Simulated Annealing in Early Layers Leads to Better Generalization

Amir M. Sarfi[1,2]   Zahra Karimpour[1]   Muawiz Chaudhary[1,2]   Nasir M. Khalid [1,2]

Mirco Ravanelli[1,2]   Sudhir Mudur[1]   Eugene Belilovsky[1,2]

[1] Concordia University   [2] Mila – Quebec AI Institute

Abstract

Recently, a number of iterative learning methods have been introduced to improve generalization. These typically rely on training for longer periods of time in exchange for improved generalization. LLF (later-layer-forgetting) is a state-of-the-art method in this category. It strengthens learning in early layers by periodically re-initializing the last few layers of the network. Our principal innovation in this work is to use Simulated annealing in EArly Layers (SEAL) of the network in place of re-initialization of later layers. Essentially, later layers go through the normal gradient descent process, while the early layers go through short stints of gradient ascent followed by gradient descent. Extensive experiments on the popular Tiny-ImageNet dataset benchmark and a series of transfer learning and few-shot learning tasks show that we outperform LLF by a significant margin. We further show that, compared to normal training, LLF features, although improving on the target task, degrade the transfer learning performance across all datasets we explored. In comparison, our method outperforms LLF across the same target datasets by a large margin. We also show that the prediction depth of our method is significantly lower than that of LLF and normal training, indicating on average better prediction performance.[1]

1. Introduction

Overfitting is a crucial challenge in supervised deep learning, which prevents a neural network from performing well on unseen data. This problem is particularly pervasive in smaller dataset regimes. Classical machine learning approaches to address this problem typically rely on explicit regularization added as part of the optimization ob-

[1]The code to reproduce our results is publicly available at: https://github.com/amiiir-sarfi/SEAL

jective [37]. A number of implicit approaches are often applied in training deep networks. These implicit methods are usually easier to design than explicit terms added to the objective function. For example, early stopping [42] and dropout [32] are classical examples of implicit regularization methods. It can be shown that these techniques can be shown to be linked directly to explicit regularization terms. Consider the case of linear regression, early stopping can be directly seen as Tikhonov regularization [42]. Indeed the use of modified optimization strategies to improve generalization is becoming pervasive in deep learning [7].

Recently, *iterative training* methods have been introduced to improve generalization in deep networks. These allow neural networks to be trained for many epochs while progressively improving generalization [9, 10, 35, 41, 46]. These works typically rely on a notion of a *generation*, in which a model is optimized towards a local minimum in a single generation. Subsequently, in a new generation, the objective function is modified, or the network is perturbed towards a high loss, requiring a new generation of optimization towards a minimum. The earliest work on this topic focused on self-distillation [10, 41], where, in each successive generation, a student network with the same architecture was initialized and trained to mimic the softmax output distribution of the previous model. This procedure was theoretically studied by [22] and provided the formal link to its explicit regularization effects.

A number of other techniques were subsequently proposed. These have been characterized by Zhou et al. [46] as instances of a forget-and-relearn scheme, where they define forgetting as any process that worsens the training accuracy of a model. In forget-and-relearn, the forgetting stage happens at the beginning of each generation where some part of the weights are perturbed (e.g., by removing [8, 9] or by re-initialization [35, 46]), and the network is normally trained for the rest of the generation. They further introduce a simple forgetting method called later-layer-forgetting (LLF), where they re-initialize the last few layers of the network. The intuition behind LLF comes from the concept of prediction depth introduced by Baldock et al. [1]. The prediction depth of a sample refers to the earliest layer, after which the layer-wise k-nearest neighbor (k-NN) probe of all layers is the same as
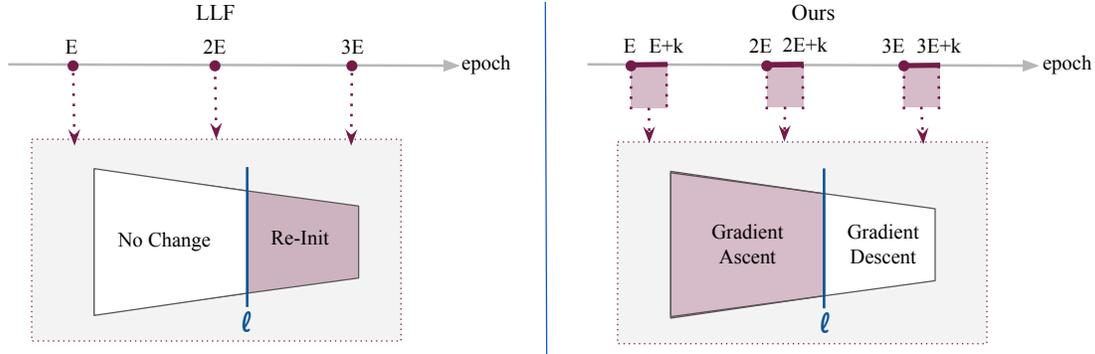
Figure 1. Our iterative training method (SEAL) compared to LLF. The model is shown with the input on the right and output on the left. E denotes the epochs in a generation. LLF re-initializes the top layers right before a new generation begin. In our approach, we do not re-initialize but perform gradient ascent in the first k epochs of a generation, only on the early layers of the network.

the model's prediction. Zhou et al. [46] empirically show that LLF improves the prediction depth.

We hypothesize that by constantly re-initializing the later layers, LLF pushes high-level information to the early layers, which explains its stronger generalization as compared to normal training. However, when it comes to transfer learning, Zhao et al. [44] have stated that it is mainly the low- and mid-level representations that are transferred. Following this insight, we analyzed features learned by LLF in a transfer learning setting. We observed that LLF performs weaker than normal training, which affirms our hypothesis. This is discussed further in the Experimental Results section 4.2.

Simulated annealing is a method for solving unconstrained and bound-constrained optimization problems. It models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. During training, gradient ascent mimics increase in defects (temperature increase), while gradient descent emulates the cooling process to reach a better minimum [17]. Cai [2] adapted the classical simulated annealing for gradient descent (SA-GD) to improve the optimization by escaping local minimums and saddle points. In SA-GD, this is done by probabilistically doing gradient ascent during training. Their method, however, is not used for iterative learning and unlike our work is not adapted to take advantage of the specific architecture biases of deep networks. Based on the hypothesis that LLF primarily benefits from pushing high-level information into the early layers (and thereby the prediction depth), in this work, we adapt the SA idea as an alternative to LLF, and we do not reset the later layers. Instead, we perform an intermittent gradient ascent procedure just on the early layers, coaxing them to find better solutions through multiple generations of training.

Our main contribution is a new iterative training method that performs gradient ascent on the initial layers of the network, for a few epochs, at the beginning of every generation to induce forgetting. By performing the ascent on only a sub-

set of layers, we ensure that the model retains information from the previous generations, rendering it suitable for long training. The intuition behind gradient ascent on the early layers is that it prohibits them from being overly specific to the task, avoiding the encoding of high-level semantics, unlike what happens in LLF. Furthermore, even though we perform simulated annealing on the early layers and LLF performs re-initialization on the later layers, the goal behind both approaches is the same; both methods try to enhance the early layers.

We have carried out extensive experiments on the Tiny-ImageNet [19] dataset and transfer learning to Flower [25], CUB [39], Aircraft [21], Dogs [16], and MIT [28] datasets, and Cross-Domain Few-Shot Learning (CD-FSL) [11]. Our experiments show that our method outperforms LLF in both in-distribution and transfer learning settings (Tables 1, 2, 4). Our method also provides better prediction depth (Fig. 2). By analyzing the statistics of Hessian eigenvalues, we observe that our method has a lower max eigenvalue and no negative eigenvalues, suggesting SEAL reaches a flatter local minima and avoids saddle points (Table 6).

2. Background and Related Work

Iterative learning Consider a neural network f parameterized by Θ. Following [35] and [46], let us define a mask M that splits a neural network's weights into fit hypothesis $\mathbf{H}_{fit}$ and forgetting hypothesis $\mathbf{H}_{forget}$ as:

$$\mathbf{H}_{fit} = M \odot \Theta \quad \text{and} \quad \mathbf{H}_{forget} = (1 - M) \odot \Theta \quad (1)$$

Forget-and-relearn iterative training methods perturb the weights in the forgetting hypothesis $\mathbf{H}_{forget}$ at the beginning of each generation and retrain the neural net for E epochs. This process is repeated for G generations; hence it is called iterative training. We now formally cast the previous iterative training methods into this framework.

In the work "Knowledge evolution in neural networks (KE)" [35] two different masking strategies were proposed.

One strategy splits the network's weights based on their location in the convolutional kernels, namely kernel-level (KELS) splitting. The other strategy splits the weights randomly, namely weight-level splitting (WELS). At the beginning of every generation, the weights in the forgetting hypothesis $\mathbf{H}_{forget}$ are re-initialized, and the fit hypothesis $\mathbf{H}_{fit}$ is kept the same. Zhou et al. [45] defined special masking strategies where weight masking and rewinding would lead to initialization that have much better-than-chance performance before retraining. Iterative magnitude pruning (IMP) [8, 9] splits the weights based on the weight magnitudes of the last generation. The forgetting hypothesis $\mathbf{H}_{forget}$ consists of weights that had low magnitude, and the rest of the weights are considered in the fit hypothesis $\mathbf{H}_{fit}$. The forgetting hypothesis $\mathbf{H}_{forget}$ weights are removed (by setting them to zero and freezing), and the fit hypothesis $\mathbf{H}_{fit}$ weights are rewound to their initial values. Thus, in each generation, a percentage of the weights is removed.

LLF [46] modifies the masking strategy of KE to improve the prediction depth of the network [1]. LLF considers a layer threshold L, and the weights in all layers before L are put into the forgetting hypothesis $\mathbf{H}_{forget}$, and the rest of the network (the last half of the network) is considered the fit hypothesis $\mathbf{H}_{fit}$. Then, similar to KE, they re-initialize the forgetting hypothesis $\mathbf{H}_{forget}$ and do not modify the fit hypothesis $\mathbf{H}_{fit}$.

Simulated Annealing: Cai [2] modified the traditional simulated annealing method for gradient descent (SA-GD) to enhance optimization by evading local minima and saddle points. Dauphing et al. [4] makes the argument that many difficulties in optimization arise from saddle points and not local minima. Jin et al. [15] develop a perturbed stochastic gradient descent procedure to deal with saddle points.

Few Shot Learning Traditionally, machine learning models require large domain-specific labeled datasets in order to correctly classify [12, 18, 30]. The goal of few shot learning (FSL) [38] is the following: given a limited number of labeled data, learn a model that rapidly generalizes to new, novel classes. Few Short Learning has been considered recently in the literature, from the point of view of meta learning and transfer approaches [5], [31], [29], [38], [33]. Typical FSL tasks are based on having an initial large dataset from which a model is either pre-trained (transfer based approaches) or meta-trained (meta-learning based approaches).

Early FSL works have focused on using natural image datasets, with another natural image dataset provided for base model training (via meta-learning or pre-training) [33,38]. On the other hand, many cases of more general interest require learning target datasets that are not natural image datasets, and moreover are distant in terms of the domain. The recently proposed Cross-Domain Few-Shot Learning (CD-FSL) [11] benchmark is aimed at providing an investigative setting for such large domain shifts from source to target training data. Four datasets of decreasing similarity to natural images are included, [23], [13], [36], [3], [40]. These include images of plant disease, aerial photos, and medical data not resembling natural images at all (skin lesions, Chest X-rays). In each dataset, we wish to predict some novel classes, whether it be rare skin diseases, airplanes, or crop diseases. A number of proposals for this challenging scenario have been suggested. [27] proposed a self-supervised learning objective coupled with a teacher-student method. They also observed that traditional meta-learning techniques fail on this task. [43] demonstrated that efficient manipulation of the batch norm layer during the training of models can lead to improved performance under such extreme domain shifts. The results of using SEAL for FSL are provided in a later section.

3. Proposed Method

In this work, we split the network into fit $\mathbf{H}_{fit}$ and forgetting hypotheses $\mathbf{H}_{forget}$, using a layer threshold, say, L. All weights prior to L are considered in the forgetting hypothesis $\mathbf{H}_{forget}$, and weights in layers $> L$ as the fit hypothesis $\mathbf{H}_{fit}$.

To induce forgetting, we perform gradient ascent on the forgetting hypothesis $\mathbf{H}_{forget}$ for k epochs. During the gradient ascent phase of the forgetting hypothesis $\mathbf{H}_{forget}$, we train the fit hypothesis $\mathbf{H}_{fit}$ normally (with gradient descent). This can be categorized under the high-level definition of forgetting that Zhou et al. [46] provide, in that it drops the training accuracy of the network to completely random. This is inspired by the simulated annealing algorithm to enhance the optimization of the network and to introduce a more definitive forgetting mechanism. Our method is different from the prior methods as they either remove [8, 9] or re-initialize [46] [35] the forgetting hypothesis $\mathbf{H}_{forget}$ (at once) before the first epoch of a new generation. We set k to $\frac{1}{4}$ of total epochs E in the generation.

We adjusted the sign of the weight decay for the layers that perform gradient ascent; so as to avoid the weights being encouraged to have a higher norm, and causing the network to diverge. However, we found that even this is not enough to stop the divergence. We noted that using the same learning rate for the ascending phase was the reason for this. Hence we toned down the ascent learning rate using a factor S:

$$\eta_a = S \times \eta \qquad (2)$$

Where η is the learning rate, and η_a is the ascent learning rate. We empirically fix $S = 0.01$. We summarize the whole process as follows:

$$\Theta_{forget}^{e,t+1} = \begin{cases} \Theta_{forget}^{e,t} + \eta_a \nabla J(\Theta_{forget}^{e,t}), & \text{if } e \% E < k \\ \Theta_{forget}^{e,t} - \eta \nabla J(\Theta_{forget}^{e,t}), & \text{otherwise} \end{cases},$$

$$\Theta_{fit}^{e,t+1} = \Theta_{fit}^{e,t} - \eta \nabla J(\Theta_{fit}^{e,t})$$

$$(3)$$

Where $J(\Theta)$ is the objective function, $\Theta_{forget}^{e,t}$ and $\Theta_{fit}^{e,t}$ refer to parameters in the forgetting and fit hypotheses, respectively, and e, t refers to the iteration t during epoch e. Again, E refers to epochs in each generation and is fixed for all generations. Figure 1 illustrates this for LLF and our proposed method (SEAL).

4. Experimental Results

4.1. Implementation Details

We use Tiny-ImageNet [19] to train models and then evaluate on both the Tiny-ImageNet test set and a wide set of downstream transfer learning tasks, including popular few shot learning benchmarks. Following LLF, we use ResNet50 and train using SGD optimizer with a momentum of 0.9 and weight decay of 5e−4. We train $G = 10$ generations for $E = 160$ epochs using a batch size of 32. As in LLF, we use cross entropy loss with label-smoothing [24, 34] with $\alpha = 0.1$. We also use cosine learning rate decay [20] with an initial learning rate of 0.01. For data augmentations, we perform horizontal flip and random crop with a padding of 4. We use layer threshold $L = 23$ for both LLF and our method (the third block in ResNet50). This means that in LLF, the first two blocks are considered the fit hypothesis $\mathbf{H}_{fit}$, while the fit hypothesis in our method is the last two blocks. In all experiments, normal training refers to $G \times E$ epochs of training with the same optimization settings.

For our few shot learning evaluation, we evaluated our models in an episodic fashion. Each episode has a train set of 5 classes with K examples each (5-way K-shot) and a test set with 15 examples for each class. These sets are sampled from a target dataset. Models are fine tuned on the train set and then used to produce predictions on the test set. The accuracies are reported over 600 episodes. Cross Domain Few Shot Learning Benchmark (CD-FSL) [11] was selected as the dataset for the task, which includes data from four different data sets, namely CropDiseases [23], EuroSAT [13], ISIC2018 [3, 36], and ChestX [40].

Overall we compare our method with the following three approaches:
Normal denotes the standard training with 160 epochs (corresponding to $G = 1$ generations under the conventions of iterative training).
Normal (long) refers to training the model with the standard settings for 1, 600 epochs (corresponding to $G = 10$ generations without any forgetting).

LLF refers to fortuitous forgetting [46] where at the beginning of each generation the later layers of the network are re-initialized.
SEAL refers to our proposal which performs a gradient ascent and subsequent descent phase during a generation.

Datasets Tiny-ImageNet consists of 200 classes and has 100, 000 training and 10, 000 validation images selected from the ImageNet dataset. These images are downsized to 64x64 colored images. For our transfer learning evaluations we use the natural image datasets Flower, CUB, Aircraft, MIT, and Stanford Dogs, the statistics of these datasets are provided in Table 3.

For the few-shot learning, we utilized the 4 datasets from the CD-FSL benchmark, which include ChestX, ISIC, CropDisease, and EuroSAT. From these, we sample training and evaluation sets with 5 classes and a varying number of samples per class.

4.2. Evaluation

We now present our primary evaluations of SEAL for both improved generalization, transfer, and few-shot transfer learning.

In-Distribution Generalization: Table 2 shows the in-distribution accuracy of different methods. We note that LLF outperforms Normal training (as reported in [46]). Our method outperforms both LLF and normal training in this setting.

Transfer Learning We now turn to evaluate the transfer learning properties of our method and that of LLF. We begin by studying the transfer learning from the Tiny-ImageNet pretrained models to 5 different image datasets. Specifically, CUB-200-2011 (CUB) [39] contains images of 200 wild bird species, Flower102 (Flower) [25] contains images from 102 flower categories, FGVC-Aircraft (Aircraft) [21] consists of 100 aircraft model variants, and Stanford Dogs dataset contains images of 120 breeds of dogs for fine-grained classification. MIT Indoor 67 (MIT) [28] is an indoor scene recognition dataset that includes 67 different scene classes.

We train linear models on top of pretrained models from each method to measure their transfer learning properties. Specifically, we re-initialize the last linear layer of the network and freeze the rest. Then, we train the linear head using the train set of the target datasets and evaluate on their test sets. For training on the target dataset, we again use SGD with a momentum of 0.9, weight decay of 1e−4, and flat learning rates of [1e−1, 1e−2, 1e−3] for 120 epochs and report the highest accuracy.

Table 1 demonstrates the transfer accuracy of LLF, normal training, and our method (SEAL). Even though LLF outperforms normal training with a 2.55% margin in the in-distribution setting, we observe that in transfer learning,

Method	Tiny-ImageNet	Flower	CUB	Aircraft	MIT	Stanford Dogs
Normal	54.37	34.31	6.49	6.24	25.67	8.99
Normal (long)	49.27	26.96	8.07	6.30	24.85	11.53
LLF	56.92	22.84	5.33	4.65	23.8	8.69
SEAL (Ours)	**59.22**	**45.68**	**8.49**	**9.81**	**35.37**	**12.61**

Table 1. Transferring tiny-imagenet learned features to other datasets using linear probe. Normal, and Normal (long) refer to $G = 1$ and $G = 10$ generations of training, respectively. LLF and SEAL were trained for $G = 10$ generations. Transfer accuracy of LLF after $1,600$ epochs is significantly lower than normal training with both 160 and $1,600$ epochs; our method after $1,600$ epochs surpasses normal training by a large margin. This demonstrates that our method learns much more generalizable features compared to Normal training and LLF.
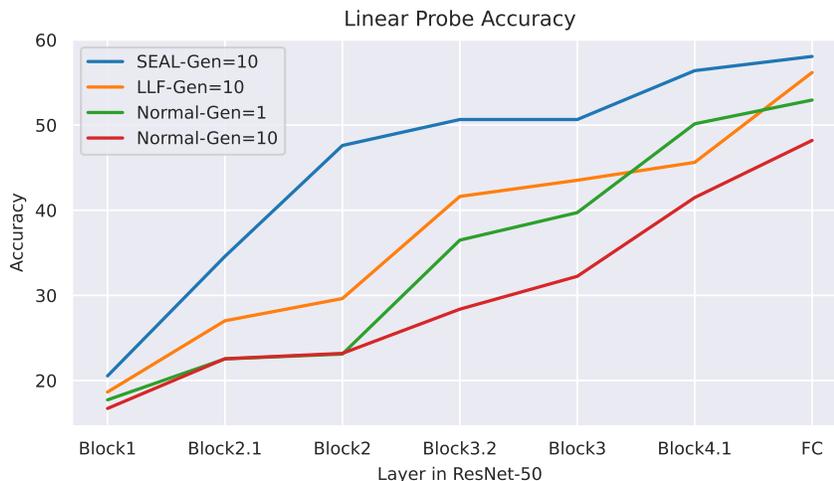

Figure 2. Comparison of layer-wise prediction depth. SEAL gives comparably much stronger predictions early on in the network. Note that block.$X.Y$ denotes the output activations of intermediate layer Y in residual block X. This indicates that our method encourages learning the difficult examples using conceptually simpler and more general features of the early layers. This leads to better overall performance as we progress deeper into the network.

Generation	Normal	LLF	Ours
Gen=1	54.37	-	-
Gen=3	51.16	56.12	**58.25**
Gen=10	49.27	56.92	**59.22**

Table 2. Comparison of our method with normal training and LLF on Tiny-ImageNet. Please note that the behavior of the first generation for all methods is the same. We significantly outperform standard long training and LLF.

	num_classes	Train	Valid	Test	Total
Flower [25]	102	1,020	1,020	6,149	8,189
CUB [39]	200	5,994	N/A	5,794	11,788
Aircraft [21]	100	3,334	3333	3,333	10,000
MIT [28]	67	5,360	N/A	1,340	6,700
Dogs [16]	120	12,000	N/A	8,580	20,580

Table 3. Summary of the datasets used in tables 2 and 1, adopted from Taha et al. [35].

LLF's performance is substantially lower than normal training for $1/10$ of epochs ($G = 1$), as well as normal training for the same number of epochs ($G = 10$). On the other hand, our method not only dominates in the in-distribution setting, but it also has a much stronger transfer learning performance than both LLF and normal training across all of the target datasets in our experiments.

Few-Shot Transfer Learning We now consider evaluating our approach for a challenging distant transfer learning task studied in [27, 43]. Here we are presented with a few shot learning problem on multiple datasets from medical imaging to satellite images, with only one dataset of natural images available for pre-training. Several approaches to this problem exist, some utilizing meta-learning methods [6] and others focused on transfer learning [43]. It has been shown in multiple works that for this distant few shot learning task, transfer learning approaches exceed meta-learning [43]. We use our Tiny-ImageNet models from the previous section as base model for FSL transfer. In transfer we train a linear head as in the standard protocol [27, 43], we also consider jointly

Parameters Updated	Base Model	ChestX	ISIC	EuroSAT	CropDisease
		5-WAY, 1-SHOT			
Linear	Normal	21.01 ± 0.35	25.7 ± 0.47	53.41 ± 0.92	50.31 ± 1.03
	Normal (long)	20.40 ± 0.23	22.59 ± 0.35	36.60 ± 0.75	27.50 ± 0.69
	LLF	20.38 ± 0.24	24.61 ± 0.46	36.58 ± 0.86	26.28 ± 0.68
	SEAL (ours)	**21.67 ± 0.36**	**27.93 ± 0.55**	**57.75 ± 0.88**	**63.64 ± 0.96**
Linear+Affines	Normal	21.14 ± 0.35	26.71 ± 0.56	47.36 ± 1.26	64.75 ± 1.13
	Normal (long)	20.90 ± 0.37	26.41 ± 0.54	45.51 ± 1.25	63.73 ± 1.05
	LLF	20.24 ± 0.24	23.28 ± 0.45	34.02 ± 1.35	35.12 ± 1.61
	SEAL (ours)	**21.3 ± 0.39**	**29.14 ± 0.57**	**55.68 ± 1.05**	**67.87 ± 0.54**
		5-WAY, 5-SHOT			
Linear	Normal	22.79 ± 0.36	33.28 ± 0.49	71.74 ± 0.75	77.05 ± 0.80
	Normal(long)	21.00 ± 0.28	28.93 ± 0.48	53.40 ± 0.77	57.10 ± 1.07
	LLF	22.03 ± 0.33	29.60 ± 0.48	60.69 ± 0.87	53.55 ± 1.12
	SEAL (ours)	**24.42 ± 0.42**	**37.58 ± 0.54**	**74.61 ± 0.65**	**85.00 ± 0.61**
Linear+Affines	Normal	20.82 ± 0.26	28.57 ± 0.78	53.35 ± 1.74	63.22 ± 2.35
	Normal(long)	21.59 ± 0.34	29.67 ± 0.82	55.79 ± 1.67	71.55 ± 2.00
	LLF	20.44 ± 0.19	22.01 ± 0.44	30.83 ± 1.45	28.06 ± 1.47
	SEAL (ours)	**22.98 ± 0.36**	**39.64 ± 0.79**	**73.83 ± 0.93**	**88.24 ± 0.54**

Table 4. Few-shot transfer results for the CFSDL benchmark (extreme distribution shift) for 1 and 5 shots. All methods make use of a ResNet50 backbone trained on Tiny-ImageNet evaluated over 600 episodes. We consider finetuning both the linear layer and the linear layer and affine parameters, the best performer in both categories highlighted in red. We observe that SEAL outperforms standard training, while LLF under-performs.

training linear head and affine parameters as suggested by [43]. Results of our evaluations for 1, 5, 20, and 50 shots are shown in Table 4 and Table 5. We observe that in this challenging benchmark LLF substantially underperforms standard training, suggesting features learned by LLF do not generalize well. We note that the training accuracies on Tiny-Imagenet of all the suggested models are 100% and the testing accuracies are the ones shown in Table 2. Although LLF has higher in-distribution performance than normal training, its FSL transfer properties are much worse. On the other hand, SEAL features generalize both in-distribution and to the distant FSL tasks.

5. Analysis and Ablation Studies

In this section we present additional analysis of our method, specifically we study the effects on the prediction depth as well as on the eigenvalues of the Hessian at the end of model training. Finally, we perform ablation studies to demonstrate that the early layers indeed benefit the most from SEAL.

Analysis of Hessian Eigenvalues We first study the eigenvalue spectra of the Hessian for the different proposed models. We utilize the same process for estimating the eigen-

values suggested in [26]. The results for the 4 models are summarized in Table 6 where we report both the maximum eigenvalues and the percentage of negative eigenvalues. We observe that the maximum eigenvalues are smaller for SEAL than for normal long training and also for LLF, suggesting a flatter minimum. Flatter minima have been previously associated with improved generalization [14].

We further observe that SEAL has no negative eigenvalues. This suggests that SEAL obtains some of its advantages by helping to avoid saddle points during training. This is consistent with prior uses of simulated annealing [2]. Following [26] we hypothesize the absence of negative eigenvalues can indicate a more robust solution.

Prediction Depth: Zhou et al. [46] proposed LLF to specifically enhance the prediction depth of the model. Their intuition was that periodically resetting the final layers would decrease the prediction depth. Following Zhou et al. [46], we approximate the prediction depth using the K Nearest Neighbor (KNN) probe (with $K = 5$) on different layers of the network. To do so, for every image in the test set, we use all of the images in the train set for the KNN.

We affirm that the prediction depth of LLF is improved over normal training (Figure 2). However, with SEAL, we achieve a much stronger prediction depth. For instance, the

Parameters Updated	Base Model	ChestX	ISIC	EuroSAT	CropDisease
		5-WAY, 20-SHOT			
Linear	Normal	25.16 ± 0.37	41.24 ± 0.50	79.14 ± 0.67	86.74 ± 0.57
	Normal (long)	21.79 ± 0.27	32.85 ± 0.50	60.95 ± 0.81	70.20 ± 1.01
	LLF	22.54 ± 0.29	32.26 ± 0.48	67.79 ± 0.85	68.43 ± 1.07
	SEAL (ours)	$\boxed{\mathbf{27.44 \pm 0.40}}$	$\mathbf{46.96 \pm 0.53}$	$\mathbf{82.45 \pm 0.53}$	$\mathbf{92.47 \pm 0.39}$
Linear+Affines	Normal	22.91 ± 0.36	49.40 ± 1.14	81.52 ± 1.38	87.43 ± 1.67
	Normal (long)	23.38 ± 0.37	47.36 ± 1.25	80.03 ± 1.53	89.63 ± 1.53
	LLF	21.10 ± 0.25	30.54 ± 1.22	46.55 ± 2.42	39.80 ± 2.30
	SEAL (ours)	$\mathbf{26.99 \pm 0.46}$	$\boxed{\mathbf{55.12 \pm 0.77}}$	$\boxed{\mathbf{87.70 \pm 0.52}}$	$\boxed{\mathbf{95.67 \pm 0.28}}$
		5-WAY, 50-SHOT			
Linear	Normal	26.55 ± 0.36	46.29 ± 0.47	82.20 ± 0.61	90.51 ± 0.45
	Normal (long)	22.56 ± 0.29	36.57 ± 0.53	67.14 ± 0.77	80.60 ± 0.78
	LLF	23.78 ± 0.31	35.59 ± 0.5	73.76 ± 0.79	79.67 ± 0.78
	SEAL (ours)	$\boxed{\mathbf{29.78 \pm 0.40}}$	$\mathbf{51.46 \pm 0.50}$	$\mathbf{84.99 \pm 0.52}$	$\mathbf{94.91 \pm 0.29}$
Linear+Affines	Normal	24.2 ± 0.40	60.27 ± 1.10	88.09 ± 1.30	93.80 ± 1.33
	Normal (long)	24.62 ± 0.45	56.98 ± 1.32	85.17 ± 1.58	89.8 ± 1.9
	LLF	22.56 ± 0.34	41.53 ± 1.65	58.69 ± 2.76	51.35 ± 2.92
	SEAL (ours)	$\mathbf{27.13 \pm 0.45}$	$\boxed{\mathbf{60.59 \pm 1.11}}$	$\boxed{\mathbf{91.94 \pm 0.53}}$	$\boxed{\mathbf{97.91 \pm 0.40}}$

Table 5. Low-shot transfer results for the CFSDL benchmark (extreme distribution shift) for 20 and 50 shots. We fine-tuned both the linear layer and the linear layer along with affine parameters. We observed that SEAL outperforms standard training and that LLF severely underperforms in this case. Tuning the affine parameters and linear layer provides consistent performance gains for both Normal training models and SEAL.

Method	Max Eigenval.	Negative % Eigenval.
Normal	889.06	4.25%
Normal (long)	2353.74	0%
LLF	1027.21	14.63%
SEAL (Ours)	847.89	0%

Table 6. In this table, we demonstrate the statistics of the Hessian eigenvalues. We observe that our method has a lower max eigenvalue which suggests flatter local minima. Furthermore, our method has no negative eigenvalues, suggesting SEAL can avoid saddle points.

KNN accuracy of our method is more than 18.54% stronger than LLF and 25.04% stronger than normal training on the outputs of the second block of the network. This is the layer threshold L used in our method and LLF. The layer-wise accuracy of the other layers indicates the superiority of our method across all layers.

Baldock et al. [1] demonstrated that example difficulty is correlated with prediction depth; decreasing the prediction depth corresponds to lower example difficulty, which is desired. They show this correlation by analyzing the speed of learning, the input and output margin, and the adversarial input margin for each data point. In Figure 3, we measure the prediction depth evolution of the three methods over different generations ($G = [1, 2, 4, 10]$). For normal training, the prediction depth gets worse over time, which explains its poor in-distribution performance. This suggests that in normal training, the early layers are becoming weaker after $G = 1$ and more samples are being classified by the later layers.

LLF slightly improves the prediction depth of the model. However, this comes with deterioration of the performance of the later layers. For instance, the KNN probe on the activations of Block4.1 in $G = 1$ has 50.15% accuracy which decreases to 45.62 in $G = 10$. On the other hand, our method does a better job of pushing more examples to be classified in the early layers than normal training and LLF. This implies that SEAL promotes relearning the more difficult samples using the simpler and more general features of the early layers. Furthermore, the later layers of the network improve over time with our method.

Ablation Study: We now investigate different strategies for the fit hypothesis $\mathbf{H}_{fit}$ during the forgetting phase. By default, during this phase, we perform gradient descent on the fit hypothesis and gradient ascent on the forgetting hypothesis. In "Ours+Reinit", following LLF [46], at the beginning

Gen	Normal	SEAL+Freeze	SEAL+Reinit	SEAL+Reverse	SEAL+Descent
Gen1	54.37	-	-	-	-
Gen3	51.16	52.45	56.82	50.25	**58.25**
Gen10	49.27	51.17	58.87	41.05	**59.22**

Table 7. Fitting hypothesis $\mathbf{H}_{fit}$ ablation study. While performing gradient ascent on the early layers during forgetting, we freeze, reinitialize, and perform gradient descent on the later layers. In reverse, we swap the fit and forgetting hypotheses. We observe that doing gradient descent on the fitting hypothesis during the forgetting phase leads to the best performance.
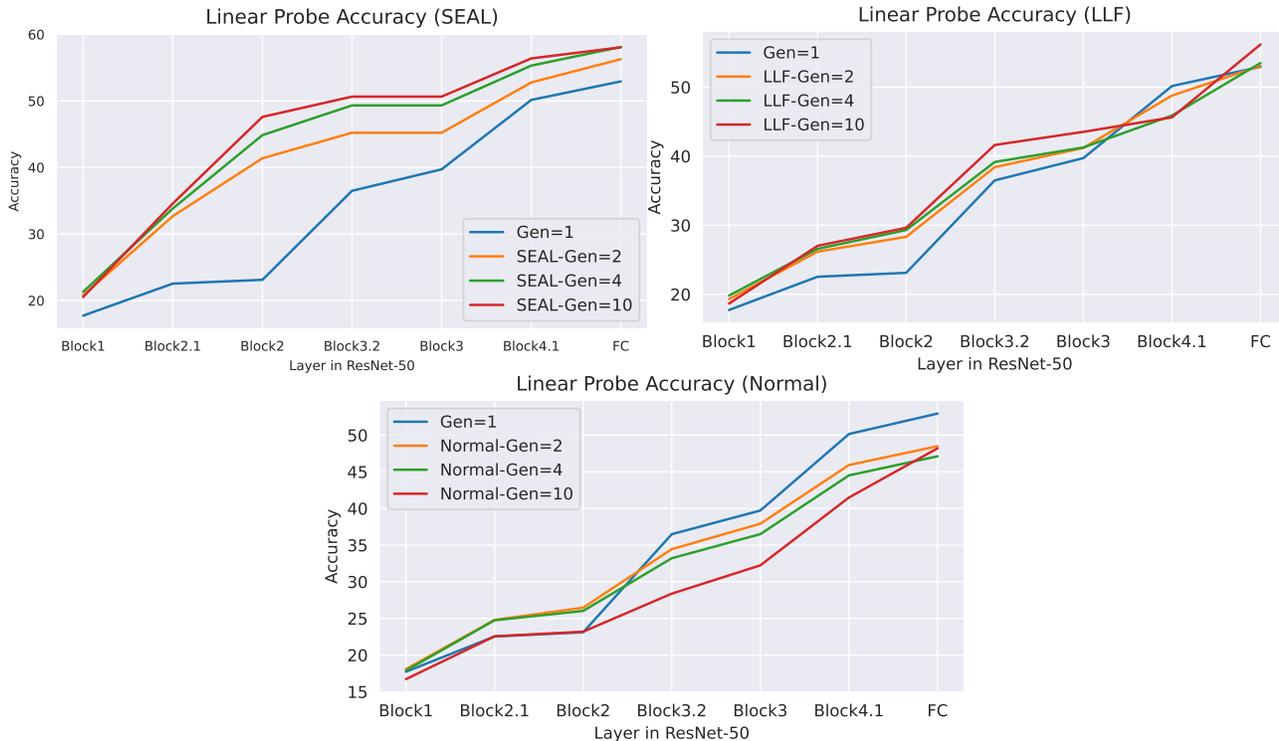
Figure 3. Evolution of Prediction Depth over Epochs for LLF, SEAL, and Normal training. Normal training worsens the prediction depth after the first generation which explains its poor in-distribution performance. LLF slightly improves the prediction depth of the model, however, it hurts the performance of the later layers of the network. SEAL shows the most significant improvement in prediction depth, while the later layers are improving over time.

of the forgetting phase, we reinitialize the fit hypothesis and during this phase, we perform gradient descent on these layers. We observe that not using the re-initialization leads to higher accuracy. Further, we demonstrate that freezing the final layers during the forgetting phase has a negative impact on the training. Finally, in "Ours+Reverse", we swap the fit and forgetting hypotheses, where we perform the gradient ascent on the later layers. We observe that performing simulated annealing in later layers fails drastically. This shows the importance of promoting the early layers and affirms the observations of Baldock et al. [1].

6. Conclusion

In this work, we used the simulated annealing concept (intermittent heating and gradual cooling) in iterative training.

We perform intermittent gradient ascent on the early layers for a few epochs. Following the iterative training literature, we do not perform gradient ascent on all the network parameters to ensure that the model maintains information from its previous state. This allowed us to obtain another perspective on the recently introduced later layer forgetting and the need to reset layers. We show that our method, SEAL, performs better than the state-of-the-art iterative training method, LLF, in an in-distribution setting. Moreover, we observed promising transfer learning performance in both natural image data and popular cross-domain few shot learning benchmarks. Investigating our approach illustrated it can greatly improve network prediction depth. Finally, we demonstrated that current iterative learning methods can have very poor generalization under transfer learning.

References

[1] Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34:10876–10889, 2021. 1, 3, 7, 8

[2] Zhicheng Cai. Sa-gd: Improved gradient descent learning strategy with simulated annealing. *arXiv preprint arXiv:2107.07558*, 2021. 2, 3, 6

[3] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. 3, 4

[4] Yann N. Dauphin, Razvan Pascanu, Çaglar Gülçehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *CoRR*, abs/1406.2572, 2014. 3

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. 3

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 5

[7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 1

[8] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 1, 3

[9] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Stabilizing the lottery ticket hypothesis. *arXiv preprint arXiv:1903.01611*, 2019. 1, 3

[10] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018. 1

[11] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. ECCV, 2020. 2, 3, 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[13] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 3, 4

[14] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997. 6

[15] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. *CoRR*, abs/1703.00887, 2017. 3

[16] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011. 2, 5

[17] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983. 2

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 3

[19] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 2, 4

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2, 4, 5

[22] Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020. 1

[23] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016. 3, 4

[24] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 4

[25] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 2, 4, 5

[26] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. 6

[27] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences, 2021. 3, 5

[28] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009. 2, 4, 5

[29] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 3

[30] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016. 3

[31] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017. 3

[32] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014. 1

[33] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017. 3

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[35] Ahmed Taha, Abhinav Shrivastava, and Larry S Davis. Knowledge evolution in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12843–12852, 2021. 1, 2, 3, 5

[36] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 3, 4

[37] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 1

[38] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. 3

[39] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 4, 5

[40] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 3, 4

[41] 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*, volume 33, pages 5628–5635, 2019. 1

[42] Y. Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007. 1

[43] Moslem Yazdanpanah, Aamer Abdul Rahman, Muawiz Chaudhary, Christian Desrosiers, Mohammad Havaei, Eugene Belilovsky, and Samira Ebrahimi Kahou. Revisiting learnable affines for batch norm in few-shot transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9109–9118, June 2022. 3, 5, 6

[44] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 2

[45] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *Advances in neural information processing systems*, 32, 2019. 3

[46] Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron Courville. Fortuitous forgetting in connectionist networks. *arXiv preprint arXiv:2202.00155*, 2022. 1, 2, 3, 4, 6, 7