Low-Pass Filtering SGD for Recovering Flat Optima in the Deep Learning Optimization Landscape (Supplementary Material)

A Network balancing

We consider balanced networks, i.e., networks where norms of weights in each layer are roughly the same. In this section, we present a normalization scheme utilized in Section 3 to balance the network. Let x be the input to the network, θ_i be the weight matrix of the i^{th} layer, $\hat{\theta}_i$ denote bias matrix and $\sigma(\ )$ denote the relu nonlinearity. The output of a network with three layers; convolution, batch normalization and relu can be written as

$$f(x) = \sigma\Big(\frac{(\theta_1 X) - E[\theta_1 X]}{Var(\theta_1 X)}\theta_2 + \hat{\theta}_2\Big). \tag{A.1}$$

Let D_i denote a diagonal normalization matrix associated with the i^{th} layer. The diagonal elements of the matrix are defined as $D_i[j,j] = \frac{1}{||\theta_i^j||_F + ||\hat{\theta}_i^j||_F}$, where θ_i^j is the weight matrix of j^{th} filter in the i^{th} layer. We normalize the parameters of the network as,

$$f(x) = \sigma\Big(\frac{(D_1 W_1 X) - E[D_1 W_1 X]}{Var(D_1 W_1 X)} D_2(W_2 + B_2) D_2^{-1}\Big) \tag{A.2}$$

Note that $\hat{W}_i = D_i W_i (D_i)^{-1} = W_i$. Since $\sigma(\lambda x) = \lambda\sigma(x)$ for $\lambda \geq 0$, we can rewrite the above equation as

$$f(x) = \sigma\Big(\frac{(D_1 W_1 X) - E[D_1 W_1 X]}{Var(D_1 W_1 X)} D_2(W_2 + B_2)\Big) D_2^{-1}. \tag{A.3}$$

We keep the multiplication with the matrix D_2^{-1} as a constant parameter in the network but it can also be combined with the parameters of the next layer. We normalize the parameters of each layer as we move from the first layer to the last layer of the network. Figure 5 shows filter wise parameter norm (D^{-1}) of LeNet and ResNet18 models trained on MNIST and CIFAR-10 data sets respectively. In Table 5, we show the mean training cross entropy loss before and after normalization.

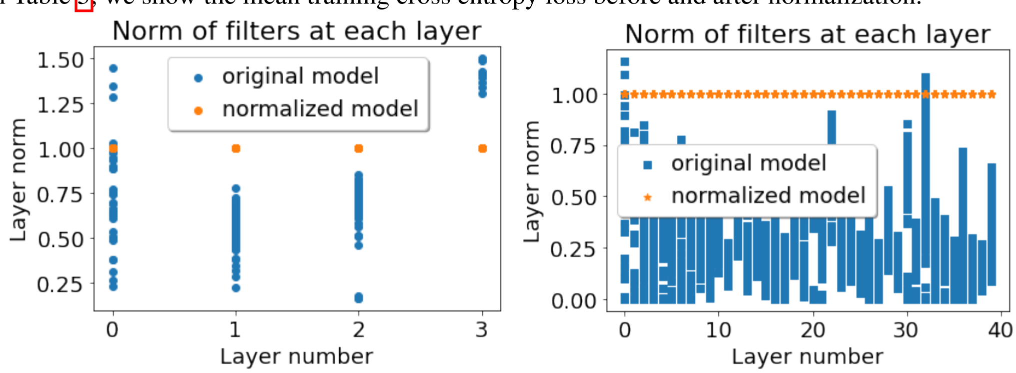

Figure 5: Norm of the filter in each layer of (**left**) LeNet and (**right**) ResNet18 networks trained on CIFAR-10 data set before and after normalization.

Model	Loss before normalization	Loss after normalization
LeNet	0.0006375186720272088	0.0006375548382939456
ResNet18	0.0014627227121118522	0.0014617550651283215

Table 5: Validation loss for CIFAR-10 data set before and after normalization.

B Computing Sharpness measures

In this section, we describe various algorithms to compute the sharpness measures presented in section 3.1.

B.1 Hessian based measures ($\lambda_{max}(H)$, $Trace(H)$, $d_{eff}(H)$, and $\|H\|_F$)

We compute 100 eigenvalues of the Hessian of the loss function using Stochastic Lanczos quadrature algorithm as described in [118]. $\lambda_{max}(H)$, $Trace(H)$, and $d_{eff}(H)$ can be easily estimated from the set of 100 eigen values. Note that for any matrix A, $||A||_F^2 = \mathbf{E}_v[||Av||_2^2]$, where $v \sim \mathcal{N}(0, I)$. Therefore, we use the algorithm the following algorithm to efficiently compute the Frobenius norm.

Algorithm $||H||_F$

Input: M: number of iterations, $hvp(v)$: Hessian-vector product
Output: $||H||_F$
  $out \leftarrow 0$
  for k = 1 to M **do**
    $v^k \sim \mathcal{N}(0, I)$
    out += $||hvp(v^k)||_2^2$
  end for
  return $\sqrt{out/M}$

B.2 Fisher Rao Norm

FRN is calculated as $\theta^{*T} hvp(\theta^*)$.

B.3 Norm of the Gradient of Local Entropy

It is prohibitive to compute the local entropy, as opposed to its gradient. We utilize the EntropySGD algorithm described in [5] however, instead of updating weight we compute the norm of the gradient.

Algorithm μ_{LE}

Input: θ^*: final weights, L: Langevin iterations, γ : scope, η : step size, ϵ : noise level
Output: μ_{LE}
  $\theta', \mu \leftarrow \theta^*$
  for k = 1 to L **do**
    B $\leftarrow$ sample mini batch
    $g = \nabla_{\theta'} L(\theta', B) - \gamma(\theta - \theta')$
    $\theta' \leftarrow \theta' - \eta g + \sqrt{\eta}\epsilon\mathcal{N}(0, I)$
    $\mu \leftarrow (1 - \alpha)\mu + \alpha\theta'$
  end for
  return $||\gamma(\theta^* - \mu)||$

B.4 ϵ-sharpness

Algorithm ϵ - sharpness

Input: θ^*: final weights, ψ: tolerance, ϵ: target deviation in loss
Output: ϵ - sharpness

```
η_max =FLOAT_EPSILON_MIN
while TRUE do
    θ = θ* + η_max ∇L(θ*)
    d = L(θ) − L(θ*)
    if d < ε then
        η_max = η_max * 10
    end if
end while
η_min =FLOAT_EPSILON_MIN
while TRUE do
    η = (η_max + η_min)/2
    θ = θ* + η∇L(θ*) \\ step in full-data gradient direction
    d = L(θ) − L(θ*)
    if ε − ψ ≤ d ≤ ε + ψ then
        return 1/||θ−θ*||
    end if
    if d < ε − ψ then
        η_min = η
    else if d > ε + ψ then
        η_max = η
    end if
end while
```

B.5 PAC-Bayes measure

Algorithm $\mu_{PAC-Bayes}$

Input: θ^*: final weights, M: MC iterations, ψ: tolerance
Output: σ

```
σ_min = FLOAT_EPSILON_MIN
σ_max = FLOAT_EPSILON_MAX
while TRUE do
    σ = (σ_min + σ_max))/2
    l̂ = 0
    for j = 1 to M do
        θ = θ* + N(0, σ²I)
        l̂+ = L̂(θ)
    end for
    l̂ = l̂/M
    d = l̂ − L(θ*)
    if ε − ψ ≤ d ≤ ε + ψ then
        return σ
    end if
    if d < ε − ψ then
        σ_min = σ
    else if d > ε + ψ then
        σ_max = σ
    end if
end while
```

B.6 Shannon Entropy

Algorithm Shannon Entropy

Input: f_{θ^*}: trained model
Output: Shannon Entropy

out $\leftarrow$ 0
for i = 1 to N **do**
  for j = 1 to K **do**
    out$+ = f_{\theta^*}(x_i)[j] \times log(f_{\theta^*}(x_i)[j])$
  end for
end for
return -out / N

B.7 LPF

Algorithm LPF

Input: θ^*: final weights, σ: standard deviation of Gaussian filter kernel, M: MC iterations
Output: $(L \circledast K)(\theta^*)$

$out \leftarrow 0.0$
for k = 1 to M **do**
  $\tau = \mathcal{N}(0, \sigma I)$
  $out+ = L(\theta^* + \tau)$
end for
return $out/ = M$

C Sensitivity of the sharpness measures to the changes in the curvature of the synthetically generated landscapes

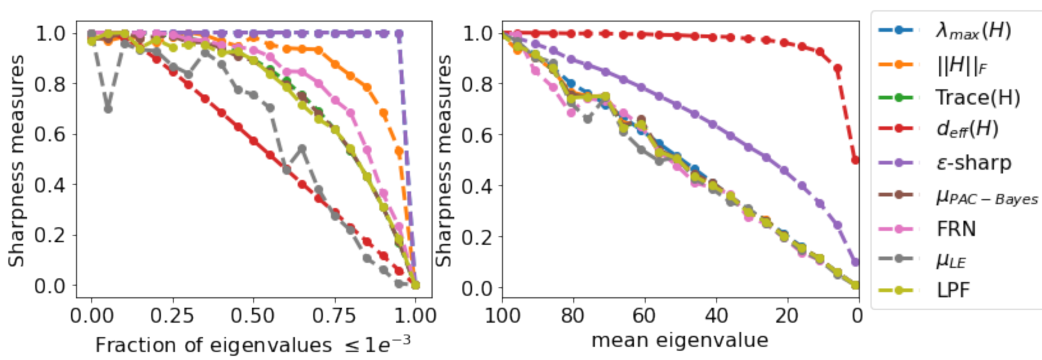

Figure 6: **Left**: The behavior of the normalized sharpness measures when the fraction of the eigenvalues of the Hessian below $1e-3$ increases from 0 to 1. **Right**: The behavior of the normalized sharpness measures when the mean eigenvalue of the Hessian is decreased from 100 to 1.

We consider a quadratic minimization problem:

$$\min_\theta f(\theta), \quad \text{where} \quad f(\theta) = \frac{\theta^T H \theta}{2}. \tag{C.1}$$

Note that $\nabla f = H\theta$, $\nabla^2 f = H$ and $\theta^* = \arg\min_\theta \frac{\theta^T H \theta}{2} = 0$.

In the first experiment, we randomly sample the Hessian matrix H of dimension 100 and set its K smallest eigenvalues uniformly in the interval $[1e-5, 1e-3]$. As the value of K is increased from 0 to 100, the loss surface becomes flatter. In the second experiment, we set the eigenvalues of Hessian H uniformly as $\mathcal{U}(K - 0.10 * K, K + 0.10 * K)$, where K is the mean eigenvalue. Intuitively, as the value of K is decreased from 100 to 1, the loss surface becomes flatter.

On the left plot in Figure 6 we show the value of the normalized sharpness measures against the fraction of eigenvalues that are $< 1e-3$. Thus as we move on the x-axis of this plot from left to right, the number of directions along which the loss landscape is flat increases. In this case LPF is the second best measure, after $d_{eff}(H)$, where by a good measure we understand the one that is sensitive to the changes in the loss landscape. On the right plot in Figure 6 we show the value of the normalized sharpness measures against the mean eigenvalue of the Hessian. Thus as we move on the x-axis of this plot from left to right, the landscape along all directions becomes flatter. In this case all measure, except ϵ-sharpness and $d_{eff}(H)$, are sensitive to the changes in the loss landscape. $d_{eff}(H)$ shows poor sensitivity to those changes. These experiments also well justify the choice of LPF based sharpness measure for the algorithm proposed in this paper.

D Training details for Section 3

D.1 Sharpness vs Generalization (training details for Section 3.2)

Following the experimental framework presented in [12, 119], we trained 2916 ResNet18 models on CIFAR-10 data set by varying different model and optimizer hyper-parameters and 3 random seeds. Each model was trained using cross entropy loss function and SGD optimizer for 300 epochs. The learning rate set to 0.1 and dropped by a factor of 0.1 at epoch 100 and 200. Since each model is trained with different hyper-parameters it is easy to overfit some models while under-fitting others. To mitigate this effect, we train each model until the cross entropy loss reaches the value of ≈ 0.01. Any model that does not reach this threshold is discarded from further analysis. We compute Kendall ranking correlation coefficient between the hyper-parameters and generalization gap and report the results in Table 6).

Measure	mo	width	wd	lr	bs	skip	bn
Emp order	-0.9712	-0.6801	-0.3135	-0.7930	0.9877	-0.2692	-0.0955

Table 6: Ranking correlation between hyper-parameter and generalization gap. The correlation sign is consistent with our intuitive understanding.

After convergence, we balance each network according to the normalization scheme presented in section A and compute sharpness measures using algorithms presented in section B. All the models were trained on NVIDIA RTX8000, V100 and RTX1080 GPUs on our high performance computing cluster. The total computational time is ~ 9000 GPU hours.

D.2 Sharpness vs Hyper-parameters (additional experimental results for Section 3.2)

As highlighted in section 3.2, we compute the Kendall ranking correlation between hyper-parameter and sharpness measures (Table 7) on ResNet18 models trained on CIFAR-10 data set as described section D.1. We observe that momentum and weight decay are strongly negatively correlated to sharpness i.e increasing both hyper-parameters leads to flatter solution. It is also widely observed that increasing both these parameters also lead to lower generalization gap. Therefore, the table can provide us guidelines on how to design or modify deep architectures. This direction of research will be investigated in the future work.

Measure	mo	width	wd	lr	bs	skip	bn
$\lambda_{\max}(H)$	-0.891	-0.063	-0.291	-0.692	0.981	0.263	0.996
$\|H\|_F$	-0.930	0.029	-0.474	-0.826	0.994	0.218	0.996
Trace (H)	-0.942	-0.127	-0.381	-0.745	0.984	-0.199	0.987
d_{eff}	-0.360	-0.137	-0.147	-0.139	0.335	-0.268	0.047
ϵ-sharpness	-0.781	0.147	-0.321	-0.772	0.967	0.509	1.000
$\mu_{PAC-Bayes}$	-0.994	0.981	-0.669	-0.971	0.996	0.322	0.996
FRN	-0.824	-0.226	-0.037	-0.545	0.855	-0.605	1.000
Shannon Entropy	-0.723	-0.174	0.246	-0.352	0.718	0.613	0.950
μ_{LE}	-0.169	0.954	-0.036	-0.112	0.117	0.013	0.241
LPF	-0.994	0.874	-0.767	-0.934	0.998	-0.543	0.954

Table 7: Kendall rank correlation coefficient between various sharpness measures (rows) and hyper-parameters (columns).

D.3 Training details and additional experimental results for Section 3.3 (Sharpness versus generalization under data and label noise)

In order to evaluate the performance of sharpness measures to explain generalization in presence of data and label noise, we trained 10 ResNet18 models with varying level of label noise and 20 ResNet18 model with varying level of data noise on the CIFAR-10 [100] dataset (Section 3.3 in the main paper). All models were trained for 350 epochs using cross entropy loss and SGD optimizer with a batch size of 128, weight decay of $5e^{-4}$ and momentum set to 0.9. The learning rate was set to 0.1 and dropped by a factor of 0.1 at epoch 150 and 200. The models were trained on NVIDIA RTX8000, V100 and RTX1080 GPUs on our high performance computing cluster. The total computational time is $\sim$ 600 GPU hours. In Figure 7, we plot the values of normalized sharpness measures and generalization gap (averaged over 5 seeds) for varying level of data noise. We also report the Kendall rank correlation coefficient in the figure title.

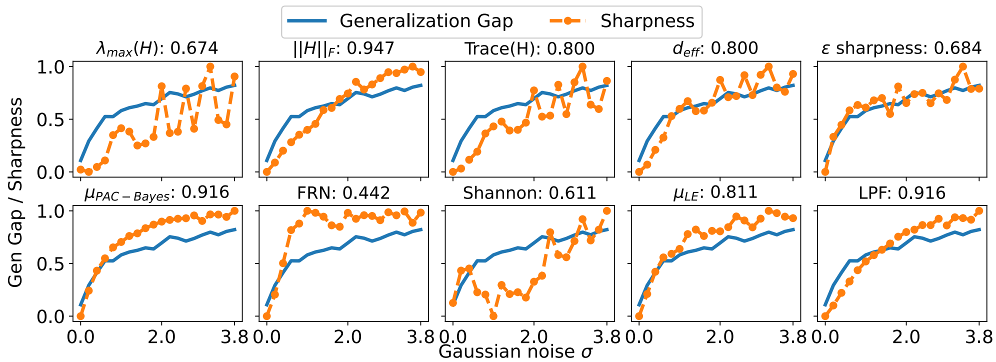

Figure 7: Normalized sharpness measures and generalization gap for varying levels of data noise. Kendall rank correlation coefficient between generalization gap and sharpness with increasing data noise are provided in the parenthesis of figure titles.

D.4 Training details for Section 3.4 (Sharpness and double descent phenomenon)

As highlighted in section 3.4 in the main paper, we evaluate sharpness measures against the double descent error curve of DNNs. We follow the experimental framework presented in [18] and set the widths of the consecutive layers of the Resnet18 model as $[k, 2k, 4k, 8k]$. The value of k is varied in the range $[1, 64]$. Note that for a standard Resnet18 model $k = 64$. The models were trained on CIFAR-10 data set for 4000 epochs with 20% label noise using an Adam optimizer [22] with a batch size of 128 and a constant learning rate set to $1e^{-4}$. All the models were trained on NVIDIA RTX8000, V100 and RTX1080 GPUs on our high performance computing cluster. The total compute time is $\sim$ 300 GPU hours.

E Training details for Section 6

We coded all our experiments in PyTorch. In all the experiments, we utilized the code for the SAM optimizer [1] available at `https://github.com/davda54/sam` that we treated as a baseline code for developing LPF-SGD. In terms of SGD, this optimizer is included in the PyTorch environment.

In the first set of experiments, we trained ResNet18 model available in torchvision [120] on TinyImageNet [121] and ImageNet [121] data sets, and a modified version of ResNet18,50,101 [99] available at `https://github.com/kuangliu/pytorch-cifar` on CIFAR-10 [100] and CIFAR-100 [110] data sets. The modification was small and was only done to accommodate 32×32 image sizes in CIFAR data set. We also train a LeNet model available at `https://github.com/pytorch/examples/blob/master/mnist/main.py` on MNIST [112] data set. The hyper-parameters common to SGD, SAM, and LPF-SGD optimizers are provided in the Table 8, while the individual hyper-parameters of SAM and LPF-SGD are provided in Table 9. The models were trained on NVIDIA RTX8000, V100 and RTX1080 GPUs on our high performance computing cluster. The total computational time is $\sim$ 1500 GPU hours.

After convergence, we balance our network using the normalization scheme described in section A and compute various sharpness measures on the best performing model. Table 10 shows normalized sharpness measures and the corresponding validation error. Note that LPF-SGD leads to a lower value of the LPF based sharpness measure and a smaller error.

Finally, the plots showing epoch vs error curves are captured in Figure 8 and Figure 9.

Dataset	Model	BS	WD	MO	Epochs	LR (Policy)
MNIST	LeNet	128	$5e^{-4}$	0.9	150	0.01 (x 0.1 at ep=[50,100])
CIFAR10, 100	ResNet-18, 50, 101	128	$5e^{-4}$	0.9	200	0.1 (x 0.1 at ep=[100,120])
TinyImageNet	ResNet-18	128	$1e^{-4}$	0.9	100	0.1 (x 0.1 at ep=[30, 60, 90])
ImageNet	ResNet-18	256	$1e^{-4}$	0.9	100	0.1 (x 0.1 at ep=[30, 60, 90])

Table 8: Training hyper-parameters common to all optimizers used for obtaining Table 3. BS: batch size, WD: weight decay, and MO: momentum coefficient.

Dataset	Model	SAM	LPF-SGD	
		ρ (policy)	M	γ (policy)
MNIST	LeNet	0.05 (fixed)	1	0.001(fixed)
CIFAR	ResNet18,50,101	0.05 (fixed)	1	0.002 (fixed)
TinyImageNet	ResNet18	0.05 (fixed)	1	0.001 (fixed)
ImageNet	ResNet18	0.05 (fixed)	1	0.0005 (fixed)

Table 9: Summary of SAM and LPF-SGD hyper-parameters used for obtaining Table 3.

Data	Model	Opt	$\|\|H\|\|_F$	ϵ-sharp	μ_{PAC}	FRN	Shannon	LPF	val-err
CIFAR10	ResNet18	SGD	1.00	0.52	1.00	1.00	1.00	1.00	11.49
		SAM	0.34	0.36	0.70	0.45	0.61	0.40	10.00
		LPF-SGD	0.71	1.00	0.58	0.46	0.21	0.17	9.04
	ResNet50	SGD	1.00	0.43	1.00	1.00	1.00	1.00	10.21
		SAM	0.37	0.41	0.66	0.64	0.60	0.45	8.81
		LPF-SGD	0.79	1.00	0.88	0.56	0.24	0.28	8.60
	ResNet101	SGD	1.00	0.59	1.00	1.00	1.00	1.00	9.49
		SAM	0.36	0.46	0.70	0.59	0.67	0.51	8.33
		LPF-SGD	0.75	1.00	0.99	0.49	0.33	0.34	8.69
CIFAR100	ResNet18	SGD	0.18	0.64	1.00	0.29	0.65	1.00	38.29
		SAM	0.09	0.47	0.78	0.20	0.50	0.52	36.17
		LPF-SGD	1.00	1.00	0.59	1.00	1.00	0.90	30.02
	ResNet50	SGD	0.54	0.50	1.00	0.62	1.00	1.00	35.55
		SAM	0.18	0.54	0.78	0.27	0.64	0.46	33.15
		LPF-SGD	1.00	1.00	0.53	1.00	0.65	0.41	30.64
	ResNet101	SGD	1.00	0.56	1.00	1.00	0.34	1.00	32.73
		SAM	0.46	0.43	0.64	0.55	0.25	0.41	30.70
		LPF-SGD	0.59	1.00	0.44	0.44	1.00	0.12	29.89

Table 10: Normalized sharpness measures and validation error for ResNet18,50,101 models trained on CIFAR-10 and CIFAR-100 data sets using standard SGD, SAM and LPF-SGD.

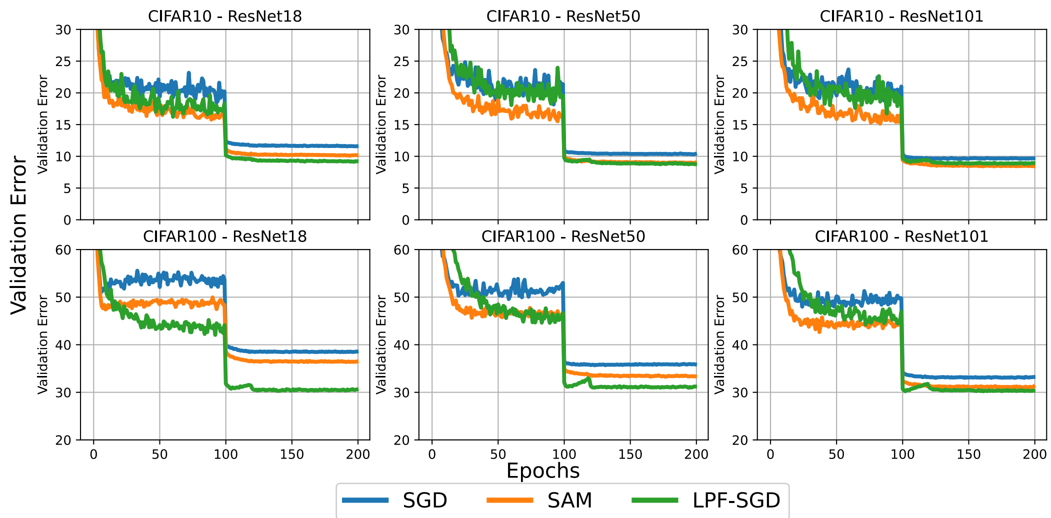

Figure 8: Validation error vs epochs for ResNet18 (left), ResNet50 (middle), and ResNet101 (right) models trained on CIFAR-10 (top) and CIFAR-100 (bottom) data sets using SGD, SAM, and LPF-SGD.

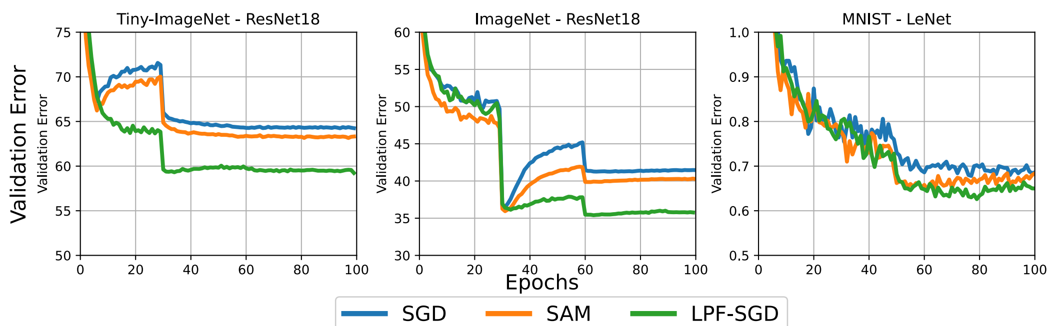

Figure 9: Validation error vs epochs for TinyImageNet (left), ImageNet (middle), and MNIST (right) data sets. TinyImage and ImageNet data sets was used to train the ResNet18 model while MNIST data set was used to train the LeNet model.

In the second set of experiments, we trained WRN16-8 and WRN28-10 [113] models available at `https://github.com/xternalz/WideResNet-pytorch`, ShakeShake (26 2x96d) [114] available at `https://github.com/hysts/pytorch_shake_shake`, and PyramidNet-110(α= 270) [115] and PyramidNet-273(α=200) [115] models available at `https://github.com/dyhan0920/PyramidNet-PyTorch`. We utilized three progressively increasing augmentation schemes: basic (random cropping and horizontal flipping), basic + cutout [116], and basic + cutout + auto-augmentation [117]. The cutout scheme is available at `https://github.com/davda54/sam` and the auto-augmentation scheme is available at `https://github.com/4uiiurz1/pytorch-auto-augment`. Table 11 shows various hyper-parameters common to SGD, SAM, and LPF-SGD optimizers, and Table 12 shows individual hyper-parameters for LPF-SGD optimizer (for SAM hyperparameter ρ is fixed to 0.05 and thus we do not report it in the table). In Figures 10, 11, 12, 13, and 14 we provide error vs epoch curves. All the models were trained on NVIDIA RTX8000, V100 and RTX1080 GPUs on our high performance computing cluster. The total computational time is $\sim$ 6000 GPU hours.

Model	BS	WD	MO	Epochs	LR(Policy)	
					CIFAR-10	CIFAR-100
WRN16-8	128	$5e^{-4}$	0.9	200	0.1($\times$ 0.2 at [60,120,160])	
WRN28-10	128	$5e^{-4}$	0.9	200	0.1($\times$ 0.2 at [60,120,160])	
ShakeShake (26 2x96d)	128	$1e^{-4}$	0.9	1800	0.2(cosine decrease)	
PyNet110	128	$1e^{-4}$	0.9	200	0.1($\times$ 0.1 at [100,150])	0.5($\times$ 0.1 at [100,150])
PyNet272	128	$1e^{-4}$	0.9	200	0.1($\times$ 0.1 at [100,150])	

Table 11: Training hyper-parameters common to all optimizers used to obtain Table 4. BS: batch size, WD: weight decay, MO: momentum coefficient, and LR: learning rate.

Model	Aug	M	CIFAR-10		CIFAR-100	
			γ_0	α (policy)	γ_0	α (policy)
WRN16-8	Basic	8	0.0005	15	0.0005	15
	Basic+Cut	8	0.0005	15	0.0005	15
	Basic+Cut+AA	8	0.0005	15	0.0005	15
WRN28-10	Basic	8	0.0005	35	0.0005	25
	Basic+Cut	8	0.0005	35	0.0005	25
	Basic+Cut+AA	8	0.0005	35	0.0007	15
ShakeShake 26 2x96d	Basic	8	0.0005	15	0.0005	15
	Basic+Cut	8	0.0005	15	0.0005	15
	Basic+Cut+AA	8	0.0005	15	0.0005	15
PyNet110	Basic	8	0.0005	15	0.0005	15
	Basic+Cut	8	0.0005	15	0.0005	15
	Basic+Cut+AA	8	0.0005	15	0.0005	15
PyNet272	Basic	8	0.0005	15	0.0005	15
	Basic+Cut	8	0.0005	15	0.0005	15
	Basic+Cut+AA	8	0.0005	15	0.0005	15

Table 12: Hyper-parameters for LPF-SGD optimizer.

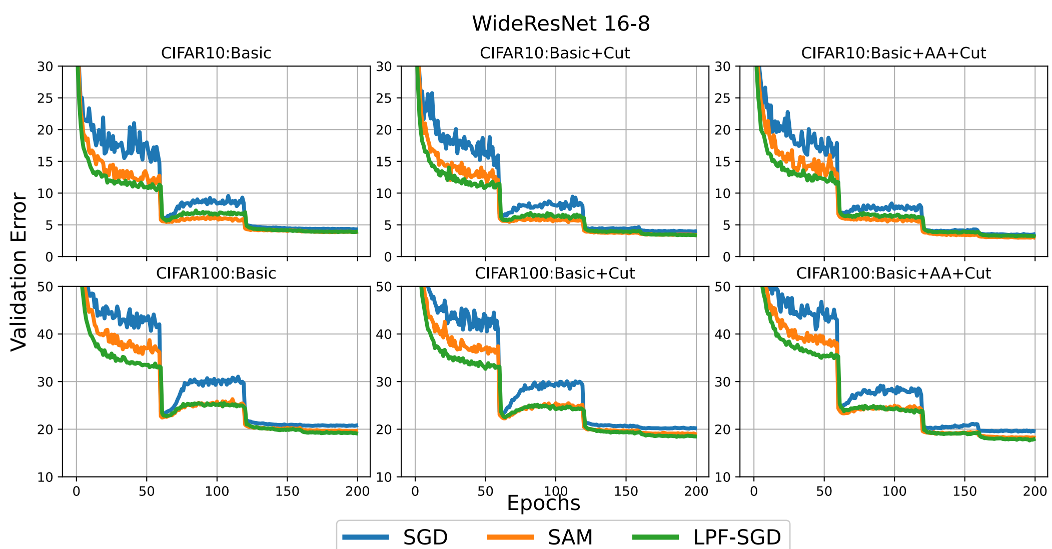

Figure 10: Validation error vs epochs for WideResNet 16-8 model trained on CIFAR-10 (top) and CIFAR-100 (bottom) data sets with Basic (left), Basic + Cutout (middle) and Basic+AutoAugmentation+Cutout (right) augmentation schemes using SGD, SAM, and LPF-SGD optimization algorithms.

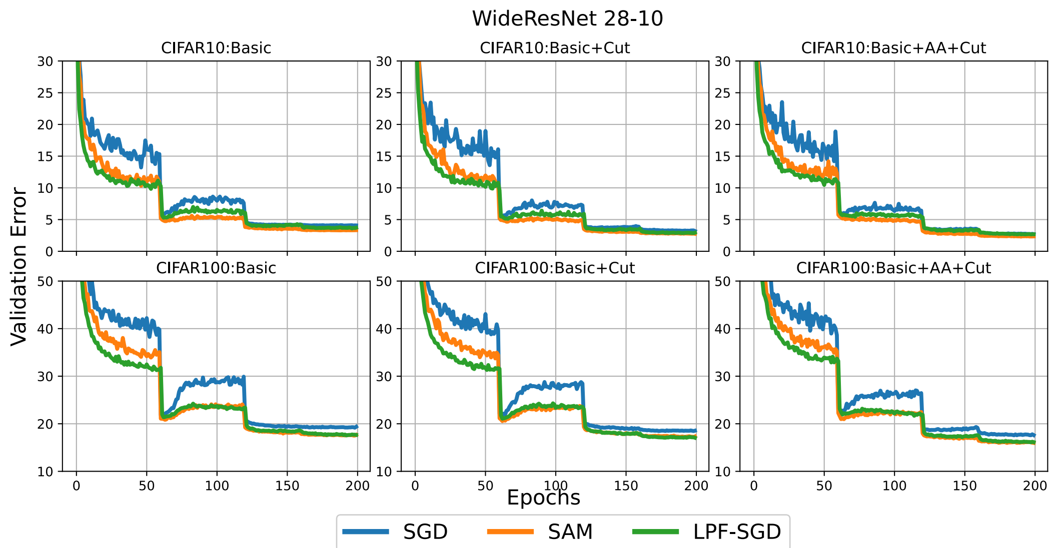

Figure 11: Validation error vs epochs for WideResNet 28-10 model trained on CIFAR-10 (top) and CIFAR-100 (bottom) data sets with Basic (left), Basic + Cutout (middle) and Basic+AutoAugmentation+Cutout (left) augmentation schemes using SGD, SAM, and LPF-SGD optimization algorithms.

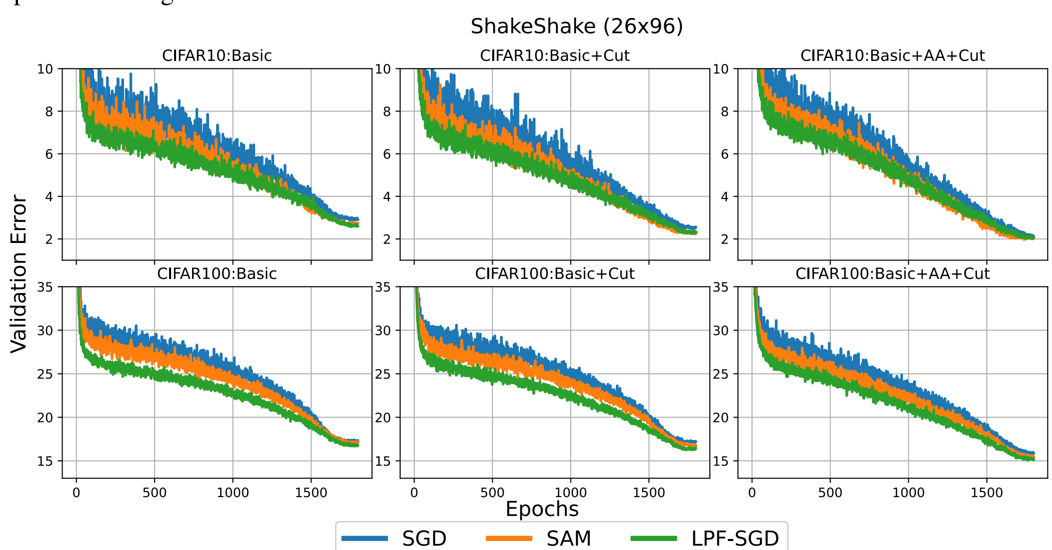

Figure 12: Validation error vs epochs for ShakeShake (26 2x96d) model trained on CIFAR-10 (top) and CIFAR-100 (bottom) data sets with Basic (left), Basic + Cutout (middle) and Basic+AutoAugmentation+Cutout (right) augmentation schemes using SGD, SAM, and LPF-SGD optimization algorithms.

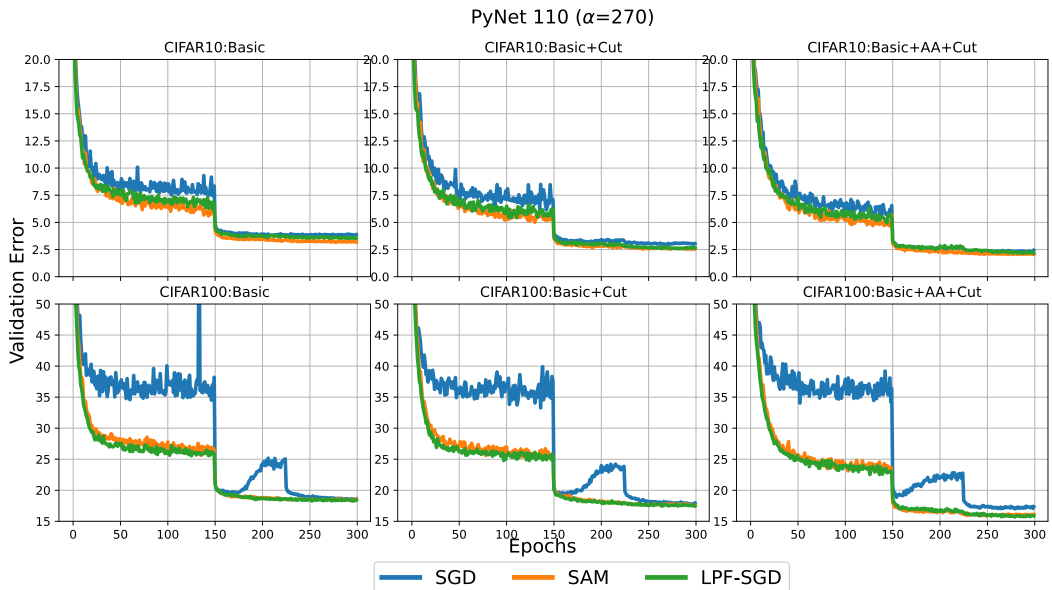

Figure 13: Validation error vs epochs for PyramidNet110 ($\alpha = 270$) model trained on CIFAR-10 (top) and CIFAR-100 (bottom) data sets with Basic (left), Basic + Cutout (middle) and Basic+AutoAugmentation+Cutout (right) augmentation schemes using SGD, SAM, and LPF-SGD optimization algorithms.

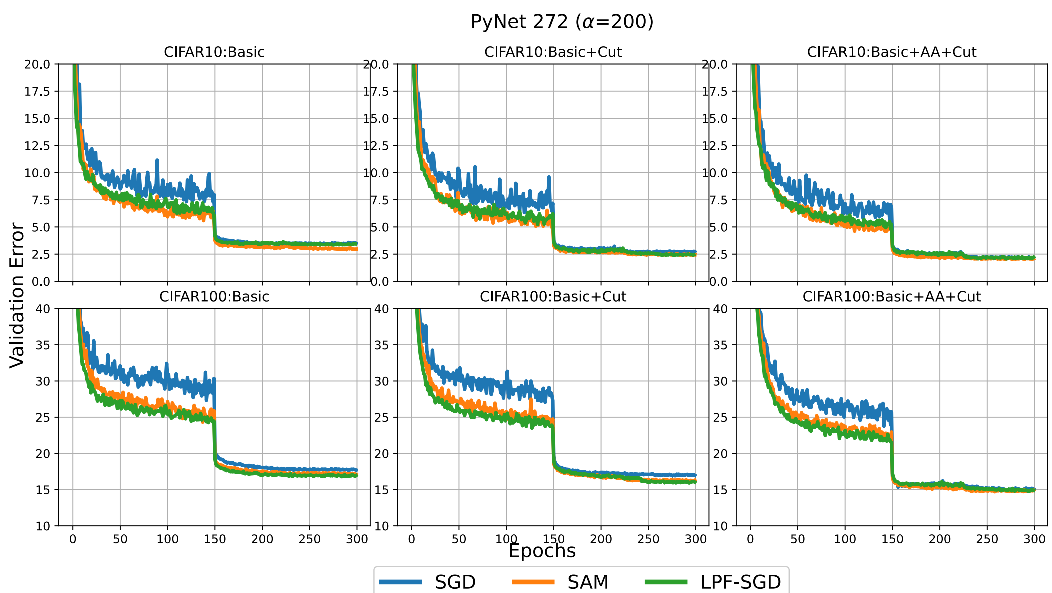

Figure 14: Validation error vs epochs for PyramidNet272 ($\alpha = 200$) model trained on CIFAR-10 (top) and CIFAR-100 (bottom) data sets with Basic (left), Basic + Cutout (middle) and Basic+AutoAugmentation+Cutout (right) augmentation schemes using SGD, SAM, and LPF-SGD optimization algorithms.

F Theoretical proofs

F.1 Proof for Theorem 1

Proof in this section is inspired by the analysis in [108].

Lemma 1. *Let $l_o(\theta;\xi)$ be α Lipschitz continuous with respect to l_2-norm. Let variable Z be distributed according to the distribution μ. Then*

$$\begin{aligned}\|\nabla l_\mu(x;\xi)-\nabla l_\mu(y;\xi)\| =&\mathbb{E}_{Z\sim\mu}[\nabla l_\mu(x+Z;\xi)-\nabla l_\mu(y+Z;\xi)] \\ \leq&\alpha\int|\mu(z-x)-\mu(z-y)|dz.\end{aligned}\tag{F.1}$$

If distribution μ is rotationally symmetric and non-increasing, the bound is tight and can be attained by the function

$$l_o(\theta;\xi)=\alpha\frac{\|x\|^2+\|y\|^2}{\|x-y\|}\left|<\frac{x-y}{\|x\|^2+\|y\|^2},\theta>-\frac{1}{2}\right|.$$

Proof. Let Z be the random variable satisfies distribution μ.

$$\begin{aligned}&\mathbb{E}_{Z\sim\mu}[\nabla l_\mu(x+Z;\xi)-\nabla l_\mu(y+Z;\xi)]\\ =&\int\nabla l_\mu(x+z;\xi)\mu(z)dz-\int\nabla l_\mu(y;\xi)\mu(z)dz\\ =&\int\nabla l_\mu(x;\xi)\mu(z)dz-\int\nabla l_\mu(y;\xi)\mu(z)dz\\ =&\int_{I_>}\nabla l_o(z)[\mu(z-x)-\mu(z-y)]dz-\int_{I_<}\nabla l_0(z)[\mu(z-y)-\mu(z-x)]dz\end{aligned}$$

where

$$\begin{aligned}I_> =&\{z\in\mathbb{R}^d|\mu(z-x)>\mu(z-y)\},\\ I_< =&\{z\in\mathbb{R}^d|\mu(z-x)<\mu(z-y)\}.\end{aligned}$$

Obviously,

$$\begin{aligned}&\|\mathbb{E}_{Z\sim\mu}[\nabla l_\mu(x+Z;\xi)-\nabla l_\mu(y+Z;\xi)]\|\\ \leq&\sup_{z\in I_>\cup I_<}\|\nabla l_o(z)\|\left|\int_{I_>}[\mu(z-x)-\mu(z-y)]dz-\int_{I_<}l(z)[\mu(z-y)-\mu(z-x)]dz\right|\\ \leq&\alpha\left|\int_{I_>}[\mu(z-x)-\mu(z-y)]dz-\int_{I_<}l(z)[\mu(z-y)-\mu(z-x)]dz\right|\\ =&\alpha\int|\mu(z-x)-\mu(z-y)|dz.\end{aligned}$$

We already prove the inequality F.1. We are going to show that the bound is tight and could be attained. Since μ is an rotationally symmetric and non-increasing, the set $I_>$ could be rewritten as

$$\begin{aligned}I_> =&\{z\in\mathbb{R}^d|\mu(z-x)>\mu(z-y)\}\\ =&\{z\in\mathbb{R}^d|\,\|z-x\|^2>\|z-y\|^2\}\\ =&\{z\in\mathbb{R}^d|\langle z,x-y\rangle>\frac{1}{2}(\|x\|^2+\|y\|^2)\},\end{aligned}$$

similarly,

$$I_<=\{z\in\mathbb{R}^d|\langle z,x-y\rangle<\frac{1}{2}(\|x\|^2+\|y\|^2)\}.$$

For given x,y, define function l_o as

$$l_o(\theta;\xi)=\alpha\frac{\|x\|^2+\|y\|^2}{\|x-y\|}\left|<\frac{x-y}{\|x\|^2+\|y\|^2},\theta>-\frac{1}{2}\right|.$$

Therefore, the gradient of function f is

$$\nabla l_o(\theta;\xi)=\begin{cases}\alpha\dfrac{x-y}{\|x-y\|} & \text{if}\langle\theta,x-y\rangle>\dfrac{1}{2}(\|x\|^2+\|y\|^2)\\ -\alpha\dfrac{x-y}{\|x-y\|} & \text{if}\langle\theta,x-y\rangle<\dfrac{1}{2}(\|x\|^2+\|y\|^2)\end{cases}\tag{F.2}$$

Hence,

$$
\begin{aligned}
&\|\mathbb{E}_{Z\sim\mu}[\nabla l_\mu(x+Z;\xi)-\nabla l_\mu(y+Z;\xi)]\| \\
=&\left\|\int_{I_>}\nabla l_o(z)[\mu(z-x)-\mu(z-y)]dz-\int_{I_<}\nabla l_o(z)[\mu(z-y)-\mu(z-x)]dz\right\| \\
=&\left\|\int_{I_>}\alpha\frac{x-y}{\|x-y\|}[\mu(z-x)-\mu(z-y)]dz+\int_{I_<}\alpha\frac{x-y}{\|x-y\|}[\mu(z-y)-\mu(z-x)]dz\right\| \\
=&\left\|\alpha\frac{x-y}{\|x-y\|}\int|\mu(z-x)-\mu(z-y)|dz\right\| \\
=&\alpha\int|\mu(z-x)-\mu(z-y)|dz\left\|\frac{x-y}{\|x-y\|}\right\| \\
=&\alpha\int|\mu(z-x)-\mu(z-y)|dz
\end{aligned}
$$

We already show that the equality holds for given function l_o. Therefore the bound is tight. □

Theorem 1. *Let μ be the $\mathcal{N}(0,\sigma^2 I_{d\times d})$ distribution. Assume the differentiable loss function $l_o(\theta;\xi):\mathbb{R}^d\to\mathbb{R}$ is α-Lipschitz continuous and β-smooth with respect to l_2-norm. The smoothed loss function $l_\mu(\theta;\xi)$ is defined as (5.1). Then the following properties hold:*

i) l_μ is α-Lipschitz continuous.

ii) l_μ is continuously differentiable; moreover, its gradient is $\min\{\frac{\alpha}{\sigma},\beta\}$-Lipschitz continuous, i.e., f_μ is $\min\{\frac{\alpha}{\sigma},\beta\}$-smooth.

iii) If l_o is convex, $l_o(\theta;\xi)\le l_\mu(\theta;\xi)\le l_o(\theta;\xi)+\alpha\sigma\sqrt{d}$.

In addition, for each bound i)-iii), there exists a function l_o such that the bound cannot be improved by more than a constant factor.

Proof. We are going the prove the properties one by one.

i) Since $\nabla l_\mu(\theta;\xi)=\mathbb{E}_{Z\sim\mu}[\nabla l_o(\theta+Z;\xi)]$, we have

$$\|\nabla l_\mu(\theta;\xi)\|=\|\mathbb{E}_{Z\sim\mu}[\nabla l_o(\theta+Z;\xi)]\|\le\mathbb{E}_{Z\sim\mu}[\|\nabla l_o(\theta+Z;\xi)\|]\le\alpha.$$

Therefore, l_μ is α-Lipschitz continuous. To prove the bound is tight, we define

$$l_o(\theta;\xi)=\frac{1}{2}v^T\theta,$$

where $v\in\mathbb{R}^d$ is a scalar. Hence, we have

$$l_\mu(\theta;\xi)=\mathbb{E}_{Z\sim\mu}[l_o(\theta+Z;\xi)]=\mathbb{E}_{Z\sim\mu}[\frac{1}{2}v^T(\theta-Z)]=\frac{1}{2}v^T\theta=l_o(\theta;\xi).$$

Both l_o and smoothed l_μ have the gradient v and l_μ is exactly α-Lipschitz.

ii) The proof scheme for this part is organized as follow: Firstly we show that l_μ is $\frac{\alpha}{\sigma}$-smooth and the bound can not be improved by more than a constant factor. Then we show that l_μ is β-smooth and the bound can not be improved by more than a constant factor as well. In all, we could draw the conclusion that l_μ is $\min\{\frac{\alpha}{\sigma},\beta\}$-smooth and the bound is tight.

By Lemma 1, for $\forall x,y\in\mathbb{R}^n$,

$$\|\nabla l_\mu(x;\xi)-\nabla l_\mu(y;\xi)\|\le\alpha\underbrace{\int|\mu(z-x)-\mu(z-y)|dz}_{I_2}. \tag{F.3}$$

Denote the integral as I_2. We follow a technique used in [107] [108]. Since $\mu(z-x) \geq \mu(z-y)$ is equivalent to $\|z-x\| \geq \|z-y\|$,

$$\begin{aligned}
I_2 =& \int |\mu(z-x) - \mu(z-y)| dz \\
=& \int_{z:\|z-x\|\geq\|z-y\|} [\mu(z-x) - \mu(z-y)] dz + \int_{z:\|z-x\|\leq\|z-y\|} [\mu(z-y) - \mu(z-x)] dz \\
=& 2\int_{z:\|z-x\|\geq\|z-y\|} [\mu(z-x) - \mu(z-y)] dz \\
=& 2\int_{z:\|z-x\|\geq\|z-y\|} \mu(z-x) dz - 2\int_{z:\|z-x\|\geq\|z-y\|} \mu(z-y) dz.
\end{aligned}$$

Denote $u = z - x$ for $\mu(z-x)$ term and $u = z - y$ for $\mu(z-y)$ term, we have

$$\begin{aligned}
I_2 =& 2\int_{z:\|u\|\geq\|u-(x-y)\|} \mu(u) dz - 2\int_{z:\|y\|\geq\|u-(x-y)\|} \mu(u) dz \\
=& 2\mathbb{P}_{Z\sim\mu}(\|Z\| \leq \|Z-(x-y)\|) - 2\mathbb{P}_{Z\sim\mu}(\|Z\| \geq \|Z-(x-y)\|).
\end{aligned}$$

Obviously,

$$\begin{aligned}
& \mathbb{P}_{Z\sim\mu}(\|Z\| \leq \|Z-(x-y)\|) \\
=& \mathbb{P}_{Z\sim\mu}(\|Z\|^2 \leq \|Z-(x-y)\|^2) \\
=& \mathbb{P}_{Z\sim\mu}(2\langle z, x-y\rangle \leq \|x-y\|^2) \\
=& \mathbb{P}_{Z\sim\mu}(2\langle z, \frac{x-y}{\|x-y\|}\rangle \leq \|x-y\|),
\end{aligned}$$

$\frac{x-y}{\|x-y\|}$ has norm 1 and $Z \sim \mathcal{N}(0, \sigma^2 I)$ implies $\langle z, \frac{x-y}{\|x-y\|}\rangle \sim \mathcal{N}(0, \sigma^2 I)$. Hence, we have

$$\begin{aligned}
& \mathbb{P}_{Z\sim\mu}(\|Z\| \leq \|Z-(x-y)\|) \\
=& \mathbb{P}_{Z\sim\mu}(\langle z, \frac{x-y}{\|x-y\|}\rangle \leq \frac{\|x-y\|}{2}) \\
=& \int_{-\infty}^{\frac{\|x-y\|}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2}) du.
\end{aligned}$$

Similarly,

$$\begin{aligned}
& \mathbb{P}_{Z\sim\mu}(\|Z\| \geq \|Z-(x-y)\|) \\
=& \mathbb{P}_{Z\sim\mu}(\langle z, \frac{x-y}{\|x-y\|}\rangle \geq \frac{\|x-y\|}{2}) \\
=& \int_{\frac{\|x-y\|}{2}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2}) du.
\end{aligned}$$

Therefore,

$$\begin{aligned}
I_2 =& \int_{-\infty}^{\frac{\|x-y\|}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2}) du - \int_{\frac{\|x-y\|}{2}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2}) du \\
=& \int_{-\frac{\|x-y\|}{2}}^{\frac{\|x-y\|}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2}) du \\
\leq& \frac{\|x-y\|}{\sigma\sqrt{2\pi}}
\end{aligned} \tag{F.4}$$

In conclusion, combine formula (F.3) and (F.4) we have

$$\|\nabla l_\mu(x;\xi) - \nabla l_\mu(y;\xi)\| \leq \alpha \frac{\|x-y\|}{\sigma\sqrt{2\pi}} \leq \frac{\alpha}{\sigma} \|x-y\|.$$

We finish proving that l_μ is $\frac{\alpha}{\sigma}$-smooth. We are going to show the bound is tight. For any given x, y, define function l_o as

$$l_o(\theta;\xi) = \alpha\frac{\|x\|^2+\|y\|^2}{\|x-y\|}\left|< \frac{x-y}{\|x\|^2+\|y\|^2}, \theta > -\frac{1}{2}\right|,$$

Uniform Lemma 1 and former proof, we know that

$$\begin{aligned}\|\nabla l_\mu(x;\xi) - \nabla l_\mu(y;\xi)\| =& \alpha\int|\mu(z-x)-\mu(z-y)|dz\\ =& \frac{\alpha}{\sigma\sqrt{2\pi}}\int_{-\frac{\|x-y\|}{2}}^{\frac{\|x-y\|}{2}}\exp(-\frac{u^2}{2\sigma^2})d\end{aligned}\tag{F.5}$$

Because

$$\frac{\alpha}{\sigma\sqrt{2\pi}}\exp(-\frac{\|x-y\|^2}{8\sigma^2})\|x-y\| \leq \frac{\alpha}{\sigma\sqrt{2\pi}}\int_{-\frac{\|x-y\|}{2}}^{\frac{\|x-y\|}{2}}\exp(-\frac{u^2}{2\sigma^2})d \leq \frac{\alpha}{\sigma\sqrt{2\pi}}\|x-y\|$$

Obviously, taking x, y such that $\|x-y\| \leq 2\sqrt{2}\sigma$,

$$\frac{\alpha}{e\sigma\sqrt{2\pi}}\|x-y\| \leq \|\nabla l_\mu(x;\xi) - \nabla l_\mu(y;\xi)\| \leq \frac{\alpha}{\sigma\sqrt{2\pi}}\|x-y\|$$

we could conclude the Lipschitz bound for ∇l_μ cannot be improved by more than a constant factor.

Then we are going to show smooth objective l_μ is β smooth and the bound is tight.

$$\begin{aligned}\|\nabla l_\mu(x;\xi) - \nabla l_\mu(y;\xi)\| &= \|\nabla\mathbb{E}_{Z\sim\mu}[l_o(x+Z)] - \nabla\mathbb{E}_{Z\sim\mu}[l_o(x+Z)]\|\\ &= \|\mathbb{E}_{Z\sim\mu}[\nabla l_o(x+Z) - \nabla l_o(y+Z)]\|\\ &= \left\|\int[\nabla l_o(x+Z) - \nabla l_o(y+Z)]\mu(z)dz\right\|\\ &\leq \int\|\nabla l_o(x+Z) - \nabla l_o(y+Z)\|\mu(z)dz\\ &\leq \int\beta\|(x+z)-(y+z)\|\mu(z)dz\\ &= \beta\|x-y\|\int\mu(z)dz\\ &= \beta\|x-y\|\end{aligned}$$

Therefore, l_μ is β-smooth. Then we are going to show the bound is tight and cannot be improved. Define α Lipschitz continuous and β-smooth function $l_o : \mathbb{R}^d \to \mathbb{R}$ as

$$l_o(\theta;\xi) = \frac{1}{2}\beta\|w\|^2 \qquad \theta \in B(0, \frac{\alpha}{\beta}).$$

Hence, we have

$$\begin{aligned}\|\nabla l_\mu(x;\xi) - \nabla l_\mu(y;\xi)\| &= \left\|\int(\beta x - \beta y)\mu(z)dz\right\|\\ &= \left\|\beta(x-y)\int\mu(z)dz\right\|\\ &= \beta\|x-y\|.\end{aligned}$$

Therefore. l_μ is exactly β-smooth.

iii) By Jensen's inequality, for left hand side:

$$l_\mu(\theta;\xi) = \mathbb{E}_{Z\sim\mu}[l_o(\theta+Z;\xi)] \geq l_o(\theta + \mathbb{E}_{Z\sim\mu}[Z];\xi) = l_o(\theta;\xi).$$

For the tightness proof, defining $l_o(\theta;\xi) = \frac{1}{2}v^T\theta$ for $v \in \mathbb{R}^d$ leads to $l_\mu = l_o$.

For right hand side:

$$\begin{aligned}
l_\mu(\theta;\xi) =& \mathbb{E}_{Z\sim\mu}[l_o(\theta+Z;\xi)] \\
\leq& l_o(\theta;\xi) + \alpha\mathbb{E}_{Z\sim\mu}[\|Z\|] \qquad (\alpha\text{-Lipchitz continuous}) \\
\leq& l_o(\theta;\xi) + \alpha\sqrt{\mathbb{E}[\|Z\|^2]} \qquad (\frac{\|Z\|^2}{\sigma^2} \sim \mathcal{X}^2(d)) \\
=& l_o(\theta;\xi) + \alpha\sigma\sqrt{d}.
\end{aligned}$$

For the tightness proof, taking $l_o(\theta;\xi) = \alpha\|\theta\|$.Since $l_\mu(\theta;\xi) \geq c\alpha\sigma\sqrt{d}$ for some constant c. Therefore, the bound cannot be improved by more than a constant factor.

□

F.2 Proof for Theorem 3

We first consider the generalization error in the context of the original loss L, and then we analyze smoothed loss function $L \circledast K$. The true loss is defined as

$$L^{true}(\theta) := \mathbb{E}_{\xi\sim D} l(\theta;\xi). \tag{F.6}$$

where l is an arbitrary loss function (i.e., l_o for SGD case and l_μ for LPF-SGD case). Since the distribution $\mathcal{D}$ is unknown, we replace the true loss by the empirical loss given as

$$L^{\mathcal{S}}(\theta) := \frac{1}{m}\sum_{i=1}^{m} l(\theta;\xi). \tag{F.7}$$

In order to bound the generalization error ϵ_g, we consider the following stability bound.

Definition 8. *[ϵ_s-uniform stability [30]] Let $\mathcal{S}$ and $\mathcal{S}'$ denote two data sets from input data distribution $\mathcal{D}$ such that $\mathcal{S}$ and $\mathcal{S}'$ differ in at most one example. Algorithm A is ϵ_s-uniformly stable if and only if for all data sets $\mathcal{S}$ and $\mathcal{S}'$ we have*

$$\sup_\xi \mathbb{E}[l(A(\mathcal{S});\xi) - l(A(\mathcal{S}');\xi)] \leq \epsilon_s. \tag{F.8}$$

The following theorem, proposed in [30], implies that the generalization error could be bounded using the uniform stability bound.

Theorem 2. *If A is an ϵ_s-uniformly stable algorithm, then the generalization error (the gap between the true risk and the empirical risk) of A is upper-bounded by the stability factor ϵ_s:*

$$\epsilon_g := \mathbb{E}_{\mathcal{S},A}[L^{true}(A(\mathcal{S})) - L^{\mathcal{S}}(A(\mathcal{S}))] \leq \epsilon_s \tag{F.9}$$

Denote the original true loss and empirical loss as:

$$L_o^{true}(\theta) := \mathbb{E}_{\xi\sim D} l_o(\theta;\xi) \quad \text{and} \quad L_o^{\mathcal{S}}(\theta) := \frac{1}{m}\sum_{i=1}^{m} l_o(\theta;\xi).$$

Denote the stability gap and generalization error of original loss function as ϵ_s^o and ϵ_g^o, respectively. Theorem 4 bounds links the stability with Lipschitz factor α, smoothing factor β, and number of iterations T of SGD. Its proof can be found in [30].

Theorem 4 (Uniform stability of SGD [30])**.** *Assume that $l_o(\theta;\xi) \in [0,1]$ is a α-Lipschitz and β-smooth loss function for every example ξ. Suppose that we rum SGD for T steps with monotonically non-increasing step size $\eta_t \leq c/t$. Then SGD is uniformly stable with the stability factor ϵ_s^o satisfying:*

$$\epsilon_s^o \leq \frac{1+1/\beta c}{n-1}(2c\alpha^2)^{\frac{1}{\beta c+1}} T^{\frac{\beta c}{\beta c+1}}. \tag{F.10}$$

Now, we have already bound the stability gap ϵ_s^o on original loss. Then we will move onto the stability gap ϵ_s^μ of loss for Gaussian LPF kernel smoothed loss function. Let μ be distribution $\mathcal{N}(0,\sigma^2 I)$. By the definition of Gaussian LPF (Definition 6), the true loss and the empirical loss with respect to the Gaussian LPF smoothed function are

$$L_\mu^{true}(\theta) := (L_o^{true} \circledast K)(\theta) = \int_{-\infty}^{\infty} L_o^{true}(\theta - \tau)\mu(\tau)d\tau = \mathbb{E}_{Z\sim\mu}[L_o^{true}(\theta + Z)], \tag{F.11}$$

$$L_\mu^{\mathcal{S}}(\theta) \quad := (L_o^{\mathcal{S}} \circledast K)(\theta) = \int_{-\infty}^{\infty} L_o^{\mathcal{S}}(\theta - \tau)\mu(\tau)d\tau = \mathbb{E}_{Z\sim\mu}[L_o^{\mathcal{S}}(\theta + Z)], \tag{F.12}$$

where K is the Gaussian LPF kernel satisfies distribution μ and Z is a random variable satisfies distribution μ. Since $L_o^{true}(\theta) := \mathbb{E}_{\xi\sim D} l_o(\theta;\xi)$ and $L_o^{\mathcal{S}}(\theta) := \frac{1}{m}\sum_{i=1}^m l_o(\theta;\xi)$, L_μ^{true} and L_μ could be rewritten as

$$L_\mu^{true}(\theta) = \int_{-\infty}^{\infty} \mathbb{E}_{\xi\sim D}[l_o(\theta-\tau;\xi)]\mu(\tau)d\tau = \mathbb{E}_{\xi\sim D}\left[\int_{-\infty}^{\infty} l_o(\theta-\tau;\xi)\mu(\tau)d\tau\right] = \mathbb{E}_{\xi\sim D}\left[l_\mu(\theta;\xi)\right] \tag{F.13}$$

$$L_\mu^{\mathcal{S}}(\theta) = \int_{-\infty}^{\infty} \frac{1}{m}\sum_{i=1}^m l_o(\theta;\xi)\mu(\tau)d\tau = \frac{1}{m}\sum_{i=1}^m \left[\int_{-\infty}^{\infty} l_o(\theta-\tau;\xi_i)\mu(\tau)d\tau\right] = \frac{1}{m}\sum_{i=1}^m l_\mu(\theta;\xi_i). \tag{F.14}$$

Compare formulas (F.13-F.14) with (F.6-F.7). We could conclude that the true and empirical Gaussian LPF smoothed loss function (L_μ^{true} and $L_\mu^{\mathcal{S}}$) is exactly the formula of original true and empirical loss (L_o^{true} and $L_o^{\mathcal{S}}$) by replacing $l_o(\theta;\xi)$ with smoothed $l_\mu(\theta;\xi)$. Since LPF-SGD is exactly performing SGD iteration on Gaussian LPF smoothed loss function instead of original loss, the generalization error and stability gap of LPF-SGD also satisfies Theorem 2 and Theorem 4 after replacing l_o with l_μ.

In section 5.1 we analyze the change of Lipschitz continuous and smooth properties of the objective function after Gaussian LPF smoothing. Therefore, by Theorem 1, l_μ is α-Lipschitz continuous and $\min\{\frac{\alpha}{\sigma},\beta\}$-smooth. Define $\hat{\beta} = \min\{\frac{\alpha}{\sigma},\beta\}$, then we could bound the stability gap for LPF-SGD as

$$\epsilon_s^\mu \le \frac{1 + 1/\hat{\beta}c}{n-1}(2c\alpha^2)^{\frac{1}{\hat{\beta}c+1}} T^{\frac{\hat{\beta}c}{\hat{\beta}c+1}}.$$

Combine Theorem 1 with Theorem 4 we could have the following proposition.

Theorem 3. *Assume that $l_o(\theta;\xi) \in [0,1]$ is a α-Lipschitz and β-smooth loss function for every example ξ. Suppose that we run SGD and LPF-SGD for T steps with non-increasing learning rate $\eta_t \le c/t$. Denote the generalization error of SGD and LPF-SGD as ϵ_g^o and ϵ_g^μ, respectively. Then the approximate ratio of generalization error is given as*

$$\rho = \frac{\epsilon_g^\mu}{\epsilon_g^o} \approx \frac{\epsilon_s^\mu}{\epsilon_s^o} \approx \frac{1-p}{1-\hat{p}}\left(\frac{2c\alpha}{T}\right)^{\hat{p}-p}, \tag{F.15}$$

where $p = \frac{1}{\beta c+1}$, $\hat{p} = \frac{1}{\min\{\frac{\alpha}{\sigma},\beta\}c+1}$, and ϵ_s^μ is the stability factor for LPF-SGD.

Finally, the following two properties hold:

i) If $T > 2c\alpha^2\left(\frac{1-p}{1-\hat{p}}\right)^{\frac{1}{\hat{p}-p}}$, $\rho \lessapprox 1$ implies $\epsilon_g^\mu \lessapprox \epsilon_g^o$.

ii) If $T > 2c\alpha^2\exp(\frac{2}{1-p})$ and $\sigma > \frac{\alpha}{\beta}$, increasing σ leads to a smaller ρ.

Proof. For easy notation, denote $\hat{\beta} = \min\{\frac{\alpha}{\sigma},\beta\}$ ϵ_S^o and ϵ_s^μ are stability gaps of SGD and LPF-SGD, respectively. From Theorem 4 and based on the facts that l_o is α-Lipschitz continuous and β-smooth and that smoothed objective l_μ is α-Lipschitz continuous and $\min\{\frac{\alpha}{\sigma},\beta\}$-smooth, the upper bounds for the stability gaps are

$$\epsilon_s^o \le \frac{1 + 1/\beta c}{n-1}(2c\alpha^2)^{\frac{1}{\beta c+1}} T^{\frac{\beta c}{\beta c+1}},$$

$$\epsilon_s^\mu \le \frac{1 + 1/\hat{\beta}c}{n-1}(2c\alpha^2)^{\frac{1}{\hat{\beta}c+1}} T^{\frac{\hat{\beta}c}{\hat{\beta}c+1}}.$$

Denote $p = \frac{1}{\beta c+1}$, $\hat{p} = \frac{1}{\hat{\beta} c+1}$, the bound could be rewritten as

$$\epsilon_s^o \leq \frac{1}{(n-1)(1-p)}(2c\alpha^2)^p T^{1-p},$$
$$\epsilon_s^\mu \leq \frac{1}{(n-1)(1-\hat{p})}(2c\alpha^2)^{\hat{p}} T^{1-\hat{p}}.$$

By Theorem 2, the generalization errors ϵ_g^o and ϵ_g^μ iare bounded by stability gaps ϵ_s^o and ϵ_s^μ:

$$\epsilon_g^o \leq \epsilon_s^o \leq \frac{1}{(n-1)(1-p)}(2c\alpha^2)^p T^{1-p},$$
$$\epsilon_g^\mu \leq \epsilon_s^\mu \leq \frac{1}{(n-1)(1-\hat{p})}(2c\alpha^2)^{\hat{p}} T^{1-\hat{p}}.$$

Because it is hard to compute the accurate value of generalization errors, we approximate the ratio ρ of generalization errors with their upper bounds instead, then we have

$$\rho = \frac{\epsilon_g^\mu}{\epsilon_g} \approx \frac{\epsilon_s^\mu}{\epsilon_s} \approx \frac{1-p}{1-\hat{p}}(\frac{2c\alpha^2}{T})^{\hat{p}-p}$$

When $T > 2c\alpha^2 \left(\frac{1-p}{1-\hat{p}}\right)^{\frac{1}{\hat{p}-p}}$, $\frac{1-p}{1-\hat{p}}(\frac{2c\alpha^2}{T})^{\hat{p}-p} \leq 1$. Therefore, $\epsilon_g^\mu \lessapprox \epsilon_g^o$ and property i) holds.

Denote $x := \hat{p} - p$, the reciprocal of approximated ratio could be re-written as

$$\frac{1}{\rho} \approx (1 - \frac{\hat{p}-p}{1-p})(\frac{T}{2c\alpha^2})^{\hat{p}-p} = (1 - \frac{x}{1-p})(\frac{T}{2c\alpha^2})^x$$

Define function $h(x) = (1-ax)b^x$, where $a = \frac{1}{1-p}$ and $b = \frac{T}{2c\alpha^2}$. Compute the derivative of function h:

$$h'(x) = (-ax - a + \ln b)b^x$$

$$h'(x_0) = 0 \Longleftrightarrow x_0 = \frac{lnb - a}{a}$$

When $x \leq \frac{lnb-a}{a}$, $h'(x) \geq 0$. Otherwise $h'(x) < 0$. Since $0 < p < \hat{p} < 1$, obviously the domain of function h is in the interval $[0, 1]$. If $\frac{lnb-a}{a} > 1$, the function h is increasing in its domain. Which means that if the difference between $\hat{p}$ and p increase, the reciprocal of approximated ratio of stability gap $\frac{1}{\rho}$ decrease, which is equivalent to the approximate ratio ρ of stability gap decrease. Because

$$\frac{\ln b - a}{a} > 1 \Longleftrightarrow \ln b > 2a \Longleftrightarrow T > 2c\alpha^2 \exp(\frac{2}{1-p}).$$

In all, we could conclude if $T > 2c\alpha^2 \exp(\frac{2}{1-p})$, $\hat{p} - p$ increase leads to the approximate ratio of stability gap ρ decrease.

What's more, we are going to analysis the relation between Gaussian filter factor σ and the difference $\hat{p} - p$. Since

$$p = \frac{1}{\beta c + 1}, \hat{p} = \frac{1}{\min\{\frac{\alpha}{\sigma}, \beta\}c + 1},$$

$\hat{p} - p$ increase is equivalent to $\hat{\beta}$ decrease. When the Gaussian factor σ is large enough ($\frac{\alpha}{\sigma} < \beta$), the smoother factor $\hat{\beta}$ for function l_μ is exactly $\frac{\alpha}{\sigma}$. Moreover, increasing the factor σ leads to the decrease of $\hat{\beta}$.

Due to the analysis above, if $T > 2c\alpha^2 \exp(\frac{2}{1-p})$ and $\frac{\alpha}{\sigma} < \beta$, increasing σ will cause the approximate ratio ρ to decrease and the generalization error will be smaller. We finish the proof for condition ii). $\square$

F.3 Non-scalar covariance version

In this section, we analysis the case when the covariance $\Sigma = \gamma * diag(||\theta_1||, ||\theta_2|| \cdots ||\theta_k||)$ for Gaussian kernel K is no longer a scalar diagonal matrix. For easy notation, we denoted $\Sigma = diag(\sigma_1^2, \cdots, \sigma_d^2)$ where $\sigma_i^2 = \gamma * ||\theta_i||$.

Theorem 5. *Let μ be the $\mathcal{N}(0, \Sigma)$ distribution, where $\Sigma = diag(\sigma_1^2, \cdots, \sigma_d^2) \in \mathbb{R}^{d \times d}$ is diagonal. Denote $\sigma_-^2 = \min\{\sigma_1^2, \cdots, \sigma_d^2\}$. Assume the differentiable loss function $l_o(\theta; \xi) : \mathbb{R}^d \to \mathbb{R}$ is α-Lipschitz continuous and β-smooth with respect to l_2-norm. The smoothed loss function $l_\mu(\theta; \xi)$ is defined as (5.1). Then the following properties hold:*

i) l_μ is α-Lipschitz continuous.

ii) l_μ is continuously differentiable; moreover, its gradient is $\min\{\frac{\alpha}{\sigma_-}, \beta\}$-Lipschitz continuous, i.e. l_μ is $\min\{\frac{\alpha}{\sigma_-}, \beta\}$-smooth.

iii) If l is convex, $l_\mu(\theta; \xi) = l(\theta; \xi) + \alpha\sqrt{tr(\Sigma)} = l(\theta; \xi) + \alpha\sqrt{\sum_{i=1}^d \sigma_i^2}$.

In addition, for bound i) and iii), there exists a function l such that the bound cannot be improved by more than a constant factor.

Proof. We are going to prove the properties one by one.

i) The proof for properties i) is exactly the same as Theorem 1.

ii) As is shown in the proof for Theorem 1, firstly, we need to first address that l_μ is $\frac{\alpha}{\sigma}$-smooth. then show that l_μ is β-smooth. Since, the proof for second part remains the same as what in Theorem 1. We will focus on demonstrating l_μ is $\frac{\alpha}{\sigma}$-smooth.

By Lemma 1, for $\forall x, y \in \mathbb{R}^n$,

$$\|\nabla l_\mu(x; \xi) - \nabla l_\mu(y; \xi)\| \le \alpha \underbrace{\int |\mu(z - x) - \mu(z - y)| dz}_{I_2}. \tag{F.16}$$

Denoted the integral as I_2. We follow the technique in [107] and [108]. Since $\mu(z - x) \ge \mu(z - y)$ is equivalent to $\|z - x\| \ge \|z - y\|$,

$$\begin{aligned}
I_2 &= \int |\mu(z - x) - \mu(z - y)| dz \\
&= \int_{z:\|z-x\| \ge \|z-y\|} [\mu(z - x) - \mu(z - y)] dz + \int_{z:\|z-x\| \le \|z-y\|} [\mu(z - y) - \mu(z - x)] dz \\
&= 2\int_{z:\|z-x\| \ge \|z-y\|} [\mu(z - x) - \mu(z - y)] dz \\
&= 2\int_{z:\|z-x\| \ge \|z-y\|} \mu(z - x) dz - 2\int_{z:\|z-x\| \ge \|z-y\|} \mu(z - y) dz.
\end{aligned}$$

Denote $u = z - x$ for $\mu(z - x)$ term and $u = z - y$ for $\mu(z - y)$ term, we have

$$\begin{aligned}
I_2 &= 2\int_{z:\|u\| \ge \|u-(x-y)\|} \mu(u) dz - 2\int_{z:\|y\| \ge \|u-(x-y)\|} \mu(u) dz \\
&= 2\mathbb{P}_{Z\sim\mu}(\|Z\| \le \|Z - (x - y)\|) - 2\mathbb{P}_{Z\sim\mu}(\|Z\| \ge \|Z - (x - y)\|).
\end{aligned}$$

Obviously,

$$\begin{aligned}
&\mathbb{P}_{Z\sim\mu}(\|Z\| \le \|Z - (x - y)\|) \\
=&\mathbb{P}_{Z\sim\mu}(\|Z\|^2 \le \|Z - (x - y)\|^2) \\
=&\mathbb{P}_{Z\sim\mu}(2\langle z, x - y\rangle \le \|x - y\|^2) \\
=&\mathbb{P}_{Z\sim\mu}(2\langle z, \frac{x - y}{\|x - y\|}\rangle \le \|x - y\|),
\end{aligned}$$

Denote $p = \frac{x-y}{\|x-y\|} \in \mathbb{R}^{d\times d}$, $\frac{x-y}{\|x-y\|}$ has norm 1 implies $\sum_{i=1}^d p_i^2 = 1$. Since $Z \sim \mathcal{N}(0, \Sigma)$ and $\Sigma = diag(\sigma_1^2, \cdots, \sigma_d^2)$, each element in vector Z satisfies $z_i \sim \mathcal{N}(0, \sigma_i^2)$. Hence, we have

$$\langle z, \frac{x-y}{\|x-y\|}\rangle = \sum_{i=1}^{d} p_i z_i \sim \mathcal{N}(0, \sum_{i=1}^{d} p_i^2 \sigma_i^2).$$

Denote $\sigma^2 = \sum_{i=1}^d p_i^2\sigma_i^2$, $\sigma_+^2 = \max\{\sigma_1^2, \cdots, \sigma_d^2\}$ and $\sigma_-^2 = \min\{\sigma_1^2, \cdots, \sigma_d^2\}$. Because $\sum_{i=1}^d p_i^2 = 1$, it is easy to know

$$\langle z, \frac{x-y}{\|x-y\|}\rangle \sim \mathcal{N}(0, \sigma^2), \quad \text{where } \sigma_-^2 \leq \sigma^2 \leq \sigma_+^2. \tag{F.17}$$

Hence, we have

$$\begin{aligned}
&\mathbb{P}_{Z\sim\mu}(\|Z\| \leq \|Z - (x-y)\|) \\
=&\mathbb{P}_{Z\sim\mu}(\langle z, \frac{x-y}{\|x-y\|}\rangle \leq \frac{\|x-y\|}{2}) \\
=&\int_{-\infty}^{\frac{\|x-y\|}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2})du.
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\mathbb{P}_{Z\sim\mu}(\|Z\| \geq \|Z - (x-y)\|) \\
=&\mathbb{P}_{Z\sim\mu}(\langle z, \frac{x-y}{\|x-y\|}\rangle \geq \frac{\|x-y\|}{2}) \\
=&\int_{\frac{\|x-y\|}{2}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2})du.
\end{aligned}$$

Therefore,

$$\begin{aligned}
I_2 &= \int_{-\infty}^{\frac{\|x-y\|}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2})du - \int_{\frac{\|x-y\|}{2}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2})du \\
&= \int_{-\frac{\|x-y\|}{2}}^{\frac{\|x-y\|}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{u^2}{2\sigma^2})du \\
&\leq \frac{\|x-y\|}{\sigma\sqrt{2\pi}} \leq \frac{\|x-y\|}{\sigma_-\sqrt{2\pi}}
\end{aligned} \tag{F.18}$$

In conclusion, combine formula (F.16) and (F.18) we have

$$\|\nabla l_\mu(x;\xi) - \nabla l_\mu(y;\xi)\| \leq \alpha\frac{\|x-y\|}{\sigma_-\sqrt{2\pi}} \leq \frac{\alpha}{\sigma_-}\|x-y\|.$$

We finish proving that l_μ is $\frac{\alpha}{\sigma_-}$-smooth. Since covariance matrix Σ for distribution μ is no longer a scalar matix and μ is not rotationally symmetric, the bound can no longer be achieved.

iii) By Jensen's inequality, for left hand side:

$$l_\mu(\theta;\xi) = \mathbb{E}_{Z\sim\mu}[l_o(\theta+Z;\xi)] \geq l_o(\theta + \mathbb{E}_{Z\sim\mu}[Z];\xi) = l_o(\theta;\xi).$$

For the tightness proof, defining $l_o(\theta;\xi) = \frac{1}{2}v^T\theta$ for $v \in \mathbb{R}^d$ leads to $l_\mu = l_o$.

For right hand side:

$$\begin{aligned}
l_\mu(\theta;\xi) =&\mathbb{E}_{Z\sim\mu}[l_o(\theta+Z;\xi)] \\
\leq& l_o(\theta;\xi) + \alpha\mathbb{E}_{Z\sim\mu}[\|Z\|] \qquad (\alpha\text{-Lipchitz continuous}) \\
\leq& l_o(\theta;\xi) + \alpha\sqrt{\mathbb{E}[\|Z\|^2]}.
\end{aligned}$$

Letting $C^T C = \Sigma$ and $V \sim \mathcal{N}(0, I)$, because $Z \sim \mathcal{N}(0, \Sigma)$, we have

$$\mathbb{E}[\|Z\|^2] = \mathbb{E}[\|CV\|^2] = \mathbb{E}[V^T C^T CV] = tr(C^T C\mathbb{E}[V^T V]) = tr(\Sigma).$$

Therefore,

$$l_\mu(\theta;\xi) = l_o(\theta;\xi) + \alpha\sqrt{tr(\Sigma)} = l_o(\theta;\xi) + \alpha\sqrt{\sum_{i=1}^{d}\sigma_i^2}.$$

For the tightness proof, taking $l_o(\theta;\xi) = \alpha\|\theta\|$. Since $l_\mu(\theta;\xi) \geq c\alpha\sqrt{tr(\Sigma)}$ for some constant c. Therefore, the bound cannot be improved by more than a constant factor.

$\square$

Theorem 6. *Let μ be the $\mathcal{N}(0, \Sigma)$ distribution, where $\Sigma = diag(\sigma_1^2, \cdots, \sigma_d^2) \in \mathbb{R}^{d\times d}$ is diagonal. Denote $\sigma_-^2 = \|\Sigma\|_\infty = \min\{\sigma_1^2, \cdots, \sigma_d^2\}$. Assume loss function $l_o(\theta;\xi) : \mathbb{R}^d \to \mathbb{R}$ is α-Lipschitz and β-smooth. The smoothed loss function l_μ is defined as (5.1). Suppose we executeing SGD and LPF-SGD for T steps with non-increasing learning rate $\eta_t \leq c/t$. Denote the stability gap and generalization error of algorithm SGD and LPF-SGD as ϵ_s^o, ϵ_s^μ, ϵ_g^o and ϵ_g^μ, respectively. Then the approximate ratio of generalization error is given as*

$$\rho = \frac{\epsilon_g^\mu}{\epsilon_g} \approx \frac{\epsilon_s^\mu}{\epsilon_s} \approx \frac{1-p}{1-\hat{p}}\left(\frac{2c\alpha}{T}\right)^{\hat{p}-p},$$

where $p = \frac{1}{\beta c+1}$, $\hat{p} = \frac{1}{\min\{\frac{\alpha}{\sigma_-},\beta\}c+1}$. Then the following 2 properties holds:

i) If $T > 2c\alpha^2\left(\frac{1-p}{1-\hat{p}}\right)^{\frac{1}{\hat{p}-p}}$, $\rho \lessapprox 1$ implies $\epsilon_g^\mu \lessapprox \epsilon_g^o$.

ii) If $T > 2c\alpha^2\exp(\frac{2}{1-p})$ and $\sigma > \frac{\alpha}{\beta}$, increasing the Gaussian factor σ leads to a smaller approximate ratio ρ.

Proof. By Theorem 5, the smoothed loss function l_μ is α-Lipschitz continuous and $\min\{\frac{\alpha}{\sigma_-}, \beta\}$-smooth. This gives as equivalency to Theorem 1 after substituting σ_- for σ. Therefore, proof of Theorem 6 is exactly the same as that of Theorem 3 after performing this substitution and therefore will be omitted.

$\square$