A Simple Normalization Technique Using Window Statistics to Improve the Out-of-Distribution Generalization on Medical Images

Chengfeng Zhou, Jun Wang, Suncheng Xiang, Member, IEEE, Feng Liu, Hefeng Huang, and Dahong Qian, Senior Member, IEEE

Abstract—Since data scarcity and data heterogeneity are prevailing for medical images, well-trained Convolutional Neural Networks (CNNs) using previous normalization methods may perform poorly when deployed to a new site. However, a reliable model for real-world clinical applications should generalize well both on in-distribution (IND) and out-of-distribution (OOD) data (e.g., the new site data). In this study, we present a novel normalization technique called window normalization (WIN) to improve the model generalization on heterogeneous medical images, which offers a simple yet effective alternative to existing normalization methods. Specifically, WIN perturbs the normalizing statistics with the local statistics computed within a window. This feature-level augmentation technique regularizes the models well and improves their OOD generalization significantly. Leveraging its advantage, we propose a novel self-distillation method called WIN-WIN. WIN-WIN can be easily implemented with two forward passes and a consistency constraint, serving as a simple extension to existing methods. Extensive experimental results on various tasks (6 tasks) and datasets (24 datasets) demonstrate the generality and effectiveness of our methods.

Index Terms—Normalization, out-of-distribution generalization, multi-center data.

I. INTRODUCTION

Despite the tremendous success of CNNs in medical image analysis, they are mainly built upon the “i.i.d. assumption”, which states that training data and test data are independent and identically distributed. The assumption rarely holds in real-world applications due to the nature of medical images, rendering the significant drop in performance of well-trained models on unseen data with distribution shifts [1], [2], [3]. Since the high costs of data acquisition and data annotation lead to data scarcity, the data used for model training can only capture a small portion of real data distribution (see Fig. 1 (a)). Meanwhile, the data heterogeneity caused by inconsistent standards (e.g., various operating procedures and imaging equipment) exacerbates the distribution shifts in medical images. Slight differences in appearance results in significant model performance dips [2]. Therefore, achieving OOD generalization on heterogeneous data is a crucial challenge for real-world clinical applications [4], [5].
The most straightforward solution is to acquire data from various sites sufficiently and train a robust model. However, it would be high-costly and even impossible in real-world applications. In practice, another economical and popular solution is data augmentation. Data augmentation enhances the breadth of seen data with predefined class-preserving operations, such as geometric augmentation (e.g., rotation, cropping, and elastic morphing [6]) and mixing operations [7], [8], [9]. With the help of the augmented samples simulating the unseen images, CNNs improve their generalization to OOD data. Although the data augmentation methods no longer require significant manual effort in designation [10], [11], their performance is less impressive for improving the OOD generalization on medical images because they are mainly developed for the natural images with large domain gaps [12], [13], [14]. Besides, they may increase computing overhead and impede model convergence.

Normalization layers have been essential components of deep models. Unlike traditional normalization techniques [15], [16], [17], they normalize feature activations (i.e., layer inputs) rather than raw inputs to aid in the training process. A lot of them are generally and over-simplistically built upon the “i.i.d. assumption”. For instance, Batch Normalization (BN) [18] constrains intermediate features within the normalized distribution using mini-batch statistical information to stabilize and accelerate training. Its fundamental flaw is train-test statistics inconsistency, which worsens under distribution shifts. Instance normalization (IN) [19] overcomes this limitation by consistently computing the normalization statistics. During the training and testing, IN normalizes features with statistics of the spatial dimension. This procedure effectively reduces the data distribution gap (i.e., by minimizing the style discrepancy) and improves the OOD generalization [20] (see Fig. 1 (b)). However, its improvement in OOD generalization is still insufficient. On this basis, subsequent studies have combined it with other mechanisms to address the OOD generalization problem, but they are complex and less efficient for medical images [20], [21], [22], [23], [24], [25]. For instance, Jin et al. [22] designed a normalization technique that consists of three components: Style Normalization and Restitution Module, Dual Causality loss, and Dual Restitution Loss.

In this work, we implement the feature-level data augmentation using a simple yet effective normalization technique called Window Normalization (WIN) to improve the OOD generalization on heterogeneous medical images. WIN uses the mean and variance of stochastic windows to perturb the normalization statistics with a mixup operation, which not only eliminates the gap between data distributions but also expands the feature space. With the benefits of WIN, we introduce a novel self-distillation scheme WIN-WIN which processes the input through different model modes twice and enforces the consistency between the outputs. WIN-WIN can be implemented with minimal coding and further improve the model’s OOD generalization. We demonstrate that our methods can generally boost OOD generalization of various tasks (such as glaucoma detection, breast cancer detection, chromosome classification, optic disc and cup segmentation, etc.), spanning 24 datasets, with free parameters and minimal impact on IND generalization. Our main contributions are summarized as follows:

- We develop a simple normalization WIN, to improve OOD generalization in heterogeneous data. WIN is a good alternative for existing normalization methods, with free parameters and a minimal impact on IND generalization.
- We propose a novel self-distillation scheme WIN-WIN based on the WIN. This method is straightforward to implement and highly effective.
- We demonstrate through extensive experiments that our methods significantly enhance OOD generalization across a variety of tasks. The code of our method is publicly available at https://github.com/joe1chief/windowNormalization.

II. RELATED WORK

A. Data Augmentation

Data augmentation is an important tool for training deep models, which effectively increases the data amount and enriches the data diversity through random operations such as translation, flipping, and cropping [26]. Intuitively, augmented inputs can help a model in learning the invariances among the data domains and improve its generalization to novel domains. Popular data augmentation techniques can be conducted at either image-level or feature-level. Image-level augmentation often requires manual designation, which is challenging due to the difficulty of simulating the data from unforeseen target domains. To this end, several methods have automatically searched for the augmentation policies in their predefined search space [10], [11]. However, as the augmentation policies are typically developed based on natural image datasets (e.g., ImageNet) with characteristics that significantly differ from those of medical images, the improvement for OOD generalization is less appealing.

In the OOD generalization literature, Lopes et al. [27] added noise to randomly selected patches on the input image. AugMix [8] and PIXMIX [9] mixed multiple augmented images, or an augmented image and dreamlike images, respectively. Zhou et al. [13] synthesized pseudo-novel domain data with a data generator. DeepAugment [1] perturbed the input image using an image-to-image network. Additionally, feature-level data augmentation also has shown their effectiveness. Verma et al. [28] simply mixed the latent features. Zhou et al. [12] developed a plug-and-play module that mixes feature statistics between instances. Li et al. [14] introduced an augmentation module that perturbs the feature with a data-independent noise and an adaptive dependent noise.

However, the aforementioned methods inevitably incur computational overhead or implement hard. Our method implements the feature-level data augmentation using the essential module in CNNs, normalization layers, and outperforms many prior data augmentation techniques for improving OOD generalization.
B. Normalization

Traditional normalization techniques are designed to bring inputs to a specific condition or state, typically applied during the image processing phase [15], [16], [17]. For example, histogram normalization [16] adjusts pixel values of an image so that its histogram matches a specified target histogram, while energy-based normalization [15] decomposes images into energy bands and standardizes each energy band with respect to a reference. With the rise of deep learning, normalization has become an essential part of CNNs, now used to normalize feature activations rather than raw inputs.

In the field of deep learning, a milestone is Batch Normalization (BN) [18], which has verified its generality on a wide variety of tasks. Although BN is empirically proven to accelerate the model convergence and combat overfitting, it exhibits several shortcomings due to its default statistics computation strategy [29], [30]. For example, it lost the instance-specific information. To this end, Li et al. [31] and Gao et al. [32] integrated a feature calibration scheme into BN to incorporate this information. Instance Normalization (IN) [19] is another significant development that normalizes features with statistics of the spatial dimension. It has been extensively applied in the field of image style transfer as it could eliminate instance-specific style discrepancy (namely, standardizing features with the instance-specific mean and variance) [33]. However, IN removes style information and resulted in inferior performance on in-distribution samples [20], [22]. Following the BN and IN, Group Normalization (GN) [29] utilizes the statistics of grouped channel dimensions to train the model with small batches. Meanwhile, several normalization techniques explore using the statistics from partial regions for normalization. Ortiz et al. [34] proposed the local context normalization (LCN), which uses unique statistics for each feature within its neighborhoods, including spatial and channel dimensions. This technique has proven effective in dense prediction tasks (including object detection, semantic segmentation, and instance segmentation).

Given the subtle decision-making involved in choosing which normalization techniques to apply in specific scenarios, many studies have explored combining multiple normalization techniques. Batch-Instance Normalization (BIN) [24] adaptively balanced the BN output and the IN output with a learnable gate parameter. This gating mechanism also used in the Switchable Normalization (SN) series [23], [25] which applied three types of statistics estimated channel-wise [19], layer-wise [35], and minibatch-wise [18]. Besides, Qiao et al. [36] introduced a different approach, stacking BN with GN instead of performing a weighted summation operation. These multi-normalization combination methods enable better adaptability and easy usage for various tasks. However, they often entail additional computational cost.

Crucially, the above normalization techniques are proposed under the assumption that training and test data follow the same distribution. Hence, numerous normalization studies have been proposed to address the distribution shifts between training and test distribution in real-world applications. For example, Mode Normalization [37] assigns the samples to different modes and normalized them using their corresponding statistics. Li et al. [5] employed local batch normalization to address non-iid training data. Tang et al. [21] expanded the training distribution with CrossNorm and bridged the gap between training and test distribution with SelfNorm. Besides, since the IN effectively minimizes style discrepancies among domains/instances, Pan et al. [20] mixed IN and BN in the same layer to reduce the distribution gaps and Jin et al. [22] introduced a style normalization and restitution module to eliminate the style variation using IN while preserving discriminative capabilities. Compared to these methods, WIN addresses the OOD generalization problem simply yet effectively. It is parameter-free and eliminates the need for cautious selection of the plug-in manner. As a variant of IN, it can be directly employed as the normalization layers, offering effortless benefits for OOD generalization.

III. METHODS

A. Background

Normalization methods typically involve feature standardization and affine transformation operations. Given the input feature $F \in \mathbb{R}^{N \times C \times H \times W}$ in 2D CNNs, where $N$, $C$, $H$, and $W$ are batch size, channel number, height, and width of the input feature, respectively. The feature standardization and affine transformation are formulated as follows:

$$\text{Standardization: } \tilde{F} = \frac{F - \mu}{\sqrt{\sigma^2 + \epsilon}},$$

$$\text{Affine: } \tilde{F} = \gamma \tilde{F} + \beta.$$  (2)

$\mu$ and $\sigma^2$ denote the mean and variance, respectively. $\epsilon$ is used to prevent division by zero. $\gamma$ and $\beta$ are the learnable parameters of an affine transformation.

In BN, the mean and variance during training are defined as:

$$\mu_{BN} = \frac{1}{NHW} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} f_{n,c,h,w},$$

$$\sigma_{BN}^2 = \frac{1}{NHW} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (f_{n,c,h,w} - \mu_{BN})^2.$$  (3)

$\mu_{BN}, \sigma_{BN}^2 \in \mathbb{R}^C$ are computed within each channel $c$ of the feature $f$ (see Fig. 2 (a)). During the evaluation, the mean and variance are computed as exponential moving averages of $\mu_{BN}, \sigma_{BN}^2$ and accumulated as:

$$\mu'_{BN} \leftarrow p \mu_{BN} + (1 - p) \mu_{BN},$$

$$\sigma'_{BN}^2 \leftarrow p \sigma_{BN}^2 + (1 - p) \sigma_{BN}^2.$$  (4)

where $p \in [0.0, 1.0]$ represents momentum. However, the running mean and variance often lead to significant performance degradation as they may not align with the statistics estimated on OOD data [30]. In addition, for the affine transformation, BN employs two learnable parameters $\gamma_{BN} \in \mathbb{R}^C$ and $\beta_{BN} \in \mathbb{R}^C$. 
Fig. 2. (a)-(c) Statistics calculation of BN, IN, and WIN. Each subplot shows the feature map $F \in \mathbb{R}^{N \times C \times H \times W}$. The mean and variance are computed by aggregating the values of blue pixels. And, all pixels share the same normalizing statistics. (d) Schematic Illustration: WIN (right) and WIN-WIN (left). WIN is adopted as the normalization layer in CNNs. It uses mixing statistics during the training and global statistics during the evaluation. WIN-WIN passes the input $x$ twice and ensures the consistency between outputs under different model mode, $\hat{y}$ and $\bar{y}$.

IN defines the mean and variance as:

$$
\mu_{IN}^{n,c} = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} f^{n,c,h,w},
$$

$$
\sigma_{IN}^{2,n,c} = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} \left( f^{n,c,h,w} - \mu_{IN}^{n,c} \right)^2.
$$

Equation (5)

$\mu_{IN}, \sigma_{IN}^{2} \in \mathbb{R}^{N \times C}$ are computed across the spatial dimensions independently for each channel and each instance (see Fig. 2 (b)). In addition, $\mu_{IN}$ and $\sigma_{IN}^{2}$ encode the instance-specific style information. Thus, standardizing the feature with $\mu_{IN}$ and $\sigma_{IN}^{2}$ constitutes the style normalization. Style normalization eliminates the feature variance caused by appearance, which benefits the OOD generalization. However, it inevitably removes some discriminative information potentially degrading the generalization [20]. In the IN, the calculation of mean and variance remains consistent during both training and evaluation, and the affine transformation which adopts $y_{IN} \in \mathbb{R}^{N \times C}$ and $\beta_{IN} \in \mathbb{R}^{N \times C}$ is usually deactivated in practice.

**B. Window Normalization**

Our main inspiration comes from Ghost Batch Normalization (GBN) [38], which improves the model generalization by calculating mean and variance on small segments of the input batch. Intuitively, GBN can be viewed as adding noises to various segments of a batch, an equivalent form of feature-level data augmentation. As IN demonstrated its effectiveness for OOD generalization on medical images [39], we introduce WIN which injects noises into each instance rather than segments of a batch. This noise injection operation serves as a regularization technique for IN, aiming to enhance the feature robustness and expand feature distribution.

Specifically, perturbation (i.e., noise injection) for each instance is conducted through the feature standardization operation. We standardize the feature $f$ using approximations of global statistics (i.e., $\mu_{IN}$ and $\sigma_{IN}^{2}$) (see Eq. 1). As shown in Fig. 2 (c), the mean and variance for each channel and instance are calculated within a randomly selected window. Formally, for a window that is specified by the top-left coordinate $(\bar{x}, \bar{y})$ and bottom-right coordinate $(\tilde{x}, \tilde{y})$, the mean and variance, $\mu_{WIN}, \sigma_{WIN}^{2} \in \mathbb{R}^{N \times C}$ are computed as:

$$
\mu_{WIN}^{n,c} = \frac{1}{\bar{H} \cdot \bar{W}} \sum_{h=\bar{x}}^{\bar{H}} \sum_{w=\bar{y}}^{\bar{W}} f^{n,c,h,w},
$$

$$
\sigma_{WIN}^{2,n,c} = \frac{1}{\bar{H} \cdot \bar{W}} \sum_{h=\bar{x}}^{\bar{H}} \sum_{w=\bar{y}}^{\bar{W}} \left( f^{n,c,h,w} - \mu_{WIN}^{n,c} \right)^2.
$$

Equation (6)

where $\bar{H}$ and $\bar{W}$ are the height and width of a window. The Window sampling strategy, as shown in Algorithm 1, involves:
Algorithm 1 Window Sampling

Data: Features $F \in \mathbb{R}^{N \times C \times H \times W}$, Threshold $T$, Window ratio $\tau$, Window width $\bar{W}$, Window height $\bar{H}$, Window center $(x, y)$.

Result: Top-left of window $(\bar{x}, \bar{y})$, Bottom-right of window $(\bar{x}, \bar{y})$.

1 repeat
   // get a squared window
   Uniformly sample a window ratio $\tau \sim U(0.0, 1.0)$;
   $\bar{W} \leftarrow W \times \text{sqrt}(\tau)$;
   $\bar{H} \leftarrow H \times \text{sqrt}(\tau)$;
   Uniformly sample a window center $(x, y)$;
   // place the window
   $\bar{x} \leftarrow \text{int}(\max(\min(x - \bar{W}/2, 0), W))$;
   $\bar{y} \leftarrow \text{int}(\max(\min(y - \bar{H}/2, 0), H))$;
   $\bar{x} \leftarrow \text{int}(\max(\min(x + \bar{W}/2, 0), W))$;
   $\bar{y} \leftarrow \text{int}(\max(\min(y + \bar{H}/2, 0), H))$;

10 until $(\bar{x} - x) \times (\bar{y} - y) \geq T \times \bar{H} \times \bar{W}$;

repeatedly generating a window denoted as $(x, y, \bar{W}, \bar{H})$ and placing this window on the feature until the size of the random window exceeds $T \times \bar{H} \times \bar{W}$. $T$ is a threshold for the ratio of window size. Notably, LCN [34] also uses the statistics of windows, but it normalizes each feature based on the unique statistics computed within its local neighborhoods (including both spatial and channel dimensions). Besides, the purpose of LCN is to exploit the spatial information to address the inconsistency of input size between the training and testing. Consequently, its primary application is in dense prediction tasks such as object detection, semantic segmentation, and instance segmentation, where the input is a local region of the image.

Considering the perturbation introduced by above strategy is marginal for the images with a large blank area (e.g., chromosome images), we propose another strategy called Block. It computes $\mu_{WIN}$ and $\sigma^2_{WIN}$ within multiple small windows. Firstly, we divide the feature $f$ into $B$ non-overlapping blocks. Each block corresponds to a patch of input images. At a certain scale $s$, the $i$-th block is denoted as $\{\bar{x}_i^s, \bar{y}_i^s, \hat{x}_i^s, \hat{y}_i^s\}$. For example, if input images have a resolution of $224 \times 224$ and patch size is $32 \times 32$, they will be divided into 49 blocks regardless of the feature scales. Then, we sample $TB$ random blocks without replacement, following a uniform distribution. The mean and variance $\mu_{WIN}, \sigma^2_{WIN}$ at the scale $s$ are computed as:

$$
\mu_{WIN}^{n,c} = \frac{1}{(TB)(s^2\bar{H}\bar{W})} \sum_{i} \sum_{h=\bar{y}_i^s} \sum_{w=\bar{x}_i^s} f^{n,c,h,w},
$$

$$
\sigma^2_{WIN}^{n,c} = \frac{1}{(TB)(s^2\bar{H}\bar{W})} \sum_{i} \sum_{h=\bar{y}_i^s} \sum_{w=\bar{x}_i^s} (f^{n,c,h,w} - \mu_{WIN}^{n,c})^2,
$$

where $i$ is the index of random selected blocks and $\hat{H}$ and $\hat{W}$ are the original height and width of the patches. This strategy prevents zero variance (e.g., a window on the consistent background) effectively and provides appropriate perturbation to enhance the OOD generalization.

To diversify the perturbations and smooth the model’s response, $WIN$ mixes the local and global statistics. The mixing statistics can be formulated as:

$$
\mu = \lambda \odot \mu_{WIN} + (1 - \lambda) \odot \mu_{IN},
$$

$$
\sigma^2 = \lambda \odot \sigma^2_{WIN} + (1 - \lambda) \odot \sigma^2_{IN},
$$

where $\lambda \in \mathbb{R}^{N \times C}$ is a random instance-specific weight drawn from a Beta distribution $\text{Beta}(a, a)$.

Lastly, to keep the evaluation deterministic, we adopt $\mu = \mu_{IN}$ and $\sigma^2 = \sigma^2_{IN}$ during evaluation as above mixing statistics are approximations of global statistics (see Fig. 2). The mixing statistics become equivalent to global statistics when $T = 1.0$ or $\lambda = 0$.

C. Training and Evaluation Discrepancy

Normalizing the features with $WIN$ has regularized the CNNs training effectively and improved the OOD generalization significantly. However, the evaluation statistics cannot be precisely aligned to training statistics and the model parameters are trained to fit features normalized by $\mu_{WIN}$, $\sigma^2_{WIN}$, which may degrade model generalization. On the other hand, normalizing features with mixing statistics (in training mode) or global statistics (in evaluation mode) produces two correlated views of the same sample. Minimizing the discrepancy between training and evaluation is a natural self-learning task to incorporate invariance across different views [40], [41].

To these ends, we introduce the $WIN-WIN$, which requires twice forward passes. As shown in Fig. 2 (d), given the input $x$ with a ground-truth label $y$, the first pass uses mixing statistics to normalize the features and outputs a logits $\hat{y}$, while the second pass uses global statistics to normalize the features and outputs a logits $\hat{y}$. $WIN-WIN$ will compel the model to minimize the Kullback-Leibler divergence between $\hat{y}$ and $\hat{y}$ and the cross-entropy between $\hat{y}$ and $y$ and between $\hat{y}$ and $y$. The Kullback-Leibler divergence loss can be written as:

$$
\mathcal{L}_{KL} = \frac{1}{2} (D_{KL}(\text{softmax}(\hat{y})||\text{softmax}(\hat{y})) + D_{KL}(\text{softmax}(\hat{y})||\text{softmax}(\hat{y}))),
$$

where the logits are converted to a probability vector via a softmax function. Meanwhile, the cross-entropy loss can be written as:

$$
\mathcal{L}_{CE} = \frac{1}{2} (\mathcal{H}(\text{softmax}(\hat{y}), y) + \mathcal{H}(\text{softmax}(\hat{y}), y))).
$$

Finally, the total loss for classification tasks can be written as:

$$
\mathcal{L}_{Total} = \mathcal{L}_{CE} + \delta \mathcal{L}_{KL}
$$

where $\delta$ is used to balance $\mathcal{L}_{CE}$ and $\mathcal{L}_{KL}$. Although $WIN-WIN$ increases the computational complexity, it effectively addresses the training and evaluation discrepancy and complements $WIN$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The International Peace Maternity Child Health Hospital of Fudan University (RHH) and 51 samples from The Obstetrics Gynecology Hospital and the Second Affiliated Hospital of Zhejiang University, BY and ZR were collected from Peking University Third Hospital and the Second Affiliated Hospital of Zhejiang University, respectively. Furthermore, the experiments were also conducted on an OOD generalization benchmark, Camelyon17. It is a breast cancer detection benchmark of WILDS [2], including five data sources. All images in Camelyon17 were resized from 96 × 96 to 256 × 256 to accommodate window sampling. Fig. 3 (a) and (b) show two benchmarks’ example cases and sample numbers. Here, the OOD generalization evaluation metric we applied was the mean clean area under the curve (m-CUE), following [39]. A dataset was split into 80% for training and 20% for validation, while other datasets were held out for testing. Before aggregating, each test dataset’s result (i.e., the area under the curve) was recalibrated based on its inherent difficulties. A higher m-CUE means better OOD generalization. We conducted 5 runs for two tasks using different random seeds.

2) Multiclass Classification: As shown in Fig. 3 (c), we collected chromosome images from two hospitals using different microscopes, aiming to cross-evaluate OOD generalization performance in chromosome classification. Two datasets include 7,931 samples from The Obstetrics Gynecology Hospital of Fudan University (RHH) and 51,151 samples from The International Peace Maternity Child Health Hospital of China welfare institute (IPMCH). A dataset from one hospital was split into training and validation sets according to an 80% : 20% proportion, while another hospital data was used for testing. The chromosomes were pre-segmented and placed on a consistent white background, making these two medical image datasets typical in multiclass classification tasks in medical image analysis. The results were reported in average Top-1 accuracy (Acc.) across 5 runs. Besides, we conducted experiments on natural image datasets, including two commonly used robustness benchmarks, CIFAR-10-C and CIFAR-100-C [42], and a widely used domain generalization benchmark, Digit-DG [13]. CIFAR-10-C and CIFAR-100-C each consist of 75 corrupted versions of their original test sets. The metric for them is the mean corruption error (mCE) calculated across all 15 corruptions and 5 severities per corruption [42]. A lower mCE is better. Digit-DG consists of four handwritten digit recognition datasets (MNIST, MNIST-M, SVHN, and SYN) with distribution shifts for font style, stroke color, and background. Following prior DG works [12], we applied the leave-one-domain-out protocol for Digit-DG and reported the average Top-1 accuracy (Acc.) over 5 runs for each target dataset. Chromosome images and images of CIFAR-10-C and CIFAR-100-C were resized to 256 × 256 and images of Digit-DG were resized to 224 × 224.

3) Image Segmentation: We employed retinal fundus images from [4] for optic cup and disc (OC/OD) segmentation. As shown in Fig. 3 (c), they were collected from 4 sites using different scanners [4], and each image was center-cropped and resized to 384 × 384. We randomly split the images from each site into training and test sets in an 80% : 20% ratio. In our experiments, the training sets from three sites (e.g., Site A, B, and C) were used for training, and the test set from fourth site (e.g., Site D) was used for testing, following the DG setting. The segmentation results were reported in the average Dice coefficient (Dice) across 5 runs. A higher Dice indicates better.

B. Implementation Details
We conducted all experiments on two NVIDIA GeForce RTX 2080Ti with PyTorch implementation. All models were trained from scratch using a cross-entropy loss, along with standard training and data augmentation policies, to avoid reliance on complex engineering techniques. The number of training epochs was adaptively determined based on the task and dataset size. Following [12], the hyper-parameters of Beta distribution used in mixing statistics were empirically set to \( \alpha = 0.1 \). And, we set the ratio threshold \( T = 0.7 \) and \( \delta = 0.3 \) for all experiments. This was done to assess the generality of our method, although better results might be achievable with task-specific hyperparameter tuning. With the exception of chromosome classification, all tasks used the Window strategy. The Block strategy was adopted for chromosome classification due to the large blank areas in these pre-processed images. More detailed settings for each task are listed as follows.

1) Binary Classification: We optimized the ResNet-50 [45] with the following settings: a SGD optimizer with Nesterov momentum of 0.9; weight decay of 1e − 5; a batch size of 64. The learning rate was initially increased linearly from...
0 to $3e^{-3}$ over the first 5 epochs and then gradually decreased to 0 according to a cosine decay schedule. Each image was horizontally flipped with a 50% probability and randomly cropped to $224 \times 224$ before being fed into the model.

2) Multiclass Classification: For the chromosome classification task, we conducted the experiments with the same setting in Binary Classification. The CIFAR-10-C and CIFAR-100-C experiments were built on the codes of PIXMIX [9].

A SGD optimizer with Nesterov momentum of 0.9 was used to train the ResNet-18 [45]. The batch size was set at 64, and the learning rate was initially set to 0.3 and decayed according to a cosine decay schedule. The augmentation policy was identical to that in Binary Classification. In the experiments of Digit-DG, the ResNet-18 model was optimized for 180 epochs with the following settings: a SGD optimizer with Nesterov momentum of 0.9; weight decay of $5e^{-4}$; a batch size of 64; and a cosine decay schedule. The code was developed using

1https://github.com/andyzoujm/pixmix
C. Comparison With Baselines

Three baseline methods, BN, GN, and IN, are introduced as they are the most popular normalization methods. Each of these methods has demonstrated their effectiveness in improving model generalization. Since our experiments were conducted on two GPUs, the BN is equivalent to GBN [38] in practice. And, the default value of num_groups in GN is 32. However, to achieve the best results in CIFAR-10-C and CIFAR-100-C experiments, we increased this value to 64. In the following sections, we first compare WIN against BN, GN, and IN. Subsequently, we compare WIN-WIN with these normalization techniques.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
For the binary classification tasks, we aggregated all datasets of the same task and trained the DeepAll. DeepAll provides a difficulty coefficient and serves as the upper bound for each dataset [39]. Table I and II present the experimental results of two tasks, namely glaucoma detection and breast cancer detection. Compared to BN, WIN surpassed it for the OOD generalization remarkably (the averaged \(m-cAUC\) of +0.202 in glaucoma detection and the averaged \(m-cAUC\) of +0.085 in breast cancer detection). Compared to GN, WIN achieved an improvement of +0.112 and +0.093 in the average \(m-cAUC\) for glaucoma detection and breast cancer detection, respectively. Compared to IN, WIN led it with +0.053 \(m-cAUC\) for glaucoma detection and approached it with +0.002 averaged \(m-cAUC\) for breast cancer detection. Moreover, their results on in-distribution (IND) data showed a slight variation, and overall were comparable. Leveraging the advantage of WIN, WIN-WIN achieved the best results in both tasks, exceeding WIN by +0.009 \(m-cAUC\) for glaucoma detection and by an average of +0.034 in \(m-cAUC\) for breast cancer detection.

Multiclass classification is a more challenging task, yet the superiority of our methods remains evident. Table III presents the results for chromosome classification. This task requires models to classify each chromosome image into one of 24 types. Notably, a few images in RHH are mislabeled by beginners, leading to distribution shifts in both input images and labels. In Table III, RHH→IPMCH denotes training on RHH and testing on IPMCH, while IPMCH→RHH denotes the reverse setting. The accuracy of WIN excelled that of BN by 19.6%, that of GN by 13.6%, and that of IN by 5.8% in RHH→IPMCH. On another setting, the performance advantage of WIN appeared to decrease, possibly due to label noise. Specifically, we observed a +7.4% increase for BN, a −0.6% decrease for GN, and a +0.3% increase for IN. Meanwhile, WIN-WIN further improved WIN by +2.4% in RHH→IPMCH and +0.3% in IPMCH→RHH. Furthermore, we verified our methods on common benchmarks, CIFAR-10-C, CIFAR-100-C, and Digit-DG. As shown in Table IV, our methods effectively improved corruption robustness compared to the baselines. In the Digit-DG experiments, three domains were used for training and one domain for testing. Our methods greatly outperformed the baselines across all target domains (refer to Table V). WIN and WIN-WIN excelled the best baseline GN by the averaged Top-1 accuracy of +0.5% and +5.8%, respectively. This strongly demonstrates that our methods can improve the OOD generalization with multi-source domain data. Besides, both WIN and WIN-WIN exhibit comparable IND performance to these baseline methods.

In addition to classification tasks, we also investigated the application of WIN and WIN-WIN in segmentation task. Table VI presents that WIN achieved a better performance than BN, GN, and IN with an average dice score improvement of +0.038, +0.032, and +0.004 on the OC/OD segmentation, respectively. Moreover, WIN-WIN yields an additional improvement to the average dice score, resulting in an additional increase of +0.02. This demonstrates once again that our method can improve the OOD generalization with multi-source domain data.

In summary, WIN is a versatile normalization technique that significantly improves OOD generalization. It surpasses BN, GN, and IN across various tasks regardless of training setting (i.e., single source domain or multi-source domains). On this basis, WIN-WIN further enhances its effectiveness in improving OOD generalization.

### Ablation Analysis
In this section, we first examined the options for \(\mu\) and \(\sigma\) in WIN. Subsequently, we investigated the effect of the mechanisms underlying WIN-WIN. Finally, we presented an analysis of hyper-parameter sensitivity. All experiments in this section were performed using the ResNet-50 trained on the LAG dataset.

First of all, we removed the statistics mixing. Secondly, we discussed the choice of statistics \(\mu\) and \(\sigma\) in WIN. We employed the global mean \(\mu\) or global variance \(\sigma\) in the third and fourth row of Table VII, respectively. Thirdly, we investigated the area contributed to \(\mu\) and \(\sigma\) (see Fig. 4). The Global computes the mean and variance using all pixels, namely IN. Block and Pixel compute these statistics using randomly selected blocks or pixels, respectively. The Mask randomly sets a certain region to zero and then computes the \(\mu\) and \(\sigma\). Lastly, we perturbed the global statistics in another way, adding the speckle noise (namely the Speckle) [1]. As shown in Table VII, the following conclusions can be drawn: 1) Statistics mixing is advantageous. 2) \(\sigma_{WIN}\) contributes more than...
Window
3) Using local statistics calculated in µZHOU et al.: SIMPLE NORMALIZATION TECHNIQUE USING WINDOW STATISTICS 2095 IN performs using global statistics (i.e., of Table VII, respectively. As shown in Table VII, each of noise does not bring gains in OOD generalization. perturbing the statistics with the mask operation or speckle we recommend configuring T 0
Fig. 5 indicates that setting T at 0.7 and δ at 0.3 yields the best results. Consequently, we employed these settings for all experiments to showcase the generality of our method. It’s worth noting that the optimal selection of T and δ may vary for different tasks, influenced by factors like data size and model complexity. Achieving better results may be possible after tuning these hyper-parameters.

E. Comparison With State-of-the-Arts
A comprehensive comparison with state-of-the-art methods is presented in Table I. All results are reported using a ResNet-50 trained for the glaucoma detection task.

We first compared our methods with several normalization methods. Among these methods, IN [18], BN [19], and GN [29] are the most popular normalization methods. Additional methods, such as LCN [34], SNR [22], BIN [24], SN [23], IBN-a [20], and IBN-b [20], are extensions IN, while RBN [32] and CNSN [21] are recently proposed methods aimed at enhancing model generalization. Overall, these normalization methods did not present any obvious advantages for either IND or OOD generalization, and our methods significantly outperformed them. It is worth noticing that LCN and CNSN are most closely related to our method. LCN normalizes each feature with the unique statistics of its neighbors. CNSN proposes the crossnorm, which involves exchanging local and global statistics between channels. Although local statistics are used in LCN and CNSN, WIN surpassed them considerably (+0.103 m-cAUC for LCN and +0.215 m-cAUC for CNSN) and this advantage further extended by WIN-WIN.

On the other hand, our methods are also relevant to augmentation-based methods. The comparison with these methods has been conducted using the ResNet-50 with BN, including image-space augmentation (i.e., AutoAugment [10], RandAugment [11], Patch Gaussian [27], randomErasing [43], mixup [7], CutMix [44], AugMix [8], and PIXMIX [9]) and feature-space augmentation (i.e., manifold mixup [28] and MixStyle [12]). As shown in Table I, these methods showed their superiority on IND data, but in terms of OOD generalization, their performance lagged significantly behind our methods. Specifically, our method outperformed the best augmentation method, RandAugment, by +0.114 and +0.123 m-cAUC for WIN and WIN-WIN, respectively.

Furthermore, it should be noted that SNR, IBN-a, IBN-b, CNSN, Patch Gaussian, AugMix, PIXMIX, and MixStyle are state-of-the-art methods in domain generalization and adaptation or model robustness, aimed at improving the OOD

\[ \mu_{WIN} \]

but the best practice is to combine them together.

3) Using local statistics calculated in Window or Block outperforms using global statistics (i.e., IN). 4) Compared to IN, perturbing the statistics with the mask operation or speckle noise does not bring gains in OOD generalization.

In WIN-WIN, we removed the statistics mixing and consistency constraint, as indicated in the first and eleventh rows of Table VII, respectively. As shown in Table VII, each of these modifications contributed to OOD generalization. The best practice is to combine them together since two mechanisms complement each other well. Furthermore, we imposed the consistency constraint to the features extracted from the penultimate layer using the NT-Xent loss [40]. This design led to degraded results, possibly because the NT-Xent loss eliminated crucial discriminative information.

Fig. 5 shows the results of different hyper-parameters settings. The m-cAUC was primarily influenced by the window ratio threshold T, as it controls the extent of perturbation applied to the feature. A smaller T results in more significant perturbations, whereas a larger T brings milder perturbations and increases the time needed for window sampling. There exists a reasonable range for the T. According to Fig. 5, we recommend configuring T within the range of 0.3 to 0.7 for optimal results. As the model optimizes, ̂y and ̂y can serve as approximations to the ground-truth y. Consequently, the objectives of LKL and LCCE are similar, leading to a relatively minor impact of δ to final results (see Eq. 9 and 10). Fig. 5 indicates that setting T at 0.7 and δ at 0.3 yields the

| Methods | Strategies | Stat. Mixing | Consist. | m-cAUC↑ |
|---------|------------|--------------|----------|---------|
| Window | \( \mu, \sigma \) | ✓ | x | 0.877±0.01 |
| Window | \( \mu, \sigma \) | x | x | 0.866±0.00 |
| Window | \( \mu \) | x | x | 0.874±0.01 |
| Window | \( \sigma \) | x | x | 0.825±0.02 |
| Global (IN) | \( \mu, \sigma \) | x | x | 0.824±0.02 |
| Block | \( \mu, \sigma \) | x | x | 0.845±0.01 |
| Pixel | \( \mu, \sigma \) | x | x | 0.808±0.02 |
| Mask | \( \mu, \sigma \) | x | x | 0.794±0.03 |
| Speckle | \( \mu, \sigma \) | x | x | 0.795±0.02 |
| WIN-WIN | \( \mu, \sigma \) | ✓ | logits | 0.883±0.01 |
| WIN-WIN | \( \mu, \sigma \) | x | logits | 0.871±0.01 |
| WIN-WIN | \( \mu, \sigma \) | ✓ | features | 0.850±0.01 |

The p-values are \( p_{1-2} = 4.84 \times 10^{-2}, p_{1-3} = 4.81 \times 10^{-3}, p_{1-4} = 2.03 \times 10^{-3}, p_{1-5} = 1.73 \times 10^{-4}, p_{1-6} = 4.73 \times 10^{-4}, p_{1-7} = 1.10 \times 10^{-4}, p_{1-8} = 2.10 \times 10^{-4}, p_{1-9} = 6.81 \times 10^{-5}, p_{1-10} = 4.77 \times 10^{-5}, p_{1-11} = 4.31 \times 10^{-2}, p_{1-12} = 5.40 \times 10^{-4} \). All p-values are < 0.05.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
generalization with a single source domain. Our methods have significantly outperformed these approaches also.

V. DISCUSSIONS

In practice, the training data represents a small population of real data, which is smaller in medical images due to the expensive costs for data acquisition and the high diversity caused by operating procedures and imaging equipment. Thereby, the discrepancy between training data and test data from unseen deploy environments is prevailing exists and leads to CNN model’s performance dips. In this paper, we mainly focus on improving the OOD generalization on heterogeneous data, which has the same labels but different appearances. A simple yet effective normalization technique WIN is proposed to address this problem with free parameters and a negligible effect for IND data. WIN conducts feature-level augmentation by perturbing the normalizing statistics with stochastic window statistics. Utilizing this feature-level augmentation technique, we propose a novel self-distillation scheme, WIN-WIN, to eliminate the train-test inconsistency and further enhance the OOD generalization. WIN-WIN can be easily implemented with twice forward passes and a consistency constraint. Despite the increased computational complexity, WIN-WIN exhibits steady improvement in model generalization. Extensive experiments have demonstrated our methods significantly and steadily boost the OOD generalization across 6 tasks spanning 24 datasets.

Here, we present a comprehensive investigation of the properties of the WIN, covering the impacts on IND data and OOD data, the model convergence, and the time costs. To ensure reproducibility and provide insight for subsequent studies, we conducted the experiments on the widely-used OOD generalization benchmark, CIFAR-10-C [42]. Fig. 6 and 7 (a) show the t-SNE visualization and the averaged cosine similarity between the raw image and its corrupted versions using the penultimate layer feature of ResNet-18, respectively. In terms of IND performance and class separation, IN and WIN outperformed BN and GN. Moreover, WIN exhibited superior feature invariance compared to these normalization techniques. In addition, we analyzed the loss curves and time costs for BN, GN, IN, and WIN. Fig. 7 (b) and (c) show that the model convergence rate and stability of WIN are better than those of GN and comparable to those of BN and IN. Regarding the time costs, we introduced two schemes: WIN (offline) and WIN (online). WIN (offline) caches the window parameters, while WIN (online) generates them on-the-fly. According to Fig. 7 (d), WIN (offline) greatly reduced the time cost but was still higher than 90.5% for BN, 41.8% for GN, and 57.7% for IN. Notably, although the experiments in this paper adopted WIN (online), their results did not differ significantly. In summary, despite the longer training time, WIN can serve as a viable alternative to existing normalization techniques, effectively eliminating the distribution gap and improves the OOD generalization.

Despite its significant improvement in OOD generalization, WIN has a limitation: window sampling is inefficient for inputs with a small size. Fortunately, this is no longer a limitation for existing CNNs. Since large input sizes are known to improve model generalization [46], existing CNNs commonly use a large input size. Lastly, as a variant of IN, a reasonable conjecture is that the advantages of IN are also retained in WIN. Therefore, in future work, we plan to apply WIN to the pixel2pixel tasks (e.g., super-resolution and style transfer) and explore its compatibility with existing techniques. Furthermore, WIN-WIN does not yield improvements over WIN in some corner cases, such as the experiment on H4 of the breast cancer detection task. Thus, we intend to investigate its applicability across a diverse range of tasks in our future work.

VI. CONCLUSION

In this paper, we propose a simple yet effective normalization technique, namely WIN, which uses stochastic local statistics to boost the OOD generalization and not sacrifice the IND generalization. Based on WIN, we propose the WIN-WIN to further improve OOD generalization, which can be
implemented with only a few lines of code. Our methods gracefully address the OOD generalization of heterogeneous data in real-world clinical practice. Extensive experiments on various tasks and datasets demonstrated their generality and superiority compared with the baselines and many state-of-the-art methods.

**ACKNOWLEDGMENT**

The authors would like to thank Songchang Chen, Chenming Xu, Chun Zhang, and Juan Ye for their contributions for data. This study utilized the chromosome data from the Obstetrics Gynecology Hospital, Fudan University and the fundus data from Peking University Third Hospital and the Second Affiliated Hospital, Zhejiang University.

**REFERENCES**

[1] D. Hendrycks et al., “The many faces of robustness: A critical analysis of out-of-distribution generalization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 8320–8329.

[2] P. W. Koh et al., “Wilds: A benchmark of in-the-wild distribution shifts,” in Proc. IJCAI, 2021, pp. 5637–5664.

[3] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, “Measuring robustness to natural distribution shifts in image classification,” in Proc. NeurIPS, vol. 33, 2020, pp. 18383–18399.

[4] S. Wang, L. Yu, K. Li, X. Yang, C.-W. Fu, and P.-A. Heng, “DoFE: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets,” IEEE Trans. Med. Imag., vol. 39, no. 12, pp. 4237–4248, Dec. 2020.

[5] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, “FedDrn: Federated learning on non-IID features via local batch normalization,” in Proc. Int. Conf. Learn. Represent., 2021, pp. 1–16.

[6] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. MICCAI, 2015, pp. 234–241.

[7] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Represent., 2018.

[8] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “Augmix: A simple data processing method to improve robustness and uncertainty,” in Proc. Int. Conf. Learn. Represent., 2019.

[9] D. Hendrycks et al., “PixMix: Dreamlike pictures comprehensively improve safety measures,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 16762–16771.

[10] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation strategies from data,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 113–123.

[11] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 3008–3017.

[12] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, “Domain generalization with mixstyle,” in Proc. Int. Conf. Learn. Represent., 2020.

[13] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, “Learning to generate novel domains for domain generalization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Glasgow, U.K., Aug. 2020, pp. 561–570.

[14] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales, “A simple feature augmentation for domain generalization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 8866–8875.

[15] R. H. H. M. Philipsen, P. Madaskar, L. Hogeweg, J. Melendez, C. I. Sánchez, and B. van Ginneken, “Localized energy-based normalization of medical images: Application to chest radiography,” IEEE Trans. Med. Imag., vol. 34, no. 9, pp. 1965–1975, Sep. 2015.

[16] S. Gündel, S. Grbic, B. Georgescu, S. Li, A. Maier, and D. Comaniciu, “Learning to recognize abnormalities in chest X-rays with location-aware dense networks,” in Proc. Iberoam. Congr. Pattern Recognit., in Lecture Notes in Computer Science, vol. 11401. Cham, Switzerland: Springer, 2019, pp. 757–785, doi: 10.1007/978-3-030-13469-3_88.

[17] D. Tellez et al., “Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology,” Med. Image Anal., vol. 58, Dec. 2019, Art. no. 101544.