7 Appendix

7.1 Extended Related Works

Other Robust Fine-Tuning Methods. WiSE-FT [14] discovers that linearly interpolating between the fine-tuned and pre-trained models after fine-tuning can improve out-of-distribution robustness. This demonstrates that a closer distance to the pre-trained model can improve robustness. However, it only applies to models with zero-shot capabilities. Another orthogonal line of research for robust fine-tuning focuses on feature distortion. LP-FT [19] shows that fine-tuning with a randomly initialized head layer distorts learned features. It proposes a simple two-stage method to train the head layer first and then fine-tune the entire model. FLYP [20] shows that fine-tuning a foundation model using the same objective as pre-training can better preserve the learned features. Our contribution is an optimization method to penalize the derivation between the fine-tuned and pre-trained models explicitly during fine-tuning, which is orthogonal to them.

7.2 Interpreting c_t as an Early Layer Selection Criterion

In previous sections, we interpreted the selection condition c_t in SPD as a measure of consistency between the current heading direction and the gradient direction. This perspective is more valid when the algorithm has accumulated some updates, i.e., $\|\theta_t - \theta_0\|_2 \gg 0$, and less justified when a heading has not been established at the beginning of training. This section discusses SPD from the perspective of *stochastic* optimization when $\|\theta_t - \theta_0\|_2$ is small at the beginning of training.

Inner product of gradients captures gradient variance. Modern deep learning models are trained by stochastic optimization techniques, e.g., mini-batch SGD, leading to stochasticity due to sampling. We first show that the inner product of gradients captures the variance of a sampling process. We invoke a common assumption in the convergence analysis of stochastic gradient descent [1, 40, 21]. Assuming that the stochastic gradient g_t is a stationary process $\mathcal{G}$ over a short period, with a small step size, successive gradients, e.g., g_t, g_{t+1}, can be seen as samples drawn from the same distribution $\mathcal{G}$. Given two successive draws g_1 and g_2, we can approximate the first and second moment of $\mathcal{G}$.

$$\mathbb{E}\left[\|g\|^2\right] \approx \frac{1}{2}(\|g_1\|^2 + \|g_2\|^2), \qquad \|\mathbb{E}\left[g\right]\|^2 \approx \|\frac{1}{2}(g_1 + g_2)\|^2. \tag{9}$$

Define the *variation of gradients* as $Var(g) := \mathbb{E}\left[\|g - \bar{g}\|^2\right]$ [41, 42], where $\bar{g} := \mathbb{E}[g]$,we can show that

$$\begin{aligned} g_1^\intercal g_2 &= 2\left(\frac{1}{4}\|g_1\|^2 + \frac{1}{4}\|g_2\|^2 + \frac{1}{2}g_1^\intercal g_2\right) - \frac{1}{2}\left(\|g_1\|^2 + \|g_2\|^2\right) \\ &\approx \|\bar{g}\|^2 - \left(\mathbb{E}\left[\|g\|^2\right] - \|\mathbb{E}\left[g\right]\|^2\right) = \|\bar{g}\|^2 - Var(g) \end{aligned} \tag{10}$$

Remarks. Eq. 10 shows that the inner product of two consecutive stochastic gradients, under certain assumptions, can be seen as the estimator for the difference between the gradient norm and the variance of gradients. When the inner product is negative, this indicates that the variance outweighs the magnitude of the gradient.

SPD prioritizes layers with higher expected gain. At the beginning of training, the heading direction $(\theta_1 - \theta_0)$ is dominated by early gradients. For example, at $t = 2$ the direction of $(\theta_1 - \theta_0)$ is the same as $-g_1$ in Adam. The sign of $-g_2^\intercal(\theta_1 - \theta_0)$ is the same as the sign of $g_2^\intercal g_1$. This shows that the condition c_t captures the difference between gradient norm and gradient variance. With this interpretation, we show that c_t reflects *expected performance gain* in stochastic optimization. To see it, we can invoke the descent lemma for SGD. For an L-smooth function $f(W)$ [41], the descent lemma for SGD states that,

Lemma 1. $\underbrace{\mathbb{E}[f(\theta_{k+1})] - f(\theta_k)}_{\textit{Expected Performance Gain}} \leq \underbrace{-\eta_k(1 - \frac{\eta_k L}{2})}_{\leq 0}\|\bar{g}_k\|^2 + \underbrace{\frac{\eta_k^2 L}{2}}_{\geq 0} Var(g_k),$

where $\eta_k \leq \frac{2}{L}$ is the learning rate.

Remarks. The term on the left hand side $\mathbb{E}[f(\theta_{k+1})] - f(\theta_k)$ is the expected performance improvement for each step. Ideally, this should be a negative quantity. On the right-hand side, we observe that improvement depends on two quantities $\|\bar{g}_k\|^2$ and $Var(g_k)$. To lower the upper bound, we want a *large* $\|\bar{g}_k\|^2$ and a *small* $Var(g_k)$. According to the decoupling Eq. 10, the inner product between successive gradients approximates this proportionality. Consequently, a negative c_t likely indicates a higher upper bound on the expected gain, meaning a smaller improvement. Therefore, SPD will prioritize layers with potentially larger expected gains.

8 Training Details

DomainNet. We use the vision transformer public repository for DEIT [37] to fine-tune all methods. Standard augmentations are used for all: weight-decay (0.1), drop-path (0.2) [43], label-smoothing (0.1) [44], Mixup (0.8) [45] and Cutmix (1.0) [46]. The learning rate is $2e-5$ and trained for 60 epochs for Tab. 1 and 30 epochs for Tab. 2. We use $\lambda = 1$ for all Adam-SPD results in Tab. 1. We use 1 A40 GPU for each experiment.

ImageNet. The same procedure as the DomainNet experiment is used for training the ImageNet models. Standard augmentations are used for all: weight-decay (0.1), drop-path (0.2) [43], label-smoothing (0.1) [44], Mixup (0.8) [45] and Cutmix (1.0) [46]. We fine-tune all methods for 30 epochs and use the best hyper-parameters reported by the prior work [11]. For Adam-SPD, we fine-tune the model with a learning rate of $3e-5$ and $\lambda = 1.4$. The regularization hyper-parameter is found through cross-validation, and the model with the best ID validation accuracy is taken. We use 2 A40 GPUs for each experiment.

Pascal Segmentation. We follow the training code released by a prior work [31]. We fine-tune all methods for 60 epochs and use the best hyper-parameters reported by the prior work. For Adam-SPD, we fine-tune the model with a learning rate of $1e-4$ and $\lambda = 2.2$. The regularization hyper-parameter is found through cross-validation, and the model with the best ID validation accuracy is taken. We use 4 2080Ti GPUs for each experiment.

Commonsense-170K. Training Details. We follow the training code released by a prior work [35]. We report the best performance from the original paper and compare them with Adam-SPD. For Adam-SPD, we fine-tune the model with an identical hyper-parameter setup as the released code and only adjust the regularization strength λ. The regularization hyper-parameter is found through cross-validation, and the model with the best ID validation loss is taken. We use 1 A40 GPU for each experiment.