their powerful capability, all foundation models may be eventually outdated due to their old knowledge or inadequate capability for new tasks in future.

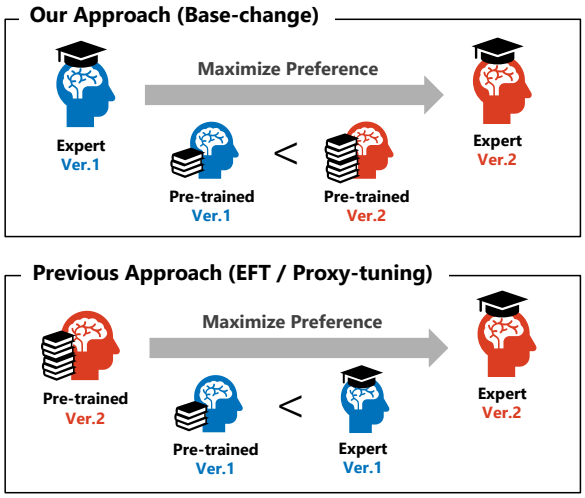


Figure 1: An overview of our approach for inference-time update of fine-tuned models to keep up with update of foundation models. It can be viewed as adjusting the fine-tuned models to prefer predictions from the new foundation model, without actual training, so that it does not degrade the accuracy for various tasks. (See Section 2.2 and 3 for details.)

When some foundation model becomes outdated, a new one would be available through re-training or continual training on updated data. For example, in the recent cases of the LLAMA series (Touvron et al., 2023a;b), LLAMA 2 follows LLAMA 1 after about a half year with increased training data, and CODE LLAMA 2 significantly enhances the coding ability of LLAMA 2. Therefore, the fine-tuned models based on the outdated foundation model also have to keep up with the new foundation model, by re-training the fine-tuned model starting from the new foundation model. However, it is costly to re-train fine-tuned models each time the foundation models are updated, and even impossible if the task-specific data is no longer available at that time.

In this paper, we focus on the research question of how we can enforce the fine-tuned models to keep up with the updates of foundation models, without actual re-training. For this purpose, we first introduce a simplified theoretical framework for sequential updates of pre-trained models and

[*]Equal contribution [1]NTT Computer and Data Science Laboratories, NTT Corporation [2]NTT Human Informatics Laboratories, NTT Corporation. Correspondence to: Daiki Chijiwa <daiki.chijiwa@ntt.com>, Taku Hasegawa <taku.hasegawa@ntt.com>.

ICML Workshop on Large Language Models and Cognition, Vienna, Austria, 2024.

fine-tuning of them, in terms of conditional probabilistic models. Under this framework with some ideal assumptions, we derive a probabilistic formula that makes it possible for the fine-tuned model to keep up with the update of the foundation model without re-training. By relaxing the formula, we propose a novel method called base-change at prediction, which modifies the probability output from the fine-tuned model with the quotient of ones from old and new pre-trained models.

Both theoretically and empirically, we compare our method with the only previous method called emulated fine-tuning (EFT; Mitchell et al. 2024) or an equivalent one called proxy-tuning (Liu et al., 2024a), which was originally proposed to fine-tune a large pre-trained model by actually tuning a small pre-trained model as a proxy. Although these previous work assume both the large and small models are pre-trained on the same data, rather than different data as in our assumption, their method itself can be applied to our problem. For the theoretical comparison of these methods, we employ the framework of offline reinforcement learning (RL) borrowed from the derivation of EFT (Mitchell et al., 2024). In this RL framework, our base-change method can be seen as RL for the fine-tuned model so that it prefers the distribution of the new pre-trained model to the old one, while EFT or proxy-tuning can be seen as RL for the new pre-trained model so that it prefers the fine-tuned model to the old pre-trained model. Experimental results show that our proposed method consistently improves accuracy of the fine-tuned model due to the updated pre-trained model, and outperforms the previous method particularly on the tasks in which the previous method fails.

2. Methods

Notations Let $\mathcal{L}$ be the set of texts, and $p_*(s_1, s_2, \cdots)$ be an ideal prior distribution over unordered, deduplicated sequences of natural texts $s_i \in \mathcal{L}$. Here we assume that a trained language model can be formulated as a conditional distribution $p_*(s|D)$ with some training data $D = \{s_1, \cdots, s_N\} \subset \mathcal{L}$, by ignoring the differences stemmed from the choice of optimization algorithms or model architectures for the trained model.

Throughout this paper, as discussed in Introduction, we focus on the situation that there are two pre-trained language models $p_1(s), p_2(s)$, where $p_2(s)$ is an updated version of $p_1(s)$ with increased training data, and the fine-tuned model $p_1^{\mathrm{ft}}(s) := p_1(s|C)$ of the older one with a task-specific data C. In this situation, $p_1(s), p_2(s)$ can be formulated as $p_1(s) := p_*(s|D_1), p_2(s) := p_*(s|D_1, D_2)$.

2.1. Base-Change at Prediction

Our goal is to simulate the fine-tuned model of the updated pre-trained model, $p_2^{\mathrm{ft}}(s) := p_2(s|C)$, without any additional training, but allowing additional inference cost instead. To achieve this goal, the following proposition plays a key role:

Proposition 2.1. *Assume that D_2, C are conditionally independent given D_1, s under the prior distribution p_*. Then it follows that*

$$p_2^{\mathrm{ft}}(s) = p_1^{\mathrm{ft}}(s)\frac{p_2(s)}{p_1(s)}. \tag{1}$$

The point of Proposition 2.1 is that we explicitly assume the relationship between two pre-trained models p_1 and p_2 as in the previous section. These assumptions enable us to validate the equation (1) by easy calculation as follows:

$$\begin{aligned}
(\mathrm{RHS}) &= p_1^{\mathrm{ft}}(s)\frac{p_2(s)}{p_1(s)} \\
&= p_1(s|C)\frac{p_1(s|D_2)}{p_1(s)} \\
&= \frac{p_1(s,C)}{p_1(C)}\frac{p_1(s,D_2)}{p_1(s)p_1(D_2)} \\
&= p_1(s)\frac{p_1(s,C)}{p_1(s)p_1(C)}\frac{p_1(s,D_2)}{p_1(s)p_1(D_2)} \\
&= p_1(s)\frac{p_1(C|s)}{p_1(C)}\frac{p_1(D_2|s)}{p_1(D_2)} \\
&= p_1(s)\frac{p_1(C,D_2|s)}{p_1(C,D_2)} \\
&= p_1(s|D_2,C) = p_2^{\mathrm{ft}}(s) = (\mathrm{LHS})
\end{aligned}$$

We can view the equation (1) as calibrating the prediction of the old fine-tuned model $p_1^{\mathrm{ft}}(s)$ by the difference between two pre-trained models $p_2(s)$ and $p_1(s)$, which can be interpreted as changing the base model of the fine-tuned model from the old one to new one at prediction. In practice, however, we note that the equation (1) may not work well since it is valid only when the ideal and simplified assumptions hold. Also, a recent language model tends to be trained as a probability of the next-tokens $p(y|x)$ following an incomplete text x, rather than the probability of an entire text $p(s)$. Therefore, for real-world language models, we propose the following relaxed version of equation (1), which we call **base-change at prediction**:

$$p_{2,\alpha}^{\mathrm{ft}}(y|x) \propto p_1^{\mathrm{ft}}(y|x) \underbrace{\left(\frac{p_2(y|x)}{p_1(y|x)}\right)^{\alpha}}_{\text{base-change factor}}, \tag{2}$$

where $\alpha \in \mathbb{R}_{>0}$ is an adjusting parameter that controls the strength of the change of bases.

Model		Source FT	Target PT	EFT / proxy-tuning	Base-change (Ours)
Source PT ⟶ Target PT			GSM: Acc. (scale α)		
LLAMA 2-7B	LLAMA 2 CHAT-7B	39.6	8.9	↓ 28.7	↓ 36.3
	CodeLlama-7B		4.8	↓ 38.9	↑ **42.3**
	LLAMA 2-13B		6.6	↑ 41.7	↑ **42.8**
	CodeLlama-13B		3.3	↑ 40.5	↑ **41.8**
	CodeLlama-34B		6.7	↑ 42.9	↑ **43.0**
			Codex Humaneval: pass@10. (scale α)		
LLAMA 2-7B	LLAMA 2 CHAT-7B	68.9	24.9	↓ 43.9	↓ 67.9
	LLAMA 2-13B		33.7	↓ 63.5	↑ **71.8**
	LLAMA 2 CHAT-13B		31.1	↓ 48.5	↑ **70.2**
			DS1000: pass@10. (scale α)		
LLAMA 2-7B	LLAMA 2 CHAT-7B	46.7	15.5	↓ 32.3	↓ 46.2
	LLAMA 2-13B		25.9	↓ 43.2	↑ **47.2**
	LLAMA 2 CHAT-13B		24.1	↓ 32.8	↑ **47.2**

Table 1: Performance on NLP tasks. For GSM task, the EFT/proxy-tuning and the base-changed model use LLAMA 2 as Source PT, and fine-tuning LLAMA 2 on GSM dataset as Source FT. For Humaneval and DS1000, they use LLAMA 2 as Source PT, and fine-tning CODE LLAMA - PYTHON-7B as Source FT. ↑: Higher than Source FT. ↓ Lower than Source FT. **Bold**: Best score.

2.2. Comparison to existing methods

Previous work proposed a similar but different approach called emulated fine-tuning, EFT (Mitchell et al., 2024), or equivalently called proxy-tuning (Liu et al., 2024a), which is formulated by

$$p^{\mathrm{EFT}}_{2,\alpha}(y|x) \propto p_2(y|x) \underbrace{\left(\frac{p^{\mathrm{ft}}_1(y|x)}{p_1(y|x)}\right)^{\alpha}}_{\text{"task-vector" factor}}. \tag{3}$$

Intuitively, by taking the logarithm of this equation, we can see the previous approach (3) as editing the logits of the new pre-trained model $\log p_2(y|x)$ by a logit-level "task-vector (Ilharco et al., 2023)", $\log p^{\mathrm{ft}}_2(y|x) - \log p_2(y|x)$, scaled with α. On the other hand, in our approach, we leverage the gap between new and old pre-trained models instead of the logit-level task-vector.

For more rigorous comparison of ours (2) and the previous method (3), we employ a viewpoint of offline reinforcement learning from the literatures of Reinforcement Learning from Human Feedback (RLHF; Ziegler et al. 2019) and the EFT paper (Mitchell et al., 2024). Here we consider the following reward maximization problem for $p(y|x)$ with KL constraint to some reference model $p^{\mathrm{ref}}(y|x)$:

$$\max_{p(y|x)} \mathbb{E}_{x\sim p_C(x)} \Big[\mathbb{E}_{y\sim p(y|x)} \big[r(x,y) \big] - \mathrm{KL}\left(p(y|x) \,\|\, p^{\mathrm{ref}}(y|x)\right) \Big], \tag{4}$$

where $r(x,y)$ is some reward function for the next token y given the task-specific text x, which is sampled from the true probability $p_C(x) := p_*(x|C)$ conditioned by the task dataset C. It is well-known (Peters & Schaal, 2007; Rafailov et al., 2024) that the closed solution for equation (4) can be described as

$$p(y|x) = \frac{1}{Z(x)} p^{\mathrm{ref}}(y|x) \exp\left(r(x,y)\right), \tag{5}$$

where $1/Z(x)$ is the normalization factor.

Both our method and the previous method can be seen as closed solutions (5) with appropriate reference model $p^{\mathrm{ref}}(y|x)$ and reward function $r(x,y)$. More specifically, our method (2) is obtained by setting

$$p^{\mathrm{ref}}(y|x) := p^{\mathrm{ft}}_1(y|x),\ r(x,y) := \alpha \log\left(\frac{p_2(y|x)}{p_1(y|x)}\right), \tag{6}$$

and the previous method (3) is obtained by setting

$$p^{\mathrm{ref}}(y|x) := p_2(y|x),\ r(x,y) := \alpha \log\left(\frac{p^{\mathrm{ft}}_1(y|x)}{p_1(y|x)}\right). \tag{7}$$

In conclusion, while EFT or proxy-tuning can be seen as RL tuning of the pre-trained model $p_2(y|x)$ to prefer p^{ft}_1 to p_1, our base-change approach as RL tuning of the fine-tuned model $p^{\mathrm{ft}}_1(y|x)$ to prefer p_2 to p_1, which leads to the stable results of our approach for various tasks in Sec. 3.

3. Experiments

In this section, we compare our method (**base-change**) with the previous method (**EFT / proxy-tuning**) on NLP and image classification tasks. We denote the source pre-trained

Source PT	Target PT	Method	Acc. (scale α) CIFAR	SVHN	RESISC45	SUN397	DTD	Cars
ViT-B16 (LION 400M)		Source FT	88.66	97.20	95.79	77.84	76.60	89.32
	ViT-B16 (LION 2B)	Target PT	76.83	50.05	68.24	19.15	56.60	88.47
		EFT / proxy-tuning	↓ 87.60 (0.7)	↓ 96.87 (0.8)	↓ 95.75 (0.9)	↓ 66.58 (0.8)	↑ 76.33 (0.5)	↑ **91.69 (0.5)**
		Base-change	↑ **88.82 (0.3)**	– 97.22 (0.1)	↑ **95.87 (0.2)**	↑ **78.02 (0.1)**	↓ 76.44 (0.4)	↑ 89.93 (0.4)
	ViT-B16 (Datacomp 1B)	Target PT	82.11	62.72	69.14	19.47	57.77	88.86
		EFT / proxy-tuning	↑ **89.31 (0.5)**	↓ 96.66 (1.5)	↓ 95.71 (0.7)	↓ 64.65 (0.8)	↑ **77.71 (0.5)**	↑ **92.29 (0.4)**
		Base-change	↑ 89.29 (0.6)	↑ **97.21 (0.1)**	↓ **95.81 (0.1)**	↑ **77.99 (0.1)**	↑ 76.91 (0.5)	↑ 89.90 (0.6)
ViT-B16 (LION 400M)	ViT-L14 (LION 400M)	Target PT	77.7	47.2	70.5	17.8	55.3	88.7
		EFT / proxy-tuning	↓ 88.00 (0.9)	↓ 96.60 (1.4)	↓ 95.68 (0.8)	↓ 66.69 (1.5)	↑ **77.82 (0.8)**	↑ **93.25 (0.5)**
		Base-change	↑ **89.33 (0.3)**	↓ 97.17 (0.4)	↑ **96.13 (0.7)**	↑ **78.08 (0.1)**	↑ **77.82 (0.5)**	↑ 92.03 (0.8)
	ViT-L14 (LION 2B)	Target PT	77.42	49.54	68.87	18.34	60.48	89.60
		EFT / proxy-tuning	↓ 87.65 (0.7)	↓ 96.20 (1.1)	↑ 95.87 (0.8)	↓ 68.20 (1.0)	↑ **78.56 (0.6)**	↑ **93.65 (0.4)**
		Base-change	↑ **89.13 (0.4)**	↑ **97.21 (0.2)**	↑ **95.94 (0.2)**	↑ **78.25 (0.1)**	↑ 77.13 (0.4)	↑ 92.33 (0.8)
ViT-L14 (LION 400M)		Source FT	90.61	97.71	95.95	80.64	81.86	93.69
	ViT-L14 (LION 2B)	Target PT	78.33	52.69	71.21	21.64	61.54	91.51
		EFT / proxy-tuning	↓ 87.31 (0.5)	↓ 94.69 (0.5)	↓ 95.48 (0.7)	↓ 66.62 (0.8)	↑ 82.13 (0.9)	↑ **94.43 (0.5)**
		Base-change	↓ 90.52 (0.1)	↓ 97.64 (0.2)	↑ **95.97 (0.1)**	↑ **80.73 (0.1)**	↑ **82.66** (0.9)	↓ 93.53 (0.9)
	ViT-L16 (Datacomp 1B)	Target PT	78.33	52.69	71.21	21.64	61.54	91.51
		EFT / proxy-tuning	↑ **91.76 (0.4)**	↓ 96.02 (1.2)	↑ **96.24 (0.5)**	↓ 65.69 (0.9)	↑ **83.51 (0.9)**	↑ **95.17 (0.5)**
		Base-change	↑ 91.43 (0.4)	↓ 97.70 (0.3)	↑ 95.98 (0.1)	↑ **80.73 (0.1)**	↑ 82.82 (1.2)	↑ 94.42 (0.7)

Table 2: Performance on image classification tasks. ↑: Higher than Source FT. ↓ Lower than Source FT. **Bold**: Best score.

model $p_1(s)$ as **Source PT**, the target pre-trained model $p_2(s)$ as **Target PT**, the source fine-tuned model $p_1^{\text{ft}}(s)$ as **Source FT**, in the equation 1.

We are mainly interested in the following questions: (1) Does base-change improve the accuracy of fine-tuned model due to the improved capability of Target PT over Source PT? (2) Beyond language models, is the base-change approach also valid for vision-language models?

3.1. Natural Language Processing

First, we evaluate base-change at prediction for NLP tasks. We use the LLAMA 2 model family (Touvron et al., 2023b), including the 7B, 13B, and 34B models, for our evaluation. The models are tested on the GSM (Cobbe et al., 2021), HumanEval (Chen et al., 2021), and DS1000 (Lai et al., 2023) datasets. Detailed settings are provided in Appendix B.

Table 1 shows that base-change successfully leverage the improved capability of Target PT over Source PT for each expert task. This also suggests that base-change effectively transfers the knowledge gained from Source FT onto Target PT model. In particular, on GSM, it is interesting to see the base-change from the vanilla LLAMA 2-7B to CODE LLAMA-7B outperforms the Source FT, even though the model scale does not change. This suggests that the reasoning capability of CODE LLAMA-7B (Madaan et al., 2022) helps the base-changed model to solve the math problems in GSM, while EFT/proxy-tuning fails to leverage it. The analysis on the effect of the scale parameter α is also provided in Appendix C.

3.2. Image Classification

Next, we empirically check that base-change at prediction is also valid for image classification tasks. We use CLIP ViT-B16/L14 (Radford et al., 2021) models pre-trained on LION400M (Schuhmann et al., 2021)/LION2B (Schuhmann et al., 2022)/Datacomp1B (Gadre et al., 2024) datasets for image classification tasks. For evaluations, we used six different datasets: CIFAR-100 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014) and Cars (Krause et al., 2013).

Even though our formula 1 is derived for language models, Table 2 shows that base-change consistently outperforms Target PT in various vision tasks, which indicates the successful knowledge transfer from the fine-tuned source model to the target model. In particular, although EFT/proxy-tuning significantly degrades the accuracy on SUN397, base-change keeps or slightly improves its accuracy even on such dataset. Even so, since some settings like DTD or Cars are still suited to EFT/proxy-tuning, we can consider the base-change and EFT/proxy-tuning complement each other.

4. Conclusion

In this work, we proposed a novel approach called base-change at prediction, based on our probabilistic formula and the viewpoint of reinforcement learning. In our experiments, although the previous method often degrades the accuracy in multiple tasks, we found that our method can consistently keep or improve the accuracy of fine-tuned models. Even so, there remain several limitations both in ours and previous methods. A major one is the requirement of the shared vocabulary set, or the shared tokenizer, between source and

target models. Also,the performance improvement over the source fine-tuned models seems still limited. Addressing these limitations is an important direction for future study.

References

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.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE, 105(10):1865–1883, 2017.

Chijiwa, D. Transferring learning trajectories of neural networks. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=bWNJFD1l8M.

Chuang, Y.-S., Xie, Y., Luo, H., Kim, Y., Glass, J. R., and He, P. Dola: Decoding by contrasting layers improves factuality in large language models. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=Th6NyL07na.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , and Vedaldi, A. Describing textures in the wild. In Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

Daheim, N., Möllenhoff, T., Ponti, E., Gurevych, I., and Khan, M. E. Model merging by uncertainty-based gradient matching. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=D7KJmfEDQP.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. Plug and play language models: A simple approach to controlled text generation. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=H1edEyBKDS.

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.

Gadre, S. Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., et al. Datacomp: In search of the next generation of multimodal datasets. Advances in Neural Information Processing Systems, 36, 2024.

Gueta, A., Venezian, E., Raffel, C., Slonim, N., Katz, Y., and Choshen, L. Knowledge is a region in weight space for fine-tuned language models. In The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. URL https://openreview.net/forum?id=vq4BnrPyPb.

Hernandez, E., Li, B. Z., and Andreas, J. Inspecting and editing knowledge representations in language models. arXiv preprint arXiv:2304.00740, 2023.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., and Rajani, N. F. GeDi: Generative discriminator guided sequence generation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 4929–4952, November 2021. URL https://aclanthology.org/2021.findings-emnlp.424.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In Proceedings of the IEEE international conference on computer vision workshops, pp. 554–561, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lai, Y., Li, C., Wang, Y., Zhang, T., Zhong, R., Zettlemoyer, L., Yih, W.-t., Fried, D., Wang, S., and Yu, T. Ds-1000: A natural and reliable benchmark for data science code generation. In International Conference on Machine Learning, pp. 18319–18345. PMLR, 2023.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. Advances in Neural Information Processing Systems, 36, 2024.

Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., and Lewis, M. Contrastive decoding: Open-ended text generation as optimization. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12286–12312, 2023. URL https://aclanthology.org/2023.acl-long.687.

Liu, A., Han, X., Wang, Y., Tsvetkov, Y., Choi, Y., and Smith, N. A. Tuning language models by proxy. arXiv preprint arXiv:2401.08565, 2024a.

Liu, T., Guo, S., Bianco, L., Calandriello, D., Berthet, Q., Llinares, F., Hoffmann, J., Dixon, L., Valko, M., and Blondel, M. Decoding-time realignment of language models. arXiv preprint arXiv:2402.02992, 2024b.

Madaan, A., Zhou, S., Alon, U., Yang, Y., and Neubig, G. Language models of code are few-shot commonsense learners. arXiv preprint arXiv:2210.07128, 2022.

Mitchell, E., Rafailov, R., Sharma, A., Finn, C., and Manning, C. D. An emulator for fine-tuning large language models using small language models. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=Eo7kv0sllr.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In NIPS workshop on deep learning and unsupervised feature learning, volume 2011, pp. 7. Granada, Spain, 2011.

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task arithmetic in the tangent space: Improved editing of pre-trained models. Advances in Neural Information Processing Systems, 36, 2024.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In Proceedings of the 24th international conference on Machine learning, pp. 745–750, 2007.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pp. 8748–8763. PMLR, 2021.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 2024.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114, 2021.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems, 35: 25278–25294, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023b.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In 2010 IEEE computer society conference on computer vision and pattern recognition, pp. 3485–3492. IEEE, 2010.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. Advances in Neural Information Processing Systems, 36, 2024.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593, 2019.

A. Related Work

To achieve our goal of enforcing fine-tuned models to keep up with updates of pre-trained models without actual re-training, there are three possible approaches as follows:

Tuning by Editing Predictions. Our method falls into this category, as well as prior work of emulated fine-tuning (EFT; Mitchell et al. 2024) and proxy-tuning (Liu et al., 2024a). EFT shows that the capability of instruction tuning can be decoupled from instruction-tuned models, by taking the difference between the instruction-tuned and its base (pre-trained) model in the logit space. Proxy-tuning is essentially same as EFT, but is validated with fine-tuning on various tasks other than instruction-tuning, like coding and math tasks. Decoding-time realignment (Liu et al., 2024b) shares the same idea with EFT, but they rather focus on tuning the regularization parameter for instruction-tuning at inference-time. Contrastive decoding (Li et al., 2023) is also categorized in this approach, which proposes to improve the text generation from a language model by subtracting a weaker model in the logit space. Also, Krause et al. (2021) proposed to leverage class-conditional language models in computing next-token distributions for controlled text generation.

Tuning by Editing Parameters. The second possible approach is directly editing parameters of fine-tuned models (Ilharco et al., 2023; Gueta et al., 2023; Ortiz-Jimenez et al., 2024; Yadav et al., 2024; Chijiwa, 2024; Daheim et al., 2024) with parameters of their pre-trained models. This approach is preferable because additional inference cost is not needed, in contrast to the first approach. However, in contrast to our setting, almost all existing methods including task vectors (Ilharco et al., 2023) assume a single shared pre-trained model. Although Chijiwa (2024) explores how to transfer task vectors or learning trajectories to the other pre-trained models, it still requires additional training for practical use.

Tuning by Editing Activations. Editing activations or hidden representations (Dathathri et al., 2020; Hernandez et al., 2023; Chuang et al., 2024; Li et al., 2024), can be seen as an intermediate approach between the above two approaches, which also enables us to control the fine-tuned models without actual training. Dathathri et al. (2020) proposed to modify the activation in a language model by the feedback of gradients from small classification models, in the plug-and-play style. Similarly Hernandez et al. (2023); Li et al. (2024) leverages external classifiers to modify activations. Chuang et al. (2024) proposed to leverage activations from different layers in a single language model in a contrastive way. One major drawback in this approach is the requirement of the same dimension for hidden representations between source and target models.

B. Detailed Experiment Settings

B.1. Settings for NLP

We use the LLAMA 2 model family (Touvron et al., 2023b), including the 7B, 13B, and 34B models, for our evaluation. The models are tested on the GSM, Codex HumanEval and DS1000 datasets.

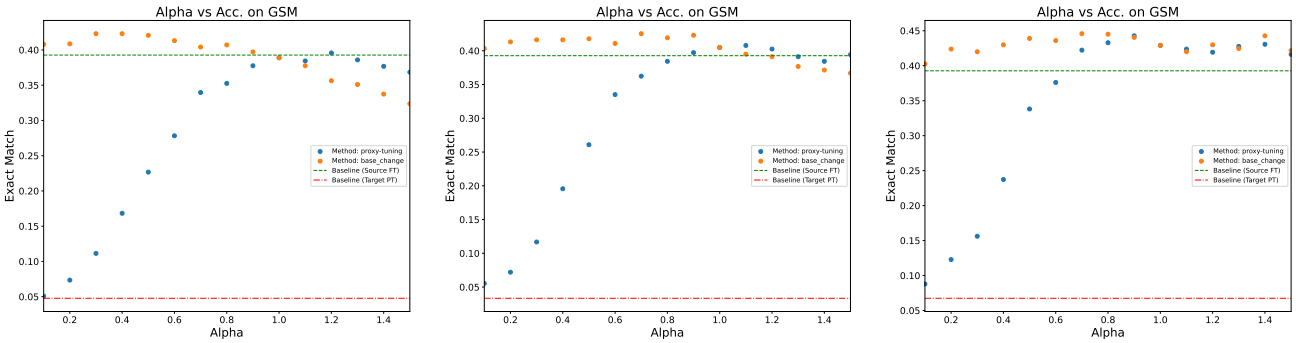
For the math task, we use the LLAMA 2 model, fine-tuned on the GSM training dataset, as the Source PT. The scale parameter α of the base-change for the math task was determined by sampling 119 samples from the GSM test set for validation.

For the code generation tasks, we use the CODE LLAMA - PYTHON model as the Source PT. The scale parameter α of the base-change for code generation tasks was determined by sampling 20 samples from the Codex HumanEval test set for validation.

Using CODE LLAMA - PYTHON-34B as the Target PT, an $\alpha = 0.4$, which showed the best performance on the validation set for math task within the range of 0.1 to 1.5, was selected. For the scale parameter α in EFT/proxy-tuning, we adopted the $\alpha = 1.0$ as proposed in (Liu et al., 2024a). Unless otherwise specified, the same α will be used in all experiments.

B.2. Settings for Image Classification

For the source model, we use a CLIP ViT pre-trained on the LION 400M dataset (Schuhmann et al., 2021) and then fine-tuned for specific image classification tasks. For the target model, if the model size is the same as the source model, we used a version pre-trained on the larger datasets, LION 2B (Schuhmann et al., 2022) and Datacomp 1B (Gadre et al., 2024). If the target model is larger than the source model, it was pre-trained on the same dataset, LION 400M, or on the LION 2B dataset as the source model.



(a) Target PT:CodeLlama-7B (b) Target PT:CodeLlama-13B (c) Target PT:CodeLlama-34B

Figure 2: Alpha values for each method. Source PT: LLAMA 2-7B, Source FT:LLAMA 2-7B finetuned on GSM, Bule: Base-change at prediction. Orange: EFT / Proxy-Tuning. Green: Source FT. Red: Target PT (Zero-shot evaluation).

Method	Source FT	Target PT	EFT/proxy-tuning	Base-change	Direct(LoRA)	Direct(Full FT)
GSM	39.6	6.6	42.6	43.2	32.4*	51.0*
Codex HumanEval	68.9	33.7	65.9	71.8	34.3	79.5*

Table 3: Performance on NLP task. The EFT/proxy-tuning base-changed model use LLAMA 2-7B as Source PT, and LLAMA 2-13B as Target PT. The scores marked with * are cited from (Liu et al., 2024a)

To determine the optimal scale parameter α for adding logits in EFT/proxy-tuning and base-change, we evaluated each α ranging from 0.1 to 1.5 in increments of 0.1 on the validation dataset. Then we report the evaluation results on the test dataset, using the best-performing α on the validation dataset for each downstream task.

C. Analysis of the Effect of the Scale Parameter α.

We examined the effect of the scale parameter alpha. Using CODE LLAMA - PYTHON 7B, 13B, and 34B as the Target PT models, we evaluated EFT/proxy-tuning and base-change on the GSM dataset, varying alpha from 0.0 to 1.5.

As shown in Figure 2, base-change demonstrated performance equal to or better than Source FT for all model sizes when alpha ranged from 0 to 1.0. On the other hand, while EFT/proxy-tuning also showed performance exceeding Source FT around alpha = 1.0, its performance was lower compared to Source FT when alpha was too small or too large.

D. Comparison with Direct Fine-Tuning of the Target Model

We compare the performance of models directly fine-tuned on target tasks with Target PT and base-change. Using the GSM dataset, we prepare and evaluate two models: one with LoRA tuning, labeled as Direct(LoRA), and one with fully fine-tuning, labeled as Direct(Full FT).

Table 3 shows that base-change, while not achieving the performance of a fully fine-tuned target pre-trained model, exceeds the performance of the target model with LoRA tuning. This suggests that base-change is a viable alternative in scenarios where full fine-tuning is impractical. It should be noted that like EFT/proxy-tuning, base-change increases the inference cost compared to using the target model alone.