ScoreMix: A Scalable Augmentation Strategy for Training GANs With Limited Data

Jie Cao, Mandi Luo, Junchi Yu, Ming-Hsuan Yang, Fellow, IEEE, and Ran He

Abstract—Generative Adversarial Networks (GANs) typically suffer from overfitting when limited training data is available. To facilitate GAN training, current methods propose to use data-specific augmentation techniques. Despite the effectiveness, it is difficult for these methods to scale to practical applications. In this article, we present ScoreMix, a novel and scalable data augmentation approach for various image synthesis tasks. We first produce augmented samples using the convex combinations of the real samples. Then, we optimize the augmented samples by minimizing the norms of the data scores, i.e., the gradients of the log-density functions. This procedure enforces the augmented samples close to the data manifold. To estimate the scores, we train a deep estimation network with multi-scale score matching. For different image synthesis tasks, we train the score estimation network using different data. We do not require the tuning of the hyperparameters or modifications to the network architecture. The ScoreMix method effectively increases the diversity of data and reduces the overfitting problem. Moreover, it can be easily incorporated into existing GAN models with minor modifications. Experimental results on numerous tasks demonstrate that GAN models equipped with the ScoreMix method achieve significant improvements.

Index Terms—Generative adversarial networks, image synthesis, data augmentation, few-shot image-to-image translation

1 INTRODUCTION

In recent years, Generative Adversarial Networks (GANs) [1] have shown much progress in numerous image synthesis tasks. However, current GAN-based methods [2], [3], [4] heavily rely on vast quantities of training data. Large-scale training datasets, e.g., the CelebA [5] and FFHQ [6] datasets, are necessary for these methods to achieve state-of-the-art results. When the amount of data is limited, recent findings [7], [8] reveal that GANs easily overfit the training set. This issue leads to drastically degraded results in scenarios where collecting sufficient training data is infeasible. Hence, improving the generalization of GANs to the data-limited regime is practically important to the image synthesis tasks.

To tackle the overfitting problem, some recent approaches develop data augmentation schemes [7], [8], [9], [10] tailored for GANs. These schemes use a set of hand-crafted image processing operations (e.g., random cropping, scaling, and color jittering) to augment the training samples. They combine these operations with adaptive [7] or automatic [8] strategies.

- Jie Cao, Mandi Luo, Junchi Yu, and Ran He are with the CRIPAC & NLPRI, Institute of Automation, Chinese Academy of Sciences, Beijing 100045, China, and also with the University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: jie.cao@cripac.is.ac.cn, luo.mandi2019, ran.he@ia.ac.cn, junchiyu@163.com.
- Ming-Hsuan Yang is with the University of California at Merced, Merced, CA 95343 USA. E-mail: mhyang@ucmerced.edu.

Manuscript received 28 November 2021; revised 19 October 2022; accepted 17 December 2022. Date of publication 23 December 2022; date of current version 5 June 2023.

This work was supported in part by the National Natural Science Foundation of China under Grants 6220073425, U21B2045, and U20A20223. The work of Ming-Hsuan Yang was supported by NSF CAREER under Grant 1149783.

(Corresponding author: Ran He.)

For some specific tasks [8], these methods can train GANs effectively with hundreds of samples, allowing decent performances.

However, the current methods have some intrinsic limitations: First, these methods are difficult to generalize across datasets. The reason is that domain knowledge is integrated into the hand-crafted augmentation operations of these methods. Such knowledge is valid only for some specific data. For instance, the color distortion strategies for horse images are not suitable for zebra images. Second, these methods include a large number of hyperparameters that require fine-tuning on specific training data. Note that unsuitable choices of the hyperparameters are even harmful to model training [7], [11]. Fine-tuning these hyperparameters, however, is prohibitively expensive due to the large search space. To reduce the computational burden, current methods assign fixed values to most of the hyperparameters based on their experience. Due to these issues, training GANs with limited data in real-world scenarios remains challenging.

To facilitate training GANs with limited data, we propose a mixing-based augmentation strategy named ScoreMix. As illustrated in Fig. 1, given a data distribution $p_{data}$, we first take a convex combination of two training samples as the mixed sample $x'$. Then, we optimize $x'$ by minimizing the L2 norm of its Stein score [12]. The Stein score of a sample, denoted as $\nabla \log p_{data}(x)$, is the gradients of the logarithmic distribution function with respect to data. During the optimization process, the mixed sample $x'$ gradually moves towards the manifold of data distribution. Under mild assumptions, for the solution $x^*$, the density probability $p_{data}(x^*)$ has the maximal value in some local neighborhood around the mixed sample $x'$. This means the augmented sample $x^*$ has the maximum likelihood that belongs to the data distribution.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We present a score-based data augmentation method that proposes a chain of augmentation operations and enforces that the augmented samples diversify training data without perturbing the data distribution. Our method reduces the overfitting problem, facilitating the training of GANs with limited data. The augmentation; Krizhevsky et al. [12] use the affine transformations, including translation, scaling, and shearing, in handwritten character recognition; Bengio et al. [28] introduce the transformations that degrade images (e.g., Gaussian noise, salt and pepper noise, and motion blur) for augmentation; Krizhevsky et al. [15] apply random cropping, horizontal flipping, and color jittering to train deep networks. Using these content-preserving transformations has become a routine data pre-processing step.

The strategies mentioned above can effectively diversify the training data, but the hyperparameters require data-specific tuning, which is non-trivial in practice. Note that an unsuitable augmentation setting even hurts model performance [7], [11]. Some methods [24], [25], [26], [27] propose to automatically search the optimal hyperparameters for generic augmentation operations. However, these methods are prohibitively expensive in many practical settings.

The mixing-based augmentation methods [19], [20], [21], [28], [29], [30], [31], [32], [33], [34] use the combinations of real samples to generate virtual training samples. These methods can regularize the models from the overfitting problem and improve generalization. The mixup method [19] is one of the most widely used approaches, which uses linear interpolations for augmentation. This method is easy to implement and requires negligible computation. However, the augmented samples look unnatural and exhibit visible artifacts, which may confuse the training model. Recent approaches have proposed various improved mixing-based methods. Guo et al. [33] present an adaptive mixing policy to reduce the generation of degraded data. The CutMix method [20] augments samples by cutting and pasting image patches within a data batch. The PuzzleMix method [31] proposes to search for an optimal mask, which reveals the most salient objects in the real images, for data mixing. Dabouei et al. [30] present the SuperMix method, which uses the knowledge from a teacher network for data augmentation. The AugMix method [21] proposes a chain of augmentation operations and enforces that the augmented image is similar to the original one. Though the existing mixing-based methods have demonstrated notable improvements over the preliminary ones, they are in three-folds: (1) The proposed data augmentation method can be applied to general image synthesis tasks. The previous approach is effective only for the image-to-image translation tasks. (2) This work proposes to produce augmented samples with score-based optimization. Moreover, we build a score estimation network based on multi-scale denoising score matching. This approach ensures that the augmented samples are close to the data manifold. In contrast, the previous approach produces augmented samples by linearly interpolating the real training samples. (3) Our approach does not rely on specific designs of the underlying GAN model. Thus, our approach facilitates the training of various GANs. The previous approach, however, requires a predefined content-preserving loss for augmentation.

2 RELATED WORK

2.1 Data Augmentation

Numerous methods have been developed to increase the amount of data for training deep learning models without overfitting. To name a few, Lecun et al. [14] use the affine transformations, including translation, scaling, and shearing, in handwritten character recognition; Bengio et al. [23] introduce the transformations that degrade images (e.g., Gaussian noise, salt and pepper noise, and motion blur) for augmentation; Krizhevsky et al. [15] apply random cropping, horizontal flipping, and color jittering to train deep networks. Using these content-preserving transformations has become a routine data pre-processing step.

The strategies mentioned above can effectively diversify the training data, but the hyperparameters require data-specific tuning, which is non-trivial in practice. Note that an unsuitable augmentation setting even hurts model performance [7], [11]. Some methods [24], [25], [26], [27] propose to automatically search the optimal hyperparameters for generic augmentation operations. However, these methods are prohibitively expensive in many practical settings.

The mixing-based augmentation methods [19], [20], [21], [28], [29], [30], [31], [32], [33], [34] use the combinations of real samples to generate virtual training samples. These methods can regularize the models from the overfitting problem and improve generalization. The mixup method [19] is one of the most widely used approaches, which uses linear interpolations for augmentation. This method is easy to implement and requires negligible computation. However, the augmented samples look unnatural and exhibit visible artifacts, which may confuse the training model. Recent approaches have proposed various improved mixing-based methods. Guo et al. [33] present an adaptive mixing policy to reduce the generation of degraded data. The CutMix method [20] augments samples by cutting and pasting image patches within a data batch. The PuzzleMix method [31] proposes to search for an optimal mask, which reveals the most salient objects in the real images, for data mixing. Dabouei et al. [30] present the SuperMix method, which uses the knowledge from a teacher network for data augmentation. The AugMix method [21] proposes a chain of augmentation operations and enforces that the augmented image is similar to the original one. Though the existing mixing-based methods have demonstrated notable improvements over the preliminary approaches, there is still room for improvement.
improvements, most methods generate augmented samples that drift off the data manifold. Using such augmented samples might lead to degraded image quality for the image synthesis tasks. To address this issue, we propose to generate the mixed data with the maximum likelihood that belongs to the real data distribution.

### 2.2 Data-Efficient GANs

GANs [1] are one of the most popular generative models. This method has been successfully applied to many computer vision applications. In particular, the GAN-based methods [35], [36], [37], [38] significantly improve the visual quality and diversity of the synthesized images. The state-of-the-art GAN models [3], [4], [6] yield impressive synthesis results on large-scale datasets. However, a fundamental challenge exists: the generalization ability and training stability of GANs are limited, especially in the data-limited regime.

These issues can be mitigated by collecting a large amount of data. But this can be prohibitively expensive or implausible. Hence, some efforts [7], [8], [9], [10], [39], [40] have recently been made to augment training data for GANs. Karras et al. [7] propose an adaptive data augmentation scheme for the discriminator. During training, this scheme controls the augmentation strength based on the degree of overfitting. The DiffAug method [8] introduces differentiable augmentation operations, which regularize GANs and improve the training stability. Tran et al. [10] propose an augmentation framework with invertible transformations. They also provide theoretical analysis to prove this approach preserves the Jensen-Shannon divergence of original GANs. These methods reduce the overfitting problem and render some plausible results. However, they use data-specific augmentation operations. Recent studies [7], [8], [11] show that these methods are sensitive to the choices of datasets, training methods, and network architectures. In this work, we propose a scalable data augmentation method for GAN training. The proposed augmentation method does not require data-specific tuning of hyperparameters. For different image synthesis tasks, we retrain the score estimation network, which guides the augmentation process.

The adaption-based approaches [41], [42], [43], [44], [45], [46], [47] train GANs with external datasets as an alternative to the target dataset. Liu et al. [41] first learn a semantically related generation model and then adapt it to the target domain. To deal with the content loss problem, Saito et al. [42] propose the content-conditioned style encoder. Moreover, several adaption-based methods focus on generating specific images, including faces [43], [44], scenes [45], human bodies [46], [47], etc. Despite the effectiveness, these approaches require additional image collection. They also assume that the external dataset and target dataset share a great similarity. In this paper, the proposed method trains GANs without any additional training data.

A recent work by Tseng et al. [17] proposes the LeCam-divergence (LC) regularization to improve the generalization of GANs trained with limited data. They impose the LC regularization on the training objective of the discriminator. The strategy can be easily added to the GAN training pipeline. While in this work, we propose to improve the training of GANs by augmenting the data, which is orthogonal to the LC regularization. We show that applying the LC regularization to the training process of our method brings additional improvements (later in our experiments).

Some recent approaches [48], [49], [50], [51], [52], [53] investigate the one-shot image synthesis task (a.k.a., single image generation). These methods learn a generation model from one sample for each image domain. The one-shot generation methods typically compute the internal statistics of patches within the training sample for model training. Several methods use domain knowledge and solve specific one-shot tasks, e.g., texture synthesis [49] and image super-resolution [48]. Towards generalized one-shot image synthesis, Shocher et al. [50] present InGAN, a unified model for various tasks and data types. Shaham et al. [51] train a pyramid of GANs using a multi-scale pipeline. They produce synthesized results with a coarse-to-fine model inference. Hinz et al. [52] propose a rescaling approach to reduce training time and a refinement method to improve image quality. To encourage content preservation, the SinCUT method [53] introduces a path-based contrastive learning scheme. In this work, the proposed data augmentation method is powerfully complementary to these approaches. Using the proposed method renders better results in the one-shot image synthesis tasks in our experiments.

### 2.3 Score-Based Generative Model

The score-based generative models [54], [55], [56] represent data distributions through scores, the vector fields which reflect the gradients of the likelihood of data. Similar to GANs, these models can produce high-quality images. They have proven highly effective in multiple image synthesis tasks. Song and Ermon [54] propose to train a neural network to estimate the scores of data. Then, they sample synthesized images using Langevin dynamics. Following this line of research, Song and Ermon [55] later improve the sampling process with a set of techniques that effectively estimate the noise scales from training data. Ho et al. [57] model the image sampling procedure as a reverse diffusion process. Then, Song et al. [56] integrate previous methods into a unified framework with stochastic differential equations. In this work, we train a deep score estimation model for data augmentation. Note that in the sampling process, we use the mixed image as initialization and produce the synthesized image with gradient-based optimization. These differences distinguish our approach from the discussed methods.

### 3 Proposed Method

The remainder of this section is organized as follows. Section 3.1 presents the overall data augmentation pipeline. In Section 3.2, we introduce the proposed score-based augmentation function. In Section 3.3, we describe how to train the score estimation network. Section 3.4 provides a comprehensive comparison between our approach and existing mixing-based methods.

#### 3.1 Overall Augmentation Pipeline

GANs [1] aim to model a data distribution \( p_{\text{data}} \) with a generator and a discriminator. GAN training is a two-player
game: the generator learns to generate samples that resemble the real samples, whereas the discriminator learns to discriminate between the real and fake samples. Formally, the objective function of GANs can be formulated as

$$
\min_G \max_D E_x[\log D(x)] + E_z[\log (1 - D(G(z)))]
$$

where \( x \in \mathbb{R}^d \) is drawn from \( p_{\text{data}} \), and \( z \) is drawn from a prior distribution. We use \( G \) and \( D \) to denote the generator and discriminator, respectively. Given a small training dataset, the discriminator \( D \) is prone to overfit the real samples. If this happens, the GAN training will collapse quickly since \( D \) does not provide useful supervision.

To address this issue, we propose a mixing-based data augmentation strategy. For unconditional GANs, as illustrated in Fig. 2a, we create augmented samples using the combinations of real samples. \( D \) regards the augmented samples as the ground truth data during training. When \( G \) conditions on the real sample, as shown in Fig. 2b, we augment the input data for \( G \). We use this strategy for practical applications such as image-to-image translation tasks.

The core component of the augmentation strategy is the mixing function. It increases the diversity of training data and prevents overfitting. Recall that one of the most classical mixing-based augmentation strategies is the mixup method [19], which uses the linear interpolation of samples. Concretely, given two real samples \( x_1, x_2 \sim p_{\text{data}} \), its mixing function can be formulated as

$$
x' = f_{\text{mixup}}(x_1, x_2) = \lambda \cdot x_1 + (1 - \lambda) \cdot x_2,
$$

where \( \lambda \in [0, 1] \) is the interpolation weight sampled from the Beta distribution, i.e., \( \lambda \sim \text{Beta}(\alpha, \alpha) \) with \( \alpha \in (0, +\infty) \) being a hyperparameter. As mentioned in Section 2.1, the mixup method may introduce characteristic image artifacts, leading to manifold intrusion. In the following, we introduce the proposed mixing function to address this issue.

### 3.2 ScoreMix Augmentation

We first introduce the notion of the Stein score. According to Stein’s method [12], given a smooth data distribution \( p_{\text{data}} \) supported on the set of real numbers, we define the Stein score of a sample \( x \sim p_{\text{data}} \) as follows:

$$
S_{p_{\text{data}}}(x) := \nabla_x p_{\text{data}}(x)/p(x) = \nabla_x \log p_{\text{data}}(x).
$$

Equation (3) shows that the Stein score can be interpreted as the gradients of the logarithmic distribution function with respect to data. Here we assume that a deep neural network \( S_0 \) can estimate the Stein score (discussed later in Section 3.3). Given two real samples \( x_1 \) and \( x_2 \), we compute the initial value \( x' \) using the mixup method [19] by Equation (2). Then, we search for \( x' \) with the following objective:

$$
x^* = \arg \min_{x \in f(\lambda|x)} \|S_0(x)\|_2^2 := f_{\text{scoremix}}(x'; S_0),
$$

where \( \|\cdot\|_2^2 \) denotes the euclidean L2 norm. We use the gradient descent method to solve Equation (4). The solution \( x' \) is a stationary point of the logarithmic distribution function \( \log p_{\text{data}}(x) \). This implies \( x' \) is also a stationary point of the distribution function \( p_{\text{data}}(x) \). Since \( x' \) produced from the mixup method is close to the manifold of \( p_{\text{data}} \), \( x' \) is a local maximum almost surely. Hence, the augmented sample \( x' \) has the maximum likelihood that belongs to \( p_{\text{data}} \) in a local neighborhood.

In Equation (4), the initial value \( x' \) provides a strong image prior, which prevents the score-based model from producing samples that deviate from the real samples. This is particularly useful when the training data is insufficient. Moreover, using \( x' \) as initialization accelerates the augmentation process. This allows the score estimation network to optimize an augmented sample from an intermediate result, which is significantly faster than the initialization with random noise in the existing methods (e.g., [54], [55], [56]).

The proposed ScoreMix augmentation follows the Vincinal Risk Minimization principle [59]. Similar to the mixup method [19], given a training data set \( \mathcal{D} \), we optimize the empirical vicinal risk. Concretely, we define the vicinity distribution \( \nu \) as follows:

$$
\nu(x'|x) = \frac{1}{n} \sum_{j=1}^{n} E_{\lambda} \left[ \delta(x' = f_{\text{scoremix}}(\lambda \cdot x_j + (1 - \lambda) \cdot x_j; S_0)) \right]
= \frac{1}{n} \sum_{j=1}^{n} E_{\lambda} \left[ \delta(x' = f_{\text{scoremix}}(x'; S_0)) \right],
$$

where
where $\delta(x' = x)$ is a Dirac mass centered at $x$. Compared with the mixup method, our approach produces augmented samples closer to the training data distribution. This is because $\propto \text{score}_{\text{mix}}$ produces a local maximum of the distribution, which implies that $p_{\text{data}}(x') \geq p_{\text{data}}(x)$.

Algorithm 1 shows the main steps to augment training data with the proposed method. We use Equation (2) to compute initial values and then use Equation (4) to push the initialized augmented samples towards the real data manifold. The proposed augmentation method can be plugged into existing GAN-based methods without changing the underlying algorithms.

Algorithm 1. The ScoreMix Augmentation Method

**Input:** Image samples $x_i$ and $x_j$;
Score estimation network $S_\theta$;
Learning rate $\eta$, optimization step $N$;
Distribution $\text{Beta}(\alpha, \alpha)$.

**Output:** Augmented sample $x'$. 

1. Sample interpolation weight $\lambda \sim \text{Beta}(\alpha, \alpha)$;
2. Compute the mixed sample $x' = \lambda \cdot x_i + (1 - \lambda) \cdot x_j$;
3. Initialize $x \leftarrow x'$, and $i \leftarrow 0$;
4. while $i < N$ do
5. Calculate loss $L(x') = \|S_\theta(x')\|^2_2$;
6. Calculate gradient $\nabla_x L(x')$;
7. $x' \leftarrow x' - \eta \cdot \nabla_x L(x')$;
8. $i \leftarrow i + 1$;
9. end
10. return $x'$

### 3.3 Score Estimation Network

We build a score estimation network $S_\theta$ to estimate the Stein score $S_{\text{data}}(x)$. Following the idea of score matching [60], we do not estimate $p_{\text{data}}(x)$ first. Instead, we directly optimize the network parameters $\theta$ by minimizing

$$
E_x \left[ \|S_\theta(x) - \nabla_x \log p_{\text{data}}(x)\|_2^2 \right],
$$

where we have $x \in \mathbb{R}^n$ drawn from $p_{\text{data}}$. However, in Equation (6), the term $\nabla_x \log p_{\text{data}}(x)$ is computationally intractable for real-world data distributions. To address this problem, we use the denoising score matching (DSM) method [13], which is robust in low-density data regions. The DSM method perturbs training samples with a prior distribution whose supports span the whole image space $\mathbb{R}^n$. This strategy ensures that the perturbed training data distribution has full support over $\mathbb{R}^n$. Hence, the score-based model gets stable training signals [60] even in low-density data regions.

Concretely, we perturb $x$ with Gaussian noise $N(0, \sigma^2)$ which has a zero mean and variance $\sigma^2$. Then, we use $S_\theta$ to estimate the Stein score of the perturbed sample $\tilde{x}$. Let $q_\sigma$ denote the perturbed data distribution, and the objective can be formulated

$$
E_\tilde{x} \left[ \|S_\theta(\tilde{x}) - \nabla_\tilde{x} \log q_\sigma(\tilde{x} | x)\|_2^2 \right],
$$

where we have $\tilde{x} \sim q_\sigma(\tilde{x} | x) = N(\tilde{x}, \sigma^2)$. The objective is tractable since we have $\nabla_\tilde{x} \log q_\sigma(\tilde{x} | x) = -\frac{1}{\sigma^2} \cdot (\tilde{x} - x)$. This immediately implies that we can rewrite the objective as

$$
E_\tilde{x} \left[ \|S_\theta(\tilde{x}) - \frac{\tilde{x} - x}{\sigma^2}\|_2^2 \right].
$$

We could minimize Equation (8) to train the estimation network. The optimal solution $S_\theta(x)$ converges to $\nabla_x \log p_{\text{data}}(x)$ as long as the noise variance $\sigma^2$ is small enough [13]. To train the estimation network $S_\theta$ more efficiently, we propose to perturb the data samples with multi-scale Gaussian noises. Specifically, the variances of the multi-scale noises are a geometric sequence $\{\sigma_n\}_{n=1}^N$, and we have

$$
\sigma_n = \sigma_1 \cdot \gamma^{n-1},
$$

where $\gamma$ denotes the common ratio, and $n \in [0, N]$. Without loss of generality, we assume that $0 < \gamma < 1$. Then, $\sigma_1$ has the maximal value, and $\sigma_N$ has the minimal value. On the one hand, the perturbed sample with the minimal noise scale is close enough to the real sample. We can regard $\nabla_\tilde{x} \log q_\sigma(\tilde{x} | x)$ as the Stein score of $x$ with negligible estimation error. On the other hand, the Stein score of the perturbed samples with a larger noise scale is easier to estimate. Hence, these samples help to stabilize the training process. Concretely, we train the estimation network $S_\theta$ by minimizing

$$
E_\tilde{x} \left[ \|S_\theta(\tilde{x}) - \nabla_\tilde{x} \log q_\sigma(\tilde{x} | x)\|_2^2 \right] = \sum_{i=1}^{N} \sigma_i^2 \cdot E_\tilde{x} \left[ \|S_\theta(\tilde{x}, \sigma_i) + \frac{\tilde{x} - x}{\sigma_i}\|_2^2 \right],
$$

where $\tilde{x} \sim q_\sigma(\tilde{x} | x) = N(\tilde{x}, \sigma_i^2)$. The score estimation network $S_\theta$ conditions on the noise variation $\sigma_i^2$ during training. Note that in Algorithm 1, we only sample the noises with the minimal variation $\sigma_N^2$. The weighting term $\sigma_i$ keeps the expectation values in Equation (10) in the same order of magnitude, which is beneficial for model training [55].

We scale the pixel values of samples to the range $[0, 1]$ and set $\sigma_N = 0.01$. We set the value of $\sigma_1$ to the maximum euclidean distance between all pairs of training samples. One may determine the value of the common ratio $\gamma$ using the heuristic rules proposed by Song et al. [55]. However, we simply use $\gamma = 0.99$ and find this setting works well in the following experiments.

### 3.4 Comparison With Existing Methods

In this subsection, we show the advantages of the ScoreMix method by making comparisons with previous mixing-based methods:

- The proposed ScoreMix method does not require any data-specific hyperparameter tuning. In contrast, previous methods such as AugMix [21] build data augmentation pipelines consisting of numerous hyperparameters. These hyperparameters need to be tuned for different tasks. Otherwise, the augmentation methods may even hurt model performance.
- We do not rely on any pre-trained model during training. Instead, we first train a score estimation network and then use it to guide the augmentation process. Methods like SaliencyMix [32] and Puzzle Mix [31] generate augmented data with the guidance
of a pre-trained saliency detection model. But how to obtain an accurate saliency detection model in the data-limited regime remains a challenging problem. The requirement of accurate pre-trained models hinders the application of these methods in real-world scenarios.

- We do not require any human annotations for data augmentation. That means the ScoreMix method is readily applicable to unsupervised image synthesis tasks. In contrast, the SuperMix method [30] and hardmix method [34] improve the data augmentation strategies with label information, which may be expensive or difficult to obtain in practice.

Fig. 3 shows some examples of the augmented results from our approach and several existing methods [19], [20], [21]. The example from the mixup method [19] exhibits characteristic artifacts. Two cats in the augmented image are displaced in relation to each other. For the CutMix method [20], it is easy to notice the edges of the pasted patch, and the augmented image looks unrealistic. We note that the mixup and CutMix methods augment samples using image arithmetic operations (e.g., addition and multiplication), which inevitably produces unrealistic details. The AugMix method [21] aims to generate augmented images without deviating too much from the original ones. However, the augmented result seems to be too similar to the original image. By contrast, the example from our approach is distinguished from the original image. We find that the fur color and pupil shape of the cat are changed. Our approach also subtly changes the shape of the stripes on the forehead of the cat. Moreover, our result expresses natural textures and details.

4 Experiments and Analysis

4.1 Implementation Details

To generate the augmented sample $x^*$, we use the Adam optimizer with a learning rate of 0.005 and linear decay scheduling. We use 100 gradient descent steps for the optimization and then finetune $x^*$ by score-based denoising. The score estimation network optimizes $x^*$ and approximates the denoising process at $t_0 = 0.25$. Similar to the mixup method [19], we set the hyperparameter $\alpha = 0.2$ for the beta distribution. We use these settings for all the experiments without hyperparameter tuning.

We build the score estimation network following the architecture design in [54]. Specifically, we adopt the RefineNet [61] architecture and use the conditional instance normalization [62] for every convolutional layer. To train the score estimation network, we use the Adam optimizer with a learning rate of 1e-4. The training batch size is set to 32.

4.2 Datasets

In this section, we introduce the datasets used in the following experiments.

The Flickr-Faces-HQ (FFHQ) dataset [6] is a benchmark dataset for unconditional image synthesis. This dataset consists of 70 thousand high-quality human face images with considerable variations. The images are numbered from 0 to 69,999, and we use the first one thousand images for model training.

The ImageNet dataset [15] consists of various images organized into 1,000 classes. It is a standard benchmark for conditional image synthesis. In this work, we train the generation models with 10% training images per class.

The MetFace dataset [7] consists of human face images from the Metropolitan Museum of Art. There are 1,336 manually selected images which depict artworks such as paintings, drawings, and statues. In the following, we train the models with the first one thousand images.

The Animal Faces-HQ v2 (AFHQ) dataset [18] provides animal faces of three domains: cat, dog, and wildlife. Each category contains about 5,000 images. Choi et al. [18] split this dataset into 14,336 samples for training and 1,467 samples for testing. We randomly choose 500 training images for each class, which is about 10% of the full training set.

4.3 Unconditional Image Synthesis

We first verify the proposed ScoreMix method by unconditional image synthesis. In this task, we train GANs to map random noise to image samples. We use the StyleGAN2 [3] as the baseline network, and the output resolution is $256 \times 256$. We implement the baseline using the released source codes.\footnote{1. https://github.com/rosinality/stylegan2-pytorch}

We evaluate the quality of synthesized images using Fréchet Inception Distance (FID) [63] and Kernel Inception Distance (KID) [64]. The FID metric measures the Wasserstein distance between two image sets. We use the FID score between the real images and the synthesized images to evaluate the sample quality: a lower score indicates better results. The KID metric measures the squared maximum mean discrepancy between the representations of the real and synthesized image sets. Similarly, a lower KID indicates that the results are better.

In the experiments, we find that the GAN training is unstable when the training data is limited. The FID score
starts to deteriorate after a certain number of training iterations. That means the overfitting problem occurs. Then, the generated samples become meaningless noises. Such observations are also reported in previous methods [4], [7], [8]. Hence, we perform the early stopping strategy to train all the models. Specially, we compute the FID score every 100,000 iterations during training. The training is stopped when the FID score increases in five consecutive epochs. Then, we use the training snapshot with the best FID score for evaluation.

Ablation: Augmentation Ratio. We aim to determine the optimal amount of augmented data. Intuitively, this amount depends on the amount of real data. Hence, we study the effect of the augmentation ratio $\mu$, i.e., the ratio of augmented to real data. In this ablation study, we investigate the following two cases:

- The augmentation ratio $\mu$ is static. We produce the augmented data before the training starts. The amount of training data increases by a factor of $(1 + \mu)$.
- The augmentation ratio $\mu$ grows during training. We augment the training batch with a certain probability at each iteration. The amount of augmented data will increase as the training goes on. Considering that the number of training iterations is huge, we assume the augmentation ratio approaches infinity.

The results are shown in Fig. 4. Overall, GANs benefit from augmented training data. We observe that the FID scores improve, and most of the training processes become more stable. These results show that augmented data can improve model performance. Models with a growing augmentation ratio still collapse quickly. One possible reason is that the amount of augmented data is overwhelming. The models have to neglect the real data to minimize the training loss. This phenomenon indicates that too many augmented samples is not helpful to model training.

The model with a larger static augmentation ratio renders better results, but we observe diminishing returns in terms of sample quality. When we crease augmentation ratio $\mu$ from 0 to 10, the FID score decreases from 58.10 to 24.21. Whereas, increasing $\mu$ to 100 only makes the FID score 1.31 lower. To balance the trade-off between sample quality and computational cost, we set $\mu = 10$ for the following tasks.

Ablation: Baseline Model. In this ablation study, we investigate whether the proposed method consistently improves model performance across different baseline models or not. Specifically, we consider the following two types of variants:

- Baselines with different model capacity. We change the network architecture to increase/reduce model capacity. Concretely, we double/halve the network width (the number of feature channels).
- Baselines with additional augmentation method. We add the DiffAug method [8], which consists of a family of the traditional augmentation operations [14], [15], [23], to the augmentation pipeline.

The evaluation results of the variants are shown in Table 1. We use the subscript wider/narrower to indicate more/fewer feature channels. For all the variants of baselines, the proposed ScoreMix method significantly improves model performance. In addition, combining our approach with the DiffAug method [8] yields further performance gain. StyleGAN2_wider performs the best among different network architectures. This shows that increasing model capacity leads to better results if we augment the training data properly. The variant (e), i.e., wider StyleGAN model + DiffAug + ScoreMix, achieves the best FID score. In the following, we consider this variant as the baseline method for unconditional image synthesis unless otherwise specified.

Evaluation Results. We evaluate the proposed ScoreMix method against existing mixing-based approaches, including the mixup [19], CutMix [20], and AugMix [21] methods. Note that for the mixup and CutMix methods, we use them to augment both the real and fake samples for the discriminator. This strategy is necessary for them to avoid the leaking problem [7].

Table 2 reports the FID and KID scores evaluated on the FFHQ [3] and MetFace [7] datasets. The score-based models achieve better FID/KID scores than the GAN model [3] under the 1K data setting. This provides empirical evidence that the score-based methods are more robust in data-limited regimes. Our approach, combining the score-based data augmentation with GANs, performs best in terms of FID and KID on both datasets.

In addition, ScoreMixLc combines the proposed method with the LC regularization in our prior work [17], and

![Fig. 4. Ablation of the augmentation ratio $\mu$. We train the StyleGAN2 model [3] with the proposed ScoreMix method on the FFHQ dataset [6] under 1K data setting. We compute Frechet Inception Distance [63] (FID, lower is better) during training.](image)

### Table 1

| Network Preprocessing method | FID score |
|-----------------------------|-----------|
| (a) StyleGAN2 [3] none     | 58.11 / 24.21 |
| (b) StyleGAN2 narrower      | 55.91 / 25.04 |
| (c) StyleGAN2 wider         | 64.38 / 23.76 |
| (d) StyleGAN2 [3] DiffAug [8] | 26.63 / 21.13 |
| (e) StyleGAN2 narrower DiffAug [8] | 26.62 / 21.26 |
| (f) StyleGAN2 wider DiffAug [8] | 26.13 / 19.41 |

We train different baselines on the FFHQ dataset [3] under 1 K data setting. We report FID [63] (lower is better). The former/later number in the table cell denotes the FID score of the baseline method without/with the proposed scheme.
ScoreMix combines the proposed method with the adaptive discriminator augmentation (ADA) method [7]. The comparison results in Table 2 show that appending the regularization method in [17] or [7] to ScoreMix both leads to further improvements in quantitative metrics.

We visualize some samples synthesized by different methods in Fig. 5. Our synthesized results are diverse and contain fewer artifacts. This indicates the generated distribution of our results is similar to the real distribution.

Table 3 shows the FID [63] scores with different amounts of training data (from 70 K to 1 K) on the FFHQ dataset [3]. The results show that our approaches consistently improve the baseline method under all the data settings. As the number of training samples increases, the baseline method benefits less from our approach. However, under the 70 K setting, ScoreMix still improves the FID score by 5.3% (4.56 → 4.32).

| Method | FFHQ-1 K | MetFace-1 K |
|--------|----------|-------------|
|        | FID | KID×10^3 | FID | KID×10^3 |
| GAN: StyleGAN2 [3] | 58.11 | 40.32 | 64.58 | 40.60 |
| score-based: NCSNv2 [55] | 38.20 | 21.73 | 44.56 | 19.48 |
| score-based: SMLD [56] | 31.23 | 17.19 | 35.70 | 13.71 |
| Baseline: [3] + [8] | 26.13 | 14.56 | 34.60 | 13.87 |
| Baseline + mixup [19] | 25.25 | 13.87 | 31.93 | 11.81 |
| Baseline + CutMix [20] | 24.72 | 13.37 | 30.11 | 12.27 |
| Baseline + AugMix [21] | 22.95 | 9.46 | 25.91 | 6.53 |
| Baseline + ScoreMix | 19.41 | 5.65 | 22.13 | 4.51 |
| Baseline + ScoreMixADA | 18.74 | 4.38 | 19.12 | 2.97 |
| Baseline + ScoreMixADA | 18.29 | 3.91 | 19.42 | 3.35 |

We report FID [63] (lower is better) and KID [64] (lower is better). ScoreMix is our proposed method. ScoreMixADA combines it with the regularization methods in [17] and [7], respectively.

Fig. 5. Unconditional FFHQ [3] (photographs) and MetFace [7] (portrait painting) samples generated by different methods under 1 K data setting. We use the StyleGAN2 model [3] with the DiffAug scheme [8] as the baseline. The results are synthesized by the baseline, baseline + mixup [19], baseline + CutMix [20], baseline + AugMix [21], and baseline + ScoreMix (ours).
TABLE 3
FID [63] Scores (Lower is Better) as a Function of Training Set Size on the FFHQ [3] Dataset

| Method                  | 70 k | 35 k | 10 k | 5 k  | 2 k  | 1 k  |
|-------------------------|------|------|------|------|------|------|
| Baseline: [3] + [8]     | 4.56 | 5.85 | 8.85 | 13.44| 17.93| 26.13|
| Baseline + ScoreMix     | 4.32 | 5.36 | 7.57 | 10.46| 13.45| 19.41|
| Baseline + ScoreMixADA  | 4.25 | 5.19 | 7.08 | 9.84 | 12.61| 18.74|
| Baseline + ScoreMixLC   | 4.29 | 5.03 | 6.85 | 9.54 | 12.04| 18.29|

ScoreMix is our proposed method. ScoreMixLC and ScoreMixADA combine it with the regularization methods in [17] and [7], respectively.

**Computational Cost.** The proposed ScoreMix method has a reasonable computational cost. In our experiments, training the baseline method (StyleGAN2 [3] + DiffAug [8]) takes 69.3 hours on 4 Tesla V100S GPUs. We follow the setting in [7] that stops the GAN training when 25,000 k real samples are shown to the discriminator. The training that combines the baseline method with our approach takes 76.0 hours, which increases the wall-clock time by around 9.5%. In addition, we take about 120 hours to train the score estimation network. During inference, our approach requires no additional computational cost.

### 4.4 Class-Conditional Image Synthesis

We also evaluate the proposed method on class-conditional image synthesis using the ImageNet dataset [65], where we train GANs to produce samples with a specified class based on a given label. Since labeled data is usually expensive to collect, solving this conditional task with limited data is important for practical applications. We use BigGAN [4] with the DiffAug method [8] as the baseline. The output resolution is $128 \times 128$. Note that we implement the baseline method on GPU, and we reduce the training batch size to 512, which is 4 times less than the size in the original work [4].

We use FID [63] and Inception Score (IS) [66] as quantitative metrics. We sample generated samples with the truncation trick proposed in [4]. Similar to the unconditional image synthesis task, a lower FID indicates a better performance. IS measures the diversity and quality of synthesized results using entropy. In this task, a higher IS score indicates a better result.

**Evaluation Results.** We evaluate our approaches against the mixing-based methods [19], [20], [21]. Table 4 reports the FID and IS scores under 100%, 50%, 25%, and 10% data settings. The comparison results show that our approaches outperform the competing methods in terms of FID and IS. The 10% data setting only has fewer than 130 training samples for each class. Such scarce training data leads to degraded performance of the baseline method. However, the proposed method significantly improves the model performance. Similar to the results in unconditional image synthesis, combining the proposed method with the LC regularization [17] or adaptive discriminator augmentation method [7] brings further performance gain. For instance, ScoreMix improves the FID score by 50.3% (49.67 → 24.67), and ScoreMixLC further improves it by 8.8% (24.67 → 22.51) under the 10% data setting.

**4.5 Image-to-Image Translation**

Image-to-image translation (I2I) aims at learning the mapping from the source domain to the target domain. In the following, we perform two specific unsupervised I2I tasks in the data-limited regime.

**4.5.1 Animal Face Translation**

In this subsection, we change the species of the input animal face (the source image). We consider the following two types of translations: (1) reference-guided translation. Given a reference animal face (the reference image), we extract a style representation from it. Then, the generator mixes the style with the content of the source image, producing the translated result. (2) latent-guided translation. Given a one-hot class label, we draw a latent code from a prior distribution conditioned on the label. The generator uses the latent code as the style representation. For this task, we use StarGAN v2 [18] as the baseline method. It can perform both reference-guided translation and latent-guided translation. The training objective is minimizing the combination of the adversarial loss [1], style reconstruction loss [68], and style...

---

**TABLE 4**

Class-Conditional Image Synthesis Results of Different Methods on the ImageNet Dataset [65]

| Methods                  | 100% data | 50% data | 25% data | 10% data |
|--------------------------|-----------|----------|----------|----------|
|                          | FID ↓ IS ↑| FID ↓ IS ↑| FID ↓ IS ↑| FID ↓ IS ↑|
| Baseline: [4] + [8]      | 9.08      | 91.82    | 12.88    | 83.27    | 17.10    | 62.15    | 49.67    | 21.30    |
| Baseline + mixup [19]    | 9.01      | 91.19    | 12.70    | 84.05    | 16.63    | 64.30    | 42.19    | 25.41    |
| Baseline + CutMix [20]   | 9.03      | 90.44    | 12.77    | 83.98    | 16.69    | 64.26    | 44.62    | 24.88    |
| Baseline + AugMix [21]   | 8.86      | 92.31    | 12.46    | 85.12    | 16.07    | 66.56    | 38.76    | 28.25    |
| Baseline + ScoreMix      | 8.60      | 94.37    | 11.85    | 87.09    | 15.12    | 71.31    | 24.67    | 37.48    |
| Baseline + ScoreMixADA   | 8.55      | 98.17    | 11.78    | 89.69    | 14.91    | 76.23    | 24.03    | 40.33    |
| Baseline + ScoreMixLC    | 8.42      | 95.55    | 11.36    | 87.14    | 14.43    | 76.74    | 22.51    | 41.12    |

We report FID [63] (lower is better) and IS [64] (higher is better). We implement BigGAN [4] on GPU with a training batch size of 512. ScoreMix is our proposed method. ScoreMixLC and ScoreMixADA combine it with the regularization methods in [17] and [7], respectively.
Fig. 6. Class-conditional ImageNet [65] samples generated under 10% data setting. We use the BigGAN model [4] with the DiffAug scheme [8] as the baseline. Applying our approach to the baseline method synthesize more realistic results.

Fig. 7. Diverse translation results on the AFHQ dataset [18]. Our model can learn to generate diverse high-quality results using only 10% data in the training set.
The resolutions of the input and output images are $256 \times 256$. To implement the baseline, we use the released official source codes.

We evaluate the quality of synthesized images using FID [63] and Learned Perceptual Image Patch Similarity (LPIPS) [67]. The LPIPS score measures the diversity of images using the L1 distance in the feature space. In this task, the higher the LPIPS score, the better the synthesized results is. We compute the FID and LPIPS scores for every pair of the image domains (e.g., dog $\rightarrow$ cat, cat $\rightarrow$ dog). Since the AFHQ dataset provides 3 different image domains, we compute these scores on 6 pairs of domains. Then, we report the average values of the scores as the evaluation metrics.

**Evaluation Results.** Given a single source image, we generate diverse results by multiple random reference images. Fig. 7 shows diverse translation results by our method. We observe that the synthesized images are obviously different breeds of animals. That means our approach can generate distinctive styles. In addition, the synthesized images preserve the content information such as pose and shape effectively.

We evaluate our method against existing mixing-based approaches, including the mixup [19], CutMix [20], AugMix [21] and ReMix (our previous conference version [22]) schemes. Fig. 8 shows some translation results. The synthesized images are supposed to preserve the low-level semantics (e.g., pose and shape) of the source image and the high-

---

1. https://github.com/clovaai/stargan-v2

2. https://github.com/clovaai/stargan-v2
Table 5
Fréchet Inception Distance (FID, Lower is Better) and Learned Perceptual Image Patch Similarity (LPIPS, Higher is Better) of Different Methods on the AFHQ Dataset [18]

| Method               | Latent-guided translation | Reference-guided translation |
|----------------------|---------------------------|----------------------------|
|                      | 100% data  | 10% data | 100% data | 10% data | 100% data | 10% data | 100% data | 10% data | 100% data | 10% data | 100% data | 10% data | 100% data | 10% data |
| Baseline: StarGAN v2 | 16.18      | 0.450    | 45.12     | 0.428    | 19.78     | 0.432    | 39.23     | 0.407    |
| Baseline + mixup     | 16.02      | 0.454    | 28.15     | 0.450    | 18.51     | 0.453    | 28.12     | 0.439    |
| Baseline + CutMix    | 15.98      | 0.453    | 41.36     | 0.444    | 17.24     | 0.449    | 26.33     | 0.442    |
| Baseline + AugMix    | 15.72      | 0.443    | 41.36     | 0.425    | 16.98     | 0.434    | 23.80     | 0.411    |
| Baseline + DiffAug   | 15.79      | 0.468    | 25.11     | 0.460    | 16.89     | 0.477    | 23.02     | 0.459    |
| Baseline + ReMix     | 15.26      | 0.482    | 21.82     | 0.472    | 16.03     | 0.480    | 22.92     | 0.458    |
| Baseline + ScoreMix  | 14.03      | 0.470    | 20.49     | 0.464    | 15.19     | 0.484    | 21.51     | 0.460    |
| ScoreMixADA          | 14.18      | 0.479    | 19.20     | 0.469    | 14.83     | 0.485    | 20.14     | 0.461    |
| Baseline + ScoreMixLC| 13.16      | 0.482    | 18.83     | 0.474    | 14.11     | 0.485    | 17.99     | 0.466    |

ScoreMix is our proposed method. ScoreMixLC and ScoreMixADA combine it with the regularization methods in [17] and [7], respectively.

Table 6
FID [63] (Lower is Better) and LPIPS [67] (Higher is Better) as a Function of Training Set Size on the AFHQ [3] Dataset

| Method               | Latent-guided translation | Reference-guided translation |
|----------------------|---------------------------|----------------------------|
|                      | 100% data  | 50% data | 25% data | 10% data | 100% data | 50% data | 25% data | 10% data | 100% data | 50% data | 25% data | 10% data |
| Baseline: [3] + [8]  | 16.18      | 19.54    | 28.34    | 45.12    | 19.78     | 22.76    | 28.11    | 39.23    |
| Baseline + ScoreMix  | 14.03      | 15.42    | 18.76    | 20.49    | 15.19     | 17.09    | 18.30    | 21.51    |
| Baseline + ScoreMixADA| 14.18    | 15.79    | 18.51    | 19.20    | 14.83     | 16.69    | 17.70    | 20.14    |
| Baseline + ScoreMixLC| 13.16      | 14.77    | 17.11    | 18.83    | 14.11     | 15.88    | 16.57    | 17.99    |

ScoreMix is the proposed method. ScoreMixLC and ScoreMixADA combine it with the regularization methods in [17] and [7], respectively.

level semantics (e.g., color and fur) of the reference image. The baseline model suffers from the overfitting problem and generates some unrealistic texture details. Overall, the proposed method synthesizes images with higher visual quality than other schemes.

Table 5 shows the FID and LPIPS scores of the evaluated methods. Our approach performs favorably against existing augmentation methods in terms of the quantitative metrics. The FID scores indicate that our results are more similar to the real data. The LPIPS score of ScoreMix with 10% data is even higher than that of the baseline with 100% data. These results demonstrate that the proposed method is effective for diverse and realistic image translation. Moreover, combining our approach with additional schemes [7], [17] further improves model performance. ScoreMixLC performs the best among all the competing methods.

Table 6 presents the FID and LPIPS scores with different amounts of training data. Under all the data settings, our approaches provide notable performance improvements. Even for the 100% data setting (about 14 K samples), our approach is still particularly beneficial: ScoreMix improves the FID score by 13.3% (16.18 \rightarrow 14.03) on latent-guided translation and 23.2% (19.78 \rightarrow 15.19) on reference-guided translation.

4.5.2 One-Shot Image-to-Image Translation

In this subsection, we conduct I2I in the single image setting. Some recent methods [50], [51], [53] have explored this challenging setting: only one source image and one target image are available for model training. Note that we do not use the pre-trained model on external data like the adaptation-based I2I approaches [41], [42].

For this task, we use the official implementation\(^3\) of the SinCUT method [53] as the baseline. This method trains the I2I model in the patch-wise manner. Concretely, the input image is randomly cropped into 16 patches. Then, these patches are resized to 128 \times 128 for model training. The translation model takes these patches as a batch of training data. The training is with the standard augmentation operations such as random scaling and horizontal flipping. In addition, we use the proposed ScoreMix method to increase the diversity of training image patches. During the test stage, the translation model takes the full resolution image as the input. This approach can translate a single image of arbitrary size.

We use Single Image Fréchet Inception Distance (SIFID) [51] to evaluate the synthesized results. This metric calculates the FID score between the internal distributions of two images. We extract the features using the Inception-v3 network [70]. Similar to the FID score, a lower SIFID score between the synthesized image and target image indicates a better result.

Evaluation Results. Fig. 9 shows some translation results and the corresponding SIFID scores. We use the collected

\(^3\) https://github.com/taesungp/contrastive-unpaired-translation
The translated images are supposed to preserve the general structure of the source image and match the textures of the target image. Although this task is challenging due to the limited data, the proposed method achieves reasonable results. The SinCUT method suffers from the overfitting problem and synthesizes distorted images. In contrast, the proposed method transfers the style of the target image more faithfully and shows fewer artifacts. Moreover, the proposed method obtains significantly lower SIFID scores. These results show that our synthesized images are more similar to the real data.

We conduct a user study to validate the proposed method. First, we use the SinCUT method and our approach to perform a total of 60 one-shot image-to-image translation tasks. The source-target image pairs for training are publicly available. Each method synthesizes 60 translated results. Then, we ask evaluators to vote for the synthesized images from the SinCUT method and our approach based on image quality. We keep the translation methods anonymous in the evaluation. The source and target images are provided for the evaluators as references. We distribute questionnaires to online users and finally get valid feedbacks from 71 human evaluators. We did not collect any private information.

Table 7 shows the competition results. We compute the SIFID scores and record user preferences. Then, we count the number of times each method performs better than its competitor and report the percentage in the table. For each task, any method with the majority of the user votes is better in the human evaluations. As for the SIFID score, lower is better. The competition results show that the proposed method generally performs better regarding both quantitative evaluations and human evaluations.

| Metrics | Human Evaluation |
|---------|------------------|
| SIFID   | Ours > SinCut    | Ours < SinCut |
| SIFID   | 83.3%            | 6.6%          |
| SIFID   | 1.7%             | 8.4%          |

We evaluate the two methods on 60 different one-shot image-to-image translation tasks. The metrics include Single Image Fréchet Inception Distance (SIFID) and human evaluation.

5 Discussion

In this work, we have verified that combining the proposed method with additional augmentation operations, including [7], [8], [14], [15], [16], further improves model performance. Incorporating recent methods that are orthogonal to this work, e.g., the RL-based augmentations [24], [25], may perform well. In addition, the proposed method could
be easily used in the semi-supervised learning framework [72], [73], [74]. We leave the study to future work.

The focus of this paper is how to generate images plausible to human visual perception. We have shown that the proposed data augmentation method improves the sample quality using automatic metrics [51], [63], [64], [67] and subjective evaluation. However, whether or not the proposed method is beneficial still depends on the applications. It might be less suited for some particular applications, such as medical imaging [75] and biology [76].

6 Conclusion
This article introduces a mixing-based data augmentation method to tackle the overfitting problem of GANs. We propose to optimize the augmented samples by minimizing the norms of the data scores. This approach prevents manifold intrusion and produces high-quality augmented samples. Hence, it increases the diversity of image samples, facilitating GAN training in image synthesis. In addition, the proposed approach is scalable to various image synthesis tasks and requires no data-specific fine-tuning. We demonstrate that our method vastly improves image quality and quantitative metrics in numerous tasks, especially when the training data is limited.

References
[1] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[2] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in Proc. Int. Conf. Learn. Representations, 2018.
[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of StyleGAN,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2020, pp. 8107–8116.
[4] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in Proc. Int. Conf. Learn. Representations, 2019.
[5] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in Proc. IEEE Int. Conf. Comput. Vis. 2015, pp. 3730–3738.
[6] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2019, pp. 4401–4410.
[7] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” in Proc. Adv. Neural Inf. Process. Syst., 2020, Art. no. 1015.
[8] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, “Differentiable augmentation for data-efficient GAN training,” in Proc. Adv. Neural Inf. Process. Syst., 2020, Art. no. 634.
[9] H. Zhang, Z. Zhang, A. Odena, and H. Lee, “Consistency regularization for generative adversarial networks,” in Proc. Int. Conf. Learn. Representations, 2019.
[10] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, “On data augmentation for GAN training,” IEEE Trans. Image Process., vol. 30, pp. 1882–1897, Jan. 2021.
[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1597–1607.
[12] C. Stein, “A bound for the error in the normal approximation to the distribution of a sum of dependent random variables,” in Proc. Berkeley Symp. Math. Statist. Probability, 1972, pp. 583–602.
[13] P. Vincent, “A connection between score matching and denoising autoencoders,” Neural Comput., vol. 23, no. 7, pp. 1661–1674, 2011.
[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
[16] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” 2017, arXiv:1708.04552.
[17] H.-Y. Tseng, L. Jiang, C. Liu, M.-H. Yang, and W. Yang, “Regularizing generative adversarial networks under limited data,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 2917–2927.
[18] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “StarGAN V2: Diverse image synthesis for multiple domains,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8188–8197.
[19] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Representations, 2018.
[20] S. Yun, D. Han, S. Chun, S. I. Oh, Y. Yoo, and J. Choe, “CutMix: Regularization strategy to train strong classifiers with localizable features,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 6022–6031.
[21] 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. Representations, 2020.
[22] T. Karras, L. Hou, M.-H. Yang, R. He, and Z. Sun, “ReMix: Towards image-to-image translation with limited data,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 113–123.
[23] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, “Fast augsyn,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 6665–6675.
[24] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “RandAugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 2020, pp. 3008–3017.
[25] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, “Adversarial autoaugment,” in Proc. Int. Conf. Learn. Representations, 2020.
[26] T. DeVries and G. W. Taylor, “Dataset augmentation in feature space,” in Proc. Int. Conf. Learn. Representations Workshop, 2017.
[27] C. Beckham et al., “Adversarial mixup resynthesizers,” in Proc. Int. Conf. Learn. Representations Workshop, 2019.
[28] A. Dabouei, S. Soleymani, F. Taberkhani, and N. M. Nasrabadi, “SuperMix: Supervising the mixing data augmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 13794–13803.
[29] J.-H. Kim, W. Choo, and H.-O. Song, “Puzzle mix: Exploiting saliency and local statistics for optimal mixup,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 5275–5285.
[30] A.-F. M. S. Uddin, M. S. M. Mura, W.-H. Shin, T. Chung, and S.-H. Bae, “SuperMix: A saliency guided data augmentation strategy for better regularization,” in Proc. Int. Conf. Learn. Representations, 2021.
[31] H. Guo, Y. Mao, and R. Zhang, “Mixup as locally linear out-of-manifold regularization,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 3714–3722.
[32] Y. Kalantidis, M. B. Sarıyüzdil, N. Pion, P. Weinzaepfel, and D. Lecours, “Hard negative mixing for contrastive learning,” in Proc. Adv. Neural Inf. Process. Syst., 2020, Art. no. 1829.
[33] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein GANs,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5767–5777.
[34] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 214–223.
[35] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, “Are GANs created equal? A large-scale study,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 698–707.
[36] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?” in Proc. Int. Conf. Mach. Learn., 2018, pp. 3481–3490.
[37] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, “Towards good practices for data augmentation in GAN training,” 2020, arXiv:2006.05338.
[38] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, “Image augmentations for GAN training,” 2020, arXiv:2006.02595.
[39] M.-Y. Liu et al., “Few-shot unsupervised image-to-image translation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10551–10560.
K. Saito, K. Saenko, and M.-Y. Liu, “COCO-FUNIT: Few-shot unsupervised image translation with a content conditioned style encoder,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 382–398.

Q. Gu, G. Wang, M. T. Chiu, Y.-W. Tai, and C.-K. Tang, “LADN: Local adversarial disentangling network for facial makeup and de-makeup,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 10480–10489.

E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, “Few-shot adversarial learning of realistic neural talking head models,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9458–9467.

M. Wang et al., “Example-guided style-consistent image synthesis from semantic labeling,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1495–1504.

T. Wang, M. Yang, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, “Few-shot video-to-video synthesis,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5014–5025.

X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, “VITON: An image-based virtual try-on network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7543–7552.

A. Shocher, N. Cohen, and M. Irani, “Zero-shot super-resolution using no. 4, pp. 695–709,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3118–3126.

Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang, “Non-stationary texture synthesis by adversarial expansion,” in Proc. Int. Conf. Comput. Graph. Interactive Techn., 2018, Art. no. 49.

A. Shocher, S. Bagon, P. Isola, and M. Irani, “InGAN: Capturing and remapping the “DNA” of a natural image,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 4501–4510.

T. R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a generative model from a single natural image,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4569–4579.

T. Hinz, M. Fisher, O. Wang, and S. Wermet, “Improved techniques for training single-image GANs,” in Proc. IEEE Winter Conf. Appl. Comput. Vis., 2021, pp. 1300–1309.

T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, “Contrastive learning for unpaired image-to-image translation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 319–345.

Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 11895–11907.

Y. Song and S. Ermon, “Improved techniques for training score-based generative models,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 4026–4037.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in Proc. Int. Conf. Learn. Representations, 2021.

J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in Proc. Adv. Neural Inf. Process. Syst., 2020, Art. no. 574.

L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou, “How does mixup help with robustness and generalization?,” in Proc. Int. Conf. Learn. Representations, 2021.

O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, “Vicinal risk minimization,” in Proc. Adv. Neural Inf. Process. Syst., 2000, pp. 395–401.

A. Hyvärinen and P. Dayan, “Estimation of non-normalized statistical models by score matching,” J. Mach. Learn. Res., vol. 6, pp. 1029–1046, 2005.

G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement network for high-resolution semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1925–1934.

V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” in Proc. Int. Conf. Learn. Representations, 2017.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6626–6637.

M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, “MMD demystifying GANs,” in Proc. Int. Conf. Learn. Representations, 2018.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, 2015, pp. 211–252.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 2234–2242.

R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 586–595.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2223–2232.

Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, “Mode seeking generative adversarial networks for diverse image synthesis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1429–1437.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 2818–2826.

F. Liu, S. Paris, E. Shechtman, and K. Bala, “Deep photo style transfer,” 2017, arXiv:1703.07511.

J. Li, R. Socher, and S. C. H. Hoi, “DivideMix: Learning with noisy labels as semi-supervised learning,” in Proc. Int. Conf. Learn. Representations, 2020.

K. Sohn et al., “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” in Proc. Adv. Neural Inf. Process. Syst., 2020, Art. no. 51.

D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “MixMatch: A holistic approach to semi-supervised learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5049–5059.

J. T. Guibas, T. S. Virdi, and P. S. Li, “Synthetic medical images from adversarial networks,” 2017, arXiv:1705.00057.

P. Costa et al., “End-to-end adversarial retinal image synthesis,” IEEE Trans. Med. Imag., vol. 37, no. 3, pp. 781–791, Mar. 2018.
Ming-Hsuan Yang (Fellow, IEEE) is a professor in electrical engineering and computer science with the University of California, Merced, California. He served as a program co-chair of IEEE International Conference on Computer Vision (ICCV), in 2019, program co-chair of Asian Conference on Computer Vision (ACCV), in 2014, and general co-chair of ACCV 2016. He served as an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011 and is an associate editor of International Journal of Computer Vision, Image and Vision Computing, and Journal of Artificial Intelligence Research. He received an NSF CAREER award, in 2012 and Google Faculty Award, in 2009. He is a fellow of the ACM.

Ran He received BE degree in computer science from the Dalian University of Technology, the MS degree in computer science from the Dalian University of Technology, and the PhD degree in pattern recognition and intelligent systems from CASIA, in 2001, 2004, and 2009, respectively. Since September 2010, he has joined NLPR, where he is currently a full professor. His research interests focus on information theoretic learning, pattern recognition, and computer vision. He served as an associate editor of IEEE Transactions on Image Processing and IEEE Transactions on Biometrics, Behavior, and Identity Science and as an area chair of CVPR/ECCV/NeurIPS. He is a fellow of the IAPR.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.