SELF-DISTILLATION FOR FURTHER PRE-TRAINING OF TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-training a large transformer model on a massive amount of unlabeled data and fine-tuning it on labeled datasets for diverse downstream tasks has proven to be a successful strategy, for a variety of vision and natural language processing tasks. However, direct fine-tuning of the pre-trained model may be suboptimal if there exist large discrepancies across data domains for pre-training and fine-tuning. To tackle this issue, several previous studies have proposed further pre-training strategies, where we continue to pre-train the model on the target unlabeled dataset before fine-tuning. However, all of them solely focus on language models and we empirically find that the continual pre-training does not improve generalization performance on Vision Transformers. In order to tackle this limitation, we propose self-distillation as a domain agnostic regularization for a further pre-training stage. Specifically, we first further pre-train the initial pre-trained model on the target unlabeled data and then consider it as a teacher for self-distillation. Then we take the same initial pre-trained model as a student and enforce its hidden representations to be close to those of the teacher while optimizing the student with a masked auto-encoding objective. We empirically validate the efficacy of self-distillation on a variety of benchmark datasets for image and text classification tasks. Experimentally, we show that our proposed method outperforms all the relevant baselines. Theoretically, we analyze the proposed method with a simplified model to understand how self-distillation for further pre-training can potentially help improve the performance in the downstream tasks.

1 INTRODUCTION

Pre-trained transformer models (Devlin et al., 2019; Brown et al., 2020; Liu et al., 2019; He et al., 2022) have shown to be effective on various vision and natural language processing tasks. The pre-trained models learn general representation from a large volume of unlabeled data so that they generalize well to various downstream tasks when they are fine-tuned on each task with labeled dataset. However, in many of real-world applications, it requires considerable amount of effort to adapt the pre-trained model to specific downstream task domain since there exists a distributional discrepancy between data for pre-training and fine-tuning stage. Moreover, it is difficult to collect a large amount of labeled data for such specific domain, which renders adaptation of the pre-trained model to downstream tasks more challenging.

Several works have proposed to tackle the problem of adaptation of pre-trained models to specific domain. A prevalent approach to handle the issue is *further pre-training* where we continue to update the parameters of the pre-trained model on curated domain specific unlabeled data with self-supervision (Beltagy et al., 2019; Lee et al., 2020) before fine-tuning it on target labeled data as described in Figure 1b. Furthermore, Gururangan et al. (2020) show the effectiveness of further pre-training on target unlabeled data without using any extra data. However, all of the works have focused on pre-trained language models and we empirically find that the further pre-training strategy is not effective for Vision Transformer (Dosovitskiy et al., 2021).

Another line of works (Chen et al., 2020a; Gouk et al., 2021; Aghajanyan et al., 2021) targeting the data scarcity problem is a regularization for fine-tuning a pre-trained model on small amount of labeled data. Intuition behind the regularization is to prevent catastrophic forgetting of the knowledge acquired from pre-training stage by enforcing the distance between the fine-tuned weight and

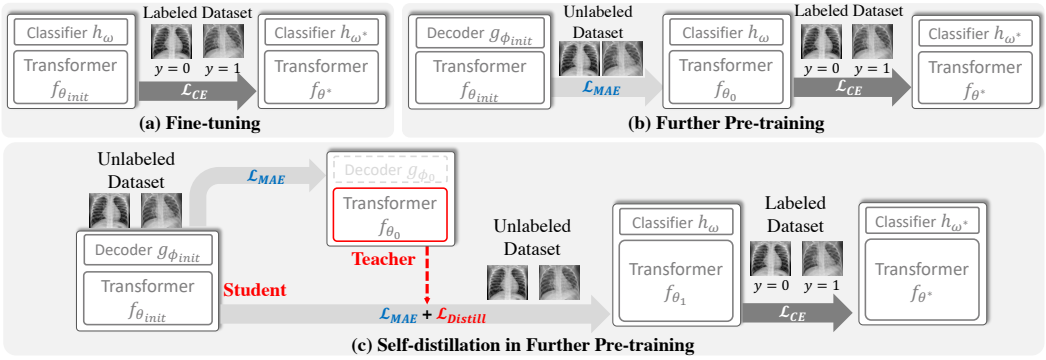


Figure 1: **Concepts.** Comparison between methods adapting pre-trained transformers to the target domain. **(a) Fine-tuning** without any further pre-training. **(b) Further pre-training** and fine-tuning. **(c) Self-distillation** in further pre-training and fine-tuning.

the initial pre-trained weight to be small. The regularization is expected to promote transfer of such knowledge to downstream task for better generalization. However, the regularization hinders adaptation of pre-trained models to downstream tasks if domain of target task is different from that of pre-training, which results in worse generalization than simple fine-tuning strategy.

In order to tackle those limitations already mentioned, we propose *self-distillation* as a regularization for further pre-training on target unlabeled dataset so that we can effectively adapt pre-trained models to downstream task with the limited number of labeled data. Note that we solely focus on masked auto-encoding as objective function for pre-training since it does not depend on any data augmentations, compared to other self-supervised learning methods (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; Zbontar et al., 2021; Chen & He, 2021) which require data augmentations to construct positive pairs for self-supervised learning objective such as contrastive learning.

Specifically, we take the initial model (transformer encoder f_{θ_0} and decoder g_{ϕ_0}) which is pre-trained on a massive amount of unlabeled data from general domain (e.g., images from ImageNet or corpora from Wikipedia) and continue to pre-train the model with masked auto-encoding objective $\mathcal{L}_{\texttt{MAE}}$ on the target unlabeled data. We discard the decoder g_{ϕ_1} and consider the encoder f_{θ_1} as a teacher for self-distillation. Then we take the same initial pre-trained model which is a student and match hidden representations of the student and those of the teacher, while optimizing the student with the masked auto-encoding objective on the target unlabeled data. Finally, we fine-tune the student on the target labeled data for downstream task. We illustrate the overview of our method in Figure 1c.

To verify efficacy of our method, we empirically show that it significantly improves the generalization performance of a pre-trained Vision Transformer and language model RoBERTA (Liu et al., 2019), and outperforms the relevant baselines on various image and text classification datasets. Moreover, we theoretically analyze the proposed method with a simplified model to understand how self-distillation for further pre-training can potentially help improve the generalization performance on the target tasks after fine-tuning.

Our contribution is threefold:

- We propose self-distillation for further pre-training on target unlabeled dataset, where we enforce the hidden representation of the student to be close to that of the teacher while training the student with masked-auto encoding objective.
- We theoretically analyze the proposed method with a simplified model to understand how self-distillation for further pre-training can potentially lead to better generalization performance on downstream tasks.
- We extensively validate our method on various image and text classification datasets with pre-trained transformers and show that ours outperforms the relevant baselines.

2 Related Work

Self-Distillation Knowledge distillation is to transfer knowledge of teacher to student by minimizing a divergence between output distribution of teacher and student (Hinton et al., 2014). When the parameterization of student and teacher is identical, we call it *self-distillation* as a special case of the knowledge distillation. Although there is no new information during self-distillation process, Furlanello et al. (2018) have shown that the student from self-distillation achieves better generalization performance than the teacher. A similar phenomenon has been consistently observed in other works (Yang et al., 2019; Ahn et al., 2019). Mobahi et al. (2020) theoretically analyze how self-distillation induces regularization and reduces overfitting in Hilbert space. However, all of them focus on self-distillation for supervised learning. Instead, we empirically and theoretically show that self-distillation in further pre-training leads to better generalization on downstream tasks after fine-tuning the model with target labeled data.

Further Pre-training Lee et al. (2020); Beltagy et al. (2019); Sun et al. (2020) have shown the success of continual pre-training language model on a large amount of corpora collected from target domain and fine-tuning the model on target labeled dataset. However, it is computationally expensive to further pre-train the model on the large amount of unlabeled text data and it may not be feasible to collect such large scale of unlabeled data on certain domains. Instead, Gururangan et al. (2020) devise a task-adaptive pre-training (TAPT) where we use only target unlabeled data for further pre-training language model before fine-tuning the model on the target labeled data. In order to improve effectiveness of the further pre-training, Kang et al. (2020); Ye et al. (2021) propose learning to mask an input for masked auto-encoding with bilevel optimization, which requires a prohibitive computational cost. However, all of them solely focus on pre-trained language models and we empirically find that naive further pre-training is not effective for Vision Transformers.

Regularization for Fine-tuning There are several works proposing regularization for fine-tuning a pre-trained model. Chen et al. (2020a) propose to modify Adam (Kingma & Ba, 2015) optimizer, called RecAdam, which enforce the fine-tuned model close to the initial pre-trained model by minimizing ℓ_2 distance between fine-tuned and initial pre-trained weight. Similarly, Gouk et al. (2021) project the fine-tuned weight for every gradient descent update such that it lies within the sphere centered on the initial pre-trained weights with the distance induced by the norm of maximum absolute row sums (MARS). Instead of explicitly minimizing the distance, motivated by trust region theory, Aghajanyan et al. (2021) propose to minimize symmetric KL-divergence between model output of an original input and that of the input perturbed with Gaussian noise.

3 Method

3.1 Preliminaries

Problem Statement We assume that we are given $\{\theta_{\texttt{init}}, \phi_{\texttt{init}}\}$ parameters of the neural network $g_{\phi_{\texttt{init}}} \circ f_{\theta_{\texttt{init}}}$ which is pre-trained on a large volume of unlabeled data with masked auto-encoding objective, where $f_{\theta_{\texttt{init}}}$ is an encoder which extracts hidden representation of an input and $g_{\phi_{\texttt{init}}}$ is an decoder reconstructing a masked input. Our goal is to fine-tune the pre-trained model $f_{\theta_{\texttt{init}}}$ with a randomly initialized task specific head h_ω on labeled dataset $\mathcal{D}^{\texttt{tr}} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ of a downstream classification task such that the model generalizes well to unseen test dataset $\mathcal{D}^{\texttt{test}}$. A typical approach is to tackle the problem is empirical risk minimization as follows:

$$\begin{aligned} &\underset{\theta,\omega}{\text{minimize}}\ \mathcal{L}_{\texttt{CE}}(\theta, \omega; \theta_{\texttt{init}}, \mathcal{D}^{\texttt{tr}}) \\ &\mathcal{L}_{\texttt{CE}}(\theta, \omega; \theta_{\texttt{init}}, \mathcal{D}^{\texttt{tr}}) = \frac{1}{|\mathcal{D}^{\texttt{tr}}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}^{\texttt{tr}}} \ell(\mathbf{x}_i, y_i, \theta, \omega) \ \text{ s.t. initializing } \theta \text{ with } \theta_{\texttt{init}} \end{aligned} \tag{1}$$

where ℓ denotes a cross-entropy loss.

Further Pre-training However, the model is prone to overfitting when it is trained on small amount of domain specific labeled data. Gururangan et al. (2020) have shown that further pre-training, where we continue to pre-train the model $g_{\phi_0} \circ f_{\theta_0}$ on the target unlabeled dataset $\mathcal{D}^u = \{\mathbf{x}^{(i)}\}_{i=1}^n$ and then fine-tune it on $\mathcal{D}^{\texttt{tr}}$, is effective for improving generalization performance when there is not enough domain specific labeled data. Note that $\mathcal{D}^u$ is the exactly same as

Algorithm 1 Self-Distillation

Require: Unlabeled dataset $\mathcal{D}^u$, initialization $\theta_{\texttt{init}}, \phi_{\texttt{init}}$, learning rate $\alpha \in \mathbb{R}_+$, round of self-distillation $T' \in \mathbb{N}_+$, masking probability $\gamma \in (0, 1)$ and batch size B.

$\theta_0 \leftarrow \text{FurtherPretrain}(\mathcal{D}^u, \theta_{\texttt{init}}, \phi_{\texttt{init}}, \alpha, \tau, B)$
for all $t \leftarrow 1, \ldots, T'$ **do**
  Initialize $\theta_t \leftarrow \theta_{\texttt{init}}$ and $\phi_t \leftarrow \phi_{\texttt{init}}$
  while not converge **do**
    Sample a mini-batch $\{\mathbf{x}^{(j)}\}_{j=1}^B$ from $\mathcal{D}^u$
    for all $j \leftarrow 1, \ldots, B$ **do**
      Sample a mask $\mathbf{z}^{(j)} \sim p_{\gamma,K}(\mathbf{z})$
      $Z^{(j)} \leftarrow \sum_{k=1}^K z_k^{(j)}$
      Get a masked input $\hat{\mathbf{x}}^{(j)}$ with $\mathbf{z}^{(j)}$
      $\ell_j^1 \leftarrow -\sum_{k=1}^K \frac{z_k^{(j)}}{Z^{(j)}} \log p_{\theta_t,\phi_t}(x_k^{(j)}|\hat{\mathbf{x}}^{(j)})$
      $\ell_j^2 \leftarrow \left\| f_{\theta_t}(\mathbf{x}^{(j)}) - \texttt{StopGrad}(f_{\theta_0}(\mathbf{x}^{(j)})) \right\|_2^2$
    end for
    $\mathcal{L}_1 \leftarrow \frac{1}{B}\sum_{j=1}^B \ell_j^1, \mathcal{L}_2 \leftarrow \frac{1}{B}\sum_{j=1}^B \ell_j^2$
    $\theta_t \leftarrow \theta_t - \alpha \frac{\partial(\mathcal{L}_1+\mathcal{L}_2)}{\partial\theta}|_{\theta=\theta_t}$
    $\phi_t \leftarrow \phi_t - \alpha \frac{\partial\mathcal{L}_1}{\partial\phi}|_{\phi=\phi_t}$
  end while
  $\theta_0 \leftarrow \theta_t$
end for
return $\theta_{T'}$

Algorithm 2 FurtherPretrain

Require: Unlabeled dataset $\mathcal{D}^u$, initialization $\theta_{\texttt{init}}, \phi_{\texttt{init}}$, learning rate $\alpha \in \mathbb{R}_+$, masking probability $\gamma \in (0, 1)$, and batch size B.

Initialize $\theta_0 \leftarrow \theta_{\texttt{init}}$ and $\phi_0 \leftarrow \phi_{\texttt{init}}$
while not converge **do**
  Sample a mini-batch $\{\mathbf{x}^{(j)}\}_{j=1}^B$ from $\mathcal{D}^u$
  for all $j \leftarrow 1, \ldots, B$ **do**
    Sample a mask $\mathbf{z}^{(j)} \sim p_{\gamma,T}(\mathbf{z})$
    $Z^{(j)} \leftarrow \sum_{k=1}^K z_k^{(j)}$
    Get a masked input $\hat{\mathbf{x}}^{(j)}$ with $\mathbf{z}^{(j)}$
    $p_k \leftarrow p_{\theta_0,\phi_0}(x_k^{(j)}|\hat{\mathbf{x}}^{(j)})$
    $\ell_j^1 \leftarrow -\sum_{k=1}^K \frac{z_k^{(j)}}{Z^{(j)}} \log p_k$
  end for
  $\mathcal{L} \leftarrow \frac{1}{B}\sum_{j=1}^B \ell_j^1$
  $\theta_0 \leftarrow \theta_0 - \alpha \frac{\partial\mathcal{L}}{\partial\theta}|_{\theta=\theta_0}$
  $\phi_0 \leftarrow \phi_0 - \alpha \frac{\partial\mathcal{L}}{\partial\phi}|_{\phi=\phi_0}$
end while
return θ_0

$\mathcal{D}^{\texttt{tr}}$ except that we remove the labels $y^{(i)}$. In this work, we focus on the masked auto-encoding (Devlin et al., 2019; He et al., 2022) as pre-training objective function since its generality compared to other self-supervised methods (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; He et al., 2020; Chen & He, 2021) which require well-defined data augmentations to construct positive pairs for self-supervised learning.

Masked Auto-Encoding We briefly describe the masked auto-encoding objective (Liu et al., 2019; He et al., 2022) for a language model such as RoBERTA (Liu et al., 2019) and Vision Transformer (ViT) (Dosovitskiy et al., 2021). Let $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_K^{(i)})$ be a sequence of patches for a image or tokens for a sentence with length K. Then we independently sample a binary mask from Bernoulli distribution with probability γ for each $x_k^{(i)}$, denoted as $\mathbf{z}^{(i)} = (z_1^{(i)}, \ldots, z_K^{(i)})$. If $z_k^{(i)} = 1$, then $x_k^{(i)}$ is replaced with a special "mask" token. Otherwise we use the same x_k for a masked input. Let $\hat{\mathbf{x}}^{(i)} = (\hat{x}_1^{(i)}, \ldots, \hat{x}_K^{(i)})$ be a masked input and let f_θ, g_ϕ be an encoder and decoder, respectively. Then the final objective for masked auto-encoding is defined as follows:

$$\mathcal{L}_{\texttt{MAE}}(\theta, \phi; \mathcal{D}^u) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\mathbf{z}^{(i)} \sim p_{\gamma,T}(\mathbf{z})} \left[-\sum_{k=1}^K \frac{z_k^{(i)}}{Z^{(i)}} \cdot \log p_{\theta,\phi}(x_t^{(i)}|\hat{\mathbf{x}}^{(i)}) \right], \; Z^{(i)} = \sum_{k=1}^K z_k^{(i)} \quad (2)$$

where $p_{\gamma,K}(\mathbf{z})$ denotes a Binomial distribution with its parameters γ for probability that $z_t = 1$ and K for the number of trials. Note that the negative log likelihood is instantiated as cross-entropy loss for language models or mean square error for Vision Transformers. See Appendix B for more detail.

3.2 Self-Distillation for Further Pre-training

Although further pre-training strategy is known to work well for text domain (Gururangan et al., 2020; Lee et al., 2020; Sun et al., 2020), we empirically find that the continual strategy barely improves generalization performance of ViT on image classification tasks. Inspired by the success of self-distillation (Furlanello et al., 2018) for supervised learning, we propose self-distillation as a regularization for further pre-training. Specifically, given a pre-trained model $g_{\phi_{\texttt{init}}} \circ f_{\theta_{\texttt{init}}}$, we first continue to train the model on the target unlabeled data $\mathcal{D}^u$ with the masked auto-encoding objective as described in equation 2 and obtain the encoder f_{θ_0} and decoder g_{ϕ_0}. We discard the decoder and consider the encoder f_{θ_0} as a teacher for self-distillation. Then we take the same pre-trained initial network $g_{\phi_{\texttt{init}}} \circ f_{\theta_{\texttt{init}}}$ which is a student and further pre-train the student with masked

auto-encoding objective but enforcing hidden representation of the student to be close to that of teacher f_{θ_0} as follows:

$$\theta_1, \phi_1 \in \arg\min_{\theta,\phi} \left(\mathcal{L}_{\texttt{MAE}}(\theta, \phi; \mathcal{D}^u) + \mathcal{L}_{\texttt{Distill}}(\theta; \theta_0, \mathcal{D}^u)\right)$$
$$\mathcal{L}_{\texttt{Distill}}\left(\theta; \theta_0, \mathcal{D}^u\right) = \frac{1}{n}\sum_{i=1}^{n} \left\| f_\theta(\mathbf{x}^{(i)}) - \texttt{StopGrad}\left(f_{\theta_0}(\mathbf{x}^{(i)})\right)\right\|_2^2 \tag{3}$$

where θ and ϕ are initialized with the pre-trained parameters $\theta_{\texttt{init}}$ and $\phi_{\texttt{init}}$, respectively and $\texttt{StopGrad}$ denotes the stop-gradient operation which considers its argument as a constant and does not back-propagate through it. As described in Algorithm 1, we can repeat this process to perform multiple rounds of self-distillation ($T' > 1$) where the student of the previous round becomes a teacher and a student is initialized with the pre-trained weights $\theta_{\texttt{init}}$ and $\phi_{\texttt{init}}$ for the next round. We empirically find that the first round of self-distillation plays the most significant role of improving the final generalization performance of downstream tasks. Furthermore, theoretical analysis shows the first round of self-distillation has the largest impact of regularization. Thus, we perform a single round of self-distillation for computational efficiency. After self-distillation, we discard the decoder g_{ϕ_1} and fine-tune the encoder of the student f_{θ_1} along with a randomly initialized task-specific head h_ω by minimizing $\mathcal{L}_{\texttt{CE}}(\theta, \omega, ; \theta_1, \mathcal{D}^{\texttt{tr}})$ as described in equation 1.

4 Theoretical Analysis

In this section, we analyze how the self-distillation affects the final model after fine-tuning in terms of generalization and regularization. This section proves a generalization bound on the supervised loss for our method and shows that the generalization bound strictly decreases as the number of self-distillation increases. Moreover, we show that the self-distillation acts as a regularizer on the distance between the initial weight before further pre-training and the final weight after fine-tuning. The regularization effect is shown to have the largest impact in the first step of self-distillation, which suggests that the first round of self-distillation plays a more significant role on the final performance when compared to other rounds.

We consider the dynamics of the weight vector $w_{t,\tau}$ over time τ of fine-tuning after t rounds of self-distillation, where $w_{0,0}$ is the result of further pretraining, and $w_{t,0} \in \mathrm{minimize}_w\, L_t(w)$ is the result of the self-distillation for $t \in \mathbb{N}_+$ with $L_t(w) = \frac{1}{n}\sum_{i=1}^{n} \|f(x_i, w) - f(x_i, w_{t-1,0})\|_2^2 + \lambda\|w\|_2^2$ for some $\lambda > 0$. After t rounds of self-distillation, we consider the dynamics of fine-tuning time τ via gradient flow (Saxe et al., 2014; Kawaguchi, 2021): $\frac{dw_{t,\tau}}{d\tau} = -\nabla\mathcal{L}(w_{t,\tau})$, with the initialization $w_{t,0}$ obtained by the self-distillation where $\mathcal{L}(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i, y_i)$ with $\ell(w, x, y) = \|f(x, w) - y\|_2^2$ and $y \in \mathbb{R}^p$. In this section, to obtain theoretical insights, we consider the regime of $d > n$ and a simple abstract model, $f(x, w) = W\varphi(x) \in \mathbb{R}^p$, with some nonlinear map φ and the weight matrix $W \in \mathbb{R}^{p\times d}$ where $w = \mathrm{vec}[W^\top] \in \mathbb{R}^{dp}$ and $\varphi(x) \in \mathbb{R}^d$. Here, $\mathrm{vec}[W^\top]$ is a vectorization of the matrix $W^\top$. Fix the fine-tuning time length T as $1 < \tau \le T < \infty$. Since $d > n$, there are infinitely many solutions to the problem of minimizing $\mathcal{L}(w)$. Thus, each of the finite length T and the over-parameterization $d > n$ implies that the initialization $w_{t,0}$ at the fine-tuning phase via self-distillation plays an important role.

Let $\delta > 0$ and $t \in \mathbb{N}_0$. We then define $\mathcal{F}_t = \{\mathcal{A}_t(s) : s \in \mathcal{S}\}$, where $\mathcal{S}$ is a set of all training datasets of size n such that $\mathbb{P}(s \notin \mathcal{S}) \le \delta$ with a training dataset $s \in \mathcal{S}$. For each training dataset $s \in \mathcal{S}$, $\mathcal{A}_t(s) = w_{t,T}$ is the final weight vector of the model after t steps of self-distillation and T time of fine-tuning. Let us define the matrix $\Phi \in \mathbb{R}^{d\times n}$ by $\Phi_{ij} = \varphi(x_j)_i$. We assume that Φ is of full rank; i.e., $\mathrm{rank}(\Phi) = n$ since $d \ge n$. This is typically satisfied since if $\mathrm{rank}(\Phi) < n$, there are some redundancy in the rows of the matrix Φ. Denote by $[I_p \otimes \Phi] \in \mathbb{R}^{dp\times np}$ the Kronecker product of the identity matrix $I_p \in \mathbb{R}^{p\times p}$ and the matrix Φ. We write its singular value decomposition by $[I_p \otimes \Phi] = U\Sigma V^\top$ where $U = [u_1, u_2 \ldots, u_{dp}] \in \mathbb{R}^{dp\times dp}$ contains the left-singular vectors $u_i \in \mathbb{R}^{dp}$ for $i \in \{1, \ldots, dp\}$ and $\Sigma \in \mathbb{R}^{dp\times np}$ is a rectangular diagonal matrix with $\Sigma_{ii} = \sigma_i \in \mathbb{R}_{\ge 0}$ for $i \in \{1, \ldots, np\}$ and $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_{np} \ge 0$. Define M to be an upper bound on the loss as $\ell(w, x, y) \le M$ for all $(w, x, y) \in \mathcal{F}_t \times \mathcal{X} \times \mathcal{Y}$. Define R to be an upper bound on the expected norm of the features as $\mathbb{E}_x\|\varphi(x)\|_2^2 \le R$. We assume that $w_{0,0} \ne 0$; if $w_{0,0} = 0$, then the target function in the self-distillation phase is always zero as $f(x_i, w_{0,0}) = 0$ for all i, which is unlikely

the case in practice. We define $w_{\text{init}} \in \mathbb{R}^{dp}$ to be the weight before further pre-training and define $Y = \text{vec}[[y_1, \ldots, y_n]^\top] \in \mathbb{R}^{np}$.

The following theorem shows that the generalization bound on the supervised loss $\ell(w_{t,T}, x, y)$ of the fine-tuning phase strictly decreases as we increase the number t of self-distillation rounds in the pre-training phase:

Theorem 1. *There exists a constant c (that only depends on M) such that with probability at least $1 - \delta$, the following holds:*

$$\mathbb{E}_{x,y}[\ell(w_{t,T}, x, y)] \leq \frac{1}{n} \sum_{i=1}^{n} \ell(w_{t,T}, x_i, y_i) + \zeta(t) \sqrt{\frac{4c^2 R^2 p}{n}} + M \sqrt{\frac{\ln(2/\delta)}{2n}}, \tag{4}$$

where the function $\zeta(t)$ is strictly decreasing in $t \in \mathbb{N}_0$.

The proofs of all results in this section are presented in Appendix A. Moreover, the following theorem shows that the tight upper bound on the distance between the initial weight w_{init} and the final weight $w_{t,T}$ after T steps of fine-tuning (i.e., $\|w_{\text{init}} - w_{t,T}\|_2$) strictly decreases as the number t of self-distillation rounds increases:

Theorem 2. *There is a function $\psi : t \mapsto \psi(t)$ such that (1) $\|w_{\text{init}} - w_{t,T}\|_2 = \psi(t)$ for some $(w_{\text{init}}, Y) \in \mathbb{R}^{dp} \times \mathbb{R}^{np}$, (2) $\|w_{\text{init}} - w_{t,T}\|_2 \leq \psi(t)$ for all $(w_{\text{init}}, Y) \in \mathbb{R}^{dp} \times \mathbb{R}^{np}$, (3) the function $\psi(t)$ is strictly decreasing in $t \in \mathbb{N}_0$, (4) the function $\psi(t)$ can be decomposed to $\psi(t) = \sqrt{c_1 + \psi_1(t) + \mathbb{1}\{t = 0\}\mathcal{B}} + c_2$ with constants $c_1, c_2 \geq 0$ in t where $\psi_1(t)$ is strictly decreasing in t and $\mathcal{B} = \sum_{i=np+1}^{dp} (u_i^\top w_{0,0})^2 \geq 0$.*

Theorem 2 shows that the self-distillation acts as a regularizer on the distance between the initial weight w_{init} and the final weight $w_{t,T}$. Since the Rademacher complexity of a set of vectors is invariant to a shift by a constant vector, this distance has been shown to control the generalization bound in previous papers in various models and settings, including deep neural networks (Bartlett & Mendelson, 2002; Bartlett et al., 2017). This suggests that the self-distillation helps generalization via a regularization effect on the distance. Moreover, the first step of self-distillation is expected to have the largest impact based on Theorem 2 since Theorem 2 shows that we can completely remove the unnecessary component $\mathcal{B}$ of $w_{0,0}$ in the first step of self-distillation. We have verified these theoretical predictions in the experiments where we show the correlation between the improvement via self-distillation and the distance that appeared in the generalization bound in the previous paper (Bartlett et al., 2017).

5 Experiment

Dataset For image classification problem, we use six datasets — FGVC Aircraft (Aircraft) (Maji et al., 2013), Caltech UCSD Birds 200 (CUB) (Wah et al., 2011), Chest X-ray (Kermany et al., 2018), Describable Textures Dataset (DTD) (Cimpoi et al., 2014), Stanford Dogs (Khosla et al., 2011), and Oxford 102 Flower (Nilsback & Zisserman, 2008). For text classification problem, we use four datasets — Chemprot (Kringelum et al., 2016), ACL-ARC (Jurgens et al., 2018), SCIERC (Luan et al., 2018), and Twitter-Emotion (Mohammad et al., 2018). Please see Appendix C for more detail.

Implementation Detail For image classification problem, we use Vision Transformer pre-trained on unlabeled ImageNet dataset with masked auto-encoding (He et al., 2022) and fine-tune it on the downstream task with AdamW optimizer (Loshchilov & Hutter, 2019) for 10,000 steps with batch size 32. Regarding TAPT and self-distillation, we continue to pre-train the model for 20,000 steps with batch size 64. We evaluate the Vision Transformers with accuracy. For text classification, following the experimental setup from Gururangan et al. (2020), we use RoBERTA (Liu et al., 2019) as a backbone network and fine-tune it on the target labeled dataset with AdamW optimizer for 10 epochs with batch size 32. In terms of further pre-training for TAPT and self-distillation, we further pre-train RoBERTA for 100 epochs with batch size 128. We evaluate the models with macro F1 for SCIERC, ACL-ARC dataset, and Twitter-Emotion, and micro F1 for Chemprot dataset.

Baselines We compare our method against following relevant baselines targeting for fine-tuning pre-trained models. All the models are initialized with the pre-trained weights $\theta_{\texttt{init}}$ and $\phi_{\texttt{init}}$.

Table 1: Average and standard deviation of accuracy with 5 runs for image classification datasets.

Method	Aircraft	CUB	Chest X-ray	DTD	Dogs	Flower
Fine-tuning	72.33 ± 1.13	55.55 ± 0.54	77.15 ± 0.52	67.56 ± 0.52	62.53 ± 0.57	88.78 ± 0.65
RecAdam	70.76 ± 1.25	55.22 ± 1.29	77.29 ± 1.32	67.59 ± 1.03	61.65 ± 0.92	88.97 ± 0.44
MARS	72.74 ± 0.57	55.35 ± 0.73	77.28 ± 1.80	66.79 ± 0.90	62.24 ± 0.96	87.93 ± 1.21
R3F	72.95 ± 0.46	55.91 ± 0.79	76.86 ± 0.97	65.32 ± 1.25	62.15 ± 0.48	88.92 ± 0.78
TAPT	73.38 ± 0.64	55.72 ± 0.46	77.79 ± 2.06	65.55 ± 1.12	62.34 ± 0.39	88.63 ± 0.35
Self-Distillation	$\mathbf{74.37 \pm 0.43}$	$\mathbf{58.06 \pm 0.90}$	$\mathbf{79.68 \pm 1.05}$	$\mathbf{68.51 \pm 0.51}$	$\mathbf{63.55 \pm 0.39}$	$\mathbf{90.28 \pm 0.44}$

Table 2: Average and standard deviation of F1 score with 5 runs for text classification datasets.

Method	SCIERC	ACL-ARC	Chemprot	Twitter-Emotion
Fine-tuning	76.63 ± 2.06	64.09 ± 4.13	80.59 ± 1.15	77.61 ± 0.83
RecAdam	79.45 ± 1.92	59.70 ± 2.68	$\mathbf{82.73 \pm 0.28}$	78.26 ± 0.88
MARS	74.69 ± 1.25	53.57 ± 6.64	80.18 ± 0.92	77.48 ± 1.69
R3F	75.61 ± 2.49	60.13 ± 2.55	79.25 ± 2.16	77.79 ± 0.81
TAPT	80.32 ± 1.25	69.73 ± 2.40	82.33 ± 0.46	78.71 ± 0.40
Self-Distillation	$\mathbf{81.79 \pm 0.75}$	$\mathbf{73.17 \pm 2.19}$	$\mathbf{82.87 \pm 0.35}$	$\mathbf{79.77 \pm 0.79}$

1. **Fine-tuning**: The model fine-tuned on target labeled dataset $\mathcal{D}^{\texttt{tr}}$ without any further pre-training or regularization except dropout and weight decay.
2. **RecAdam (Chen et al., 2020a)**: The model trained with RecAdam optimizer which is a variant of Adam optimizer (Kingma & Ba, 2015) and additionally penalize L_2 distance between the fine-tuned and the initial pre-trained weight.
3. **MARS (Gouk et al., 2021)**: The model trained to minimize the cross-entropy loss along with the regularization projecting the fine-tuned weight to lie within a sphere centered on the initial pre-trained weights. For each layer, the distance induced by Maximum Absolute Row Sum (MARS) matrix norm ($\max_j \sum_{i=1} |W_{j,i} - U_{j,i}|$), is used for the regularization.
4. **R3F (Aghajanyan et al., 2021)**: The model trained to minimize the cross-entropy loss as well as symmetric KL-divergence between softmax output of the original input and that of the input perturbed with Gaussian noise.
5. **TAPT (Gururangan et al., 2020)**: Task adaptive pre-training where we further pre-train the model on unlabeled target dataset $\mathcal{D}^u$ with masked auto-encoding objective and fine-tune it on the target labeled dataset $\mathcal{D}^{\texttt{tr}}$.
6. **Self-Distillation**: This is our model which is further pre-trained on unlabeled target dataset $\mathcal{D}^u$ with equation 3 and fine-tuned on the target labeled dataset $\mathcal{D}^{\texttt{tr}}$.

5.1 Main Results

As shown in Table 1, self-distillation consistently outperforms all the regularization methods and the further pre-training method TAPT on image datasets. Notably, our method significantly improves the performance on Chest X-ray dataset consisting of grey-scaled images for diagnosis of pneumonia. In addition, self-distillation effectively tackles Flower dataset which contains only 2,040 labeled examples. In contrast, the other baselines do not show consistent improvement across all the image datasets. For instance, TAPT is effective for Aircraft dataset, but significantly degrades the test accuracy on DTD dataset. For regularization methods such as RecAdam, MARS, and R3F, they barely improve generalization performance on most datasets or underperform the simple Fine-tuning method on certain datasets. This empirical evidence supports that the regularizations enforcing the fine-tuned models close to the initial pre-trained weight is not effective for adapting a pre-trained model to the target datasets of specific domains.

Furthermore, as shown in Table 2, we provide additional experimental results for text classification tasks. Again, self-distillation significantly outperforms most of the baselines across all three datasets, except RecAdam in Chemprot dataset. Contrast to the previous experiment, the model TAPT which continues to pre-train on target unlabeled dataset improves test F1 score of the simple Fine-tuning method, yet it still underperforms our model. For regularization methods — RecAdam, MARS, and R3F, they do not achieve consistent improvement across all three datasets. RecAdam moderately improves F1 score on SCIERC dataset and Chemprot, but significantly degrades the

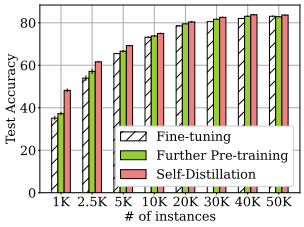


Figure 2: Accuracy with varying the number of training data.

Table 3: Ablation on CUB and SCIERC.

Model	CUB	SCIERC
Full Model	**58.06** $\pm$ **0.90**	**81.79** $\pm$ **0.75**
w/o $\mathcal{L}_{\texttt{MAE}}$	57.60 $\pm$ 0.81	80.66 $\pm$ 0.62
w/o $\mathcal{L}_{\texttt{Distill}}$	55.72 $\pm$ 0.46	80.32 $\pm$ 1.25
Further Pre-train$\times$2	53.41 $\pm$ 0.75	80.52 $\pm$ 0.98
Prediction-Matching	55.27 $\pm$ 1.07	81.09 $\pm$ 1.07
Weight-Matching (ℓ_2)	53.70 $\pm$ 0.91	80.54 $\pm$ 1.21
Weight-Matching (MARS)	54.82 $\pm$ 0.60	80.95 $\pm$ 0.71

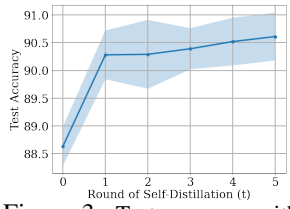


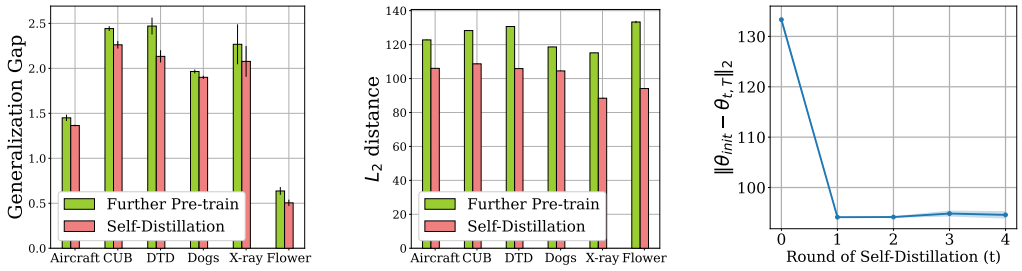
Figure 3: Test accuracy with varying self-distillation round.

generalization performance on ACL-ARC dataset. Both MARS and R3F show poor performance on SCIERC and ACL-ARC dataset, and their performance slightly is worse than Fine-tuning method on Chemprot dataset.

Result for Low Resource Data We further perform experiments to show how self-distillation effectively handles low resources of labeled data. Given a full CIFAR-100 dataset (Krizhevsky et al., 2009) which contains 50,000 training pairs of an image and corresponding label, we plot the test accuracy of each model by varying the number of training instances. Note that we also reduce the number of unlabeled images used for further pre-training or self-distillation. As shown in Figure 2, self-distillation consistently improves the generalization performance of both Fine-tuning model and the one which is further pre-trained on the images from the CIFAR-100 dataset. Notably, the gain by self-distillation becomes larger when the models are trained with an extremely small number of instances. For example, self-distillation achieves 13% and 6% improvement of test accuracy compared to the model with simple fine-tuning when there are 1,000 and 2,500 labeled examples, respectively. These empirical results verify that self-distillation can effectively adapt the pre-trained model to the target dataset even if there are extremely small amounts of labeled data.

Ablation Study We perform ablation study to verify the effectiveness of each component of self-distillation. In Table 3, we show empirical results on both CUB dataset and SCIERC data set while removing or replacing various components of self-distillation. Firstly, we remove masked auto-encoding objective $\mathcal{L}_{\texttt{MAE}}$ and train the model with only distillation loss $\mathcal{L}_{\texttt{Distill}}$ before fine-tuning. On image dataset CUB, it does not make a significant difference, however, removing the masked auto-encoding objective degrades the generalization performance of language model on text classification dataset SCIERC. Alternatively, we remove the distillation loss $\mathcal{L}_{\texttt{Distill}}$ in equation 3, which results in TAPT model. Furthermore, we continue to pre-train TAPT model for twice longer steps than original TAPT model, denoted as Further Pre-train$\times$2, to show that higher test accuracy of self-distillation is not a consequence of longer pre-training. Both of the models significantly underperform self-distillation, which shows the effectiveness of the self-distillation loss. Lastly, we perform experiments for variants of distillation loss $\mathcal{L}_{\texttt{Distill}}$ in equation 3. Instead of matching representation of teacher and student, we enforce the reconstruction of masked inputs by teacher and student to be consistent, i.e., minimizing $\|g_{\phi_0} \circ f_{\theta_0}(\hat{\mathbf{x}}) - g_{\phi_1} \circ f_{\theta_1}(\hat{\mathbf{x}})\|_2^2$ for ViT or $\sum_{t=1}^{T} D_{\mathrm{KL}}\left(p_{\theta_1,\phi_1}(x_t|\hat{\mathbf{x}}) \parallel p_{\theta_0,\phi_0}(x_t|\hat{\mathbf{x}})\right)$ for RoBERTA, denoted as Prediction-Matching. Furthermore, we replace the distillation loss with the one minimizing L_2 or MARS distance between the parameters of student and teacher, denoted as Weight-Matching. As shown in Table 3, minimizing distance in the weight space is not effective compared to minimizing distance between hidden representation of student and teacher.

Multi-Round of Self-Distillation Lastly, we empirically show that the first round of self-distillation plays the most significant role of improving generalization performance. Specifically, we fine-tune each model after t round of self-distillation and plot test accuracy on Oxford 102 Flower dataset, where 0 round of self-distillation ($t = 0$) denotes the model with further pre-training, TAPT. As shown in Figure 3, the first round of self-distillation significantly improves test accuracy of the model with further pre-training and the gain by self-distillation becomes marginal after the first round. Considering extra computational cost and marginal improvement of multi-round self-distillation, we perform a single round of self-distillation for all the experiments.



(a) (b) (c)

Figure 4: **(a) Generalization gap**: Difference between supervised test loss and training loss. **(b) Effect of self-distillation on distance:** Distance between the initial pre-trained weights and the final fine-tuned weights of further pre-training and self-distillation. **(c) Effect of multi-round self distillation:** Distance between the initial pre-trained weights and the final fine-tuned weights for each round of self-distillation t.

5.2 Further Analysis

In this subsection, we present numerical experiments to analyze why self-distillation can potentially help improve generalization performance of downstream tasks compared to further pre-training and empirically show that Theorem 1 and 2 can be extended to deep neural networks, transformers.

(a) Generalization gap: In Figure 4a, we plot generalization gap, which is test loss minus training loss on each labeled dataset, of self-distillation and further pre-training method. Self-distillation improves the generalization gap of further pre-training method across all the datasets. It is consistent with Theorem 1 showing that self-distillation with a simplified model strictly decreases the generalization bound on the supervised loss of the fine-tuning stage.

(b) Effect of self-distillation on distance: In order to empirically validate Theorem 2 about regularization by self-distillation on L_2 distance between initial pre-trained and the final fine-tuned weight, we plot the distance obtained from self-distillation and further pre-training. As shown in Figure 4b, self-distillation consistently decreases the distance and the reduced distance correlates with better generalization gap. These empirical results also confirms the connection between the L_2 distance from the initialization and generalization (Nagarajan & Kolter, 2019).

(c) Effect of multi-round self-distillation: Lastly, we empirically verify part of Theorem 2 which shows that the first round of self-distillation plays the most critical role of regularization on the L_2 distance between the initial pre-trained weight $\theta_{\texttt{init}}$ and the final weight $\theta_{t,T}$ denoted as the parameter after t round of self-distillation and T steps of gradient descent for fine-tuning. As shown in Figure 4c, self-distillation significantly decreases the distance at the first round $(t = 1)$ and the regularization effect on the distance diminishes afterwards, where 0 round of self-distillation $(t = 0)$ denotes the model with further pre-training but without self-distillation.

6 Conclusion

In order to effectively adapt pre-trained transformers to a target domain, we proposed self-distillation for further pre-training. Specifically, we first took the initial pre-trained transformer and continued to pre-train it with masked auto-encoding objective on the target unlabeled dataset and considered the model as a teacher for self-distillation. Then we took the same initial pre-trained model which was a student and enforced the hidden representation of the student to be close to that of the teacher while optimizing the student with masked auto-encoding objective on the target unlabeled dataset. Finally, we fine-tuned the student on the labeled data from the target task. We empirically validated our method on various image and text classification benchmark datasets, showing that self-distillation consistently improved the generalization performance compared to the relevant baselines we considered. Lastly, we provided theoretical analysis of the proposed method with a simplified model to understand how self-distillation for further pre-training can potentially help improving generalization performance in the downstream tasks.

REFERENCES

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*, 2021.

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3615–3620, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7870–7881, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018.

Henry Gouk, Timothy Hospedales, and Massimiliano Pontil. Distance-based regularisation of deep networks for fine-tuning. In *International Conference on Learning Representations*, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS 2014 Deep Learning Workshop*, 2014.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406, 2018.

Minki Kang, Moonsu Han, and Sung Ju Hwang. Neural mask generator: Learning to generate adaptive word maskings for language model adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6102–6120, 2020.

Kenji Kawaguchi. On the theory of implicit deep learning: Global convergence with implicit layers. In *International Conference on Learning Representations*, 2021.

Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. Chemprot-3.0: a global chemical biology diseases mapping. *Database*, 2016, 2016.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *arXiv preprint*, 2009.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3219–3232, 2018.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pp. 3–17. Springer, 2016.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pp. 1–17, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for open-vocabulary image classification. *arXiv preprint arXiv:2111.10050*, 2021.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8968–8975, 2020.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-UCSD birds-200-2011. Technical report, California Institute of Technology, 2011.

Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5628–5635, 2019.

Qinyuan Ye, Belinda Z Li, Sinong Wang, Benjamin Bolte, Hao Ma, Wen-tau Yih, Xiang Ren, and Madian Khabsa. On the influence of masking policies in intermediate pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7190–7202, 2021.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

APPENDIX

A PROOFS

We also define the model output vector $f_t \in \mathbb{R}^{n\times p}$ by $(f_t)_{ij} = f(x_i, w_t)_j$. For example, f_0 is the initial teacher label matrix. Let $[n] = \{1, \dots, n\}$. Denote the rank of $[I_p \otimes \Phi]$ by $r = \mathrm{rank}([I_p \otimes \Phi]) \leq np$. Define $\tilde{U} = [u_1, u_2 \dots, u_r] \in \mathbb{R}^{dp\times r}$ and $\mathbf{P}_r = I - \tilde{U}\tilde{U}^\top$, which is the projection matrix onto the null space of $\tilde{U}^\top$. We first prove the following lemma, which will be used in the proofs of Theorem 1 and Theorem 2 later:

Lemma 1. *For any $t \in \mathbb{N}_0$, $w_{t,0} = \sum_{i=1}^r \alpha_{i,t}\tilde{u}_i + \mathbb{1}\{t=0\}v$, where $\alpha_{i,t} = \frac{1}{\sigma_i}\left(\frac{1}{1+(n\lambda/\sigma_i^2)}\right)^t \in \mathbb{R}$, $\tilde{u}_i = \tilde{y}_i u_i \in \mathbb{R}^{dp}$, $\tilde{y}_i = (V^\top \mathrm{vec}[f_0])_i \in \mathbb{R}$, and $v = \mathbf{P}_r w_{0,0}$.*

Proof of Lemma 1. Define $w_t \coloneqq w_{t,0}$ for $t \in \mathbb{N}_+$. The necessary condition on the solution w_t at step t is that $\nabla L(w_t) = 0$. Thus, by solving $\nabla L(w_t) = 0$ for w_t, we have that $w_t = ([I_p \otimes \Phi][I_p \otimes \Phi]^\top + n\lambda I)^{-1}[I_p \otimes \Phi]\,\mathrm{vec}[f_{t-1}]$. By using the singular value decomposition $[I_p \otimes \Phi] = U\Sigma V^\top$, since $UU^\top = U^\top U = I$ and $V^\top V = I$, we have that

$$\begin{aligned} w_t = (U\Sigma\Sigma^\top U^\top + n\lambda I)^{-1}U\Sigma V^\top \mathrm{vec}[f_{t-1}] &= (U(\Sigma\Sigma^\top + n\lambda I)U^\top)^{-1}U\Sigma V^\top \mathrm{vec}[f_{t-1}] \\ &= U(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma V^\top \mathrm{vec}[f_{t-1}]. \end{aligned}$$

Therefore, $w_t = U(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma V^\top \mathrm{vec}[f_{t-1}]$. Using this and $[I_p \otimes \Phi] = U\Sigma V^\top$,

$$\begin{aligned} \mathrm{vec}[f_t] = \mathrm{vec}[\Phi^\top W_t^\top I_p] = [I_p \otimes \Phi]^\top w_t &= [I_p \otimes \Phi]^\top U(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma V^\top \mathrm{vec}[f_{t-1}] \\ &= V\Sigma^\top(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma V^\top \mathrm{vec}[f_{t-1}]. \end{aligned}$$

Therefore, $\mathrm{vec}[f_t] = VAV^\top \mathrm{vec}[f_{t-1}]$ where $A = \Sigma^\top(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma$. Repeating this process for $\mathrm{vec}[f_{t-1}]$, since $V^\top V = I$,

$$\mathrm{vec}[f_t] = VAV^\top VAV^\top \cdots VAV^\top \mathrm{vec}[f_0] = VA^tV^\top \mathrm{vec}[f_0].$$

Plugging this equation of $\mathrm{vec}[f_{t-1}] = VA^{t-1}V^\top \mathrm{vec}[f_0]$ into the equation of $w_t = U(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma V^\top \mathrm{vec}[f_{t-1}]$, we have that

$$w_t = U(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma V^\top VA^{t-1}V^\top \mathrm{vec}[f_0] = UBA^{t-1}V^\top \mathrm{vec}[f_0]$$

where $B = (\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma$. Here, we can rewrite the matrix $B \in \mathbb{R}^{dp\times np}$ as

$$B = \begin{bmatrix} \bar{B} \\ \mathbf{0}_{(dp-np)\times np} \end{bmatrix}$$

where $\mathbf{0}_{(dp-np)\times np}$ is the $(dp-np)$ by np matrix with all entries being zero, and $\bar{B} \in \mathbb{R}^{np\times np}$ is a diagonal matrix defined by

$$\bar{B}_{ii} \coloneqq \sigma_i(\sigma_i^2 + n\lambda)^{-1}.$$

Using this B in the above equation of $w_t = UBA^{t-1}V^\top \mathrm{vec}[f_0]$,

$$\begin{aligned} w_t &= U\begin{bmatrix} \bar{B} \\ \mathbf{0}_{(dp-np)\times np} \end{bmatrix} A^{t-1}V^\top \mathrm{vec}[f_0] \\ &= [u_1 \quad u_2 \quad \cdots \quad u_{dp}]\begin{bmatrix} \bar{B}A^{t-1} \\ \mathbf{0}_{(dp-np)\times np} \end{bmatrix} V^\top \mathrm{vec}[f_0] \\ &= \bar{U}\bar{B}A^{t-1}V^\top \mathrm{vec}[f_0] \end{aligned}$$

where $\bar{U} = [u_1, u_2 \dots, u_{np}] \in \mathbb{R}^{dp\times np}$. Since the matrix $A = \Sigma^\top(\Sigma\Sigma^\top + n\lambda I)^{-1}\Sigma \in \mathbb{R}^{np\times np}$ is a diagonal matrix with its entry being $A_{ii} = \sigma_i^2(\sigma_i^2 + n\lambda)^{-1}$, this can be further simplified as

$$
\begin{aligned}
w_t &= \bar{U}\bar{B}A^{t-1}V^\top \operatorname{vec}[f_0] \\
&= \bar{U}\begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+n\lambda} & & \\ & \ddots & \\ & & \frac{\sigma_{np}}{\sigma_{np}^2+n\lambda} \end{bmatrix}\begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+n\lambda} & & \\ & \ddots & \\ & & \frac{\sigma_{np}^2}{\sigma_{np}^2+n\lambda} \end{bmatrix}^{t-1}\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{np} \end{bmatrix} \\
&= \begin{bmatrix} u_1 & u_2 & \cdots & u_{np} \end{bmatrix}\begin{bmatrix} \sigma_1(\sigma_1^2+n\lambda)^{-1}(\sigma_1^2(\sigma_1^2+n\lambda)^{-1})^{t-1}\tilde{y}_1 \\ \sigma_2(\sigma_2^2+n\lambda)^{-1}(\sigma_2^2(\sigma_2^2+n\lambda)^{-1})^{t-1}\tilde{y}_2 \\ \vdots \\ \sigma_{np}(\sigma_{np}^2+n\lambda)^{-1}(\sigma_{np}^2(\sigma_{np}^2+n\lambda)^{-1})^{t-1}\tilde{y}_{np} \end{bmatrix} \\
&= \sum_{i=1}^{np} \sigma_i(\sigma_i^2+n\lambda)^{-1}(\sigma_i^2(\sigma_i^2+n\lambda)^{-1})^{t-1}\tilde{y}_i u_i \\
&= \sum_{i=1}^{r} \sigma_i(\sigma_i^2+n\lambda)^{-1}(\sigma_i^2(\sigma_i^2+n\lambda)^{-1})^{t-1}\tilde{y}_i u_i
\end{aligned}
$$

where the last line follows from the fact that $\sigma_i(\sigma_i^2+n\lambda)^{-1}(\sigma_i^2(\sigma_i^2+n\lambda)^{-1})^{t-1} = 0$ for all $i > r$. Since $\sigma_i(\sigma_i^2+n\lambda)^{-1}(\sigma_i^2(\sigma_i^2+n\lambda)^{-1})^{t-1} = \sigma_i^{2t-1}(\sigma_i^2+n\lambda)^{-t} = \frac{1}{\sigma_i}(\frac{\sigma_i^2}{\sigma_i^2+n\lambda})^t$ for $i \leq r$, this implies that

$$
w_t = \sum_{i=1}^{r} \frac{1}{\sigma_i}\left(\frac{\sigma_i^2}{\sigma_i^2+n\lambda}\right)^t \tilde{y}_i u_i = \sum_{i=1}^{r} \frac{1}{\sigma_i}\left(\frac{1}{1+(n\lambda/\sigma_i^2)}\right)^t \tilde{y}_i u_i
$$

Since $t \in \mathbb{N}_+$ was arbitrary, this holds for any $t \in \mathbb{N}_+$. This proves the first statement of the theorem for any $t \in \mathbb{N}_+$. For $t = 0$, since

$\tilde{y}_i = (V^\top \operatorname{vec}[f_0])_i = (V^\top [I_p \otimes \Phi]^\top w_{0,0})_i = (V^\top V \Sigma^\top U^\top w_{0,0})_i = (\Sigma^\top U^\top w_{0,0})_i = \sigma_i u_i^\top w_{0,0}$,

we have that

$$
w_0 = \sum_{i=1}^{r} \frac{1}{\sigma_i}\left(\frac{1}{1+(n\lambda/\sigma_i^2)}\right)^t \tilde{y}_i u_i = \left(\sum_{i=1}^{r} u_i u_i^\top\right) w_{0,0} = \tilde{U}\tilde{U}^\top w_{0,0}.
$$

Thus,

$$
w_{0,0} = w_0 + (I - \tilde{U}\tilde{U}^\top) w_{0,0}.
$$

This completes the first statement of the theorem for any $t \in \mathbb{N}_0$.

□

A.1 Proof of Theorem 1

Proof. Define $Z \coloneqq [I_p \otimes \Phi]^\top \in \mathbb{R}^{\bar{n}\times\bar{d}}$ where $\bar{n} = np$ and $\bar{d} = dp$. Then,

$$
\mathcal{L}(w) = \frac{1}{n}\sum_{i=1}^{n} \|f(x_i, w) - y_i\|_2^2 = \frac{1}{n}\|Zw - Y\|_2^2
$$

where $Y = \operatorname{vec}[[y_1, \ldots, y_n]^\top] \in \mathbb{R}^{\bar{n}}$. Since $\nabla\mathcal{L}(w_{t,\tau}) = \frac{1}{2n}Z^\top(Zw_{t,\tau} - Y)$,

$$
\frac{dw_{t,\tau}}{d\tau} = -\frac{1}{2n}Z^\top(Zw_{t,\tau} - Y)
$$

Since $\operatorname{rank}(\Phi) = n$ and $d \geq n$, we have $\operatorname{rank}(Z) = \bar{n}$ by the property of the Kronecker product with the identity matrix. Since $\operatorname{rank}(Z) = \bar{n}$, there exists $v \in \mathbb{R}^{\bar{d}}$ such that $Y = Zv$. Thus,

$$
\begin{aligned}
\frac{dw_{t,\tau}}{d\tau} &= -\frac{1}{2n}Z^\top(Zw_{t,\tau} - Zv) \\
&= -\frac{1}{2n}Z^\top Z(w_{t,\tau} - v) \\
&= -\frac{1}{2n}Z^\top Z(w_{t,\tau} - v).
\end{aligned}
$$

Since $Z^\top = U\Sigma V^\top$, we have $Z^\top Z = U\Sigma\Sigma^\top U^\top = \sum_{i=1}^{\bar{n}} \sigma_i^2 u_i u_i^\top$. Thus,

$$\frac{dw_{t,\tau}}{d\tau} = -\frac{1}{2n}\left(\sum_{i=1}^{\bar{n}} \sigma_i^2 u_i u_i^\top\right)(w_{t,\tau} - v) = -\frac{1}{2n}\sum_{i=1}^{\bar{n}} \sigma_i^2 u_i u_i^\top (w_{t,\tau} - v).$$

Since the columns of U forms the basis of $\bar{d}$ and $w, v \in \mathbb{R}^{\bar{d}}$, we can write $w_{t,\tau} = \sum_{k=1}^{\bar{d}} c_k^{(t,\tau)} u_k$ and $v = \sum_{k=1}^{\bar{d}} q_k u_k$ for some $c_k^{(t,\tau)}$ and q_k. Thus,

$$\begin{aligned}
\frac{dw_{t,\tau}}{d\tau} &= -\frac{1}{2n}\sum_{i=1}^{\bar{n}} \sigma_i^2 u_i u_i^\top \sum_{k=1}^{\bar{d}} (c_k^{(t,\tau)} - q_k) u_k \\
&= -\frac{1}{2n}\sum_{i=1}^{\bar{n}}\sum_{k=1}^{\bar{d}} \sigma_i^2 (c_k^{(t,\tau)} - q_k) u_i u_i^\top u_k \\
&= -\frac{1}{2n}\sum_{i=1}^{\bar{n}} \sigma_i^2 (c_i^{(t,\tau)} - q_i) u_i.
\end{aligned}$$

Using $w_{t,\tau} = \sum_{k=1}^{\bar{d}} c_k^{(t,\tau)} u_k$ for the right-hand side too, we have that

$$\frac{d}{d\tau}\sum_{i=1}^{\bar{d}} c_i^{(t,\tau)} u_i = -\frac{1}{2n}\sum_{i=1}^{\bar{n}} \sigma_i^2 (c_i^{(t,\tau)} - q_i) u_i.$$

This implies that for all $i \in \{1, \ldots, \bar{n}\}$,

$$\frac{d}{d\tau} c_i^{(t,\tau)} = -\sigma_i^2 (c_i^{(t,\tau)} - q_i),$$

and $\frac{d}{d\tau} c_i^{(t,\tau)} = 0$ for all $i \notin \{1, \ldots, \bar{n}\}$. This can be also seen by the fact that $\frac{dw_{t,\tau}}{d\tau} = -\frac{1}{n} Z^\top (Z w_{t,\tau} - Zv)$ with $Z^\top = U\Sigma V^\top$ and thus the dynamics only adds components of u_i for $i \in \{1, \ldots, \bar{n}\}$, and not for $i \notin \{1, \ldots, \bar{n}\}$. Thus, for components of u_i for $i \notin \{1, \ldots, \bar{n}\}$, the initial values stays. In other words, for $i \notin \{1, \ldots, \bar{n}\}$,

$$c_i^{(t,\tau)} = c_i^{(t,0)}.$$

On the other hand, for $i \in \{1, \ldots, \bar{n}\}$, since $\frac{d}{d\tau} q_i = 0$,

$$\frac{d}{d\tau}(c_i^{(t,\tau)} - q_i) = \frac{d}{d\tau} c_i^{(t,\tau)} = -\sigma_i^2 (c_i^{(t,\tau)} - q_i).$$

Solving this for $(c_i^{(t,\tau)} - q_i)$, we have that for $i \in \{1, \ldots, \bar{n}\}$,

$$c_i^{(t,\tau)} - q_i = (c_i^{(t,0)} - q_i) e^{-\sigma_i^2 \tau}.$$

This implies that

$$c_i^{(t,\tau)} = q_i + (c_i^{(t,0)} - q_i) e^{-\sigma_i^2 \tau} = q_i (1 - e^{-\sigma_i^2 \tau}) + c_i^{(t,0)} e^{-\sigma_i^2 \tau}.$$

Combining these with $w_{t,T} = \sum_{k=1}^{\bar{d}} c_k^{(t,T)} u_k$,

$$w_{t,T} = \sum_{i=1}^{\bar{d}} c_i^{(t,T)} u_i = \sum_{i=1}^{\bar{n}} q_i (1 - e^{-\sigma_i^2 T}) u_i + \sum_{i=1}^{\bar{n}} c_i^{(t,0)} e^{-\sigma_i^2 T} u_i + \sum_{i=\bar{n}+1}^{\bar{d}} c_i^{(t,0)} u_i. \tag{5}$$

Therefore, for any particular $s \in \mathcal{S}$, since $U = [u_1, u_2 \ldots, u_{dp}] \in \mathbb{R}^{dp \times dp}$ is an orthogonal matrix,

$$\|\mathcal{A}_t(s)\|_2^2 = \|w_{t,T}\|_2^2 \leq \sum_{i=1}^{\bar{n}} \left(q_i (1 - e^{-\sigma_i^2 T})\right)^2 + \sum_{i=1}^{\bar{n}} (c_i^{(t,0)})^2 e^{-2\sigma_i^2 T} + \sum_{i=\bar{n}+1}^{\bar{d}} (c_i^{(t,0)})^2. \tag{6}$$

where q_i, σ_i, and $c_i^{(t,0)}$ all depend on s.

By using Lemma 4 of (Pham et al., 2021) and taking union bound with $\mathbb{P}(s \notin \mathcal{S}) \leq \delta$, with probability at least $1-\delta$, we have that $w_{t,T} \in \mathcal{F}_t$ and the following holds:

$$\mathbb{E}_{x,y}[\ell(w_{t,T}, x, y)] \leq \frac{1}{n}\sum_{i=1}^{n} \ell(w_{t,T}, x_i, y_i) + 2\mathcal{R}_n(\mathcal{F}_t) + M\sqrt{\frac{\ln(2/\delta)}{2n}}, \tag{7}$$

where $\mathcal{R}_n(\mathcal{F}_t) = \mathbb{E}_{s,\xi}[\sup_{w\in\mathcal{F}_t} \frac{1}{n}\sum_{i=1}^{n} \xi_i \|W\varphi(x_i) - y_i\|_2^2]$, $s = ((x_i, y_i))_{i=1}^n$, $w = \mathrm{vec}[W]$, and $\xi_1, \ldots, \xi_m$ are independent uniform random variables taking values in $\{-1, 1\}$. By using Corollary 4 of (Maurer, 2016), there exits a constant c (only depending on M) such that,

$$\begin{aligned}
\mathcal{R}_n(\mathcal{F}_t) &\leq \frac{c}{n}\mathbb{E}_{s,\xi}[\sup_{w\in\mathcal{F}_t} \sum_{i=1}^{n}\sum_{k=1}^{p} \xi_{ik} W_k \varphi(x_i)] \\
&= \frac{c}{n}\mathbb{E}_{s,\xi}[\sup_{w\in\mathcal{F}_t} \sum_{k=1}^{p} W_k \sum_{i=1}^{n} \xi_{ik}\varphi(x_i)] \\
&= \frac{c}{n}\mathbb{E}_{s,\xi}[\sup_{w\in\mathcal{F}_t} w^\top h]
\end{aligned}$$

where ξ_{ik} are independent uniform random variables taking values in $\{-1, 1\}$, $h = \mathrm{vec}[H] \in \mathbb{R}^{dp}$, and $H \in \mathbb{R}^{d\times p}$ with $H_{jk} = \sum_{i=1}^{n} \xi_{ik}\varphi(x_i)_j$. Thus,

$$\mathcal{R}_n(\mathcal{F}_t) \leq \frac{c}{n}\mathbb{E}_{s,\xi}[\sup_{w\in\mathcal{F}_t} \|w\|_2\|h\|_2] = \frac{c(\sup_{w\in\mathcal{F}_t}\|w\|_2)}{n}\mathbb{E}_{s,\xi}[\|h\|_2]$$

Here,

$$\begin{aligned}
\mathbb{E}_{s,\xi}[\|h\|_2] = \mathbb{E}_{s,\xi}\sqrt{\sum_{j=1}^{d}\sum_{k=1}^{p}\left(\sum_{i=1}^{n}\xi_{ik}\varphi(x_i)_j\right)^2} &\leq \sqrt{\sum_{j=1}^{d}\sum_{k=1}^{p}\mathbb{E}_{s,\xi}\left(\sum_{i=1}^{n}\xi_{ik}\varphi(x_i)_j\right)^2} \\
&= \sqrt{\sum_{k=1}^{p}\sum_{i=1}^{n}\mathbb{E}_s\sum_{j=1}^{d}(\varphi(x_i)_j)^2} \\
&= \sqrt{\sum_{k=1}^{p}\sum_{i=1}^{n}\mathbb{E}_s\|\varphi(x_i)\|^2} \\
&\leq R\sqrt{pn}
\end{aligned}$$

Thus,

$$\mathcal{R}_n(\mathcal{F}_t) \leq \frac{cR\sqrt{p}(\sup_{w\in\mathcal{F}_t}\|w\|_2)}{\sqrt{n}}. \tag{8}$$

Define

$$\zeta_t(s) := \sqrt{\sum_{i=1}^{\bar{n}}\left(q_i(1-e^{-\sigma_i^2 T})\right)^2 + \sum_{i=1}^{\bar{n}}(c_i^{(t,0)})^2 e^{-2\sigma_i^2 T} + \sum_{i=\bar{n}+1}^{\bar{d}}(c_i^{(t,0)})^2}.$$

where q_i, σ_i, and $c_i^{(t,0)}$ all depend on s. With this, we define

$$\zeta(t) := \sup_{s\in\mathcal{S}} \zeta_t(s).$$

Then, by combining equation 6, equation 7, and equation 8, with probability at least $1-\delta$, the following holds:

$$\mathbb{E}_{x,y}[\ell(w_{t,T}, x, y)] \leq \frac{1}{n}\sum_{i=1}^{n}\ell(w_{t,T}, x_i, y_i) + \zeta(t)\sqrt{\frac{4c^2R^2p}{n}} + M\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Finally, from Lemma 1, for any $t \in \mathbb{N}_0$ and $i \in \{1, \ldots, \bar{n}\}$,

$$(c_i^{(t,0)})^2 = \left(\frac{1}{\sigma_i}\left(\frac{1}{1+(n\lambda/\sigma_i^2)}\right)^t \tilde{y}_i\right)^2.$$

Since $\frac{1}{1+(n\lambda/\sigma_i^2)} < 1$ (because $n\lambda/\sigma_i^2 > 0$), the value of $\left(\frac{1}{1+(n\lambda/\sigma_i^2)}\right)^{2t}$ strictly decreases as t increases. Since $\frac{1}{\sigma_i^2} > 0$, $\tilde{y}^2 \geq 0$, and $c_0 \geq 0$, this implies that $(c_i^{(t,0)})^2$ is strictly decreasing in $t \in \mathbb{N}_0$ unless $c_i^{(t,0)} = 0$. Moreover, from Lemma 1, for any $t \in \mathbb{N}_0$ and $i \in \{\bar{n}+1, \ldots, \bar{d}\}$,

$$(c_i^{(t,0)})^2 = \mathbb{1}\{t = 0\}(u_i^\top w_{0,0})^2,$$

since

$$(I - \tilde{U}\tilde{U}^\top)w_{0,0} = \sum_{i=1}^{\bar{d}}(u_i^\top w_{0,0})u_i - \sum_{i=1}^{\bar{n}}(u_i^\top w_{0,0})u_i = \sum_{i=\bar{n}+1}^{\bar{d}}(u_i^\top w_{0,0})u_i.$$

These implies that $\zeta(t)$ is strictly decreasing in $t \in \mathbb{N}_0$ unless $w_{0,0} = 0$. □

A.2 Proof of Theorem 2

Proof. In this proof, we continue the proof of Theorem 1 using the results and the notation from the proof of Theorem 1. By using equation 5 in the proof of Theorem 1, we have that

$$\|w_{\text{init}} - w_{t,T}\| = \|w_{\text{init}} - v_t\|,$$

where

$$v_t = \sum_{i=1}^{\bar{n}} q_i(1 - e^{-\sigma_i^2 T})u_i + \sum_{i=1}^{\bar{n}} c_i^{(t,0)} e^{-\sigma_i^2 T} u_i + \sum_{i=\bar{n}+1}^{\bar{d}} c_i^{(t,0)} u_i.$$

If $w_{\text{init}} = -\alpha v_t$ for some $\alpha > 0$ and Y is set such that $q_i = 0$ for all $i \in [\bar{n}]$ for which $c_i \neq 0$, then

$$\|w_{\text{init}} - v_t\| = \|v_t + \alpha v\| = \|v_t\| + \|\alpha v\|$$
$$= \sqrt{\sum_{i=1}^{\bar{n}} q_i^2(1 - e^{-\sigma_i^2 T})^2 + \sum_{i=1}^{\bar{n}}(c_i^{(t,0)})^2 e^{-2\sigma_i^2 T} + \sum_{i=\bar{n}+1}^{\bar{d}}(c_i^{(t,0)})^2} + \|w_{\text{init}}\|.$$

On the other hand, for any $w_{\text{init}} \in \mathbb{R}^{dp}$,

$$\|w_{\text{init}} - v\| \leq \|v_t\| + \|w_{\text{init}}\| \leq \sqrt{\sum_{i=1}^{\bar{n}} q_i^2(1 - e^{-\sigma_i^2 T})^2 + \sum_{i=1}^{\bar{n}}(c_i^{(t,0)})^2 e^{-2\sigma_i^2 T} + \sum_{i=\bar{n}+1}^{\bar{d}}(c_i^{(t,0)})^2} + \|w_{\text{init}}\|.$$

Thus, setting $\psi(t)$ to be the following function satisfies conditions (1) and (2) in the statement:

$$\psi(t) = \sqrt{\sum_{i=1}^{\bar{n}} q_i^2(1 - e^{-\sigma_i^2 T})^2 + \sum_{i=1}^{\bar{n}}(c_i^{(t,0)})^2 e^{-2\sigma_i^2 T} + \sum_{i=\bar{n}+1}^{\bar{d}}(c_i^{(t,0)})^2} + \|w_{\text{init}}\|$$

Finally, from Lemma 1, for any $t \in \mathbb{N}_0$ and $i \in \{1, \ldots, \bar{n}\}$,

$$(c_i^{(t,0)})^2 = \left(\frac{1}{\sigma_i}\left(\frac{1}{1 + (n\lambda/\sigma_i^2)}\right)^t \tilde{y}_i\right)^2.$$

which is strictly decreasing in $t \in \mathbb{N}_0$ unless $c_i^{(t,0)} = 0$ for all $i \in \{1, \ldots, \bar{n}\}$ as shown in the proof of Theorem 1. Moreover, from Lemma 1, for any $t \in \mathbb{N}_0$ and $i \in \{\bar{n}+1, \ldots, \bar{d}\}$,

$$(c_i^{(t,0)})^2 = \mathbb{1}\{t = 0\}(u_i^\top w_{0,0})^2.$$

That is,

$$\psi(t) = \sqrt{G_1 + \sum_{i=1}^{\bar{n}}\left(\frac{1}{\sigma_i}\left(\frac{1}{1 + (n\lambda/\sigma_i^2)}\right)^t \tilde{y}_i\right)^2 e^{-2\sigma_i^2 T} + \sum_{i=\bar{n}+1}^{\bar{d}} \mathbb{1}\{t = 0\}(u_i^\top w_{0,0})^2} + G_2,$$

where

$$G_1 = \sum_{i=1}^{\bar{n}} q_i^2(1 - e^{-\sigma_i^2 T})^2,$$

and

$$G_2 = \|w_{\text{init}}\|.$$

These implies that $\zeta(t)$ is strictly decreasing in $t \in \mathbb{N}_0$ unless $w_{0,0} = 0$. □

B MASKED AUTO-ENCODING

In this section, we describe the masked auto-encoding objective from equation 2 in more detail. Given a sequence $\mathbf{x} = (x_1, \ldots, x_K)$ with length K, we sample mask $\mathbf{z} = (z_1, \ldots, z_K)$ from a Binomial distribution $p_{\gamma,K}$ with probability for success $\gamma \in (0, 1)$ and the number of trials K. For each x_k, we replace it with the special token "mask" if $z_k = 1$. Otherwise we use the same x_k for an masked input. Let $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_K)$ be a masked input and let f_θ, g_ϕ be encoder and decoder, respectively. We want to compute the log-likelihood of the reconstructed input $\sum_{k=1}^{K} z_k \log p_{\theta,\phi}(x_k|\hat{\mathbf{x}})$.

For language models, reconstruction of the masked input $\hat{x}_k$ is to predict which token is masked out of pre-defined vocabulary of which size is V, where each token is represented as an integer from $\{1, \ldots, V\}$. Thus the conditional probability of $x_k \in \{1, \ldots, V\}$ given $\hat{\mathbf{x}}$ is parameterized as follows:

$$p_{\theta,\phi}(x_k|\hat{\mathbf{x}}) = \frac{\exp(u_{x_k}))}{\sum_{j=1}^{V} \exp(u_j)}$$

$$\text{where } (u_1, \ldots, u_V) = g_\phi(\mathbf{h}_k) \in \mathbb{R}^V, \quad \begin{bmatrix} | & & | \\ \mathbf{h}_1 & \cdots & \mathbf{h}_K \\ | & & | \end{bmatrix} = f_\theta(\hat{\mathbf{x}}) \in \mathbb{R}^{h \times K}.$$

For Vision Transformers, the sequence $\mathbf{x}$ consists of image patches and the reconstruction of the masked input is to predict pixel values for each masked patches, which is a regression problem. Thus, we parameterize the conditional probability of a patch $x_k = (x_{k,1}, \ldots, x_{k,l}) \in \mathbb{R}^m$ given $\hat{\mathbf{x}}$ as follows:

$$p_{\theta,\phi}(x_k|\hat{\mathbf{x}}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_{k,i} - \mu_{k,i})^2}{2\sigma^2}\right)$$

$$\text{where } \mu_k = (\mu_{k,1}, \ldots, \mu_{k,m}) \in \mathbb{R}^m, \quad \begin{bmatrix} | & & | \\ \mu_1 & \cdots & \mu_K \\ | & & | \end{bmatrix} = f_\theta(\hat{\mathbf{x}}) \in \mathbb{R}^{m \times K}.$$

Since $\sigma > 0$ is constant,

$$\underset{\theta,\phi}{\text{minimize}} - \log p_{\theta,\phi}(x_k|\hat{\mathbf{x}}) = \underset{\theta,\phi}{\text{minimize}} \, \|x_k - \mu_k\|_2^2 \, .$$

C DATASET

We describe statistics of all the image and text classification datasets used for our experiments in Table 4 and 5.

Table 4: The number of training instances and classes for each image classification dataset.

	Aircraft	**CUB**	**Chest X-ray**	**DTD**	**Dogs**	**Flower**
# of instances	6,667	5,594	5,216	4,230	12,000	2,040
# of classes	100	200	2	47	120	102

Table 5: The number of training instances and classes for each text classification dataset.

	SCIERC	**ACL-ARC**	**Chemprot**	**Twitter-Emotion**
# of instances	3,219	1,688	4,169	4230
# of classes	7	6	13	47