UNITY OF OPPOSITES: SELFNORM AND CROSSNORM FOR MODEL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Studies have demonstrated the fundamental behavioral difference between human vision and Convolutional Neural Networks (CNNs). Human vision can make robust object recognition mainly using object shape (content). In contrast, CNNs are not only texture-biased but also sensitive to texture (style) noises, giving rise to security concerns in practice. To reduce CNNs' texture sensitivity and bias, this paper proposes SelfNorm and CrossNorm, respectively. SelfNorm uses attention to recalibrate feature styles by emphasizing primary styles and suppressing trivial ones. CrossNorm exchanges styles between feature channels to perform style augmentation, diversifying the content and style mixtures. SelfNorm and CrossNorm explore the opposite directions in utilizing style. Nevertheless, they complement each other and seek the same goal: model robustness. Surprisingly, they can boost the performance of corrupted and clean test data simultaneously. Extensive experiments on different tasks (classification and segmentation) and settings (supervised and semi-supervised) show their effectiveness.

1 INTRODUCTION

It is well-known that biological neural networks inspire the design of artificial counterparts. However, they perform quite differently on robustness evaluation. For object recognition, a recent study (Hendrycks & Dietterich, 2019) shows that CNNs are sensitive to common texture noises, e.g., brightness, which human vision can easily ignore. Another study (Geirhos et al., 2019) reveals that CNNs are overly texture-biased through cue conflict experiments. In contrast, human vision has a talent for capturing shape. Although texture sensitivity is partially due to texture bias, they focus on different aspects. Texture sensitivity is caused by overfitting to trivial textures with limited generalization ability, whereas texture bias means insufficient shape information in decision-making.

The concepts of shape and texture, also known as content and style, are widely used in the style transfer field (Gatys et al., 2016). In particular, the normalization-based style transfer methods (Dumoulin et al., 2016; Ulyanov et al., 2017; Huang & Belongie, 2017) point out that the channel-wise mean and variance statistics carry style information. This paper aims to manipulate style to address the two robustness problems: *i) texture sensitivity, ii) texture bias*.

First, to decrease texture sensitivity, we propose to recalibrate style by *SelfNorm*. The key idea is to use attention to adjust channel-wise mean and variance. *The intuition behind SelfNorm is that attention can help emphasize essential styles and suppress trivial ones.* Consequently, the model can focus on informative textures and become robust to texture noises. Second, to reduce texture bias, we propose *CrossNorm* to perform efficient style augmentation. The key idea is to swap the channel-wise mean and variance between two instances, or even feature maps within one instance. *The insight beneath CrossNorm is that each instance, or feature map, has its unique style.* The style augmentation can increase the texture uncertainty, forcing CNNs to learn more shape information.

It is interesting to analyze the distinction and connection between SelfNorm and CrossNorm. At first glance, they take opposite actions (style recalibration *v.s.* style augmentation) to tackle different problems (texture sensitivity *v.s.* texture bias) at separate stages (testing *v.s.* training) with distinct modes (single style *v.s.* style pair). Even so, they use the same tool: style and pursue the same goal: model robustness. Besides, CrossNorm can increase the capacity of SelfNorm by style augmentation. SelfNorm, with the help from CrossNorm, can generalize better to new styles in testing. In summary, the key contributions are:

- We propose SelfNorm to diminish the model sensitivity against texture perturbations.
- We also introduce CrossNorm to reduce the model bias to texture in contrast to shape.
- SelfNorm and CrossNorm form a unity of opposites in using style for model robustness.
- They are domain agnostic and can advance state-of-the-art robustness performance with different settings (fully or semi-supervised) and tasks (classification and segmentation).

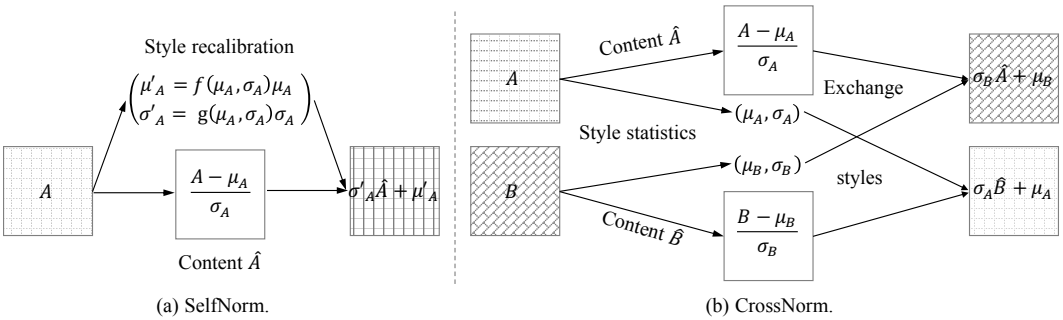
2 Related Work

Model robustness. Although the current deep models continue to break records on benchmark datasets, they still struggle to generalize under common corruptions (Hendrycks & Dietterich, 2019) and domain gaps (Richter et al., 2016). To improve the robustness against corruption, Stylized-ImageNet (Geirhos et al., 2019) conducts style augmentation to reduce the texture bias of CNNs. Compared with it, CrossNorm has two main advantages. First, CrossNorm is efficient as it transfer styles directly in the feature space of the target CNNs. However, Stylized-ImageNet relies on external style datasets and pre-trained style transfer models. Besides, CrossNorm can advance the performance on both clean and corrupted data, while Stylized-ImageNet hurts clean generalization. Because the styles within one dataset are much more consistent than the external ones, which can result in massive distribution shifts. Recently, AugMix (Hendrycks et al., 2020b) trains robust models by mixing multiple augmented images, based on random image primitives or image-to-image networks (Hendrycks et al., 2020a). Adversarial noises training (ANT) (Rusak et al., 2020) can also improve the robustness against corruption. CrossNorm is domain agnostic and orthogonal to AugMix and ANT, making it possible for their joint application. Moreover, different from the data augmentation methods, SelfNorm recalibrates style to boost model robustness.

Besides common corruptions, generalization with domain gaps also suffers from problems. IBN (Pan et al., 2018) mixes instance and batch normalization to shrink the domain distances. SelfNorm can bridge the domain gaps by style recliabration. Domain randomization (Yue et al., 2019) uses style augmentation for domain generalization on segmentation datasets. It suffers from the same issues of Stylized-ImageNet as it also uses pre-trained style transfer models and additional style datasets. By contrast, CrossNorm is more efficient and balances better between the source and target domains' performance.

Feature standardization and attention. Several standardization techniques, e.g., batch normalization (Ioffe & Szegedy, 2015), instance normalization (Ulyanov et al., 2016), and layer normalization (Ba et al., 2016), play important roles in training modern deep neural networks. Recently, some works integrate attention (Hu et al., 2018) into feature normalization. Mode normalization (Deecke et al., 2018) and attentive normalization (Li et al., 2019) use attention to weigh a mixture of batch normalizations. Examplar normalization (Zhang et al., 2020) learns to combine multi-type normalizations by attention. By contrast, SelfNorm uses attention with only instance normalization. Besides, SelfNorm is different from SE (Hu et al., 2018), though they use similar attention. First, SelfNorm learns to recalibrate style (channel-wise mean and variance) instead of directly channel features in SE. Second, SE models the interdependency between channels, while SelfNorm deals with each channel's style independently. Also, a SelfNorm unit, with $O(n)$, is more lightweight than a SE one, of $O(n^2)$, where n denotes the channel number.

Data augmentation. Data augmentation has drawn increasing attention as the deep models become powerful. Current data augmentation techniques are either label-preserving or label-perturbing. For image data, most state-of-the-art methods (Cubuk et al., 2019a; Lim et al., 2019; Ho et al., 2019) learn to compose image primitives, e.g., rotation and color, to conduct data augmentation. The image primitives are known not to change the image labels. The label-perturbing approaches mix both the data and their labels. Mixup (Zhang et al., 2017) linearly interpolates the image pixels and labels. CutMix (Yun et al., 2019) moves a region from an image to another and also interpolates labels. Two problems exist for the above data augmentation techniques. First, their goal is to improve generalization to clean data instead of data with corruption or domain gaps. Besides, they mainly work for classification and may have trouble in more broad applications, e.g., segmentation. In contrast, CrossNorm can boost clean generalization, corruption robustness, and domain generalization, and it is also readily applicable to diverse tasks and settings.



(a) SelfNorm. (b) CrossNorm.

Figure 1: SelfNorm (**left**) and CrossNorm (**right**). SelfNorm uses attention to emphasize essential styles and suppress trivial ones, lessening texture sensitivity. CrossNorm performs style augmentation by exchanging styles between feature maps. It can help the model focus more on shape information, reducing texture bias. SelfNorm and CrossNorm aim to decrease and increase the content and style mixture diversity in testing and training, respectively. Although they oppose each other in several aspects, they both manipulate style to improve model robustness and can work together.

3 SelfNorm and CrossNorm

Background. This paper uses the mean and variance of a feature map to represent its style, following the style transfer practices (Dumoulin et al., 2016; Ulyanov et al., 2017; Huang & Belongie, 2017). Technically, SelfNorm and CrossNorm share the same origin: instance normalization (Ulyanov et al., 2016). In 2D CNNs, each instance has C feature maps of size $H \times W$. Given $\boldsymbol{A} \in \mathbb{R}^{H \times W}$, instance normalization first normalizes the feature map and then conducts affine transformation:

$$\gamma \frac{\boldsymbol{A} - \mu_{\boldsymbol{A}}}{\sigma_{\boldsymbol{A}}} + \beta, \tag{1}$$

where $\mu_{\boldsymbol{A}}$ and $\sigma_{\boldsymbol{A}}$ are the mean and standard deviation; γ and β denotes learnable affine parameters.

SelfNorm. SelfNorm replaces β and γ with recalibrated mean $f(\mu_{\boldsymbol{A}}, \sigma_{\boldsymbol{A}})\mu_{\boldsymbol{A}}$ and standard deviation $g(\mu_{\boldsymbol{A}}, \sigma_{\boldsymbol{A}})\sigma_{\boldsymbol{A}}$, where f and g are the attention functions. Thus, the adjusted channel becomes:

$$g(\mu_{\boldsymbol{A}}, \sigma_{\boldsymbol{A}})\boldsymbol{A} + (f(\mu_{\boldsymbol{A}}, \sigma_{\boldsymbol{A}}) - g(\mu_{\boldsymbol{A}}, \sigma_{\boldsymbol{A}}))\mu_{\boldsymbol{A}}. \tag{2}$$

As f and g learn to scale $\mu_{\boldsymbol{A}}$ and $\sigma_{\boldsymbol{A}}$ based on themselves, $\boldsymbol{A}$ *normalizes itself by self-gating, hence SelfNorm*. SelfNorm is inspired by the fact that attention can help model emphasize informative features and suppress less useful ones. In terms of style recalibration, SelfNorm expects to highlight the discriminative styles and understate trivial ones. In practice, we use a fully connected (FC) network to wrap attention functions f and g. The architecture is efficient as its input and output are both two scalars. Since each channel has its independent style, SelfNorm recalibrates each channel separately using C such lightweight FC networks, hence the complexity of $O(C)$.

CrossNorm. CrossNorm exchanges $\mu_{\boldsymbol{A}}$ and $\sigma_{\boldsymbol{A}}$ of channel A with $\mu_{\boldsymbol{B}}$ and $\sigma_{\boldsymbol{B}}$ of channel $\boldsymbol{B}$, i.e., changing β and γ to each other's μ and σ:

$$\sigma_{\boldsymbol{B}} \frac{\boldsymbol{A} - \mu_{\boldsymbol{A}}}{\sigma_{\boldsymbol{A}}} + \mu_{\boldsymbol{B}} \qquad \sigma_{\boldsymbol{A}} \frac{\boldsymbol{B} - \mu_{\boldsymbol{B}}}{\sigma_{\boldsymbol{B}}} + \mu_{\boldsymbol{A}}, \tag{3}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ serve as content and style features simultaneously. $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *seem to normalize each other, hence CrossNorm.* Figure 1 gives its illustration. CrossNorm is motivated by the key observation that a target dataset, such as a classification dataset, has rich, though subtle, styles. *Specifically, each instance, or even every channel, has its unique style*. Exchanging the styles can perform efficient style augmentation for content features. Note that CrossNorm aims to reduce the texture bias in CNNs, rather than remove the texture dependence. In mini-batch training, we turn on CrossNorm with some probability.

Unity of Opposites. SelfNorm and CrossNorm both start from instance normalization but head in opposite directions. SelfNorm recalibrates styles to focus on only necessary styles, reducing content and style mixtures' diversity. In contrast, CrossNorm transfers styles between channels, enriching

the combinations of content and style. They perform opposite operations mainly because they target at different stages. SelfNorm dedicates to style recalibration during testing, while CrossNorm, a data augmentation technique, functions only in training. Besides, they aim to address different problems: texture sensitivity and bias. However, they both can improve the texture-related model robustness, e.g., testing with corruptions or domain gaps. Another interesting comparison is that SelfNorm uses a single style (a pair of channel-wise mean and variance), whereas CrossNorm requires pair styles.

Further, CrossNorm can boost the performance of SelfNorm. SelfNorm, as a learnable module, may suffer from the overfitting issue. Fortunately, the style augmentation of CrossNorm can prevent SelfNorm from overfitting to specific styles. Overall, although SelfNorm and CrossNorm oppose each other in several aspects, they form a unity of using style to advance model robustness.

3.1 CrossNorm Variants

The core idea of CrossNorm is to swap mean and variance between channel features. For 2D CNNs, given one instance $\mathbf{X}, \in R^{C\times H\times W}$, CrossNorm can exchange styles between its C channels:

$$\{(\boldsymbol{A}, \boldsymbol{B}) \in (\mathbf{X}_{i,:,:}, \mathbf{X}_{j,:,:}) \mid i \neq j, 0 < i, j < C\}, \tag{4}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ refer to the channel pair in Equation 3. If two instances $\mathbf{X}, \mathbf{Y} \in R^{C\times H\times W}$ given, CrossNorm can swap styles between their corresponding channels, i.e., $\boldsymbol{A}$ and $\boldsymbol{B}$ become:

$$\{(\boldsymbol{A}, \boldsymbol{B}) \in (\mathbf{X}_{i,:,:}, \mathbf{Y}_{i,:,:}) \mid 0 < i < C\}. \tag{5}$$

Compared with one-instance CrossNorm, the two-instance one tends to consider instance-level style instead of channel-level.

Besides, distinct spatial regions probably have different mean and variance statistics, i.e., styles. To promote the style diversity, we propose to crop regions for CrossNorm:

$$\{(\boldsymbol{A}, \boldsymbol{B}) \in (\text{crop}(\boldsymbol{A}), \text{crop}(\boldsymbol{B})) \mid r_{\text{crop}} \geq t\} \tag{6}$$

where the crop function returns a square with area ratio r no less than a threshold $t (0 < t \leq 1)$. The whole channel is a special case in cropping. There are three cropping choices: content only, style only, and both. For content cropping, we crop $\boldsymbol{A}$ only when it acts as the content features. In other words, no cropping applies to $\boldsymbol{A}$ when it provides its style to $\boldsymbol{B}$. Cropping both means cropping $\boldsymbol{A}$ and $\boldsymbol{B}$ no matter they are content or style feature maps. The cropping strategy can produce diverse styles for both the two-instance and one-instance CrossNorms, hence improved performance.

3.2 Modular Design

SelfNorm and CrossNorm can naturally work in the feature space, making it flexible to plug them into many network locations. Two questions come: how many units are necessary and where to place them? To simplify the questions, we turn to the modular design by embedding them into a network cell. For example, in ResNet (He et al., 2016), we put them into a residual module. The search space significantly shrinks for the limited positions in a residual module. We will investigate the position choices in experiments. The modular design allows using multiple SelfNorms and CrossNorms in a network. Although repetitive style recalibrations (Hu et al., 2018) are helpful for model robustness, excessive style augmentations are harmful (Geirhos et al., 2019). To get a decent amount of style augmentation, we randomly turn on only some CrossNorms in a forward process. Random sampling encourages diverse augmentations even though the same data pass multiple times.

4 Experiment

In this section, we evaluate SelfNorm and CrossNorm through generalization to clean data, corrupted data, and data with domain gaps. The evaluation uses not only supervised and semi-supervised settings but also image classification and segmentation tasks.

Image classification datasets. We use benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100, and ImageNet(Deng et al., 2009). To evaluate the model robustness against corruption, we use the datasets: CIFAR-10-C, CIFAR-100-C, and ImageNet-C (Hendrycks & Dietterich, 2019).

Table 1: mCE (%) comparison of CIFAR-10-C and CIFAR-100-C. SelfNorm&CrossNorm (SNCN) obtains lower errors than most previous methods with different backbones. Albeit some higher errors than AugMix, it is totally domain agnostic without relying on the image primitives, e.g., rotation, in AugMix. As SNCN and AugMix are orthogonal, their joint usage brings new state-of-the-art results.

CIFAR-10-C	Basic	Cutout	Mixup	CutMix	AutoAug	AdvTr.	AugMix	SNCN	SNCN+AugMix
AllConvNet	30.8	32.9	24.6	31.3	29.2	28.1	15.0	17.2	**11.8**
DenseNet	30.7	32.1	24.6	33.5	26.6	27.6	12.7	18.5	**10.4**
WideResNet	26.9	26.8	22.3	27.1	23.9	26.2	11.2	16.9	**9.9**
ResNeXt	27.5	28.9	22.6	29.5	24.2	27.0	10.9	15.7	**9.1**
Mean	29.0	30.2	23.5	30.3	26.0	27.2	12.5	17.0	**10.3**
CIFAR-100-C	Basic	Cutout	Mixup	CutMix	AutoAug	AdvTr.	AugMix	SNCN	SNCN+AugMix
AllConvNet	56.4	56.8	53.4	56.0	55.1	56.0	42.7	42.8	**36.8**
DenseNet	59.3	59.6	55.4	59.2	53.9	55.2	39.6	48.5	**37.0**
WideResNet	53.3	53.5	50.4	52.9	49.6	55.1	35.9	43.7	**33.4**
ResNeXt	53.4	54.6	51.4	54.1	51.3	54.4	34.9	40.8	**30.8**
Mean	55.6	56.1	52.6	55.5	52.5	55.2	38.3	43.5	**34.7**

Table 2: Clean error and mCE (%) of ResNet50 trained 90 epochs on ImageNet. Compared with Stylized-ImageNet, CrossNorm (CN) holds a better balance between clean and corruption errors. SelfNorm also works well in decreasing the corruption error while maintaining low clean error. Jointly applying SNCN with AugMix can produce the lowest clean and corruption errors.

		Noise			Blur				Weather				Digital				
Aug.	Clean	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	mCE
Standard	23.9	79	80	82	82	90	84	80	86	81	75	65	79	91	77	80	80.6
Patch Uniform	24.5	67	68	70	74	83	81	77	80	74	75	62	77	84	71	71	74.3
Random AA*	23.6	70	71	72	80	86	82	81	81	77	72	61	75	88	73	72	76.1
MaxBlur pool	23.0	73	74	76	74	86	78	77	77	72	63	56	68	86	71	71	73.4
SIN	27.2	69	70	70	77	84	76	82	74	75	69	65	69	80	64	77	73.3
AugMix*	23.4	66	66	66	69	80	65	68	72	72	66	60	63	78	66	71	68.4
CN	23.4	73	75	75	78	89	79	82	79	75	66	61	69	97	69	74	75.3
SN	23.7	69	71	69	77	87	77	80	75	77	70	61	73	83	61	71	73.4
SNCN+AugMix	**22.3**	61	62	60	70	77	62	68	62	65	63	55	43	73	55	66	**62.8**

They are the original test data poisoned by 15 everyday image corruptions from 4 general types: noise, blur, weather, and digital. Each noise has 5 intensity levels when injected into images.

Image segmentation datasets. We further validate our method using a domain generalization setting, where the models are trained without any target domain data and are tested on the unseen domain. We use the synthetic dataset Grand Theft Auto V (GTA5) (Richter et al., 2016) as the source domain and generalize to the real-world dataset Cityscapes (Cordts et al., 2016). GTA5 has the training, validation, and test divisions of 12,403, 6,382, and 6,181, more than those of 2,975, 500, and 1,525 from Cityscapes. Despite the differences, their pixel categories are compatible with each other, making it possible to evaluate models' generalization capability from one to another.

Metric. For classification, we use test errors to measure the robustness. Given corruption type c and severity s, let E_s^c denote the test error. For CIFAR datasets, we use the average over 15 corruptions and 5 severities: $1/75 \sum_{c=1}^{15} \sum_{s=1}^{5} E_{c,s}$. In contrast, for ImageNet, we normalize the corruption errors by those of AlexNet (Krizhevsky et al., 2012): $1/15 \sum_{c=1}^{15} (\sum_{s=1}^{5} E_s^c / \sum_{s=1}^{5} E_{c,s}^{AlexNet})$. The above two metrics follow the convention (Hendrycks et al., 2020b) and are denoted as mean corruption errors (mCE) whether they are normalized or not. Different from classification, segmentation use the mean Intersection over Union (mIoU) over all categories as metric.

Hyper-parameters. In the experiments, a SelfNorm unit uses one fully connected layer, followed by batch normalization and a sigmoid layer. When applied, it is plugged into every cell in a network, e.g., each residual module in a ResNet. Besides, we also place a CrossNorm unit in each cell. During training, we turn on only some CrossNorms with probability to avoid excessive data augmentation. Usually, we conduct a grid search on four combinations of active numbers (1,2) and probability (0.25, 0.5). For CrossNorm with cropping, we sample the bounding box ratio uniformly and set the threshold $t = 0.1$.

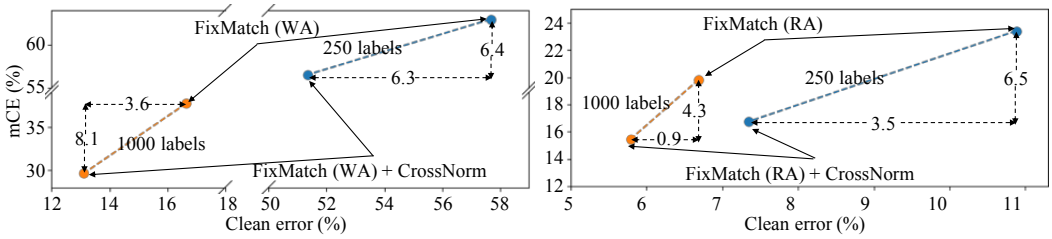


Figure 2: CrossNorm for semi-supervised CIFAR-10 classification. We apply CrossNorm on top of FixMatch with weak augmentation (WA) (**Left**), or strong RandAugment (RA) (**Right**). For either case, CrossNorm (CN) can substantially reduce both clean and corruption errors. Compared with RA, CrossNorm performs domain agnostic data augmentation, easily applicable to new domains.

4.1 Robustness against Unseen Corruptions

Supervised training on CIFAR. Following AugMix (Hendrycks et al., 2020b), we evaluate SelfNorm and CrossNorm with four different backbones: an All Convolutional Network (Springenberg et al., 2014), a DenseNet-BC (k = 12, d = 100) (Huang et al., 2017), a 40-2 Wide ResNet (Zagoruyko & Komodakis, 2016), and a ResNeXt-29 (32×4) (Xie et al., 2017). We also use the same hyper-parameters in the AugMix Github repository[1].

According to Table 1, SelfNorm and CrossNorm can decrease the mean error by ∼12% on both CIFAR-10-C and CIFAR-100-C, outperforming most previous approaches. The unseen corruptions bring uncertainty to data textures. SelfNorm and CrossNorm can reduce texture sensitivity and bias, significantly improving the robustness against unseen corruptions. Besides, the domain-agnostic SelfNorm and CrossNorm are orthogonal to AugMix, which relies on domain-specific operations. Their joint application can continue to lower the mCEs by 2.2% and 3.6% on top of AugMix.

Supervised training on ImageNet. Following the AugMix Github repository, we train a ResNet-50 for 90 epochs with weight decay 1e-4. The learning rate starts from 0.1, divided by 10 at epochs 30 and 60. Note that AugMix reports the results of 180 epochs in their paper. For a fair comparison, we also train 90 epochs in our experiments. Stylized-ImageNet finetunes a pre-trained ResNet-50 for 45 epochs with double data (stylized and original ImageNets) in each epoch. To compare CrossNorm with Stylized-ImageNet, we perform the finetuning for 90 epochs using only the original ImageNet. Besides, we also add Instance-batch normalization (IBN) (Pan et al., 2018) in the final combination with AugMix. It was initially designed for domain generalization, but can also boost model robustness against corruption.

Table 2 gives the results on ImageNet. We can find that SelfNorm and CrossNorm can effectively decrease the corruption error by 7.2% and 5.3%. Their clean errors also have small drops (0.2% and 0.5%) comparing with the baseline. Although Stylized-ImageNet has 2% lower corruption error than CrossNorm, its clean error is 3.8% higher. Because the external styles in Stylized-ImageNet cause large distribution shifts, impairing its clean generalization. In contrast, The more consistent yet diverse internal styles help CrossNorm decreases both corruption and clean errors. Moreover, applying SelfNorm and CrossNorm on top of AugMix can further significantly lower its clean and corruption errors by 1.1% and 5.6%, achieving state-of-the-art performance. IBN also makes some contributions here since it is complementary to other components.

Semi-supervised training on CIFAR. Apart from supervised training, we also evaluate CrossNorm in semi-supervised learning. Following state-of-the-art FixMatch (Sohn et al., 2020) setting, we train a 28-2 Wide ResNet for 1024 epochs on CIFAR-10. The SGD optimizer applies with Nesterov momentum 0.9, learning rate 0.03, and weight decay 5e-4. The probability threshold to generate pseudo-labels is 0.95, and the weight for unlabeled data loss is 1. We sample 250 and 4,000 labeled data with random seed 1, leaving the rest as unlabeled data. In each experiment, we apply CrossNorm to all data or only the unlabeled, and choose the better one. Our experiments use the Pytorch FixMatch implementation, [2], which has higher errors than the FixMatch reported. We choose it because it has the most stars among all the Pytorch implementations on Github.

[1] https://github.com/google-research/augmix

[2] https://github.com/kekmodel/FixMatch-pytorch

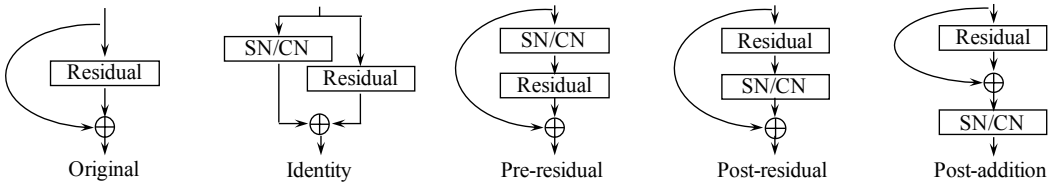


Figure 3: Illustration of SelfNorm (SN) and CrossNorm (CN) positions in a residual module.

Table 3: Segmentation results (mIoU) on GTAV-Cityscapes domain generalization using a FCN with ResNet50. SelfNorm (SN) and CrossNorm (CN) are comparable with IBN and domain randomization (DR) on target domain. Combining SN and CN can achieve state-of-the-art performance.

Methods	Baseline	IBN	DR	SN	CN	SNCN
Source	63.7	64.2	49.0	**64.6**	61.2	63.5
Target	21.4	29.6	32.7	29.9	32.0	**36.5**

Figure 2 shows the semi-supervised training results. We run FixMatch with the strong RandAugment (Cubuk et al., 2019b) or only weak random flip and crop augmentations. With either FixMatch version, CrossNorm can always decrease both the clean and corruption errors, demonstrating its effectiveness in semi-supervised training. Especially, with the help of CrossNorm, training with 250 labels even has 3% lower corruption error than with 1000 labels, according to the right sub-figure. Besides, two points are noteworthy here. First, we try FixMatch with only weak augmentations to simulate more general situations. For new domains other than natural images, humans may have limited domain knowledge to design advanced augmentation operations. Fortunately, CrossNorm is domain-agnostic and easily applicable to such situations. Moreover, previous semi-supervised methods mainly focus on clean generalization. Here we introduce corruption robustness as another metric for more comprehensive evaluation.

4.2 Generalization with Domain Shift

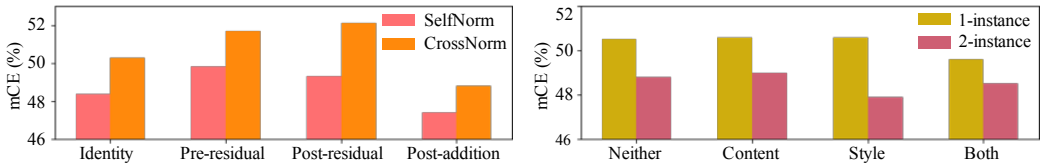
Besides the (semi-)supervised settings, we also test SelfNorm and CrossNorm on domain generalization from synthetic to realistic data, which is meaningful in lowering the cost of data annotation.

Training setup. We perform domain generalization from GTA5 to Cityscapes, following the setting of IBN (Pan et al., 2018). It uses 1/4 training data in GTA5 to match the data scale of Cityscapes. We train the FCN (Long et al., 2015) with ResNet50 backbone in source domain GTA5 for 80 epochs with batch size 16. The network is initialized with ImageNet pretrained weights. Then we test the model on both source domain and target domain. The training uses random scaling, flip, rotation, and cropping (713×713) for data augmentation. We use the 2-instance CrossNorm with style cropping in this setting. Besides, we re-implement the domain randomization (Yue et al., 2019) and make the training iterations to be the same as ours. It transfers the synthetic images to 15 auxiliary domains with ImageNet image styles.

Results. Based on Table 3, SelfNorm and CrossNorm both can substantially increase the segmentation accuracy on the target domain by 8.5% and 10.6%. SelfNorm learns to highlight the discriminative styles that are likely to share across domains. CrossNorm performs style augmentation to make CNNs focus more on domain-invariant features instead of texture. SelfNorm and CrossNorm get comparable generalization performance as state-of-the-art IBN (Pan et al., 2018) and domain randomization (Yue et al., 2019). However, CrossNorm significantly outperforms the domain randomization method by 12.2% on the source accuracy. Because the domain randomization transfers external styles to the source training data, causing dramatic distribution shifts. Moreover, combining SelfNorm and CrossNorm gives state-of-the-art generalization performance while still maintaining high source accuracy.

4.3 Ablation Study

All the experiments here use a 40-2 Wide ResNet, measured by the mCE on CIFAR-100-C.



(a) mCEs of SelfNorm and CrossNorm at 4 residual positions. (b) mCEs of 1 and 2-instance CrossNorm with 4 cropping types.

Figure 4: Study of Modular positions (**left**) and CrossNorm variants (**right**). SelfNorm and CrossNorm both get the lowest errors at the post-addition position. Second, the 2-instance mode consistently outperforms the 1-instance one, and proper cropping can decrease the error.

Table 4: Block choices of CrossNorm. We compare the mCEs when applying CrossNorm to image space or different blocks in a WideResNet-40-2. The choice of all blocks gives the lowest error.

CrossNorm stages	N/A	Image	Block 1	Block 2	Block 3	All
mCE(%)	53.3	54.3	52.2	51.2	51.5	**48.8**

Table 5: Ablation study of SE, SelfNorm(SN), CrossNorm(CN), cropping, and consistency regularization(CR). SN obtains much lower corruption error than SE, justifying its robustness superiority. Besides, SN and CN can work together in reducing mCE, and cropping and CR can help further.

	Basic	SE	SE(post)	SN	CN	SNCN	SNCN+Crop	SNCN+Crop+CR
mCE(%)	53.3	52.3	51.0	47.4	48.8	46.6	44.5	**43.7**

CrossNorm variants. CrossNorm can be in 1-instance or 2-instance mode with four cropping options. Figure 4 compares their mCEs. The 2-instance mode constantly gets lower errors than the 1-instance one. Besides, cropping can help decrease the error since it can encourage the style augmentation diversity. Note that style cropping may not always be superior. We find that the best cropping choice may change over backbones, datasets, and joint application with SelfNorm.

Modular positions. Here we investigate four positions in a residual module, illustrated in Figure 3. We can find that, in Figure 4, both SelfNorm and CrossNorm work the best at the post-addition position. Thus, we plug them there for the CIFAR experiments. For positions in other module types, please refer to the supplementary.

CrossNorm Blocks. After narrowing down the CrossNorm position in a cell, we still need to study which blocks in a network should build on the CrossNorm cells. According to Table 4, CrossNorm in image space performs poorly. However, it performs the best when plugged into all blocks.

Ablating components. SelfNorm learns to recalibrate test styles to decrease texture sensitivity, while CrossNorm augments training styles to reduce texture bias. Although focusing on distinct aspects, they can lower the corruption error both separately and jointly, according to Table 5. On top of them, cropping and the consistency regularization in AugMix can further advance model robustness. Besides, SelfNorm outperforms SE (Hu et al., 2018) for model robustness because it takes advantage of the style information.

5 Conclusion

In this paper, we have presented SelfNorm and CrossNorm, two simple yet effective methods to improve model robustness. They form a unity of opposes as they confront and conform to each other in terms of approach (style usage) and goal (model robustness). Beyond their extensive applications, they may also shed light on developing domain agnostic methods for broad generalization under unseen corruption, domain gaps, and even more. Given the simplicity of SelfNorm and CrossNorm, we believe there is substantial room for improvement. One possible direction is to explore better style representations. The current mean and variance of feature maps may not be optimal.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pp. 113–123, 2019a.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv preprint arXiv:1909.13719*, 2019b.

Lucas Deecke, Iain Murray, and Hakan Bilen. Mode normalization. *arXiv preprint arXiv:1810.05466*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proceedings of the International Conference on Learning Representations*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020a.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *Proceedings of the International Conference on Learning Representations*, 2020b.

Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules. *arXiv preprint arXiv:1905.05393*, 2019.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pp. 1097–1105, 2012.

Xilai Li, Wei Sun, and Tianfu Wu. Attentive normalization. *arXiv preprint arXiv:1908.01259*, 2019.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *NeurIPS*, pp. 6662–6672, 2019.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 464–479, 2018.

Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pp. 102–118. Springer, 2016.

Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. *arXiv preprint arXiv:2001.06057*, 2020.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6924–6932, 2017.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2100–2110, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Ruimao Zhang, Zhanglin Peng, Lingyun Wu, Zhen Li, and Ping Luo. Exemplar normalization for learning deep representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12726–12735, 2020.

6 SUPPLEMENTARY

6.1 ILLUSTRATION

Figure 5 gives an illustration of 15 corruptions used in robustness evaluation on CIFAR and ImageNet. Besides, Figure 6 shows some synthetic images from GTA5 and realistic ones from Cityscapes. The visualization of CrossNorm applied to image space is provided in Figure 7.

6.2 ABLATION STUDY ON IMAGENET

Table 6 reports the results of applying SelfNorm and CrossNorm with IBN. Besides, in Table 7, we uses ImageNet to investigate the SelfNorm positions in a residual module.

6.3 MORE ABLATION STUDY ON CIFAR

In addition to the residual module, we also investigate the positions of SelfNorm and CrossNorm in the cells of AllConvNet, DenseNet, and ResNeXt. Figure 8 illustrates the positions in the AllConvNet and DenseNet cells. The results of SelfNorm and CrossNorm on two datasets CIFAR-10-C and CIFAR-100-C are given in Tables 8, 9, 10, and 11. We also report the results of SelfNorm and CrossNorm with different cropping choices in Tables 13 and 12. The full component ablation studies with different backbones are given in Tables 15 and 14.

6.4 ABLATION STUDY ON DOMAIN GENERLIZATION

Table 16 reports more detailed comparison on domain generalization from GTA5 to Cityscapes.

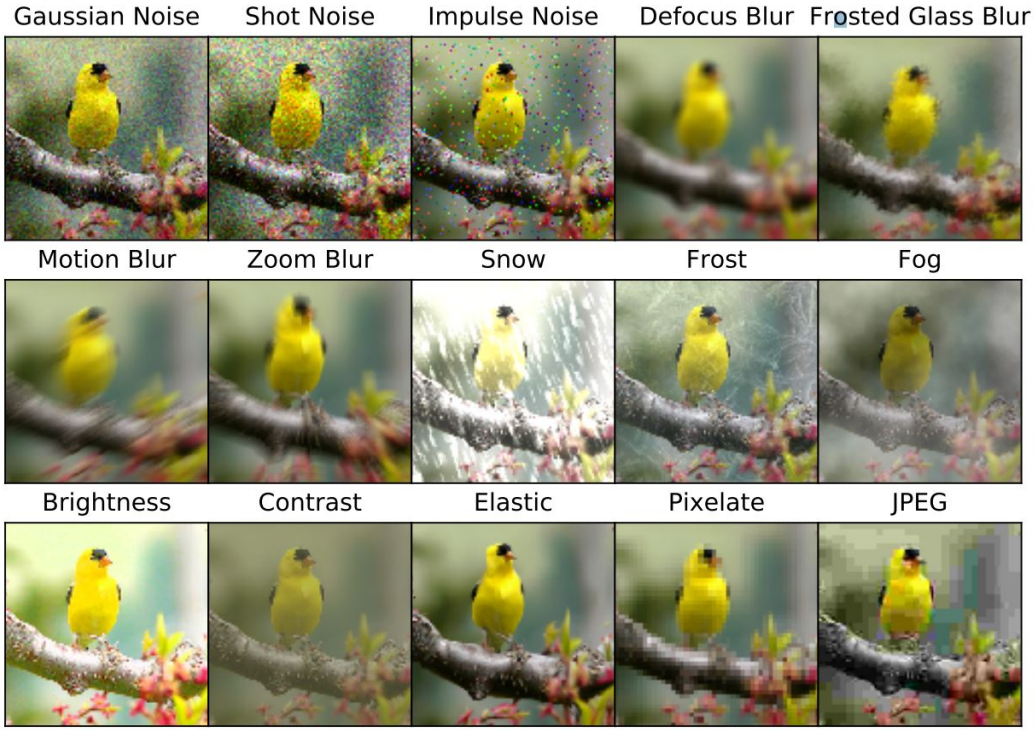


Figure 5: A demostration of corrupted images in ImageNet-C dataset (Hendrycks & Dietterich, 2019). 15 types of algorithmically generated corruptions from noise, blur, weather, and digital categories are applied to images to create corrupted dataset.

Figure 6: Some examples of segmentation dataset. The first row are images from Cityscapes dataset, while the second row are images from GTA5 dataset.

Figure 7: A visualization of CrossNorm with crop style used on image level. The two images on the first column are the original images in the GTA5 dataset and are in the same training batch. We applied CrossNorm to these two images several times and got the following three pairs of images. By calculating style from random crops, CrossNorm can perform a variety of style augmentation.It should be noted that we use CrossNorm in both image and feature level, we only visualize in image level here for simplicity.

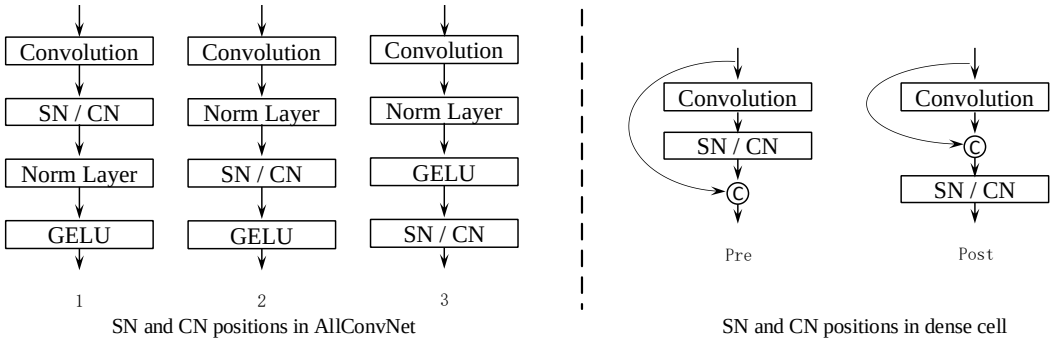


Figure 8: Illustration of SelfNorm (SN) and CrossNorm (CN) positions in AllConvNet block, and dense cells in DenseNet. For blocks in AllConvNet, we name the position after convolution layer as *1*, after normalization layer as *2*, and after GELU layer as *3*. For dense cells in DenseNet, we label the position before feature concatenation as *Pre*, and after concatenation as *Post*.

Table 6: Ablation study of IBN, SelfNorm(SN), CrossNorm(CN), consistency regularization(CR), and AugMix(AM) on ImageNet-C with ResNet50. IBN, originally designed for domain generalization, can also decrease mCE. Both SN and CN can further lower the error based on IBN. Combining them with AM gives the best robustness performance.

	ResNet50	ResNet50+IBN(a)				ResNet50+IBN(b)			
	Basic	Basic	+CN	+CN+CR	+CN+AM	Basic	+SN	+SN+AM	+SNCN+AM
Clean err(%)	23.9	23.2	23.1	22.6	-	24.0	23.5	22.3	22.3
mCE(%)	80.6	75.1	73.2	73.6	-	74.1	72.6	64.1	62.8

Table 7: Position investigation of SelfNorm in a residual module of ResNet50 trained on ImageNet.

Modular Position	Identity	Pre-Residual	Post-Residual	Post-Addition
Clean error (%)	24.0	**23.0**	23.2	23.7
mCE(%)	75.5	75.8	74.8	**73.4**

Table 8: Ablation study of the impact of different SelfNorm position for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-10-C measured by mCE

Position	1	2	3	-
AllConvNet	**24.01**	26.38	25.56	-
Position	Pre	Post	-	-
DenseNet	23.40	**21.96**	-	-
Position	Residual	Post	Pre	Identity
WideResNet	22.69	21.29	**20.78**	22.29
Position	Residual	Post	Pre	Identity
ResNeXt	21.94	24.76	**21.49**	21.99

Table 9: Ablation study of the impact of different CrossNorm position for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-10-C measured by mCE

Position	1	2	3	-
AllConvNet	**25.99**	26.27	26.84	-
Position	Pre	Post	-	-
DenseNet	**24.72**	29.17	-	-
Position	Residual	Post	Pre	Identity
WideResNet	25.20	**21.62**	24.91	23.27
Position	Residual	Post	Pre	Identity
ResNeXt	26.71	**22.37**	23.75	26.92

Table 10: Ablation study of the impact of different SelfNorm position for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-100-C measured by mCE

Position	1	2	3	-
AllConvNet	**50.30**	51.63	51.03	-
Position	Pre	Post	-	-
DenseNet	**53.86**	54.67	-	-
Position	Residual	Post	Pre	Identity
WideResNet	49.28	**47.44**	49.82	48.43
Position	Residual	Post	Pre	Identity
ResNeXt	**47.56**	49.00	50.86	50.44

Table 11: Ablation study of the impact of different CrossNorm position for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-100-C measured by mCE

Position	1	2	3	-
AllConvNet	**52.20**	52.49	52.69	-
Position	Pre	Post	-	-
DenseNet	**55.44**	57.57	-	-
Position	Residual	Post	Pre	Identity
WideResNet	52.06	**48.76**	51.74	50.26
Position	Residual	Post	Pre	Identity
ResNeXt	51.47	**46.95**	47.87	50.22

Table 12: Ablation study of the impact of SelfNorm + CrossNorm with different cropping, including neither, content, style and both, for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-100-C measured by mCE

Backbone	Neither	Content	Style	Both
AllConvNet, 1	44.17	46.92	**43.86**	46.08
DenseNet, Conv1 Pre	51.38	49.40	49.13	**48.97**
WideResNet, Post	46.58	45.08	45.82	**44.46**
ResNeXt, Post	**41.03**	44.85	42.96	46.52

Table 13: Ablation study of the impact of SelfNorm + CrossNorm with different cropping, including neither, content, style and both, for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-10-C measured by mCE

Backbone	Neither	Content	Style	Both
AllConvNet, 1	18.98	20.29	**18.82**	20.28
DenseNet, Conv1 Pre	18.75	**18.23**	18.70	18.83
WideResNet, Post	17.93	17.99	**16.77**	17.47
ResNeXt, Post	**17.73**	18.52	18.37	18.59

Table 14: Ablation study of each component of our methods, including SelfNorm, CrossNorm with cropping, consistency and combination with AugMix, for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-100-C measured by mCE

Backbone	Basic	SN	CN	SN+CN +Crop	SN+CN+Crop +Consistency	AugMix	SNCN+Crop +AugMix
AllConvNet, 1, style	56.40	50.30	52.20	43.86	42.83	42.7	**36.80**
DenseNet, Conv1 Pre, both	59.30	53.86	55.44	48.97	48.48	39.60	**36.95**
WideResNet, Post, both	53.30	47.44	48.76	44.46	43.70	35.90	**33.38**
ResNeXt, Post, neither	53.40	47.56	46.95	41.03	40.84	34.9	**30.76**

Table 15: Ablation study of each component of our methods, including SelfNorm, CrossNorm with cropping, consistency and combination with AugMix, for AllConvNet, DenseNet, WIdeResNet and ResNeXt on CIFAR-10-C measured by mCE

Backbone	Basic	SN	CN	SN+CN +Crop	SN+CN+Crop +Consistency	AugMix	SNCN+Crop +AugMix
AllConvNet, 1, style	30.80	24.01	25.99	18.82	17.24	15.00	**11.79**
DenseNet, Conv1 Pre, both	30.70	21.96	24.72	18.83	18.53	12.7	**10.40**
WideResNet, Post, both	26.90	20.78	21.62	17.47	16.93	11.2	**9.94**
ResNeXt, Post, neither	27.50	21.49	22.37	17.73	15.69	10.90	**9.09**

Table 16: Ablation on GTAV-Cityscapes domain generalization. Mean IoU for both within-domain and cross-domain evaluation are reported. All methods use FCN with ResNet50 as backbone network. We use style-only crop for segmentation.

Methods	FCN Baseline	SN	CN	CN+Crop	SNCN	SNCN+Crop
Source	63.7	64.6	61.8	61.2	**65.0**	63.5
Target	21.4	29.9	31.5	32.0	34.1	**36.5**