Appendix

This appendix is structured as follows:

- In Appendix A we provide more training details. In particular, we report the hyperparameters used for the CIFAR experiments in A.1 and for the ImageNet experiments in A.2. In A.3 we provide more details and a formal definition of the SAM-variants used throughout this paper.
- In Appendix B we show additional experimental results for: CIFAR in B.1, ImageNet in B.3, and a machine translation task in B.5. In B.2 we provide additional ablation studies for sparse perturbation *SSAM* approaches and in B.4 we extend the discussion on adversarial robustness. To gain a better understanding of SAM-ON, we further investigate: the weight distribution shift induced by *SAM-ON* (B.6), the effect of SAM when fixing the normalization parameters during training (B.7), SAM's performance when only training the normalization layers (B.8), and ablations on weight decay and dropout (B.9). Finally, we provide an extended discussion on the sharpness evaluation and more ablations in B.10.
- In Appendix C we provide a convergence analysis for *SAM-ON*.

A Training Details

A.1 CIFAR training details

For our CIFAR experiments, we consider a range of SAM-variants which differ either in the norm ($p \in \{2, \infty\}$) or in the definition of the normalization operator. We use SGD, the original SAM with no normalization and $p = 2$, Fisher-SAM and the following ASAM-variants: elementwise-ℓ_∞, layerwise-ℓ_2, and elementwise-ℓ_2. For the ViT-experiments, we use AdamW instead of SGD. For each of the ASAM-variants, we normalize both bias and weight parameters and set $\eta = 0$. Additionally, we employ the original ASAM-algorithm, where the bias parameters are not normalized and $\eta = 0.01$. We train all models on a single GPU for 200 epochs, and m-sharpness is not employed (unless indicated otherwise). For ResNets, we follow [37] and adopt a learning rate of 0.1, momentum of 0.9, weight decay of 0.0005 and use label smoothing with a factor of 0.1. We use both basic augmentations (random cropping and flipping) and strong augmentations (basic+AutoAugment). For ViTs we use AdamW with learning rate 0.0001, batchsize 64 and only strong augmentations, the other settings remain unchanged. The ResNet results were computed on 2080ti-GPUs and the ViT results on A100s. The values of ρ we considered for each method can be found in Table 7. The ResNet-networks we considered for the CIFAR-experiments in the main paper are ResNet56 (RN56) [25], ResNeXt-29-32x4d (RNxT) [55], and WideResNet-28-10 (WRN) [59]. We adopted the ViTs to CIFAR by setting the image-size to 32 and patch-size to 4.

Table 7: Search-space for ρ. The values used for the the experiments in Tables 1,3 and 10 are marked in bold.

		CIFAR-10 RN	CIFAR-100 RN	CIFAR-10/100 ViT
SAM	all	0.05, **0.1**, 0.25	0.05, **0.1**, 0.5, 1.	0.025, 0.05, **0.1**, 0.25, 0.5
SAM	ON	0.1, **0.5**, 1	0.1, 0.5, **1.**, 5.	1., 2.5, 5., **10.**, 25
el. l_2	all	0.5, 1, **2**, 3, 5	0.5, **1**, 2.5, 5., 10.	0.5, 1., **2.5**, 5, 10
el. l_2	ON	0.5, 1, 2, **3**, 5	0.5, 1., **2.5**, 5., 10.	1., 2.5, 5.,**10.**, 25
el. l_2, orig.	all	0.1, **0.5**, 1, 5, 10	0.5, **1**, 2.5, 5	0.5, 1., **2.5**, 5, 10
el. l_2, orig.	ON	0.1, **0.5**, 1, 5, 10	0.5, **1.**, 2.5, 5	1., 2.5, 5.,**10.**, 25
el. l_∞	all	0.001, **0.005**, 0.01, 0.05	0.001, 0.005, **0.01**, 0.05	0.0005, 0.001, **0.0025**, 0.005, 0.01
el. l_∞	ON	0.01, **0.025**, 0.05, 0.1	0.01, **0.05**, 0.1, 0.5	0.025, 0.05, **0.1**, 0.25, 0.5
layer l_2	all	0.005, 0.01, **0.025**, 0.05, 0.1	0.001, **0.01**, 0.05, 0.1	0.001, 0.0025, **0.005**, 0.01, 0.025
layer l_2	ON	0.05, 0.1, **0.25**, 0.5, 1	0.1, **0.2**, 0.5, 1.	0.05, 0.1, 0.25, **0.5**, 1.
Fisher	all	0.05, **0.1**, 0.5, 1,5	0.05, **0.1**, 0.5, 1	0.05,5 **0.1**, 0.5, 1,5
Fisher	ON	0.1, **0.5**,1 ,5 , 10	0.1, 0.5,**1** ,5 , 10	0.1, 0.5,1 ,5 , **10**

A.2 ImageNet training details

Table 8 shows the hyperparameters for all variants used for ImageNet training. For the ResNet-50 with SGD, SAM and elementwise-ℓ_2 we used the hyperparameters from [23] and [37]. For the layerwise ℓ_2 and elementwise-ℓ_∞ we tried two ρ-values per configuration and report the results of the better one (named ρ (reported) in the table). ρ (discarded) refers to the ρ value we probed, but found to perform worse than the other one. For the ViT-S (additional fine-tuning experiments in Appendix B.3), we tried at least three values of ρ per SAM-configuration and reported the best one.

Table 8: Hyperparameters for training on ImageNet. Top: ResNet-50 from scratch, center: ViT-S from scratch, bottom: finetuning the ViT-S.

param	SGD	SAM	elem. ℓ_2		ResNet-50	elem. ℓ_∞		layer ℓ_2	
	all	all	all	onlyNorm		all	onlyNorm	all	onlyNorm
train epochs					90				
warm-up epochs					3				
cool-down epochs					10				
batch-size					512				
augmentation					inception-style				
lr					0.2				
lr decay					Cosine				
weight decay					0.0001				
ρ (reported)		0.05	1	1		0.001	0.005	0.005	0.05
ρ (discarded)						0.01	0.05	0.05	0.5
Input Resolution					224×224				
m					64				
GPU Type					8×2080-ti				

param	AdamW	AdamW+SAM		ViT-S scratch	Lion	Lion+SAM	
	all	all	onlyNorm		all	all	onlyNorm
train epochs				300			
warm-up epochs				10			
cool-down epochs				0			
batch-size				128			
augmentation				inception-style			
lr				0.001			
lr decay				Cosine			
weight decay				0.1			
ρ (reported)	–	1	15		–	1	10
ρ (discarded)	–	0.05,0.1,0.5,2	10,20		–	0.5,2	5,20
Input Resolution				224×224			
m				128			
GPU Type				1×A100			

param	SGD	SAM		elem. ℓ_2		ViT-S FT	elem. ℓ_∞		layer ℓ_2	
	all	all	onlyNorm	all	onlyNorm		all	onlyNorm	all	onlyNorm
train epochs						9				
warm-up epochs						1				
cool-down epochs						0				
batch-size						896				
augmentation						inception-style				
lr						0.017				
lr decay						Cosine				
weight decay						0.0001				
ρ (reported)	–	0.01	0.1	0.1	1		10^{-4}	10^{-2}	10^{-3}	10^{-3}
ρ (discarded)		0.1	0.01	0.01	0.1		10^{-3}	10^{-3}	10^{-2}	10^{-2}
ρ (discarded)		0.001	1.	1.	10		10^{-5}	10^{-1}	10^{-4}	10^{-1}
Input Resolution						224×224				
m						128				
GPU Type						7×A100				

A.3 SAM variants

Here, we provide a more comprehensive overview of the SAM-variants used throughout the experiments. To this end, we first recall the definition of the (A)SAM-perturbation (Eq. (5) in the main paper):

$$\epsilon_2 = \rho \frac{T_w^2 \nabla L(\mathbf{w})}{||T_w \nabla L(\mathbf{w})||_2} \text{ for } p = 2, \qquad \epsilon_\infty = \rho T_w \text{sign}\big(\nabla L(\mathbf{w})\big) \text{ for } p = \infty.$$

with the normalization operator T_w^i, which is diagonal for all variants. We note that *SAM-ON* can be formally defined as using the conventional (A)SAM-algorithm but setting all entries $T_w^i = 0$ if w_i is not a normalization parameter. This leads to a change of the perturbation ϵ according to Eq. (5). Importantly, the magnitude of ϵ is still ρ, since both the nominator and the denominator of Eq. (5) change. We provide an overview over all (A)SAM-variants and their respective perturbation models in Table 9.

Table 9: The definition of T_w^i for the considered SAM-variants.

variant		T_w^i	p	η
SAM	all	1	2	0
	ON	$\begin{cases} 1 & \text{if } w_i \text{ is a normalization parameter} \\ 0 & \text{else} \end{cases}$	2	0
el. ℓ_2	all	$\lvert w_i \rvert$	2	0
	ON	$\begin{cases} \lvert w_i \rvert & \text{if } w_i \text{ is a normalization parameter} \\ 0 & \text{else} \end{cases}$	2	0
el. ℓ_2, orig.	all	$\begin{cases} \lvert w_i \rvert + \eta & \text{if } w_i \text{ is a weight parameter} \\ 1 + \eta & \text{if } w_i \text{ is a bias parameter} \end{cases}$	2	0.01
	ON	$\begin{cases} \lvert w_i \rvert + \eta & \text{if } w_i \text{ is a normalization weight} \\ 1 + \eta & \text{if } w_i \text{ is a normalization bias} \\ 0 & \text{else} \end{cases}$	2	0.01
el. ℓ_∞	all	$\lvert w_i \rvert$	∞	0
	ON	$\begin{cases} \lvert w_i \rvert & \text{if } w_i \text{ is a normalization parameter} \\ 0 & \text{else} \end{cases}$	∞	0
layer ℓ_2	all	$\lvert\lvert \mathbf{W}_{\text{layer[i]}} \rvert\rvert_2$	2	0
	ON	$\begin{cases} \lvert\lvert \mathbf{W}_{\text{layer[i]}} \rvert\rvert_2 & \text{if } w_i \text{ is a normalization parameter} \\ 0 & \text{else} \end{cases}$	2	0
Fisher	all	$\left(1 + \eta \left(\partial_{w_i} L_{Batch}(\mathbf{w})\right)^2\right)^{-0.5}$	2	1
	ON	$\begin{cases} \left(1 + \eta \left(\partial_{w_i} L_{Batch}(\mathbf{w})\right)^2\right)^{-0.5} & \text{if } w_i \text{ is a normalization parameter} \\ 0 & \text{else} \end{cases}$	2	1

B Further Experimental Results

B.1 SAM-ON on CIFAR

We omitted the results for ResNet-like models on CIFAR-10 in the main paper. Those are thus reported in Table 10. Due to the already very high accuracies, the differences between *SAM-ON* and *SAM-all* are smaller, yet on average *SAM-ON* is still clearly the better method. We further plot all considered SAM-variants for different values of ρ in Figure 6 for a WRN-28 and in Figure 7 for a ViT-S on CIFAR-100. We show results for various VGG-models [47] and DenseNet-100 [30] for CIFAR-10/100 in Table 11 and observe that *SAM-ON* consistently improves over *SAM-all*.

B.2 Additional ablation studies for sparse SAM

In this section we provide additional ablation studies for sparsified perturbation approaches as discussed in Section 5.1. Mi et al. [42] proposed two sparsified SAM (*SSAM*) approaches: Fisher SSAM (*SSAM-F*) and Dynamic SSAM (*SSAM-D*). As an extension to Figure 4 for ResNet-18 on CIFAR-10 data in the main paper we provide an accompanying Figure 8 which includes error bars

Table 10: ***SAM-ON* improves over *SAM-all* for BatchNorm and ResNets on CIFAR-10**: Test accuracy for ResNet-like models on CIFAR-10. Bold values mark the better performance between *SAM-ON* and *SAM-all* within a SAM-variant, and underline highlights the overall best method per model and augmentation

	SAM variant	RN-56		RNxT		WRN-28	
		all	onlyNorm	all	onlyNorm	all	onlyNorm
basic aug.	SGD	$94.28^{\pm 0.2}$		$95.37^{\pm 0.1}$		$96.20^{\pm 0.1}$	
	SAM	$94.94^{\pm 0.1}$	**95.18**$^{\pm 0.1}$	$96.35^{\pm 0.2}$	**96.48**$^{\pm 0.1}$	$97.08^{\pm 0.1}$	**97.10**$^{\pm 0.0}$
	elem. ℓ_2	**94.96**$^{\pm 0.1}$	$94.94^{\pm 0.2}$	$96.41^{\pm 0.1}$	**96.53**$^{\pm 0.1}$	$96.98^{\pm 0.2}$	**97.06**$^{\pm 0.0}$
	elem. ℓ_2, orig.	$95.14^{\pm 0.1}$	**95.21**$^{\pm 0.1}$	$96.40^{\pm 0.1}$	**96.41**$^{\pm 0.1}$	**97.10**$^{\pm 0.1}$	$97.07^{\pm 0.1}$
	elem. ℓ_∞	$94.93^{\pm 0.1}$	**94.96**$^{\pm 0.0}$	$96.06^{\pm 0.2}$	**96.22**$^{\pm 0.1}$	$96.95^{\pm 0.2}$	**97.00**$^{\pm 0.1}$
	Fisher	$95.01^{\pm 0.1}$	**95.03**$^{\pm 0.1}$	$96.31^{\pm 0.0}$	**96.55**$^{\pm 0.0}$	$96.95^{\pm 0.0}$	**97.13**$^{\pm 0.1}$
	layer. ℓ_2	$94.95^{\pm 0.2}$	**95.07**$^{\pm 0.1}$	$96.07^{\pm 0.3}$	**96.46**$^{\pm 0.1}$	**97.02**$^{\pm 0.0}$	$96.96^{\pm 0.1}$
basic aug. + AA	SGD	$94.70^{\pm 0.1}$		$96.19^{\pm 0.2}$		$97.01^{\pm 0.0}$	
	SAM	$95.25^{\pm 0.1}$	**95.40**$^{\pm 0.1}$	$96.98^{\pm 0.1}$	**97.22**$^{\pm 0.3}$	$97.57^{\pm 0.1}$	**97.58**$^{\pm 0.0}$
	elem. ℓ_2	**95.12**$^{\pm 0.0}$	$94.82^{\pm 0.2}$	$97.01^{\pm 0.0}$	**97.21**$^{\pm 0.1}$	$97.61^{\pm 0.0}$	**97.69**$^{\pm 0.0}$
	elem. ℓ_2, orig.	$95.39^{\pm 0.1}$	**95.60**$^{\pm 0.1}$	$97.24^{\pm 0.0}$	**97.33**$^{\pm 0.1}$	**97.60**$^{\pm 0.0}$	$97.56^{\pm 0.0}$
	elem. ℓ_∞	$95.12^{\pm 0.1}$	**95.48**$^{\pm 0.3}$	$96.70^{\pm 0.2}$	**96.91**$^{\pm 0.2}$	$97.52^{\pm 0.1}$	**97.62**$^{\pm 0.1}$
	Fisher	$95.19^{\pm 0.0}$	**95.38**$^{\pm 0.1}$	$96.77^{\pm 0.0}$	**97.24**$^{\pm 0.1}$	$97.53^{\pm 0.0}$	**97.65**$^{\pm 0.1}$
	layer. ℓ_2	**95.43**$^{\pm 0.3}$	$95.28^{\pm 0.1}$	$96.80^{\pm 0.1}$	**96.88**$^{\pm 0.1}$	**97.60**$^{\pm 0.0}$	$97.48^{\pm 0.1}$

Table 11: ***SAM-ON* improves over *SAM-all* for BatchNorm and more ResNet models**: Bold values mark the better performance between *SAM-ON* and *SAM-all* within a SAM-variant, and underline highlights the overall best method per model and augmentation

		VGG-13		VGG-16		VGG-19		DenseNet-100	
	SAM variant	all	onlyNorm	all	onlyNorm	all	onlyNorm	all	onlyNorm
CIFAR-100	SGD	$75.44^{\pm 0.2}$		$74.43^{\pm 0.4}$		$73.40^{\pm 0.2}$		$77.00^{\pm 0.2}$	
	SAM	$76.74^{\pm 0.2}$	**77.57**$^{\pm 0.1}$	$75.81^{\pm 0.2}$	**76.86**$^{\pm 0.1}$	$74.08^{\pm 0.6}$	**75.60**$^{\pm 0.1}$	$79.42^{\pm 0.6}$	**79.90**$^{\pm 0.3}$
	elem. ℓ_2	$76.65^{\pm 0.1}$	**77.49**$^{\pm 0.1}$	$75.95^{\pm 0.2}$	**76.45**$^{\pm 0.2}$	$74.72^{\pm 0.2}$	**75.12**$^{\pm 0.1}$	$78.90^{\pm 0.2}$	**79.83**$^{\pm 0.3}$
	elem. $\ell_2, \eta = 0.01$	$77.27^{\pm 0.2}$	**77.37**$^{\pm 0.2}$	$76.65^{\pm 0.1}$	**76.66**$^{\pm 0.3}$	$75.00^{\pm 0.5}$	**75.44**$^{\pm 0.2}$	$79.94^{\pm 0.4}$	**80.14**$^{\pm 0.1}$
	elem. ℓ_∞	$76.82^{\pm 0.3}$	**77.62**$^{\pm 0.2}$	$75.43^{\pm 0.4}$	**76.68**$^{\pm 0.1}$	$72.74^{\pm 0.2}$	**74.50**$^{\pm 0.4}$	$79.47^{\pm 0.3}$	**79.64**$^{\pm 0.2}$
	Fisher, $\eta = 1.$	$76.76^{\pm 0.2}$	**77.68**$^{\pm 0.4}$	$75.85^{\pm 0.2}$	**76.99**$^{\pm 0.1}$	$74.03^{\pm 0.2}$	**74.96**$^{\pm 0.3}$	$79.68^{\pm 0.2}$	**80.38**$^{\pm 0.3}$
	layer. ℓ_2	$76.76^{\pm 0.2}$	**77.91**$^{\pm 0.2}$	$75.99^{\pm 0.2}$	**77.12**$^{\pm 0.2}$	$74.65^{\pm 0.5}$	**75.28**$^{\pm 0.2}$	$78.25^{\pm 0.2}$	**79.86**$^{\pm 0.3}$
CIFAR-10	SGD	$94.29^{\pm 0.0}$		$93.85^{\pm 0.3}$		$93.82^{\pm 0.0}$		$94.51^{\pm 0.1}$	
	SAM	$94.88^{\pm 0.1}$	**95.19**$^{\pm 0.2}$	$94.96^{\pm 0.0}$	**95.02**$^{\pm 0.1}$	$94.58^{\pm 0.1}$	**94.81**$^{\pm 0.2}$	$95.84^{\pm 0.2}$	**95.89**$^{\pm 0.0}$
	elem. ℓ_2	$94.97^{\pm 0.1}$	**95.08**$^{\pm 0.0}$	$95.01^{\pm 0.1}$	**95.02**$^{\pm 0.1}$	$94.68^{\pm 0.0}$	**94.99**$^{\pm 0.1}$	$95.76^{\pm 0.2}$	**95.86**$^{\pm 0.2}$
	elem. $\ell_2, \eta = 0.01$	$94.95^{\pm 0.0}$	**95.13**$^{\pm 0.1}$	$94.87^{\pm 0.1}$	**95.12**$^{\pm 0.1}$	$94.66^{\pm 0.1}$	**94.87**$^{\pm 0.2}$	**95.92**$^{\pm 0.3}$	$95.85^{\pm 0.1}$
	elem. ℓ_∞	$94.96^{\pm 0.1}$	**95.06**$^{\pm 0.0}$	$94.74^{\pm 0.2}$	**94.91**$^{\pm 0.0}$	$94.68^{\pm 0.1}$	**94.73**$^{\pm 0.1}$	$95.56^{\pm 0.2}$	**95.91**$^{\pm 0.1}$
	Fisher, $\eta = 1.$	$95.07^{\pm 0.0}$	**95.17**$^{\pm 0.0}$	$94.77^{\pm 0.0}$	**95.10**$^{\pm 0.2}$	$94.55^{\pm 0.0}$	**94.91**$^{\pm 0.1}$	$95.65^{\pm 0.1}$	**96.00**$^{\pm 0.1}$
	layer. ℓ_2	$94.78^{\pm 0.1}$	**95.09**$^{\pm 0.1}$	$94.54^{\pm 0.1}$	**95.08**$^{\pm 0.1}$	$66.21^{\pm 48.7}$	**94.96**$^{\pm 0.1}$	$95.48^{\pm 0.2}$	**95.82**$^{\pm 0.1}$

and comparisons with the dynamic sparse perturbation approach (*SSAM-D*) [42]. We also provide additional results for *SSAM-D* for a WideResNet-28 on CIFAR-100 data in Table 12. We found optimal performance for *SSAM* for 50% sparsity and $\rho = 0.1$ on CIFAR-10 and $\rho = 0.2$ on CIFAR-100 (as also observed in [42] for slightly different training settings). We find that although both *SSAM* approaches can perform on par or even outperform regular *SAM*, they are less effective than our *SAM-ON* approach. The generalization gap increases even further when considering the same high sparsity levels as for *SAM-ON*.

Table 12: Although *SSAM-F* and *SSAM-D* [42] with different sparsity levels can outperform *SAM-all* on CIFAR-100 with WRN-28, they are less effective than *SAM-ON*.

	SAM	SAM-ON	SAM-rand	SSAM-F		SSAM-D	
Sparsity	0%	99.95%	99.95%	50%	99.95%	50%	99.95%
Accuracy	$83.11_{\pm 0.3}$	**84.19**$_{\pm 0.2}$	$80.97_{\pm 0.2}$	$83.94_{\pm 0.1}$	$83.14_{\pm 0.1}$	$83.53_{\pm 0.1}$	$81.01_{\pm 0.1}$

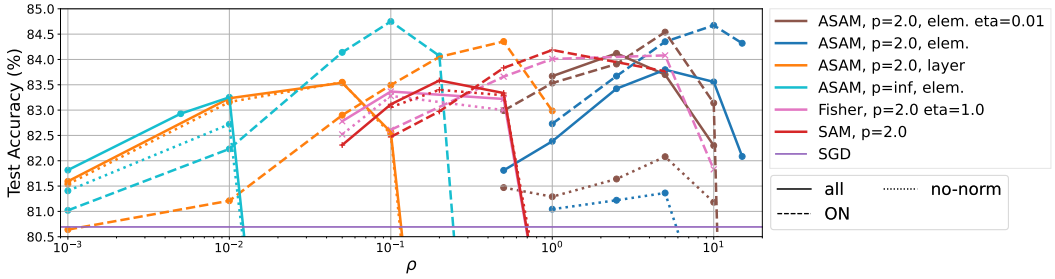

Figure 6: All considered SAM-variants and their *SAM-ON* counterpart for a WRN-28 on CIFAR-100.

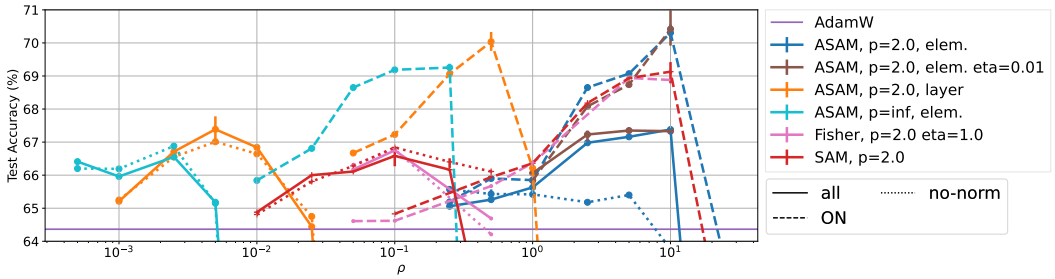

Figure 7: All considered SAM-variants and their *SAM-ON* counterpart for a ViT-S on CIFAR-100.

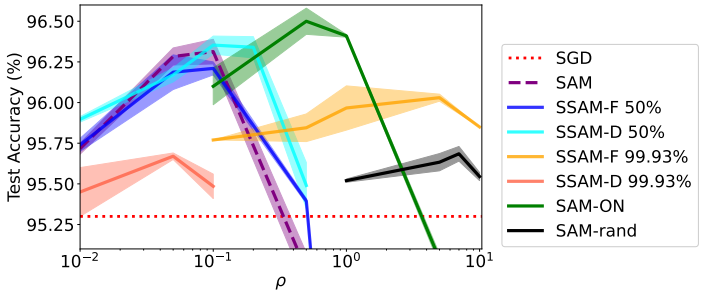

Figure 8: *SAM-ON* outperforms *SSAM-F* and *SSAM-D* [42] (with different sparsity levels) and random mask *SAM-rand* (same sparsity level 99.93% as *SAM-ON*) sparse perturbation approaches on CIFAR-10 for ResNet-18.

B.3 Finetuning from ImageNet-21k

Since ViTs are commonly trained on large-scale datasets and then fine-tuned, we investigate this scenario for *SAM-ON*. In particular, we consider a ViT-S pretrained on ImageNet-21k from [48]. We fine-tune for 9 epochs with SGD for a range of SAM-variants with their respective *SAM-ON* counterpart. For each setup, we probe three values of ρ and report the best result in Table 13. We find in this setting that *SAM-ON* performs on par with *SAM-all* although there are small differences across SAM variants: for layerwise-ℓ_2 *SAM-ON* performs slightly worse, whereas for all other variants *SAM-ON* performs equally well or slightly better than *SAM-all*. In all cases *SAM-ON* outperforms plain SGD.

Table 13: Results for ImageNet-1k fine-tuning of a ViT-S-224 from a ImageNet-21k model.

SGD	SAM		ASAM elem. ℓ_2		ASAM layer. ℓ_2		ASAM elem. ℓ_∞	
	all	ON	all	ON	all	ON	all	ON
81.62	**81.75**	**81.75**	81.73	**81.75**	**81.79**	81.75	**81.84**	**81.84**

B.4 Adversarial robustness

Here, we provide additional results and extend the discussion on adversarial robustness from Section 4.2. In a study by Wei et al. [52] SAM-trained models showed non-trivial robustness to small adversarial perturbations [50]. Since there are several works highlighting the role of normalization layers for adversarial robustness [9, 54], it is interesting to investigate whether the robustness properties of SAM can be preserved when training with SAM-ON instead of SAM-all. In Table 15 we report the adversarial robustness of the ViT-S trained from scratch on ImageNet (as reported in Section 4.2 evaluated with the two white-box attacks from APGD, but for more radii). The SAM-ON models are not only better than the base optimizer, but consistently outperform the SAM-all models by a small margin. For a WRN-28-10 on CIFAR-100 the differences are less pronounced and often within the standard deviation (reported over 3 seeds in Table 14). SAM-ON also improves over SAM-all, but for the ASAM-elementwise-ℓ_∞ the all-variant is slightly better than the ON-variant. Overall, we find that in order to get SAM-like improvements for adversarial robustness (as shown in [52]) it is enough to only perturb the normalization layers in SAM, illustrating again their special role.

Table 14: **Adversarial robustness CIFAR-100:** Reported is robust accuracy (in %) for a WRN-28 trained from scratch on CIFAR-100. Adversarial robustness is evaluated with the two whitebox APGD attacks from autoattack [16].

		SGD	SAM		ASAM-el.-l_∞	
threat model	ϵ		all	ON	all	ON
ℓ_2	0.10	$18.14^{\pm 0.11}$	$28.14^{\pm 1.09}$	**31.28** $^{\pm 0.50}$	$30.33^{\pm 0.80}$	$30.16^{\pm 0.26}$
ℓ_2	0.20	$2.33^{\pm 0.11}$	$5.39^{\pm 0.34}$	**6.62** $^{\pm 0.07}$	**6.63** $^{\pm 0.12}$	$6.10^{\pm 0.18}$
ℓ_∞	$1/255$	$10.29^{\pm 0.04}$	$17.96^{\pm 1.08}$	**19.56** $^{\pm 0.33}$	**20.69** $^{\pm 0.81}$	$18.63^{\pm 0.30}$
ℓ_∞	$2/255$	$0.67^{\pm 0.01}$	$1.96^{\pm 0.17}$	**2.16** $^{\pm 0.07}$	**2.62** $^{\pm 0.01}$	$2.05^{\pm 0.17}$
Clean acc.		$80.7^{\pm 0.2}$	$83.1^{\pm 0.3}$	$\mathbf{84.2}^{\pm 0.2}$	$83.3^{\pm 0.2}$	$\mathbf{84.1}^{\pm 0.2}$

Table 15: **Adversarial robustness ImageNet:** Reported is robust accuracy (in %) for a ViT-S trained from scratch on ImageNet, as reported in Table 4. Adversarial robustness is evaluated with the two whitebox APGD attacks from autoattack [16].

		AdamW			Lion		
	ϵ	vanilla	SAM-all	SAM-ON	vanilla	SAM-all	SAM-ON
ℓ_2	0.25	$19.67^{\pm 0.47}$	$37.53^{\pm 0.69}$	**41.16** $^{\pm 0.24}$	$22.01^{\pm 0.78}$	$38.52^{\pm 0.66}$	**43.12** $^{\pm 0.97}$
ℓ_2	0.50	$5.47^{\pm 0.18}$	$17.71^{\pm 0.61}$	**22.72** $^{\pm 0.25}$	$6.63^{\pm 0.46}$	$19.03^{\pm 0.92}$	**24.27** $^{\pm 1.34}$
ℓ_2	1.00	$0.43^{\pm 0.09}$	$3.34^{\pm 0.36}$	**5.58** $^{\pm 0.19}$	$0.57^{\pm 0.07}$	$3.98^{\pm 0.28}$	**6.64** $^{\pm 0.69}$
ℓ_∞	$0.25/255$	$33.45^{\pm 0.80}$	$48.08^{\pm 0.14}$	**49.34** $^{\pm 0.08}$	$35.31^{\pm 0.08}$	$49.57^{\pm 0.60}$	**51.37** $^{\pm 0.99}$
ℓ_∞	$0.5/255$	$14.98^{\pm 0.18}$	$29.68^{\pm 0.09}$	**32.46** $^{\pm 0.15}$	$15.86^{\pm 0.13}$	$31.68^{\pm 0.62}$	**34.23** $^{\pm 1.73}$
ℓ_∞	$1/255$	$2.61^{\pm 0.16}$	$8.64^{\pm 0.01}$	**10.82** $^{\pm 0.56}$	$2.93^{\pm 0.29}$	$10.02^{\pm 0.56}$	**12.03** $^{\pm 1.30}$
Clean acc.		$66.89^{\pm 0.04}$	$71.47^{\pm 0.12}$	$71.37^{\pm 0.026}$	$68.20^{\pm 0.02}$	$71.90^{\pm 0.19}$	$\mathbf{72.64}^{\pm 0.14}$

B.5 Machine translation task

To probe the effectiveness of *SAM-ON* outside the vision domain, we apply it to the IWSLT'14 DE-EN machine translation task, following the setup of Kwon et al. [37]. We report the resulting Bleu scores in Table 16: *SAM-all* and *SAM-ON* perform similar (within standard deviations reported over 3 random seeds), both improving over the vanilla optimizer. While being very limited in its scope, this experiment is a first hint that *SAM-ON* might also be effective outside the vision domain. Proper evaluations, as for instance done in [7], are required to confirm this for large-scale settings.

Table 16: IWSLT-DE-EN Bleu scores. Reported over 3 random seeds.

vanilla	SAM-all	SAM-ON
$34.56^{\pm 0.11}$	$34.83^{\pm 0.10}$	$34.95^{\pm 0.16}$

B.6 Weight distribution after training

In order to get a better understanding of the impact of *SAM-ON* on γ and β (as defined in Eq. 1), we train a WideResNet-28-10 with different SAM-variants and both *SAM-ON* and *all*. We show the distribution of $|w_i|$, i.e. the parameter magnitudes, at the end of training for different layer types in Figure 9. Different to the discussion in Section 5.3, we show the y-axis on log-scale, in order to inspect more nuanced differences. For elementwise ℓ_2 there is no strong change in the distribution of the BatchNorm parameters between *all* and *SAM-ON*. For elementwise ℓ_∞, layerwise ℓ_2 and SAM, however, the magnitude of the BatchNorm parameters shifts clearly towards larger values, especially for the weight parameters. We note that this resembles a pattern we observed when comparing the optimal ρ-value for *all* and *SAM-ON* in Table 7: The optimal ρ of elementwise ℓ_2 did not change much for ResNet architectures, whereas for the other considered methods, it shifted towards larger values for *SAM-ON*. Additionally and in contrast to the other methods, the elementwise ℓ_2 variant showed a strong performance decrease in *no-norm* (Figure 1), indicating that it implicitly focuses on perturbing the BatchNorm layers already. We note that larger BatchNorm parameters do not necessarily indicate a functionally different network, since there are many reparameterization invariances in ReLU networks, some of which ASAM tries to leverage in its perturbation definition Eq. (4). Nevertheless, the scale of the network still has an impact on the training dynamics, since other methods like e.g. weight decay depend on it. We discuss the impact of weight decay further in Appendix B.9.

B.7 Removing the affine parameters

Frankle et al. [24] found for SGD that fixing the normalization parameters typically decreases the generalization performance of networks. As an ablation, we therefore study the effect of SAM when the normalization weights are non-trainable. This is, we set $\gamma = 1$ and $\beta = 0$ and train the remaining parameters with SAM. The results are shown for a WRN-28 in Figure 10, where it can be seen that fixing the normalization parameters (*fix-norm*) does *not* lead to a decrease in the performance of SAM. We thus hypothesize that in certain settings, SAM might not leverage the expressive power of the normalization layers, which might contribute to the improved performance of *SAM-ON*.

B.8 Training BatchNorm and only BatchNorm

The affine parameters of the normalization layers are relatively understudied in the literature. Recently, Frankle et al. [24] were able to obtain surprisingly high performance for ResNet architectures by only training the BatchNorm layers (freezing all other parameters), illustrating their expressive power. We study the effect of SAM in this setting (i.e. when all parameters except for the BatchNorm layers are frozen) for a ResNet-101 and a WRN-28 on CIFAR-10 and find that SAM still aids generalization in this setting (Table 17).

Table 17: Effect of SAM when training *only* BatchNorm layers, for networks trained on CIFAR-10.

Model	SGD	SAM $\rho = 0.01$	SAM $\rho = 0.05$
ResNet-101	78.75	78.63	**79.27**
WRN-28	63.49	**64.48**	62.70

B.9 Weight decay and dropout

Here, we explore potential connections of *SAM-ON* with weight decay and dropout. Since weight decay is sometimes applied to all network parameters, and sometimes normalization layers are omitted, it is worth investigating if the benefits of *SAM-ON* can be attributed to its interaction with weight decay. To this end, we train a WRN-28 with SGD, *SAM-all* and *SAM-ON*, and apply weight decay to either all parameters, all *except* the normalization layers, or not at all (Figure 11, right). For each setting *SAM-ON* outperforms SAM, outlining that its success should not be attributed to the interaction with weight decay.

We further test if *SAM-ON*-like performance can be achieved by simply applying stronger regularization and stochasticity to the normalization parameters. To this end, we apply dropout solely on the normalization layers (Figure 11, left) and find that this is not the case.

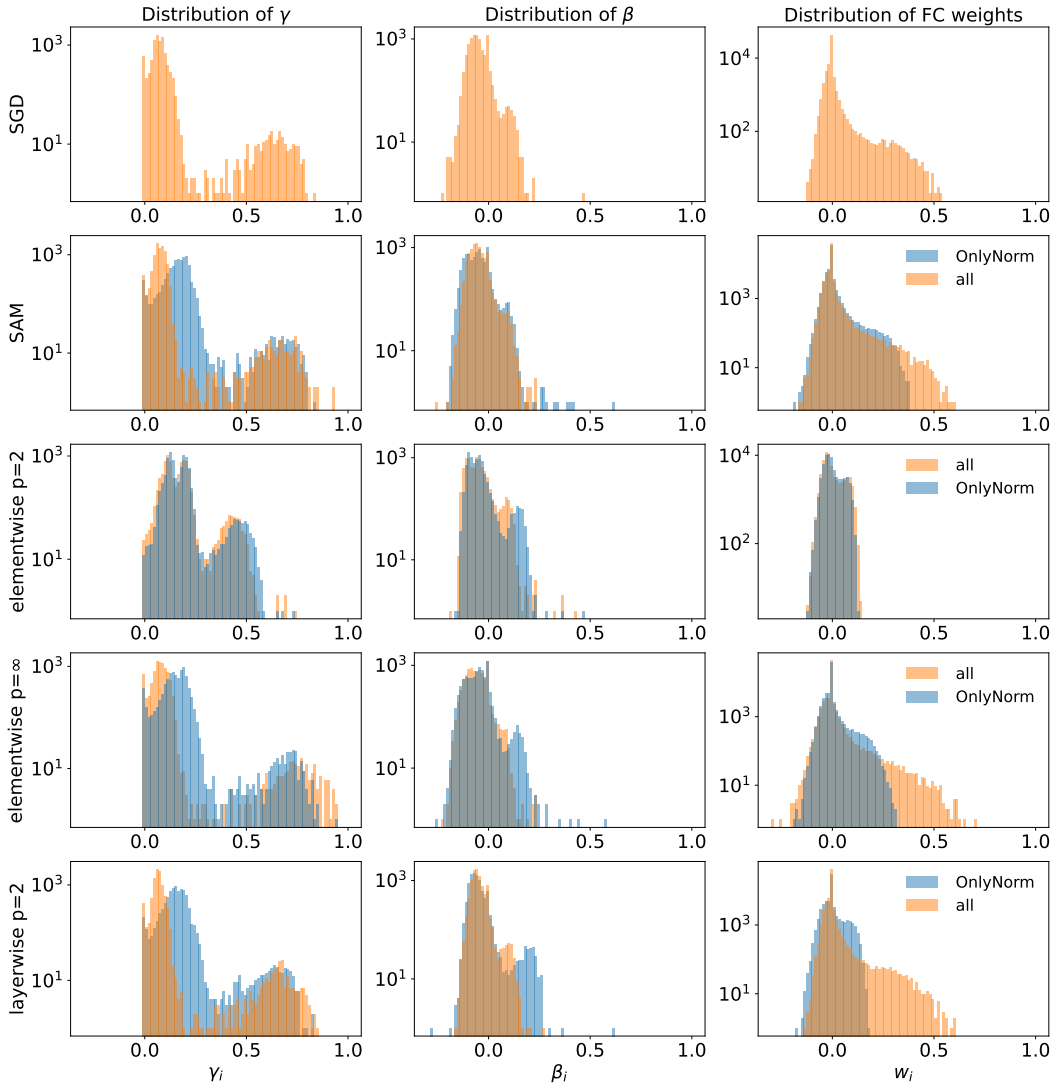

Figure 9: SAM-ON leads to a shift in the distribution of γ.

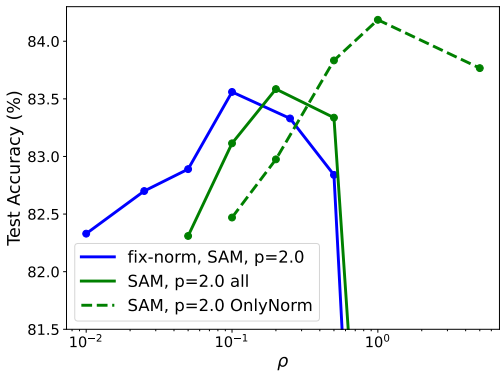

Figure 10: When training with SAM, fixing $\gamma = 1, \beta = 0$ (*fix-norm*) barely changes the performance of the network. WRN-28, CIFAR-100.

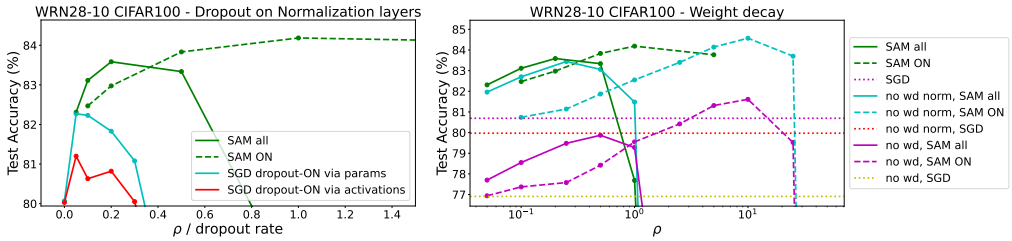

Figure 11: **Left:** Applying dropout only to the normalization layers (blue/red) performs worse than *SAM-ON*. **Right:** *SAM-ON* improves over *SAM-all* irrespective of whether weight decay is applied to all parameters (green), all *except* the normalization layers (blue) or not at all (yellow).

B.10 Details on sharpness evaluation

For the following discussion, we note that the term *generalization* is sometimes used as the difference between train and test error, while in other cases people use it as a synonym for test error. Since in CIFAR settings the models achieve train error close to zero, the two definitions become equivalent.

Many studies have attempted to better understand the possible connection between the generalization of deep neural networks and the flatness of the loss-surface [26, 32, 22, 34, 18]. Recently, Andriushchenko et al. [4] conducted a large-scale study for a range of models, datasets, and sharpness-definitions, finding that "while there definitely exist restricted settings where correlation between sharpness and generalization is significantly positive (e.g., for ResNets on CIFAR-10 with a specific combination of augmentations and mixup) it is not true anymore when we compare all models jointly" and concluding "that one should avoid blanket statements like *flat minima generalize better*". In order to evaluate sharpness, we therefore adopt their setup and choose the best-performing sharpness measure for CIFAR from their study, which is logit-normalized elementwise-adaptive worst-case-ℓ_∞-m-sharpness. This is, m-sharpness s_w^m is defined as the largest possible change in loss within the adaptive perturbation model defined in 4,

$$s_w^m = \mathbb{E}_{x,y\sim D_m} \max_{||T_w^{-1}\epsilon||_p \leq \rho} L(\mathbf{w}+\epsilon) - L(\mathbf{w}) \tag{6}$$

where $T_w^i = |w_i|$, $p = \infty$ and D_m returns data batches of size m. ρ here denotes the size of the ball over which sharpness is evaluated and is not to be confused with the ρ from the SAM-algorithm. Like for ASAM [37], the motivation behind adaptive sharpness measures is to make them invariant to reparameterizations of the network. Further, the logit-outputs of the network are normalized with respect to their ℓ_2-norm in order to mitigate the scale-sensitivity of classification losses. In practice, Andriushchenko et al. [4] compute s_w^m over a subset of the train set of size 1024 and use $m = 128$, i.e. average 8 batches. We use a subset of size 2048 in order to obtain more reliable sharpness estimates, and adopt $m = 128$. The maximization in (6) is performed with AutoPGD [16], a hyperparameter-free method designed for accurate estimation of adversarial robustness. It is to note that except for the logit-normalization, the sharpness definition reported in Table 6 corresponds exactly to the perturbation model that ASAM elementwise ℓ_∞ uses, and hence the 1-step sharpness reported should be fairly close the the objective that ASAM elementwise ℓ_∞ actually minimizes during training. While ASAM elementwise ℓ_∞ yields slightly smaller sharpness values than the conventional SAM algorithm, the differences are rather small when compared to the significantly sharper *SAM-ON* models. For the results in Table 6 in the main paper we tuned the sharpness radius ρ such that we obtain sharpness values similar to those reported to yield the highest correlation in Andriushchenko et al. [4]. In Table 18 we report sharpness values for a ResNeXt-model, in addition to the WRN-28 from the main paper. In all cases the *SAM-ON* models are sharper than the *SAM-all* models yet generalize better. In Table 19 we further report other sharpness measures without logit-normalization for a WRN-28. SAM-ON is sharper than SAM-all with respect to most metrics, although there exist some exceptions. It should however be stressed that many of those metrics did not show good correlation with generalization in the study by Andriushchenko et al. [4].

Table 18: Sharpness evaluation of both a WRN-28 and a ResNeXt. *SAM-ON* is sharper than *SAM-all* in all cases. Shown is 20-step logit-normalized ℓ_∞ sharpness from [2], averaged over three models per method. Dataset considered is CIFAR-100.

		SGD	SAM		ASAM-el.-ℓ_∞	
			all	ON	all	ON
WRN-28	Test Accuracy (%)	$80.71^{\pm 0.2}$	$83.11^{\pm 0.3}$	$\mathbf{84.19}^{\pm 0.2}$	$83.25^{\pm 0.2}$	$\mathbf{84.14}^{\pm 0.2}$
	ℓ_∞-sharpness, $\rho = 0.003$	$0.071^{\pm 0.000}$	$\mathbf{0.048}^{\pm 0.001}$	$0.090^{\pm 0.005}$	$\mathbf{0.048}^{\pm 0.001}$	$0.078^{\pm 0.004}$
	ℓ_∞-sharpness, $\rho = 0.005$	$0.201^{\pm 0.001}$	$\mathbf{0.139}^{\pm 0.004}$	$0.296^{\pm 0.018}$	$\mathbf{0.124}^{\pm 0.002}$	$0.283^{\pm 0.011}$
	ℓ_∞-sharpness, $\rho = 0.007$	$0.433^{\pm 0.002}$	$\mathbf{0.309}^{\pm 0.011}$	$0.585^{\pm 0.018}$	$\mathbf{0.255}^{\pm 0.005}$	$0.580^{\pm 0.020}$
ResNeXt	Test Accuracy (%)	$80.16^{\pm 0.3}$	$81.79^{\pm 0.4}$	$\mathbf{82.22}^{\pm 0.2}$	$81.02^{\pm 0.6}$	$\mathbf{82.38}^{\pm 0.3}$
	ℓ_∞-sharpness, $\rho = 0.001$	$0.036^{\pm 0.001}$	$\mathbf{0.029}^{\pm 0.000}$	$0.034^{\pm 0.000}$	$\mathbf{0.026}^{\pm 0.002}$	$0.034^{\pm 0.001}$
	ℓ_∞-sharpness, $\rho = 0.003$	$0.164^{\pm 0.005}$	$\mathbf{0.117}^{\pm 0.004}$	$0.140^{\pm 0.002}$	$\mathbf{0.099}^{\pm 0.010}$	$0.147^{\pm 0.001}$
	ℓ_∞-sharpness, $\rho = 0.005$	$0.383^{\pm 0.011}$	$\mathbf{0.252}^{\pm 0.008}$	$0.291^{\pm 0.005}$	$\mathbf{0.203}^{\pm 0.021}$	$0.312^{\pm 0.001}$

Table 19: Additional sharpness measures. WRN-28 (no logitnorm).

		SGD	SAM		ASAM-el.-ℓ_∞	
	adaptive		all	ON	all	ON
Test Accuracy (%)		$80.71^{\pm 0.2}$	$83.11^{\pm 0.3}$	$\mathbf{84.19}^{\pm 0.2}$	$83.25^{\pm 0.2}$	$\mathbf{84.14}^{\pm 0.2}$
ℓ_2 avg, $\rho = 0.005$	False	$1.358^{\pm 0.049}$	**0.515** $^{\pm 0.020}$	$2.372^{\pm 0.071}$	**0.569** $^{\pm 0.012}$	$2.141^{\pm 0.045}$
ℓ_2 avg, $\rho = 0.1$	True	$0.042^{\pm 0.001}$	**0.019** $^{\pm 0.001}$	$0.022^{\pm 0.001}$	$0.040^{\pm 0.001}$	**0.019** $^{\pm 0.001}$
ℓ_∞ avg, $\rho = 0.01$	False	$2.643^{\pm 0.097}$	**1.264** $^{\pm 0.028}$	$3.455^{\pm 0.050}$	**1.304** $^{\pm 0.007}$	$3.259^{\pm 0.031}$
ℓ_∞ avg, $\rho = 0.2$	True	$0.078^{\pm 0.001}$	$0.035^{\pm 0.001}$	$0.034^{\pm 0.004}$	$0.068^{\pm 0.003}$	**0.031** $^{\pm 0.001}$
ℓ_2-worst, $\rho = 0.05$	False	$0.501^{\pm 0.048}$	$0.655^{\pm 0.277}$	$0.701^{\pm 0.057}$	$0.768^{\pm 0.141}$	**0.313** $^{\pm 0.044}$
ℓ_2-worst, $\rho = 0.25$	True	$0.065^{\pm 0.008}$	$0.033^{\pm 0.004}$	$0.037^{\pm 0.017}$	$0.056^{\pm 0.006}$	$0.062^{\pm 0.001}$
ℓ_∞-worst, $\rho = 1e-05$	False	$0.149^{\pm 0.003}$	**0.055** $^{\pm 0.002}$	$0.144^{\pm 0.005}$	**0.050** $^{\pm 0.002}$	$0.123^{\pm 0.007}$
ℓ_∞-worst, $\rho = 0.004$	True	$0.537^{\pm 0.023}$	**0.262** $^{\pm 0.009}$	$0.600^{\pm 0.053}$	**0.255** $^{\pm 0.011}$	$0.505^{\pm 0.027}$

C Convergence Analysis

We provide in this section a convergence analysis for *SAM-ON* in the non-convex setting. Using standard assumptions we obtain a theorem which resembles findings for closely related methods such as found in [2, 42].

Our assumptions:

Assumption C.1. We assume function $f : \mathbb{R}^n \to \mathbb{R}$ to be L-smooth: there exists $L > 0$ such that

$$\|\nabla f(v) - \nabla f(w)\|_2 \leq L\|v - w\|_2, \ \forall v, w \in \mathbb{R}^n. \tag{7}$$

Assumption C.2. There exists $M > 0$ for any sample x_i such that

$$\|\nabla f_{x_i}(w)\|_2^2 \leq M, \ \ \forall w \in \mathbb{R}^n. \tag{8}$$

Remark C.3. If Assumption C.1 holds (L-smoothness), then $\forall v, w \in \mathbb{R}^n$:

$$|f(v) - (f(w) + \nabla f(w)^T(v - w))| \leq \frac{L}{2}\|v - w\|_2^2. \tag{9}$$

This well-known result can be derived using the fundamental theorem of calculus and Cauchy-Schwartz.

Remark C.4. Assumption C.2 guarantees that the variance of the stochastic gradient is less than M.

SAM-ON. In the following we shall denote the true gradient as $\nabla f(w)$ and the noisy observation gradient as $g(w)$. The gradient of the loss of the ith training example is denoted as $g_{x_i}(w)$. We partition the neural network parameters layer-wise as $w = \{w_N, w_A\}$, with $w_N \in \mathbb{R}^{n_N}, w_A \in \mathbb{R}^{n_A}$, $n = n_N + n_A$, where w_N represent the normalization layer parameters and w_A all other layers. The

iteration for w_N is:

$$w_N^{t+1/2} = w_N^t + \rho \frac{g_{N,x_i}(w^t)}{\|g_{N,x_i}(w^t)\|}$$
$$w_N^{t+1} = w_N^t - h\, g_{N,x_i}\left(w^{t+1/2}\right) \tag{10}$$

and for w_A is:

$$w_A^{t+1/2} = w_A^t$$
$$w_A^{t+1} = w_A^t - h\, g_{A,x_i}\left(w^{t+1/2}\right). \tag{11}$$

Theorem C.5. *Assuming C.1 and C.2, $h \leq 1/L$, we obtain:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(w^t)\|^2\right] \leq \frac{2(f(w^0) - f(w^*))}{hT} + 2LhM + L^2\rho^2(1+Lh), \tag{12}$$

with w^* the optimal solution to $f(w)$.

Proof. From Assumption C.1 and thus Remark C.3 it follows that:

$$f(w^{t+1}) \leq f(w^t) + \nabla f(w^t)\cdot(w^{t+1} - w^t) + \frac{L}{2}\|w^{t+1} - w^t\|^2 \tag{13}$$
$$\leq f(w^t) - h\nabla f(w^t)\cdot g_{x_i}\left(w^{t+1/2}\right) + \frac{h^2 L}{2}\left\|g_{x_i}\left(w^{t+1/2}\right)\right\|^2 \tag{14}$$
$$= f(w^t) - h\nabla f(w^t)\cdot g_{x_i}\left(w^{t+1/2}\right)$$
$$+ \frac{h^2 L}{2}\left(\|\nabla f(w^t) - g_{x_i}(w^{t+1/2})\|^2 - \|\nabla f(w^t)\|^2 + 2\left(\nabla f(w^t)\cdot g_{x_i}(w^{t+1/2})\right)\right)$$
$$= f(w^t) - \frac{Lh^2}{2}\|\nabla f(w^t)\|^2 + \frac{Lh^2}{2}\|\nabla f(w^t) - g_{x_i}(w^{t+1/2})\|^2$$
$$- (1-Lh)h\left(\nabla f(w^t)\cdot g_{x_i}(w^{t+1/2})\right) \tag{15}$$
$$\leq f(w^t) - \frac{Lh^2}{2}\|\nabla f(w^t)\|^2 + Lh^2\|\nabla f(w^t) - g_{x_i}(w^t)\|^2$$
$$+ Lh^2\|g_{x_i}(w^t) - g_{x_i}(w^{t+1/2})\|^2 - (1-Lh)h\left(\nabla f(w^t)\cdot g_{x_i}(w^{t+1/2})\right). \tag{16}$$

Taking the double expectation gives (because unbiased gradient and Assumption C.2 and Remark C.4):

$$\mathbb{E}[f(w^{t+1})] \leq \mathbb{E}[f(w^t)] - \frac{Lh^2}{2}\mathbb{E}\|\nabla f(w^t)\|^2 + Lh^2 M$$
$$+ \underbrace{Lh^2\|g(w^t) - g(w^{t+1/2})\|^2}_{\mathcal{A}} - (1-Lh)h\underbrace{\mathbb{E}\left[\nabla f(w^t)\cdot g(w^{t+1/2})\right]}_{\mathcal{B}}. \tag{17}$$

For term $\mathcal{A}$ we obtain using Assumption C.1:

$$\mathcal{A} \leq L^3h^2\|w^t - w^{t+1/2}\|^2 = L^3h^2\rho^2. \tag{18}$$

For term $\mathcal{B}$ we obtain:

$$\mathcal{B} = \mathbb{E}\left[\{\nabla f_N(w^t), \nabla f_A(w^t)\}\cdot\{g_N(w^{t+1/2}), g_A(w^{t+1/2})\}\right] \tag{19}$$
$$= \mathbb{E}[\nabla f_A(w^t)\cdot(g_A(w^{t+1/2}) - g_A(w^t) + g_A(w^t))]$$
$$+ \mathbb{E}[\nabla f_N(w^t)\cdot(g_N(w^{t+1/2}) - g_N(w^t) + g_N(w^t))] \tag{20}$$
$$= \mathbb{E}\left[\|\nabla f(w^t)\|^2\right]$$
$$+ \underbrace{\mathbb{E}[\nabla f_A(w^t)\cdot(g_A(w^{t+1/2}) - g_A(w^t))] + \mathbb{E}[\nabla f_N(w^t)\cdot(g_N(w^{t+1/2}) - g_N(w^t))]}_{\mathcal{C}}. \tag{21}$$

Using $xy \leq \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|y\|_2^2$ and Assumption C.1 we get for $\mathcal{C}$:

$$|\mathcal{C}| \leq \frac{1}{2}\mathbb{E}\left[\|\nabla f(w^t)\|^2\right] + \frac{L^2}{2}\|w^{t+1/2} - w^t\|^2 = \frac{1}{2}\mathbb{E}\left[\|\nabla f(w^t)\|^2\right] + \frac{L^2\rho^2}{2}. \tag{22}$$

Plugging this into (17) gives:

$$\begin{aligned}\mathbb{E}[f(w^{t+1})] &\leq \mathbb{E}[f(w^t)] - \frac{Lh^2}{2}\mathbb{E}\|\nabla f(w^t)\|^2 + Lh^2M + L^3h^2\rho^2 - (1-Lh)h\mathbb{E}\|\nabla f(w^t)\|^2 \\ &\quad + (1-Lh)h\left(\frac{1}{2}\mathbb{E}\|\nabla f(w^t)\|^2 + \frac{L^2\rho^2}{2}\right) \qquad (23) \\ &\leq \mathbb{E}[f(w^t)] - \frac{h}{2}\mathbb{E}\|\nabla f(w^t)\|^2 + Lh^2M + \frac{1}{2}hL^2\rho^2(1+Lh). \qquad (24)\end{aligned}$$

In T iterations we obtain using a telescoping sum:

$$\begin{aligned}f(w^*) - f(w^0) &\leq \mathbb{E}[f(w^T)] - f(w^0) \\ &\leq -\frac{h}{2}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w^t)\|^2\right] + Lh^2MT + \frac{1}{2}hL^2\rho^2(1+Lh)T. \qquad (25)\end{aligned}$$

This gives Theorem C.5. □