Lookaround Optimizer: k steps around, 1 step average
— *Supplementary Material* —

Anonymous Author(s)
Affiliation
Address
`email`

A Exploration in the Loss Landscape

To demonstrate the effectiveness of Lookaround during training, we set up an experiment and visualize the loss landscape of the models under different data augmentations in Figure 1. By observing the loss landscape, we can gain a clearer understanding of the role played by the weight averaging at different stages during the training process.

Training Process and Parameter Settings. We first train a ResNet50 on the CIFAR100 dataset. The learning rate is initialized to 0.1 and decay at 60, 120, and 160 epochs using a MultiStepLR scheduler with a decay factor 0.2. The batch size is set to 128, we use stochastic gradient descent with momentum to optimize the model and use random crops and random vertical flips augmentation to enhance the training datasets. We use model checkpoints at epochs 50, 110, and 150 as our three pretrained models(V-network). The three pretrained models correspond to learning rates of 0.1, 0.02 and 0.004, respectively. Using these pretrained models as the starting point, we finetune the model with 1, 10, 100, 1000, 10000 iterations under the corresponding learning rate and the setting of random horizontal flipping (H-network) or RandAugment (R-network).

Visualization Method. We use the visualization method in [2] to plot the loss landscape. In this method, the weights of the three models are flatten respectively as one-dimensional vectors w_v, w_h, w_r, and then two orthogonal vectors are calculated between the three vectors as the X-axis direction and the Y-axis direction: $u = (w_h - w_v), v = (w_r - w_v, w_h - w_v)/\|w_h - w_v\|^2 \cdot (w_h - w_v)$. Then the normalized vectors $\hat{u} = u/\|u\|, \hat{v} = v/\|v\|$ foam an orthonormal basis in the plane contain w_v, w_h, w_r. Then a point P with coordinates (x, y) in the plane would be given by $P = w_v + x \cdot \hat{u} + y \cdot \hat{v}$.

Discussion and Inspiring. Under different learning rates and different around steps k, Lookaround has the tendency to lead the model trained on different data augmentation to the near-constant loss manifold. In such circumstances, the "average step" can lead the model into the center of the loss basin to get a lower test loss. However, weight averaging does not necessarily work in all cases. For example, the network obtained after weight averaging gets a larger loss under a large learning rate with a large around step (e.g., $lr = 0.1, k = 10000$). In this case, the model is located on a peak between different basins rather than in the center of a basin. Moreover, in the case of around step $k = 1$, the weight averaging also does not achieve better performance. Nevertheless, such extreme cases do not prevent weight averaging from being a useful tool to speed up the training process. The center of the basin in loss landscape, which requires multi-step gradient descent to reach, can be reached by only one weight averaging step. At other learning rates and around steps k, the models after weight averaging all result in a lower loss than the individual model. Such phenomena encourage us to explore more methods to find more optimal solutions in the loss landscape in the future.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

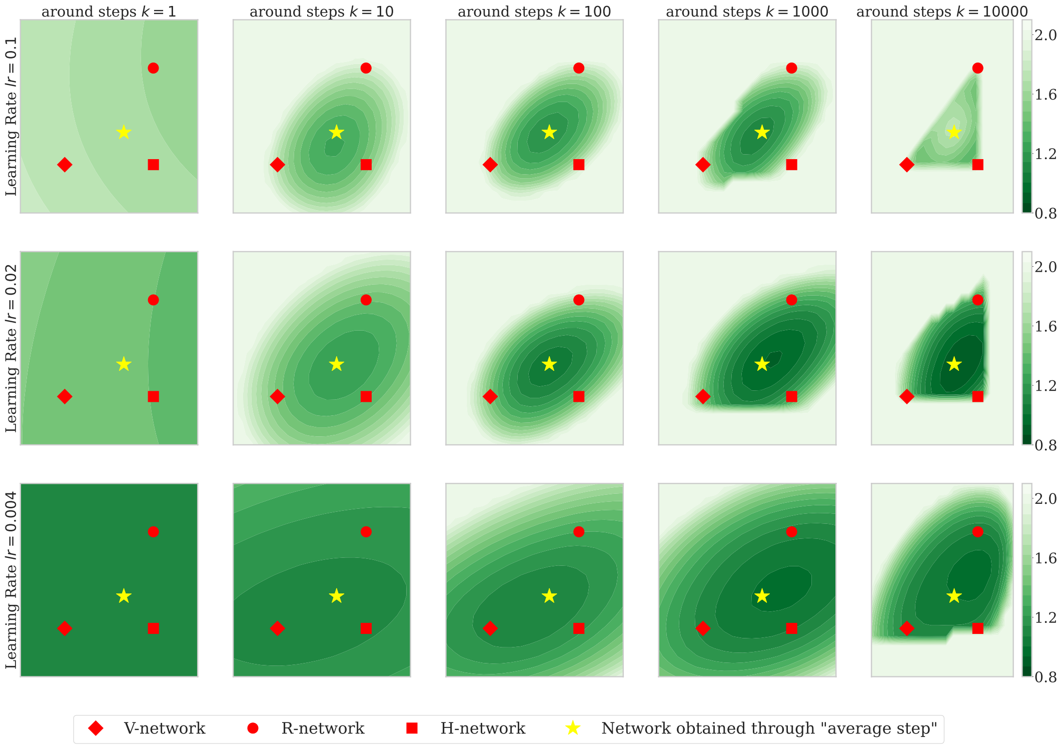

Figure 1: The test loss landscape of ResNet50 on CIFAR100. The diamond block (V-network) represents the pretrained model trained with random vertical flipping. Then we can use random horizontal flipping and RandAugment to finetune V-network to get H-network and R-network.

B Steady-state and Convergence Analysis of Lookaround

We use quadratic functions to analyze the steady-state and convergence analysis of Lookaround. First, we present the proof of Proposition 1.

Proposition 1 (Steady-state risk)**.** *Let $0 < \gamma < 1/L$ be the learning rate satisfying $L = \max_i a_i$. One can obtain that in the noisy quadratic setup, the variances of SGD, Lookahead [7] and Lookaround converge to the following fixed matrix:*

$$V^*_{SGD} = \frac{\gamma^2 \mathbf{A}^2 \Sigma^2}{\mathbf{I} - (\mathbf{I} - \gamma \mathbf{A})^2}, \tag{1}$$

$$V^*_{Lookahead} = \underbrace{\frac{\alpha^2(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k})}{\alpha^2(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k}) + 2\alpha(1-\alpha)(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{k})}}_{\preccurlyeq \mathbf{I},\ \textit{if}\ \alpha\in(0,1)} V^*_{SGD}, \tag{2}$$

$$V^*_{Lookaround} = \underbrace{\frac{\alpha^2(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k}) + 2\alpha(1-\alpha)(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{k})}{\alpha^2(d\mathbf{I} - (d-1)(\mathbf{I} - \gamma\mathbf{A})^{2k})}}_{\preccurlyeq \mathbf{I},\ \textit{if}\ d\geq 3\ \textit{and}\ \alpha\in[1/2,1)} V^*_{Lookahead}. \tag{3}$$

respectively, where α denotes the average weight factor of models with varying trajectory points.

From Wu et al. [6], we have the following conclusions about the dynamics of SGD with learning rate γ:

$$\mathbb{E}[\theta^{(t+1)}] = (\mathbf{I} - \gamma\mathbf{A})\,\mathbb{E}[\theta^{(t)}],$$
$$\mathbb{V}[\theta^{(t+1)}] = (\mathbf{I} - \gamma\mathbf{A})^2\,\mathbb{V}[\theta^{(t)}] + \gamma^2\mathbf{A}^2\Sigma.$$

Lemma 1. *The expectation and variance of lookaround have the following iterates:*

$$\mathbb{E}[\boldsymbol{\phi}^{(t+1)}] = (\mathbf{I} - \gamma\mathbf{A})^k\, \mathbb{E}[\boldsymbol{\phi}^{(t)}], \tag{4}$$

$$\mathbb{V}[\boldsymbol{\phi}^{(t+1)}] = \frac{d-1}{d}(\mathbf{I} - \gamma\mathbf{A})^{2k}\, \mathbb{V}[\boldsymbol{\phi}^{(t)}] + \frac{\gamma^2\mathbf{A}^2\Sigma(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k})}{d(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^2)}. \tag{5}$$

$\boldsymbol{\phi}_t$ yields $\boldsymbol{\phi}_{t+1}$ by performing an around step and an average step.

Proof. The expected iterate follows from SGD:

$$\begin{aligned}\mathbb{E}[\boldsymbol{\phi}^{t+1}] &= \mathbb{E}[\frac{1}{d}\sum_i \theta_{t,i,k}] = \sum_i \frac{1}{d}\,\mathbb{E}[\theta_{t,i,k}] \\ &= \sum_i \frac{1}{d}(\mathbf{I} - \gamma\mathbf{A})^k\, \mathbb{E}[\theta_{t,i,0}] = (\mathbf{I} - \gamma\mathbf{A})^k\, \mathbb{E}[\boldsymbol{\phi}^t].\end{aligned}$$

For the variance of $\boldsymbol{\phi}^{t+1}$, we can break it down into two parts as follows:

$$\mathbb{V}[\boldsymbol{\phi}^{t+1}] = \mathbb{V}[\frac{1}{d}\sum_i \theta_{t,i,k}] = \sum_i^d \frac{1}{d^2}\,\mathbb{V}[\theta_{t,i,k}] + \sum_{i \neq j, 1 \leq i,j \leq d} \frac{1}{d^2}\text{cov}(\theta_{t,i,k}, \theta_{t,j,k}).$$

The covariance of the different models can be calculated in the following way:

$$\begin{aligned}\text{cov}(\theta_{t,i,k}, \theta_{t,j,k}) &= \mathbb{E}[\theta_{t,i,k}\theta_{t,j,k}] - \mathbb{E}[\theta_{t,i,k}]\,\mathbb{E}[\theta_{t,j,k}] \\ &= \mathbb{E}[(\mathbf{I} - \gamma\mathbf{A})^{2k})(\boldsymbol{\phi}^t)^2] - (\mathbf{I} - \gamma\mathbf{A})^{2k}\,\mathbb{E}[\boldsymbol{\phi}^t]^2 \\ &= (\mathbf{I} - \gamma\mathbf{A})^{2k}\,\mathbb{V}[\boldsymbol{\phi}^t].\end{aligned}$$

After permuting and regrouping again, we can obtain the iterate with respect to the variance.

$$\begin{aligned}\mathbb{V}[\boldsymbol{\phi}^{t+1}] &= \sum_{i \neq j, 1 \leq i,j \leq d} \frac{1}{d^2}\text{cov}(\theta_{t,i,k}, \theta_{t,j,k}) + \sum_i^d \frac{1}{d^2}\,\mathbb{V}[\theta_{t,i,k}] \\ &= \frac{1}{d^2}(d^2 - d)(\mathbf{I} - \gamma\mathbf{A})^{2k}\,\mathbb{V}[\boldsymbol{\phi}^t] + \frac{1}{d}[\sum_{i=0}^{k-1}(\mathbf{I} - \gamma\mathbf{A})^{2i}\gamma^2\mathbf{A}^2\Sigma] \\ &= \frac{d-1}{d}(\mathbf{I} - \gamma\mathbf{A})^{2k}\,\mathbb{V}[\boldsymbol{\phi}^t] + \frac{\gamma^2\mathbf{A}^2\Sigma(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k})}{d(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^2)}.\end{aligned}$$

The proof is now complete. □

Remark. From Equation 4, the expectation term for $\boldsymbol{\phi}$ in Lookaround eventually converges to 0, as does Lookahead and SGD.

From Zhang et al. [7], we have the following analysis about the variance fixed point of lookahead with learning rate γ and weight factor α, which represents the average weight factor of models with different trajectory points in the Lookahead optimizer, which is generally (0, 1):

$$V^*_{Lookahead} = \frac{\alpha^2(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k})}{\alpha^2(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k}) + 2\alpha(1-\alpha)(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^k)} V^*_{SGD}. \tag{6}$$

We now derive the fixed point of the variance, proceed with the proof of Proposition 1:

$$V^*_{Lookaround} = \frac{\alpha^2(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^{2k}) + 2\alpha(1-\alpha)(\mathbf{I} - (\mathbf{I} - \gamma\mathbf{A})^k)}{\alpha^2(d\mathbf{I} - (d-1)(\mathbf{I} - \gamma\mathbf{A})^{2k})} V^*_{Lookahead}.$$

Proof.

$$V^*_{Lookaround} = \frac{d-1}{d}(\mathbf{I}-\gamma\mathbf{A})^{2k}V^*_{Lookaround} + \frac{\gamma^2\mathbf{A}^2\Sigma(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k})}{d(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^2)}.$$

$$\begin{aligned}
\Rightarrow V^*_{Lookaround} &= \frac{1}{\mathbf{I}-\frac{d-1}{d}(\mathbf{I}-\gamma\mathbf{A})^{2k}} \frac{\gamma^2\mathbf{A}^2\Sigma(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k})}{d(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^2)} \\
&= \frac{\gamma^2\mathbf{A}^2\Sigma[\mathbf{I}-(\mathbf{I}-\gamma\mathbf{I})^{2k}]}{[d\mathbf{I}-(d-1)(\mathbf{I}-\gamma\mathbf{A})^{2k}][\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^2]} \\
&= \frac{\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k}}{d\mathbf{I}-(d-1)(\mathbf{I}-\gamma\mathbf{A})^{2k}}V^*_{SGD}.
\end{aligned}$$

According to Equation 6, we can deduce that

$$\begin{aligned}
V^*_{Lookaround} &= \frac{\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k}}{d\mathbf{I}-(d-1)(\mathbf{I}-\gamma\mathbf{A})^{2k}}V^*_{SGD} \\
&= \frac{\alpha^2(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k}) + 2\alpha(1-\alpha)(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{k})}{\alpha^2(d\mathbf{I}-(d-1)(\mathbf{I}-\gamma\mathbf{A})^{2k})}V^*_{Lookahead}.
\end{aligned}$$

The proof is now complete. □

B.1 Comparing the dynamics of Lookahead

We now proceed with the proof for the range of constraints variable α in Equation 3. When $d \geq 3$, and $\alpha \in [0.5, 1)$, the Lookaround method can obtain a smaller variance than the Lookahead method:

Proof. Let $B = (\mathbf{I}-\gamma\mathbf{A})^k$, due to $0 < \gamma < 1/L, L = \max_i a_i$, so we can have $B \preccurlyeq \mathbf{I}$. Substituting the matrix B into the expressions for the variance fixed point relation of Lookaround and Lookahead, we can obtain

$$\begin{aligned}
V^*_{Lookaround} &= \frac{\alpha^2(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k}) + 2\alpha(1-\alpha)(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{k})}{\alpha^2(d\mathbf{I}-(d-1)(\mathbf{I}-\gamma\mathbf{A})^{2k})}V^*_{Lookahead} \\
&= \frac{\mathbf{I}-B^2+2\frac{1-\alpha}{\alpha}(\mathbf{I}-B)}{d\mathbf{I}-(d-1)B^2}V^*_{Lookahead} \\
&= \frac{\frac{2-\alpha}{\alpha}\mathbf{I}-B^2-\frac{2-2\alpha}{\alpha}B}{d\mathbf{I}-(d-1)B^2}V^*_{Lookahead}.
\end{aligned}$$

Let the coefficient matrix be denoted as C, when $d \geq 3$, for each diagonal element C_{ii} of C, we can scale the denominator of this expression as follows:

$$C_{ii} \leq \frac{\frac{2-\alpha}{\alpha}-B_{ii}^2-\frac{2-2\alpha}{\alpha}B_{ii}}{3-2B_{ii}^2},$$

Then, we can derive the range of α by restricting the right-hand side expression to be less than or equal to 1.

$$\frac{\frac{2-\alpha}{\alpha}-B_{ii}^2-\frac{2-2\alpha}{\alpha}B_{ii}}{3-2B_{ii}^2} \leq 1,$$

As $0 \leq B_{ii} \leq 1$, we can multiply both sides of the inequality by the denominator, then rearrange and combine like terms to obtain the following form:

$$B_{ii}^2 - \frac{2-2\alpha}{\alpha}B_{ii} + \frac{2-4\alpha}{\alpha} \leq 0.$$

Skipping the detailed steps, we can obtain $\alpha \geq 0.5$ by solving the quadratic equation. Therefore, in the case where $\alpha \in [0.5, 1)$ and $d \geq 3$ ($\alpha < 1$ is subject to Lookahead's settings), the coefficient matrix $C \preccurlyeq \mathbf{I}$, so the convergence speed of Lookaround is slower than Lookahead.

□

B.2 Deterministic quadratic convergence

Since our method samples data under multiple data augmentations, it is approximately seen as the average sampling of historical trajectories in the convergence analysis of quadratic functions. In such a perspective, we compare the convergence rates of Lookaround and Lookahead.

We first show the state transition equation for the classical momentum method in quadratic functions:

$$\mathbf{v}_{t+1} = \beta \mathbf{v}_t - \nabla_\theta f(\theta_t) = \beta \mathbf{v}_t - \mathbf{A}\theta_t, \tag{7}$$

$$\theta_{t+1} = \theta_t + \gamma \mathbf{v}_{t+1} = \gamma\beta \mathbf{v}_t + (\mathbf{I} - \gamma \mathbf{A})\theta_t. \tag{8}$$

Here, $\mathbf{v}$ stands for the momentum term. We can generalize this to matrix form:

$$\begin{bmatrix} \mathbf{v}_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} \beta & -\mathbf{A} \\ \gamma\beta & \mathbf{I} - \gamma\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{v}_t \\ \theta_t \end{bmatrix}.$$

Thus, given the initial θ_0, we can obtain the convergence rate with respect to θ by the maximum eigenvalue of the matrix. Referring to Zhang et al. [7] and Lucas et al. [4], we obtain the state transition matrix regarding our algorithm as follows:

$$\begin{bmatrix} \boldsymbol{\theta}_{t,0} \\ \boldsymbol{\theta}_{t-1,k} \\ \vdots \\ \boldsymbol{\theta}_{t-1,1} \end{bmatrix} = LB^{(k-1)}T \begin{bmatrix} \boldsymbol{\theta}_{t-1,0} \\ \boldsymbol{\theta}_{t-2,k} \\ \vdots \\ \boldsymbol{\theta}_{t-2,1} \end{bmatrix},$$

where L, B and T denote the average weight matrix, the single-step transfer matrix and the position transformation matrix respectively:

$$L = \begin{bmatrix} \frac{1}{k+1}I & \frac{1}{k+1}I & \cdots & \frac{1}{k+1}I & \frac{1}{k+1}I \\ I & 0 & \cdots & \cdots & 0 \\ 0 & I & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & I & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} (1+\beta)I - \eta A & -\beta I & 0 & \cdots & 0 \\ I & 0 & \cdots & \cdots & 0 \\ 0 & I & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & I & 0 \end{bmatrix},$$

$$T = \begin{bmatrix} I - \eta A & \beta I & -\beta I & 0 & \cdots & 0 \\ I & 0 & \cdots & \cdots & 0 & \vdots \\ 0 & I & \ddots & \cdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ \vdots & \cdots & 0 & I & 0 & 0 \\ 0 & \cdots & 0 & 0 & I & 0 \end{bmatrix}.$$

After specifying the appropriate parameters and performing matrix multiplication to obtain the state transition matrix, the convergence rate ρ can be obtained by calculating the largest eigenvalue of the matrix. Note that since this linear dynamical system corresponds to k updates, we finally have to compute the k^{th} root of the eigenvalues to recover the correct convergence bounds.

C Experimental Detail

C.1 Random Initialization

C.1.1 CIFAR10 and CIFAR100

Data augmentation details. For the CIFAR10 dataset, we use [RandomCrop + $*$] for data augmentation, and for the CIFAR100 dataset, we use [RandomCrop + $*$ + RandomRotation] for data augmentation. $*$ can be replaced by three different data augmentation methods of random horizontal flip, random vertical flip, or RandAugment [1].

Training details. For the CIFAR10 and CIFAR100 datasets, we have applied some common settings. The initial learning rate is set to 0.1, and the batch size is set to 128. Additionally, a warm-up phase of 1 epoch is implemented. Subsequently, different learning rate schedulers are used based on the specific dataset. For the CIFAR100 dataset, we utilize the MultiStepLR scheduler. The learning rate is decayed at the 60, 120, and 160 epochs, with a decay factor of 0.2. For the CIFAR10 dataset, we employ the CosineAnnealingLR learning rate scheduler. In comparison with other optimizers or optimization methods, we use the default settings of the other methods in the corresponding papers.

C.1.2 ImageNet.

For the ImageNet dataset, we use [RandomResizedCrop + $*$ + RandomRotation] for data augmentation. $*$ can be replaced by three different data augmentation methods of random horizontal flip, random vertical flip, or RandAugment [1]. We train the model with the following settings: an initial learning rate of 0.1, a batch size of 256, and a warm-up phase of 1 epoch. We train the model for a total of 90 epochs. We utilize the MultiStepLR scheduler with decay steps at 30 and 60 epochs, and a decay factor of 0.1.

C.2 Finetuning

In this stage, all images are resized to 224*224 pixels to fit the input size of the pretrained model, all models use the ImageNet-1k pretrained weights from the PyTorch library, and other settings remain the same as in C.1.1. For the training of ViT-B/16, we utilize the Adam optimizer with an initial learning rate of 0.001. β_1 is set to 0.9, β_2 is set to 0.999, and the weight decay is set to 0.00005. To reduce memory consumption, we employed a batch size of 64.

C.3 Compared with Ensemble Method

We compare Lookaround with Logit Ensemble and Snapshot Ensemble [3]. In the setting of Logit Ensemble, we train multiple models separately using different data augmentation methods, and then average the outputs of these models for prediction. However, this approach requires more inference time. In the setting of Snapshot Ensemble, we use the CosineAnnealingWarmRestarts learning rate scheduler to collect four snapshots during the training process. Then, we average the outputs of these different snapshots for prediction. This approach also requires more inference time.

C.4 Ablation Study

In ablation experiments, we consider the effectiveness of independent components Data Augmentation (DA) and Weight Averaging (WA) for Lookaround, respectively. In the ablation experiments without data augmentation, in order to improve the competitiveness of the corresponding experiments, we choose to train different models independently using different data augmentation methods, and then select the model that performs best on the test dataset.

C.5 Additional Analysis

In the robustness experiments with the number of Data Augmentation methods, the six data augmentation methods are given as: RandomVerticalFlip, RandomHorizontalFlip, RandAugment, AutoAugment, RandomPerspective, RandomEqualize. All the Augmentation methods are from the Pytorch library. When more data augmentation methods are used, the training time will be correspondingly increased in our method. Therefore, in the main experiments in this paper, in order to reduce the time consumption, we only select three data augmentation methods for comparison.

D Relationship with Model Soups

Model Soups [5] is a framework for finetuning a common pretrained model using different hyperparameters and then averaging the weights of different finetuned models to improve model performance and generalization. We conduct experiments under two data augmentation methods to compare the performance gap between Lookaround and Model Soups.

As shown in Figure 2, Model Soups is effective with a few finetuning epochs, achieving better performance than individual models (θ_1 or θ_2). However, as the number of training epochs increases, its robustness significantly decreases, leading to a result that is unacceptable. We can observe that the accuracy after 50 epochs is almost near 0. Different models in Model Soups easily fall into different loss basins and cannot be connected linearly. On the contrary, Lookaround continuously performs weight averaging during training to maintain the locality of different models for linear mode connectivity, resulting in better robustness and generalization performance.

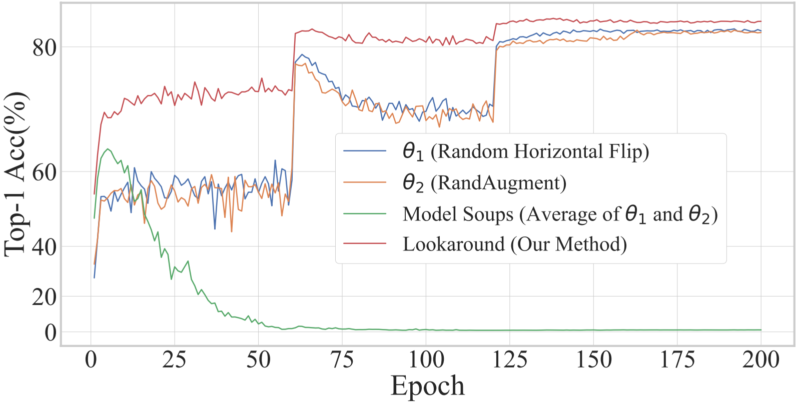

Figure 2: Top-1 accuracy curves of ResNet50 on CIFAR100 under Lookaround and Model Soups. The initial weights of the ResNet50 model are obtained from the pretrained weights on the ImageNet.

References

[1] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *Conference on Neural Information Processing Systems*, 2019.

[2] Timur Garipov, Pavel Izmailov, D. A. Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Conference on Neural Information Processing Systems*, 2018.

[3] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *International Conference on Learning Representations*, 2017.

[4] James Lucas, Shengyang Sun, Richard Zemel, and Roger Grosse. Aggregated momentum: Stability through passive damping. In *International Conference on Learning Representations*, 2018.

[5] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

[6] Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger Grosse. Understanding short-horizon bias in stochastic meta-optimization. *arXiv preprint arXiv:1803.02021*, 2018.

[7] Michael R. Zhang, James Lucas, Jimmy Ba, and Geoffrey E. Hinton. Lookahead optimizer: k steps forward, 1 step back. *Conference on Neural Information Processing Systems*, 2019.