Instance Enhancement Batch Normalization: An Adaptive Regulator of Batch Noise

Senwei Liang,[1][*] **Zhongzhan Huang,**[2][*] **Mingfu Liang,**[3] **Haizhao Yang**[1,4][†]

[1]Purdue University, [2]Tsinghua University, [3]Northwestern University, [4]National University of Singapore[‡]
{liang339, haizhao}@purdue.edu, hzz_dedekinds@foxmail.com,
mingfuliang2020@u.northwestern.edu

Abstract

Batch Normalization (BN) (Ioffe and Szegedy 2015) normalizes the features of an input image via statistics of a batch of images and hence BN will bring the noise to the gradient of training loss. Previous works indicate that the noise is important for the optimization and generalization of deep neural networks, but too much noise will harm the performance of networks. In our paper, we offer a new point of view that the self-attention mechanism can help to regulate the noise by enhancing instance-specific information to obtain a better regularization effect. Therefore, we propose an attention-based BN called Instance Enhancement Batch Normalization (IEBN) that recalibrates the information of each channel by a simple linear transformation. IEBN has a good capacity of regulating the batch noise and stabilizing network training to improve generalization even in the presence of two kinds of noise attacks during training. Finally, IEBN outperforms BN with only a light parameter increment in image classification tasks under different network structures and benchmark datasets.

Introduction

Mini-batch Stochastic Gradient Descent (SGD) is an effective method in large-scale optimization by aggregating multiple samples at each iteration to reduce operation and memory costs. However, SGD is sensitive to the choice of hyperparameters and it may cause training instability (Luo, Xiong, and Liu 2019). Normalization is one possible choice to remedy SGD methods for better stability and generalization. Batch Normalization (BN) (Ioffe and Szegedy 2015) is a frequently-used method that normalizes the features of an image using the mean and variance of the features of a batch of images during training. Meanwhile, the tracked mean and variance that estimate statistics of the whole dataset are used for normalization during testing. It has been shown that BN is an effective module to regularize parameters (Luo et al. 2019), stabilize training, smooth gradients (Santurkar et al. 2018), and enable a larger learning rate (Bjorck et al. 2018;

Cai, Li, and Shen 2019) for faster convergence. Further, the cause of noise and the usage of both SGD and BN are inseparable. Two kinds of noise effected in SGD and BN are concerned,

Estimation Noise. In BN, the mean and variance of a batch are used to estimate those of the whole dataset; in SGD, the gradient of loss over a batch is applied to approximate that of the whole dataset. These estimation errors are called estimation noise.

Batch Noise. In the forward pass, the features of an instance are incorporated with batch information via the normalization with batch statistics. The gradient of the loss of an instance is disturbed by the batch information due to the forward pass. These disturbances to an instance caused by the batch information are referred to as batch noise.

The randomness of BN and SGD has been well-known to improve the performance of deep networks. BN with the estimation noise can work as an adaptive regularizer of parameters (Luo et al. 2019; Shekhovtsov and Flach 2019) and the moderate noise can help escape bad local minima and saddle points (Jin et al. 2017; Ge et al. 2015). There exists extensive study on optimizing effectiveness via tuning batch size. On the one hand, a small batch size will lead to a high variance of statistics and weaken the training stability (Wu and He 2018; Huang et al. 2019a). On the other hand, a large batch size can reduce the estimation noise but it will incur a sharp landscape of loss (Keskar et al. 2016) making the optimization problem more challenging. Therefore, appropriate batch size is required to make a good balance. In practice, we choose a relatively large batch size and the estimation noise becomes moderate when the batch size is not small (Luo et al. 2019). At that time, the batch noise turns into the primary noise and our paper mainly discusses the batch noise.

The appropriate batch noise is necessary for a model. In fact, too much noise may harm the performance of the network. In our paper, we design two kinds of additional noise injected into the network to heighten the batch noise. Table 2, Table 3 and Fig. 4 show the performance of the BN networks under the noise attack. The instability of training and degradation of performance indicate that the BN networks can not resist augmented batch noise.

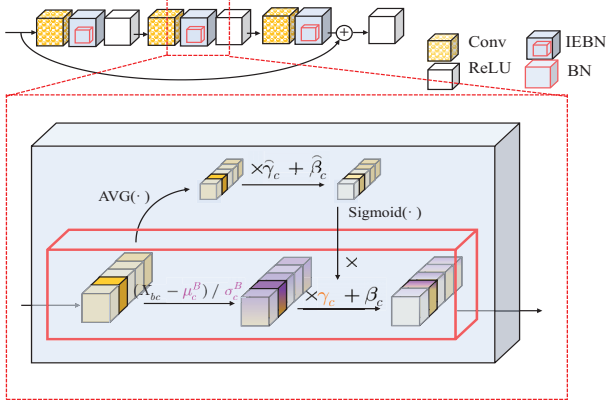It remains a problem on how to infuse a model with the

---

Figure 1: The illustration of IEBN. The top shows a block of ResNet. The bottom is the showcase of IEBN, where the box with red border is the basic flow of BN. $AVG(\cdot)$ means the average pooling over a channel.

Algorithm 1 The forward pass of IEBN during training

Input: X is a batch input of size $B \times C \times H \times W$;
Parameters: γ_c, β_c, $\hat{\gamma}_c$ and $\hat{\beta}_c$, $c = 1, \cdots, C$.
Output: $\{Y = \textbf{IEBN}_{\gamma_c, \beta_c, \hat{\gamma}_c, \hat{\beta}_c}(X)\}$.

1:   $\hat{\gamma}_c \leftarrow 0; \hat{\beta}_c \leftarrow -1$
2:   **for** channel c from 1 to C **do**
3:      $\mu_c^B \leftarrow \frac{1}{B \cdot H \cdot W} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{bchw}$
4:      $\sigma_c^B \leftarrow \sqrt{\frac{1}{B \cdot H \cdot W} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{w=1}^{W} (X_{bchw} - \mu_c^B)^2 + \epsilon}$
5:      **for** instance b from 1 to B **do**
6:         $\delta_{bc} \leftarrow Sigmoid(\textbf{AVG}(X_{bc}) \times \hat{\gamma}_c + \hat{\beta}_c)$
7:         $\hat{X}_{bc} \leftarrow (X_{bc} - \mu_c^B)/\sigma_c^B$
8:         $Y_{bc} \leftarrow \hat{X}_{bc} \times (\gamma_c \times \delta_{bc}) + \beta_c$
9:      **end for**
10: **end for**

---

appropriate batch noise. Through experiments on style transfer, we find that the self-attention mechanism is an adaptive batch noise regulator for the model by enhancing instance specificity, which motivates us to design a new normalization to combine the advantage of BN and self-attention. This paper proposes an attention-based BN which adaptively emphasizes instance information called Instance Enhancement Batch Normalization (IEBN). The idea behind IEBN is simple. As shown in Fig. 1, IEBN extracts the instance statistic of a channel before BN and applies it to rescale the corresponding output channel of BN with a pair of additional parameters. IEBN costs a light parameter increment and it can be plugged into existing architectures readily. The extended experiment shows that IEBN outperforms BN on benchmark datasets over popular architectures for image classification. Our contribution is summarized as followed,

1. We offer a point of view that the self-attention mechanism can regulate the batch noise adaptively.

2. We propose a simple-yet-effective and attention-based BN called as Instance Enhancement Batch Normalization (IEBN). We demonstrate the effectiveness of it on benchmark datasets with different network architectures.

Related Work

This session reviews the works in two related fields: normalization and self-attention mechanism, especially those combining these techniques.

Normalization. Normalization is an important component of a deep network. Multiple normalization methods have been proposed for different tasks. Batch Normalization (Ioffe and Szegedy 2015) which normalizes input by mini-batch statistics has been a foundation of visual recognition tasks (He et al. 2016). Instance Normalization (Ulyanov, Vedaldi, and Lempitsky 2017) performs one instance BN-like normalization and is widely used in generative model (Johnson, Alahi, and Fei-Fei 2016; Zhu et al.

2017). There are some variants of BN, such as Conditional Batch Normalization (De Vries et al. 2017) for Visual Questioning and Answering, Group Normalization (Wu and He 2018) and Batch Renormalization (Ioffe 2017) for small-batch-size training. Also, there are Adaptive Batch Normalization (Li et al. 2018) for domain adaptation and Switchable normalization (Luo, Ren, and Peng 2018) which learns to select different normalizers for different layers. Among them, Conditional Batch Normalization and Batch Renormalization adjust the trainable parameters in the reparameterization step of BN, and they are similar to our work that modifies the trainable scaling parameter.

Self-Attention Mechanism. The self-attention mechanism selectively focuses on the most informative components of a network via self-information processing and has gained a promising performance on vision tasks. The procedure can be divided into three parts. First, the added-in module extracts internal information of a networks which can be squeezed channel-wise information (Hu, Shen, and Sun 2018; Li et al. 2019) or spatial information (Wang et al. 2018; Li, Hu, and Yang 2019). Next, the module processes the extraction and generates a mask to measure the importance of features via a fully connected layer (Hu, Shen, and Sun 2018), convolution layer (Wang et al. 2018) or LSTM (Huang et al. 2019b). Last, the mask is applied back to the features to adjust feature importance.

Comparison with Recent Works. The cooperation of BN and attention dates back to Visual Questioning and Answering (VQA). For this task, Conditional Batch Normalization (De Vries et al. 2017) is proposed to adjust the feature extraction of an image via the features collected from the question. Note that for VQA, the features from question can be viewed as external attention to guide the training of the overall network. In our work, IEBN can be viewed as a kind of Conditional Batch Normalization but the guidance of the network training is carried out by the internal attention since we extract the attention information from the image itself. Instance Meta Normalization (ILM) (Jia, Chen,

and Chen 2019) is a recent work that combines normalization and self-attention. Different from ILM that generates a rescaling coefficient of normalization by modeling the relationship between channels, we generate the rescaling coefficient independently for each channel.

BN and SE Module

This session first reviews BN and then reviews Squeeze-and-Excitation (SE) module (Hu, Shen, and Sun 2018). SE is regarded as the representative of the self-attention mechanism and is used for the analysis of the self-attention mechanism in our paper.

We consider a batch input $X \in \mathbb{R}^{B \times C \times H \times W}$, where B, C, H and W stand for batch size, number of channels, height and width respectively. For simplicity, we denote $X_{bchw} = X[b, c, h, w]$ as the value of pixel (h, w) at the channel c of the instance b and $X_{bc} = X[b, c, :, :]$ as the tensor at the channel c of the instance b.

Review of BN and SE Module

BN. The computation of BN can be divided into two steps: batch-normalized step and reparameterization step. Without loss of generality, we perform BN on the channel c of the instance b, i.e., X_{bc}. In the batch-normalized step, each channel is normalized using mean and variance of a batch,

$$\hat{X}_{bc} = (X_{bc} - \mu_c^B)/\sigma_c^B, \tag{1}$$

where μ_c^B, σ_c^B are defined in Step 3 and Step 4 in Alg. 1 as the estimation of mean and standard derivation of the whole dataset respectively. Then in the reparameterization step, a pair of learnable parameters γ_c, β_c scale and shift the normalized tensor $\hat{X}_{bc}$ to restore the representation power,

$$\hat{Y}_{bc} = \hat{X}_{bc} \times \gamma_c + \beta_c. \tag{2}$$

As mentioned in Introduction, the batch noise mainly comes from the batch-normalized step where the feature of the instance b is mixed with information from the batch, i.e., μ_c^B and σ_c^B.

SE Module. SE module utilizes the relationship between all channels in the same layer and generates a reweighing coefficient for each channel. Consider that SE module is connected to X and recalibrates the channel information of X.

First, the SE module squeezes the information of channels for the instance b by taking the average over each channel,

$$m_{bc} = \text{AVG}(X_{bc}) = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{bchw}, \tag{3}$$

where $c = 1, \cdots, C$ goes through all channels. Then, a fully connected layer stacking of a linear transformation, ReLU activation and a linear transformation with learnable parameters $\mathbf{W}$ is used to fuse the information of all the channels. Then Sigmoid function (i.e., $\text{sig}(z) = 1/(1 + e^{-z})$) is applied to the value and we get the rescaling coefficient for each channel. We denote the fully connected layer as $\text{FC}(\cdot; \mathbf{W})$ which maps $\mathbb{R}^C$ to $\mathbb{R}^C$. The rescaling coefficient is

$$[\hat{\delta}_{b1}; \cdots; \hat{\delta}_{bC}] = \text{sig}(\text{FC}([m_{b1}; \cdots; m_{bC}]; \mathbf{W})). \tag{4}$$

Finally, the rescaling coefficient is applied to its corresponding channel to adjust the channel, and we obtain $X_{bc} \times \hat{\delta}_{bc}$ with $c = 1, \cdots, C$.

Adaptive Batch Noise Regulation via SE Module

We explore the role of self-attention mechanism on regulating the batch noise through the style transfer task (Gatys, Ecker, and Bethge 2016). We use the style transfer method which generates a stylized image by a network called transformation network (Johnson, Alahi, and Fei-Fei 2016).

It has been empirically shown that the type of normalization in the network has an impact on the quality of image generation (Ulyanov, Vedaldi, and Lempitsky 2017; Huang and Belongie 2017; Dumoulin, Shlens, and Kudlur 2016). The style transform task is sensitive to the batch noise. Instance Normalization (IN), a "batch-noise-free" normalization, has been proved to have a significant advantage over BN in style transfer tasks (Ulyanov, Vedaldi, and Lempitsky 2017). The formulation of IN is followed,

$$\left(\frac{X_{bc} - \mu(X_{bc})}{\sigma(X_{bc})}\right) \cdot \gamma + \beta = \frac{\gamma}{\sigma(X_{bc})} \cdot X_{bc} + \beta - \frac{\mu(X_{bc})}{\sigma(X_{bc})} \cdot \gamma, \tag{5}$$

where $\mu(X_{bc})$ and $\sigma(X_{bc})$ denote the mean and standard deviation of the instance b at the channel c. Similarly, the formulation of BN can be written in this form,

$$\left(\frac{X_{bc} - \mu_c^B}{\sigma_c^B}\right) \cdot \gamma + \beta = \frac{\gamma}{\sigma_c^B} \cdot X_{bc} + \beta - \frac{\mu_c^B}{\sigma_c^B} \cdot \gamma. \tag{6}$$

γ and β are learned parameters and both are closely related to the target style (Dumoulin, Shlens, and Kudlur 2016). From Eqn. 5 and Eqn. 6, IN or BN directly associates the scaling of γ that affects the style of images. Different from BN, IN affects the style by self-information instead of batch information. Fig. 2 compares the quality of images generated by the BN or IN transformation network. Note that, the style transfer task is noise-sensitive, which coincides with (Salimans and Kingma 2016). When the batch noise is added by BN, the style of the generated images becomes confused.

We add the SE module to the BN transformation network (denoted as SE+BN) to find its effectiveness of regulating batch noise. In Fig. 2, the attention mechanism (SE) visually improves the effect of style transfer and the quality of the generated images is closer to that of IN. Fig. 3 shows the training loss against the number of iterations by applying the style Mosaic. The SE+BN network achieves smaller style loss and smaller content loss than the original BN network and is closer to an IN network (see Appendix for more results about the loss by applying other styles).

To analyze the effect of the self-attention mechanism, we consider such SE+BN structure, i.e., the SE module is plugged after BN layer. The formulation of plugging SE module into BN network is as followed,

$$\left(\frac{X_{bc} - \mu_c^B}{\sigma_c^B}\right) \cdot \gamma\hat{\delta}_{bc} + \beta\hat{\delta}_{bc} = \frac{\gamma\hat{\delta}_{bc}}{\sigma_c^B} \cdot X_{bc} + \beta\hat{\delta}_{bc} - \frac{\mu_c^B}{\sigma_c^B} \cdot \gamma \cdot \hat{\delta}_{bc}, \tag{7}$$

where $\hat{\delta}_{bc}$ is rescaling coefficient generated by applying SE module to $\hat{Y}_b$ in Eqn. 2. Although BN introduces batch noise

to an instance, we can see that $\hat{\delta}_{bc}$ helps adjust the batch statistics after applying the SE module to BN in Eqn. 7. The attention mechanism such as the SE module may be good at alleviating the batch noise and we will investigate it further.

Instance Enhancement Batch Normalization

Through experiments on style transfer shown in Fig. 2 and Fig. 3, we find that the self-attention mechanism is an adaptive batch noise regulator for the network by enhancing instance specificity, which motivates us to design a new normalization to combine the advantage of BN and self-attention. This session introduces IEBN and analyzes its role in enhancing instance information in style transfer. Last, we evaluate the performance of IEBN empirically on the image classification task.

Formulation of IEBN

The showcase of IEBN is shown in Fig. 1, where we highlight the instance enhancement process of one channel. The detailed computation of the forward pass in training phase can be found in Alg. 1. IEBN is based on the adjustment of the trainable scaling parameter on BN and its implementation consists of three operations: global squeezing, feature processing, and instance embedding.

Global Squeezing. The global reception field of a feature map is captured by average pooling $\text{AVG}(\cdot)$. We obtain a shrinking feature descriptor m_{bc} of the channel c for the instance b, which is the same as Eqn. 3. This is,

$$m_{bc} = \text{AVG}(X_{bc}) = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{bchw}. \quad (8)$$

Feature Processing. Compared with Eqn. 4, we introduce an addition pair of parameters $\hat{\beta}_c$, $\hat{\gamma}_c$ for the c_{th} channel, which serve as scale and shift respectively to linearly transform m_{bc}. Then Sigmoid function $(\text{sig}(\cdot))$ is applied to the value after linear transformation as a gating mechanism:

$$\delta_{bc} = \text{sig}(\hat{\gamma}_c m_{bc} + \hat{\beta}_c). \quad (9)$$

Specially, the parameters $\hat{\gamma}_c, \hat{\beta}_c$ are initialized by constant 0 and -1 respectively. We will discuss it in Ablation Study.

Instance Embedding. δ_{bc} works as a weight coefficient to adjust the scaling in the reparameterization step of BN for the instance b. We embed the recalibration δ_{bc} to compensate the instance information in Eqn. 2,

$$Y_{bc} = \hat{X}_{bc} \times (\gamma_c \times \delta_{bc}) + \beta_c. \quad (10)$$

The δ_{bc} is composed of nonlinear activation function and an additional pair of parameters, which helps improve the nonlinearity of reparameterization of BN.

We conduct IEBN on all channels, i.e., $c = 1, 2, \cdots, C$. Compared with BN, the parameter increment comes from the additional pair of parameters for generating coefficient for each channel. The total number of parameter increment is equal to twice the number of channels.

Forward Pass in Testing Phase. During testing, the batch-normalized step as Eqn. 1 in IEBN uses the same operations as in the original BN (Ioffe and Szegedy 2015),

where μ_c^B and σ_c^B are replaced with the moving average of the mean and standard derivation acting as the estimation of the statistics of the whole dataset.

Instance Enhancement in Style Transfer

The main purpose of this section is to illustrate that the self-attention mechanism, including IEBN and SE+BN, can regulate the batch noise adaptively.

IEBN is a BN equipped with the self-attention and Fig. 2 shows the similarity of the generated images of the SE module and IEBN. In fact, we consider IEBN:

$$\left(\frac{X_{bc} - \mu_c^B}{\sigma_c^B} \right) \cdot \gamma \delta_{bc} + \beta = \frac{\gamma \delta_{bc}}{\sigma_c^B} \cdot X_{bc} + \beta - \frac{\mu_c^B}{\sigma_c^B} \cdot \gamma \cdot \delta_{bc}, \quad (11)$$

where δ_{bc} is defined in Eqn. 9 and δ_{bc} contains information from the instance b. The added-in δ_{bc} seems to be only directly applied to scaling parameter γ of BN, but it does scale the batch information (i.e., μ_c^B, σ_c^B) to regulate the batch information via supplement of instance information. This adjustment of batch information via δ_{bc} makes the Eqn. 11 closer to Eqn. 5 than Eqn. 6 and also leads to the similar results in style transfer between IN and IEBN. We can also illustrate the effect of instance enhancement from experiment. The comparison in Fig. 2 shows that both IEBN and SE+BN can provide comparable generated images to IN and they are better than the BN network. Also, Fig. 3 shows that SE+BN and IEBN achieve a smaller style loss and a smaller content loss than the BN network. The performance of SE+BN or IEBN is closer to IN and better than BN, supporting our main purpose.

Experiments

In this section, we evaluate the effect of IEBN in image classification and empirically demonstrate its effectiveness.

Dataset and Model. We conduct experiments on CIFAR10, CIFAR100 (Krizhevsky and Hinton 2009), and ImageNet 2012 (Russakovsky et al. 2015). CIAFR10 or CIFAR100 has 50k train images and 10k test images of size 32 by 32 but has 10 and 100 classes respectively. ImageNet 2012 (Russakovsky et al. 2015) comprises 1.28 million images for training and 50k for validation from 1000 classes, and the random cropping of size 224 by 224 is used in our experiments. We evaluate our method on popular networks, including ResNet (He et al. 2016), PreResNet (He et al. 2016), ResNeXt (Xie et al. 2017), DenseNet (Huang et al. 2017). In our experiments, we replace all the BNs in the original networks with IEBN. The implementation details can be found in Appendix.

Image Classification. As shown in Table 1, IEBN improves the testing accuracy over BN for different datasets and different network backbones. For small-class-dataset CIFAR10, the performance of the networks with BN is good enough, so there is not large space for improvement. However, for CIFAR100 and ImageNet, the networks with IEBN achieve a significant testing accuracy improvement over BN. In particular, the performance improvement of ResNet with the IEBN is most remarkable. Also, IEBN has comparable performance to SE but has the advantage over SE in the aspect of low parameter increment.

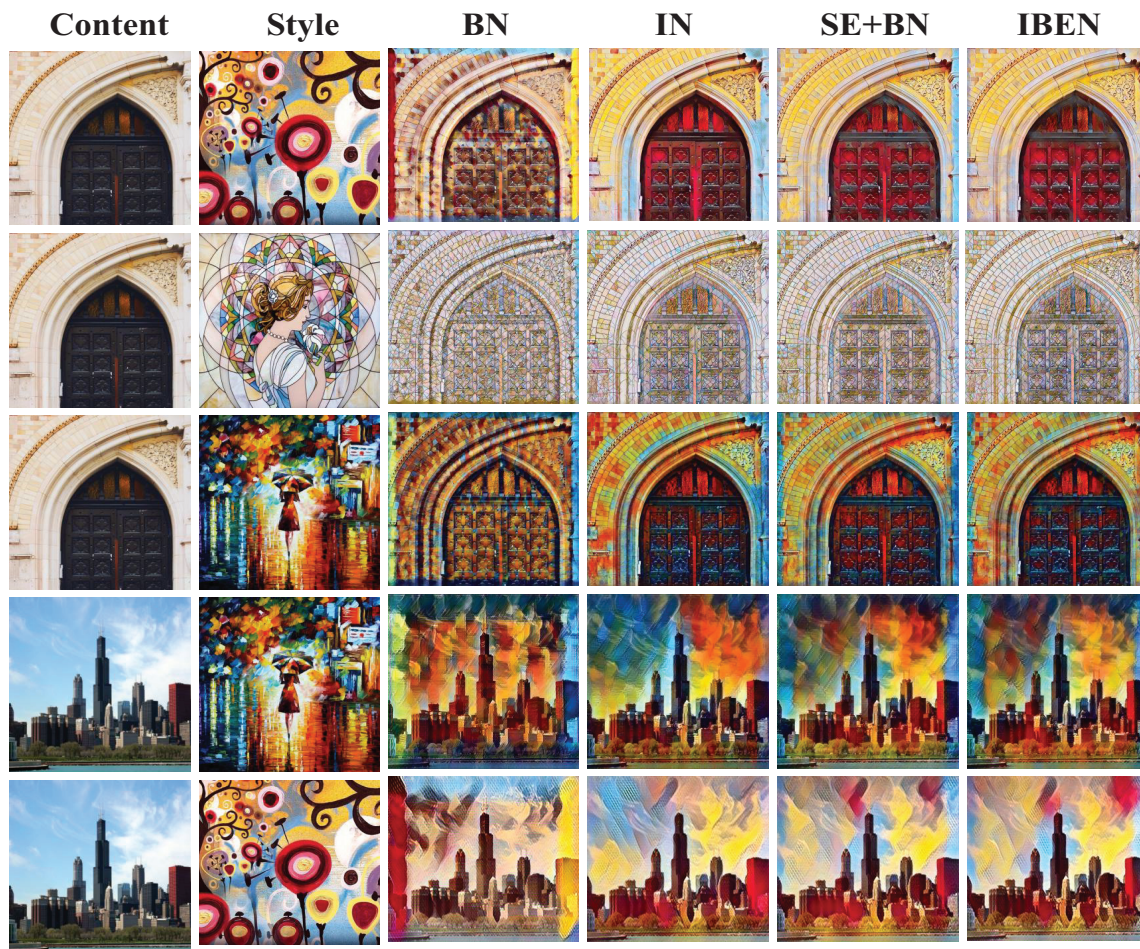| Content	Style	BN	IN	SE+BN	IBEN

Figure 2: Stylization results obtained by applying style (second column) to the content images (first column) with different normalization methods. The style of the images generated by the BN network appears more confused, while the stylized images generated by SE+BN or IEBN are quite similar to IN visually.
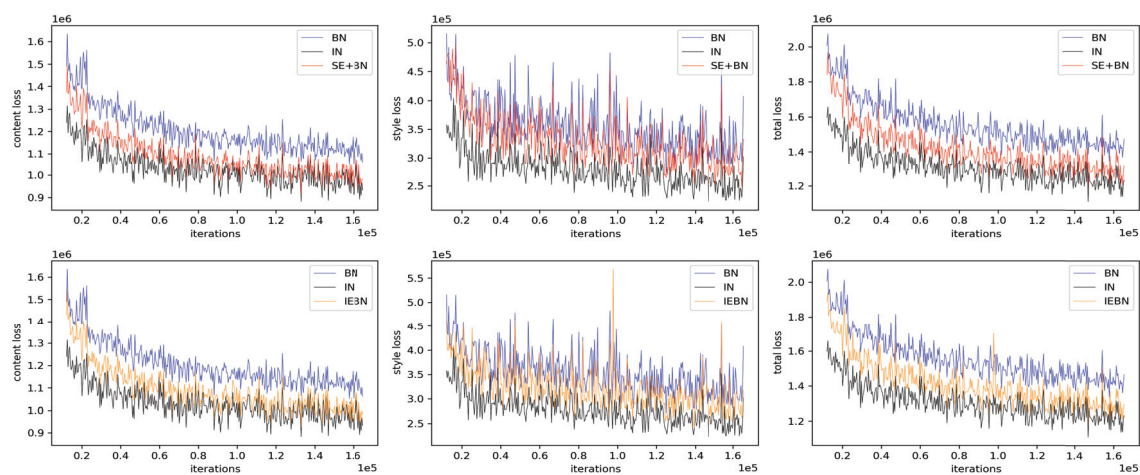

Figure 3: Training curves of style transfer networks with Mosaic style and different normalization methods.

Network	Dataset	BN		SE		IEBN	
		#P(M)	top1-acc.	#P(M)	top1-acc.	#P(M)	top1-acc.
ResNet164	CIFAR100	1.73	74.29	1.93	75.80	**1.75**	**77.09**
PreResNet164	CIFAR100	1.73	76.56	1.92	**77.41**	**1.75**	77.27
DenseNet100-12	CIFAR100	0.80	77.23	**0.81**	78.43	0.82	**78.57**
ResNeXt29,8x64	CIFAR100	34.52	81.47	35.04	**83.05**	**34.57**	82.45
ResNet164	CIFAR10	1.70	93.93	1.91	94.65	**1.73**	**95.03**
Preresnet164	CIFAR10	1.70	95.01	1.90	**95.18**	**1.73**	95.09
DenseNet100-12	CIFAR10	0.77	95.29	**0.78**	95.76	0.79	**95.83**
ResNeXt29,8x64	CIFAR10	34.43	96.11	34.94	**96.30**	**34.48**	96.26
ResNet34	ImageNet	21.81	73.91	21.97	**74.39**	**21.82**	74.38
ResNet50	ImageNet	25.58	76.01	28.09	76.61	**25.63**	**77.10**
ResNet152	ImageNet	60.27	77.58	66.82	78.36	**60.41**	**79.17**
ResNeXt50	ImageNet	25.03	77.19	27.56	**78.04**	**25.09**	77.99

Table 1: Accuracy (%) on benchmark datasets with different architectures using BN, SE module or IEBN.

Analysis

In this session, we further explore the role of the self-attention mechanism on enhancing instance information and regulating the batch noise. We analyze through the experiments with two kinds of noise attack designed in our paper.

Constant-Noise Attack

We add the constant noise into every BN of the network at the batch-normalized step as followed,

$$\hat{X}_{bc} = [(X_{bc} - \mu_c^B)/\sigma_c^B] \cdot N_a + N_b, \qquad (12)$$

where (N_a, N_b) is a pair of constants working as the constant noise. The constant noise added in Eqn. 12 is equivalent to disturbing μ_c^B and σ_c^B such that we will use the inaccurate estimation of mean and variance respectively of the whole dataset in training.

We denote $(X_{bc} - \mu_c^B)/\sigma_c^B$ as Δ. Then in the reparameterization step of BN, we introduce the learnable parameters γ and β and get

$$\begin{aligned} \hat{X}_{bc} &= (\Delta \cdot N_a + N_b) \cdot \gamma + \beta \\ &= \Delta \cdot (N_a \cdot \gamma) + (N_b \cdot \gamma + \beta). \end{aligned} \qquad (13)$$

From the form of Eqn. 13, the impact of constant noise can be easily neutralized by the linear transformation of γ and β because N_a and N_b are just constants. Table 2 shows the testing accuracy of ResNet164 on CIFAR100 under different pairs of constant noise and Fig. 4 shows the training curve under the constant noise attack. The results indicate that the BN network is not good at handling most constant noise. The trainable γ and β of BN do not have enough capacity to reduce the impact of the constant noise. Due to the forward pass, the noise will accumulate as the depth increases and a certain amount of noise leads to training instability and poor performance. As shown in Table 2, SE module can partly alleviate this problem, but not enough as we can see the high variance of the testing accuracy under most pairs of constant noise. For IEBN, we can rewrite Eqn. 13 as

$$\hat{X}_{bc} = \Delta \cdot (N_a \cdot \gamma \cdot \delta_{bc}) + (N_b \cdot \gamma \cdot \delta_{bc} + \beta), \qquad (14)$$

where δ_{bc} denotes the attention learned in IEBN. Compared to Eqn. 13 with Eqn. 14, δ_{bc} from IEBN has successfully adjusted constant noise and even achieved better performance under partial noise configuration. If δ_{bc} only excites β, we can rewrite Eqn. 14 as

$$\hat{X}_{bc} = \Delta \cdot \underbrace{N_a \cdot \gamma}_{\gamma'} + \underbrace{N_b \cdot \gamma + \beta \cdot \delta_{bc}}_{\beta'}, \qquad (15)$$

where δ_{bc} can only adjust the noise in β' instead of γ'. But if applied to γ, δ_{bc} can handle the noise of scale and bias simultaneously. It may be the reason why the result about only exciting β is worse than the other in Table 4 (Right), but better than the BN network in Table 1.

(N_a, N_b)	BN	SE	IEBN
(1.0,0.0)	$74.29_{(\pm0.64)}$	$75.80_{(\pm0.25)}$	$77.09_{(\pm0.15)}$
(0.8,0.8)	$45.42_{(\pm31.42)}$	$73.18_{(\pm0.66)}$	$75.42_{(\pm0.08)}$
(0.8,0.5)	$46.10_{(\pm31.91)}$	$71.59_{(\pm1.77)}$	$77.39_{(\pm0.09)}$
(0.5,0.5)	$35.77_{(\pm34.76)}$	$74.61_{(\pm0.56)}$	$77.00_{(\pm0.29)}$
(0.5,0.2)	$73.10_{(\pm1.72)}$	$75.72_{(\pm1.47)}$	$77.11_{(\pm0.08)}$

Table 2: The testing accuracy (mean $\pm$ std %) of ResNet164 on CIFAR100. (N_a, N_b) is a pair of constants added to BN at the batch-normalized step as stated in Eqn. 12. $(1.0, 0.0)$ means we do not add the noise.

Mix-Dataset Attack

In this part, we consider interfering with μ_c^B and σ_c^B by simultaneously training on the datasets with different distributions in one network. Unlike constant noise which is added to networks directly, this noise is implicit and is generated when BN computes the mean and variance of training data from different distributions. These datasets differ widely in their distribution and causes severe batch noise. Compared with the constant noise, this noise is not easy to eliminate by linear transformation of γ and β.
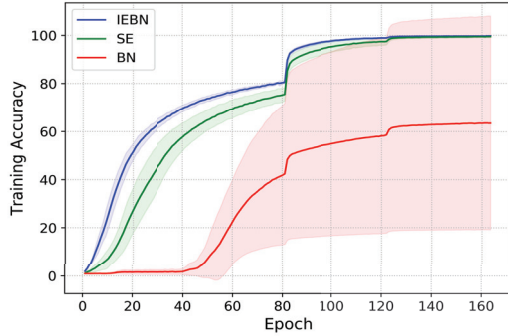
Figure 4: The training curve (mean $\pm$ std %) of ResNet164 on CIFAR100 under the constant noise attack $(N_a, N_b) = (0.5, 0.5)$.

Operator	Test Acc.		Position	Test Acc.
Linear	**77.09($\pm$0.15)**		γ	**77.09($\pm$0.15)**
Identity	67.53($\pm$2.49)		β	75.03($\pm$0.54)
FC	76.11($\pm$0.28)		γ and β	77.02($\pm$0.08)

Table 4: (**Left**.) Testing accuracy (%) with different ways to process the squeezed features. (**Right**.) Testing accuracy (%) with different positions that the δ_{bc} excites. γ and β are the parameters in reparameterization step of BN on CIFAR100.

In our experiments, we train ResNet164 on CIFAR100 but mix up with MINIST (LeCun and Cortes 2010) or Fashion-MINIST (Xiao, Rasul, and Vollgraf 2017) in a batch and compare the performance of BN and IEBN. Table 3 shows the test accuracy on CIFAR100. As k increases, the batch noise becomes more severe for CIFAR100 since μ_c^B and σ_c^B contain more information about MNIST or FashionMnist. In most cases, despite the severe noise like "C+2$\times$", the IEBN network still performs better than the BN network training merely on CIFAR100. On the other hand, the decline in the accuracy of IEBN or SE is smaller than that of BN. These phenomena illustrate that the model with the attention module has a stronger ability to resist the batch noise even if under the influence of MINIST or FashionMINIST and alleviates the degradation of network generalization. Note that, this result again supports our argument that the self-attention mechanism can regulate the batch noise adaptively.

	BN	SE	IEBN
dataset	test acc.	test acc.	test acc.
C	74.29 (0.00)	75.8 (0.00)	77.09 (0.00)
C+2$\times$M	73.13 (1.16)	75.3 (0.50)	76.65 (0.44)
C+3$\times$M	71.54 (2.75)	74.94 (0.86)	76.03 (1.06)
C+2$\times$F	71.56 (2.73)	74.3 (1.51)	75.57 (1.52)
C+3$\times$F	71.27 (3.02)	73.67 (2.13)	74.26 (2.83)

Table 3: Test accuracy (accuracy drop %) on CIFAR100 with ResNet-164. "C+$k\times$ M/F" means we sample a batch consisted of 100 images from CIFAR100 (C) and 120$\times k$ images from MNIST (M) or FashionMNIST (F) at each iteration during training. "accuracy drop" means the drop of accuracy compared with network trained merely on CIFAR100.

Ablation Study

In this section, we conduct experiments to explore the effect of different configurations of IEBN. All experiments are performed on CIFAR100 with ResNet164 using 2 GPUs.

The Way of Generating δ_{bc}. This part we study different ways to process the squeezed features to generate δ_{bc}. As shown in Alg. 1, IEBN squeezes the channel through global average pooling AVG($\cdot$) and processes the squeezed feature by linear transformation (i.e. AVG(X_{bc})$\times\hat{\gamma}_c + \hat{\beta}_c$) for each channel, denoted as "Linear". We also consider another two alternative ways to process the the squeezed feature. The first one is that we remove the additional trainable parameters $\hat{\gamma}_c$ and $\hat{\beta}_c$ for linear transformation in IEBN and directly apply the squeezed feature after sigmoid function to the channel, denoted as "Identity". The second one is that we use a fully connected layer stacking of a linear transformation, a ReLU layer, and a linear transformation to fuse the squeezed features of all channels $\{\text{AVG}(X_{bc})\}_{c=1}^C$, denotes as "FC". "FC" is similar to the configuration as SE module introduced in (Hu, Shen, and Sun 2018). As shown in Table 4 (Left), "FC" operator provides more nonlinearity than "Linear" operator (IEBN), but such nonlinearity may lead to overfitting and the "Linear" operator (IEBN) simplifies the squeezed feature processing and has better generalization ability. Furthermore, the result of "Identity" indicates that it is not enough to simply and directly use instance information to enhance self-information without any trainable parameters. The operators with sufficient trainable parameters, such as "Linear" (IEBN) and "FC", are needed to process the instance information such that the adaptive and advantageous noise during training can be regulated to improve the performance.

Excitation Position. We study the influence of different positions that δ_{bc} excites. For the self-attention mechanism like SENet (Hu, Shen, and Sun 2018), the rescaling coefficient usually excites both the trainable parameter γ and β of BN. In IEBN, the δ_{bc} is only applied to adjust the scaling parameter γ in BN. To differentiate the influence of the excitation positions, Table 4 (Right) shows the testing accuracy with different positions which the δ_{bc} excites. We show that the performance is unsatisfied when the δ_{bc} is merely exciting β. Moreover, there is a slight difference between exciting only γ and exciting both γ and β, and the former excitation position has better performance. From the point of view of adjusting noise, Eqn. 14 and Eqn. 15 can explain the result shown in Table 4 (Right).

Initialization of $\hat{\gamma}_c$ and $\hat{\beta}_c$. We study the initialization of parameters $\hat{\gamma}_c$ and $\hat{\beta}_c$ which are used to process the squeezed feature in IEBN. Specifically, we use constant 1, 0 and -1 for
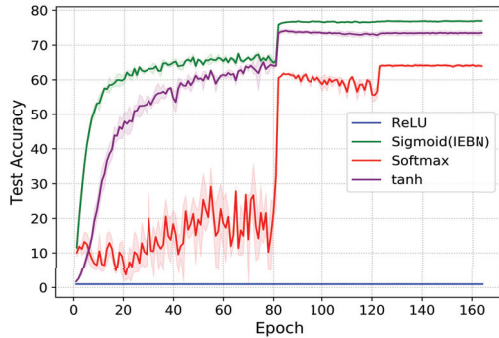
Figure 5: The training curve (mean $\pm$ std%) of different activation functions in IEBN.

grid search to find the best pair of initialization for $\hat{\gamma}_c$ and $\hat{\beta}_c$. The initialization of the trainable parameters of IBEN $\hat{\gamma}_c$ and $\hat{\beta}_c$ have a significant impact on the model. The best choice of $\hat{\gamma}_c$ and $\hat{\beta}_c$ are 0 and -1. The theoretical nature behind the best initialization configuration will be our future work.

$\hat{\gamma}_c \setminus \hat{\beta}_c$	1	0	-1
1	68.5	66.96	69.53
0	75.86	76.21	**77.09**
-1	74.64	74.73	75.31

Table 5: Test accuracy (%) with different constant initialization for trainable parameters scaling $\hat{\gamma}_c$ and shift $\hat{\beta}_c$ in IEBN.

Activation Function. We consider four options for activation function: sigmoid, tanh, ReLU and Softmax in IEBN. The testing accuracy results are reported in Fig. 5. Among these options, sigmoid may be the best choice, which is also used in many attention-based methods like SENet (Hu, Shen, and Sun 2018) to generate attention maps as a gate mechanism. The testing accuracy of different choices of activation functions in Fig. 5 shows that sigmoid helps IEBN as a gate to rescale channel features better. The similar ablation study in the SENet paper (Hu, Shen, and Sun 2018) also shows the performance of different activation functions like: sigmoid > tanh > ReLU (bigger is better), which coincides to our reported results.

Conclusion

In this paper, we introduce two kinds of noise brought by BN and offer a point of view that the self-attention mechanism can regulate the batch noise adaptively. We propose a simple-yet-effective and attention-based BN called Instance Enhancement Batch Normalization (IEBN). We demonstrate empirically the effectiveness of IEBN on benchmark datasets with different network architectures and also provide the ablation study to explore the effect of different configurations of IEBN.

References

Bjorck, N.; Gomes, C. P.; Selman, B.; and Weinberger, K. Q. 2018. Understanding batch normalization. In *Advances in Neural Information Processing Systems*, 7694–7705.

Cai, Y.; Li, Q.; and Shen, Z. 2019. A quantitative analysis of the effect of batch normalization on gradient descent. In *International Conference on Machine Learning*, 882–890.

De Vries, H.; Strub, F.; Mary, J.; Larochelle, H.; Pietquin, O.; and Courville, A. C. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, 6594–6604.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

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

Ge, R.; Huang, F.; Jin, C.; and Yuan, Y. 2015. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, 797–842.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

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

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *Computer Vision and Pattern Recognition*.

Huang, L.; Zhou, Y.; Zhu, F.; Liu, L.; and Shao, L. 2019a. Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4874–4883.

Huang, Z.; Liang, S.; Liang, M.; and Yang, H. 2019b. Dianet: Dense-and-implicit attention network. *CoRR* abs/1905.10671.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate

shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 448–456. JMLR.org.

Ioffe, S. 2017. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in neural information processing systems*, 1945–1953.

Jia, S.; Chen, D.-J.; and Chen, H.-T. 2019. Instance-level meta normalization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin, C.; Ge, R.; Netrapalli, P.; Kakade, S. M.; and Jordan, M. I. 2017. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1724–1732. JMLR. org.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database. *http://yann.lecun.com/exdb/mnist/*.

Li, Y.; Wang, N.; Shi, J.; Hou, X.; and Liu, J. 2018. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition* 80:109–117.

Li, X.; Wang, W.; Hu, X.; and Yang, J. 2019. Selective kernel networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, X.; Hu, X.; and Yang, J. 2019. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *CoRR* abs/1905.09646.

Luo, P.; Wang, X.; Shao, W.; and Peng, Z. 2019. Towards understanding regularization in batch normalization. In *International Conference on Learning Representations*.

Luo, P.; Ren, J.; and Peng, Z. 2018. Differentiable learning-to-normalize via switchable normalization. *arXiv preprint arXiv:1806.10779*.

Luo, L.; Xiong, Y.; and Liu, Y. 2019. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*.

NSCC. 2019. The computational work for this article was partially performed on resources of the national supercomputing centre, singapore (https://www.nscc.sg). *https://www.nscc.sg*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3):211–252.

Salimans, T., and Kingma, D. P. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, 901–909.

Santurkar, S.; Tsipras, D.; Ilyas, A.; and Madry, A. 2018. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, 2483–2493.

Shekhovtsov, A., and Flach, B. 2019. Stochastic normalizations as bayesian learning. In Jawahar, C. V.; Li, H.; Mori, G.; and Schindler, K., eds., *Computer Vision – ACCV 2018*, 463–479. Cham: Springer International Publishing.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. 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*, 6924–6932.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *CVPR*, 7794–7803.

Wu, Y., and He, K. 2018. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR* abs/1708.07747.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.