Improving Out-of-Distribution Robustness of Classifiers Through Interpolated Generative Models

Anonymous authors
Paper under double-blind review

Abstract

Out-of-distribution (OoD) generalization is one of the major challenges for deploying machine learning systems in the real world. Learning representations that disentangle the underlying structure of data is of key importance for improving OoD generalization. Recent works suggest the proprieties of disentangled representation in the latent space of GAN models. In this work, we investigate when and how GAN models can be used to improve OoD robustness in classifiers. Generative models are expected to be able to generate realistic images and increase the diversity of the training set to improve the model's ability to generalize. However, training the conventional GAN models for data augmentation preserves the correlations in the training data. This hampers training a robust classifier against distribution shifts since spurious correlations from the biased training data are unrelated to the causal features of interest. Besides, Training GAN models directly on multiple source domains are fallible and suffer from mode collapse. In this paper, we employ interpolated generative models to generate OoD samples at training time via data augmentation. Specifically, we use the StyleGAN2 model as the source of generative augmentation, which is pre-trained on one source training domain. We then fine-tune it on other source domains with frozen lower layers of the discriminator. Then, we apply linear interpolation in the parameter space of the multiple correlated networks on multiple source domains and control the augmentation in the training time via the interpolation coefficients. A style-mixing mechanism is further introduced to improve the diversity of the generated OoD samples. Our experiments show that our proposed framework explicitly increases the diversity of training domains and achieves consistent improvements over baselines on both synthesized MNIST and many real-world OoD datasets.

1 Introduction

Deep learning achieves superior performances in various practical applications, such as computer vision (Krizhevsky et al., 2012), natural language processing (Devlin et al., 2018), recommendation systems (Zhang et al., 2019) and autonomous driving (Caesar et al., 2020). The standard setting of deep learning assumes that the training and test data are drawn independently and identically distributed (i.i.d.) from the same distribution. However, in the real world, the mismatch of training and test data distributions is widely observed (Koh et al., 2021; Gulrajani & Lopez-Paz, 2020), and it hurts the performance of many deep learning systems (Geirhos et al., 2020). This challenge is known as an out-of-distribution (OoD) generalization problem. How to improve the robustness of classifiers against distribution shifts is still a challenging problem, as the true underlying data distributions are significantly underrepresented or misrepresented by the limited training data with selection bias (Beery et al., 2018).

Generative models, such as GANs, can synthesize photo-realistic images (Goodfellow et al., 2014). One intuitive idea is to use generative models as a data source to increase the diversity of the training data. Previous work (Jahanian et al., 2021; Antoniou et al., 2017) has explored this idea of generating multiple views of the same content for better representation learning. Such an idea should be also applicable to improve the OoD robustness of classifiers. However, training the conventional generative adversarial network (GAN) models preserves and even amplifies the correlations in the training set (Tan et al., 2020). Directly training classifiers on the generated data from GANs may suffer from over-fitting when the test data come from another distribution, as spurious correlations from the biased training set are unrelated to the causal reasons of target objects (Arjovsky et al., 2019). Besides, training GAN models directly on multiple source domains is fallible and may suffer from mode collapse. A proper training strategy of GAN models on multiple domains and improving the fairness and diversity of the generated data distributions are needed for improving OoD robustness using generative models.

In this paper, we propose to use interpolated generative models to generate OoD samples for improving OoD robustness. The core idea is to learn conditional generator networks in a pretraining and fine-tune manner, that effectively and efficiently models the multiple source domains, and then linearly interpolate the multiple correlated networks in the parameter space to generated diversified OoD data. Both the source domain data and the generated OoD samples are used to train the robust classifiers. To be specific, we leverage StyleGAN2 (Karras et al., 2020) as the data source which is pre-trained on one source domain. We then fine-tune the model on the other domains with limited distribution ranges by freezing the lower layers of the discriminator (Mo et al., 2020). However, modeling the multiple source domains using correlated GANs still preserves the bias in the training set. Therefore, we adopt the network interpolation method (Wang et al., 2019) to interpolate the model parameters of the correlated generative networks from the multiple source domains. The interpolated generative models generate continuous additional vicinity of the training data with the same class, which will consistently lead to better generalization ability (Simard et al., 1998). The generated OoD samples distribution also covers larger diversity ranges. Besides, the layer-wise generative representations emerge in GANs. We further perform style-mixing mechanisms to control the semantic augmentation process to alleviate the over-fitting problem to the spurious features (e.g., color) in the training set. This helps classifiers to learn features that focus more on the shape than texture, which results in better OoD robustness (Geirhos et al., 2018). The augmentation process can be controlled in fine-grained detail through the interpolation coefficients.

Our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first attempt to adopt the interpolated deep generative models for OoD generalization, where correlated conditional generators are trained and fine-tuned on the multiple source domains and linearly interpolated in the parameter space to explicitly increases the diversity of source domains.
- We take a step to understand OoD generalization from a data augmentation perspective. We provide further analysis of the classifiers trained on the generated OoD samples. Our practice shows that data diversity does influence OoD robustness in classifiers.
- Our experimental results show that our proposed framework can explicitly generate diversified OoD samples and achieves consistent improvements over baselines on both synthesized MNIST and real-world OoD datasets.

2 Methodology

In this section, we present preliminaries on data augmentation via Mixup and the layer-wise generative representations in GANs (Section 2.1). Then, we introduce the details of our proposed framework of interpolated GANs for OoD generalization in classifiers (Section 2.2). We provide further discussion of improving generalization ability through interpolated generative models in Section 2.3

2.1 Preliminaries

Data Augmentation via Mixup. In OoD scenarios, we are interested in augmenting the training data with similar but different additional virtual examples, which can be described by the Vicinal Risk Minimization (VRM) principle (Chapelle et al., 2001). The additional virtual examples that are drawn from the vicinity of the training data with the same class consistently result in better generalization ability (Simard et al., 1998). Mixup (Zhang et al., 2017) proposes generating virtual vectors from a generic vicinal distribution: $\tilde{x} = \lambda x_i + (1-\lambda)x_j$, $\tilde{y} = \lambda y_i + (1-\lambda)y_j$, where x_i, x_j are input vectors, y_i, y_j are one-hot label encodings. The weights λ are sampled from the Beta distribution. The neural network trained with linear interpolation of examples and corresponding

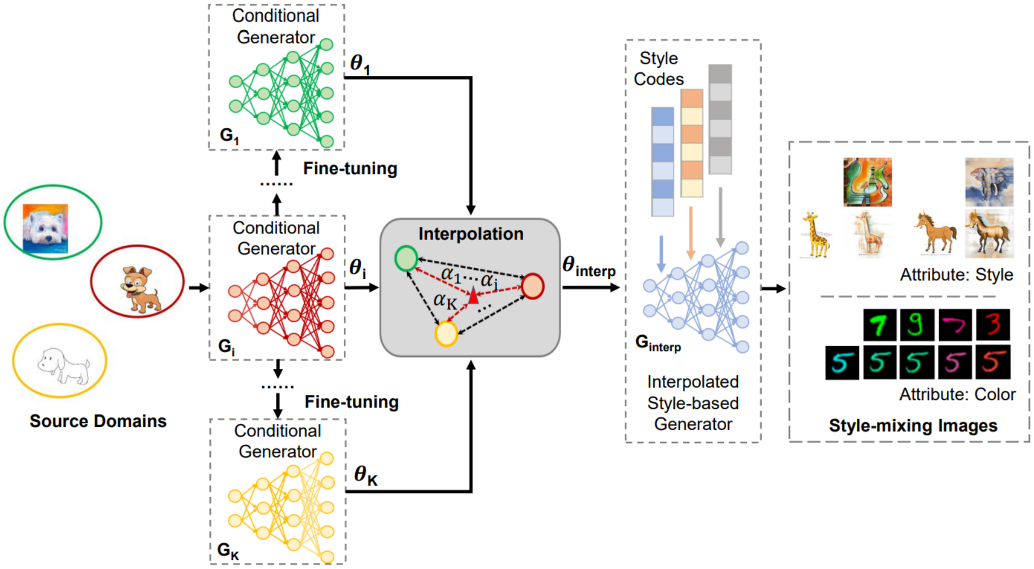


Figure 1: The framework of the proposed method. We are provided with K source domains. The target is to train a classifier that can generalize to the unseen domain. The conditional generator networks are learned in a pretraining and fine-tune manner, which effectively and efficiently models the multiple source domains. We apply linear interpolation on the multiple correlated networks in the parameter space to generate diversified OoD data. A style-mixing mechanism is further introduced to get semantic augmented samples.

labels pairs is more stable for model predictions and improves the generalization of neural network architectures. However, Mixup produces locally ambiguous and unnatural samples, which misleads the model, especially for recognition (Yun et al., 2019)

Layer-wise Generative Representations in GANs. The well-trained GANs are able to synthesize photo-realistic images, which can be used as an unlimited data source for data augmentation. As shown in Figure 1, recent advanced GAN models, such as StyleGAN2, takes layer-wise stochastic latent codes to all generator layers, which naturally encodes multi-level semantics as the layer-wise generative representations (Yang et al., 2021b). In the following section, we present how GAN models with layer-wise generative representations can be used to generate controllable virtual vicinity of the training data for better generalization ability.

2.2 From Mixup to Interpolated Generative Models

In the OoD generalization scenarios, we are provided with K source domains. The target is to learn a robust classifier that can generalize well to the unseen target domain. In the following descriptions, conditional generators $G_1(\cdot), G_2(\cdot)...G_K(\cdot)$ denotes the K correlated conditional generators for the K source domains, which takes data $\boldsymbol{x}$ as the input. Let $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}...\boldsymbol{\theta_K}$ denote the parameters for the multiple generators. Let $\ell_{\text{classifier}}$ be the training loss function for the classifiers. The details of the proposed framework are illustrated in Figure 1.

Training GANs on Multiple Source Domains. To effectively and efficiently train conditional GANs for improving OoD generalization in classifiers, a conditional generator $G_i(\cdot)$ with parameters $\boldsymbol{\theta_i}$ is pretrained on source domain i. Then, we fine-tune the pre-trained conditional generator $G_i(\cdot)$ on other source domains with frozen lower layers of the discriminator to obtain the conditional generator $G_j(\cdot)$ with parameters $\boldsymbol{\theta_j}$, where $j \neq i, 1 < j < K$. Training GANs on source domains in a pretraining and fine-tune manner is different from directly training one conditional generator on the multiple discrete source domains, which is fallible and easily mode collapses, as shown in our experiments. This facility the GANs training on multiple discrete source domains and speeds up the training process. Directly training GANs on the source domains still preserves the correlations

Algorithm 1 Interpolated deep generative models for OoD generalization

```
Require: Training set $\mathcal{D}$, batch size $n$, learning rate $\mu$, conditional generators $\mathrm{G}_1, \mathrm{G}_2, ..., \mathrm{G}_K$.
Ensure: $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K, \boldsymbol{\omega}$.
Initialize $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K, \boldsymbol{\omega}$;
Training $\boldsymbol{\theta}_i$ on source domain $i$;
Update $\boldsymbol{\theta}_j$ by fine-tuning $\boldsymbol{\theta}_i$ on source domain $j$;
Calculate $\boldsymbol{\theta}_{\text{interp}} = \alpha_1 \boldsymbol{\theta_1} + \alpha_2 \boldsymbol{\theta_2} + ... + \alpha_K \boldsymbol{\theta_K}$, according to equation 1;
repeat
    Sample a mini-batch of training images $\{(x_i, y_i)\}_{i=1}^{n}$;
    Sample a mini-batch of synthesized OoD samples: $x_i^{\text{syn}} \leftarrow \mathrm{G}(x_i; \boldsymbol{\theta}_{\text{interp}})$;
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \mu \cdot \nabla_{\boldsymbol{\omega}} \ell_{\text{classifier}}(\boldsymbol{\omega}, x_i, x_i^{\text{syn}})$, according to equation 2;
until convergence;
```

in training data. Thus, we introduce the following network interpolation in the parameter space to disregards spurious features that are correlated but not causal for training a robust classifier.

Network Interpolation for GANs. Inspires by the priors literature (Wang et al., 2019), we propose to perform linear interpolation for the parameters in the networks of the multiple correlated generators with the same architecture from pretraining and fine-tuning on the K source domains. The generators $G_1(\cdot), G_2(\cdot)...G_K(\cdot)$ trained on different source domains $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}...\boldsymbol{\theta_K}$, which has a close correlation with each other, are mixed together via interpolation in the parameter space:

$$\boldsymbol{\theta}_{\text{interp}} = \alpha_1 \boldsymbol{\theta_1} + \alpha_2 \boldsymbol{\theta_2} + ... + \alpha_K \boldsymbol{\theta_K}, \tag{1}$$

where $\alpha_1 + \alpha_2 + ... + \alpha_K = 1$, and there is a constrain for α_i that $\alpha_i \geq 0$. This is a convex combination of the parameter vectors of $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$,..., $\boldsymbol{\theta_K}$. Diverse and continuous OoD samples synthesizing can be realized by adjusting (α_1, α_2, ..., α_K). The interpolation operation is applied on the layers in the parameter space including all the convolutional layers and the normalization layers. The generated OoD data can be controlled by these interpolation coefficients.

Style-mixing Strategy for Improving Data Diversity. The interpolated conditional generators compose multi-level hierarchical semantics in the latent space, which preserves the layer-wise generative representations. We exchange certain layers of the layer-wise latent vectors for two given batches of images. This further controls the semantic augmentation process and increases the data diversity. As the interpolated generator is conditioned on the category labels, the style-mixing strategy can generate additional examples in the vicinity of the training data with the same class.

Interpolated GANs as Data Source. The interpolated GANs as a data source can be used to increase the diversity of training data. To train the classifier and improve OoD robustness, we apply classification loss to the classifier on both real images and the synthesized OoD samples:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{classifier}}^{i}(\boldsymbol{\omega}, x_i, x_i^{\text{syn}}), \tag{2}$$

where $\mathcal{L}_{\text{classifier}}$ denotes the cross-entropy loss, $\boldsymbol{\omega}$ is the parameters of the classifier, x_i be the real input data, and x_i^{syn} denotes the synthesized OoD samples. The stochastic gradient descent algorithm (SGD) can be performed to optimize the objective. The algorithm of the proposed framework is outlined in Algorithms 1.

2.3 Discussion

What does model learn from InterpolatedGAN? We have mentioned that augmenting the training data with similar but different additional virtual examples improves the generalization ability. The motivation is inspired by the VRM (Chapelle et al., 2001) principle. As shown in Figure 4, the interpolated generative models are indeed able to generate more realistic images than the Mixup (Zhang et al., 2017) algorithm with preserved class labels. Besides, we observe that some attributes, e.g. style, color, are successfully being changes with shape remains the same, as shown in Figure 1. This may be due to the emergence of semantic hierarchy in the latent space of interpolated generative models. The classifiers trained with these mixed samples help to alleviate the over-fitting problem to the spurious feature (e.g., color) in the training set and focus more on the shape feature of digits. The neural network models, which focus more on shape than texture, have better generalization ability (Geirhos et al., 2018).

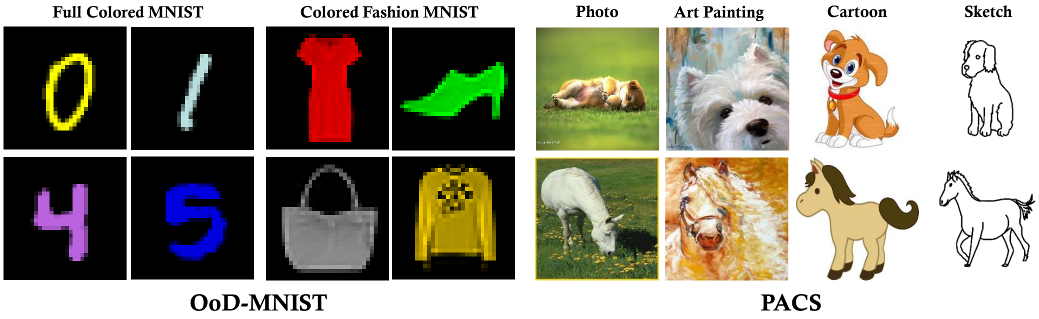


Figure 2: Typical examples of the OoD datasets.

Table 1: Classification accuracy on the full colored MNIST. The backbone for the baselines is MLP. The baselines are implemented by ourselves.

Model	In Distribution	Real Data	Syn. $1K$	Syn. $10K$	Syn. $20K$	Syn $25K$
ERM	✓	92.86±0.02	92.58±0.03	91.51±0.03	91.27±0.04	91.27±0.02
	✗	55.93±0.06	56.73±0.07	61.00±0.07	73.50±0.06	**64.26±0.02**
Mixup	✓	91.67±0.01	91.58±0.06	91.08±0.04	90.71±0.03	90.75±0.04
	✗	46.75±0.08	49.06±0.33	55.50±0.11	59.38±0.11	**60.28±0.08**
IRM	✓	93.58±0.02	93.39±0.03	92.54±0.00	92.42±0.02	92.36±0.00
	✗	60.04±0.05	60.87±0.06	65.33±0.02	67.40±0.06	**68.04±0.05**
REx	✓	92.86±0.03	92.56±0.02	91.43±0.01	91.13±0.02	91.00±0.03
	✗	56.41±0.04	57.13±0.08	60.98±0.03	63.27±0.12	**63.87±0.04**

Analysis of generalization ability. We analyze the effect of the proposed interpolated generative models on improving the OoD performance of classifiers. As shown in Table 3.2, training on the synthesized OoD samples achieve consistent improvements over baselines, in terms of out-of-distribution accuracy. This confirms the improved generalization ability via increasing data diversity. We also compare FID results of the visual quality for the generated data. We train GANs on the multiple source domains with joint training on different domains, training from scratch on one domain, and the proposed pretraining and fine-tune strategy (see Table 3.4). We observe that the proposed pretraining and fine-tune strategy achieves lower FID and speeds up the GANs training process.

3 Experiments

In this section, we evaluate our proposed method on different typical OoD datasets. We present the implementation details and baselines in Section 3.1. Section 3.2 provides the experimental results and discussion on the benchmarks. Section 3.3 presents the detailed ablation study of the proposed framework. We provide a detailed analysis of the generated OoD data in Section 3.4.

3.1 Implementation

Datasets. We evaluate our proposed method on various OoD benchmarks: Full Colored MNIST, Colored Fashion MNIST, PACS (see Figure 2). The challenging Full Colored MNIST dataset contains 60000 images with resolution (32×32), which includes 10 digits ranging from 0 to 9. The digits were colored with 10 colors based on different correlations with the labels to construct different environments, i.e., 80% and 90% for the training environments and 10% for the test environment. This is different from the original Colored MNIST dataset in (Arjovsky et al., 2019). This original Colored MNIST construct a binary classification problem with only two colors, which is a much simpler setting, compared with the Full Colored MNIST used in this work. Colored Fashion MNIST is a more challenging classification task than MNIST digit, where we assign colors correlated with

Table 2: Classification accuracy on the fashion MNIST. The backbone for the baselines is MLP. The baselines are implemented by ourselves.

Model	In Distribution	Real Data	Syn. $1K$	Syn. $10K$	Syn. $20K$	Syn $25K$
ERM	✓	94.41±0.08	92.55±0.09	89.91±0.05	88.47±0.03	87.84±0.09
	✗	45.06±0.14	45.46±0.38	57.18±0.21	61.12±0.33	**62.46±0.14**

Table 3: Classification accuracy on PACS dataset compared with different methods with ResNet-18.

Model	Art	Cartoon	Sketch	Photos	Average
IRM	70.31	73.12	75.51	84.73	75.92
REx	76.22	73.76	66.00	95.21	77.80
Mixup	82.01	72.58	72.48	93.29	80.09
MTL	76.76	71.87	76.73	92.65	79.50
MMD	79.34	73.76	72.61	94.19	79.97
DRO	78.09	74.18	77.00	93.45	80.68
ERM	77.85	74.86	67.74	95.73	79.05
Ours	81.18	77.65	78.80	95.33	**83.24**

(a) ImageNet pre-trained

Model	Art	Cartoon	Sketch	Photos	Average
IRM	30.08	41.85	35.56	39.10	36.65
REx	31.93	45.95	35.84	44.19	39.48
Mixup	35.16	47.87	42.12	53.59	44.69
MTL	38.48	49.06	46.55	51.98	46.52
MMD	39.89	51.11	43.09	53.41	46.88
DRO	38.92	46.72	46.73	54.67	46.76
ERM	38.87	48.93	41.10	58.38	46.82
Ours	**41.55**	**52.52**	**48.13**	**59.46**	**50.42**

(b) Training from scratch

labels to the original Fashion MNIST dataset (Xiao et al., 2017) to build different environments. The PACS dataset consists of 9991 images with resolution (227 × 227).This dataset contains 7 categories and 4 domains (photo, art painting, cartoon, sketch). In our experiments, we follow the same leave-one-domain-out validation protocol (Li et al., 2017), which means that we use three domains for training and the remaining domain for testing.

Baselines. Our proposed framework can be implemented on any GAN framework. In our experiment, we use the state-of-the-art StyleGAN2 (Karras et al., 2020) model to demonstrate the effectiveness of our method. We compare our proposed method of multiple OoD algorithms, including empirical risk minimization (ERM) (Arjovsky et al., 2019), invariant risk minimization (IRM) (Arjovsky et al., 2019), mixup (Mixup) (Zhang et al., 2017), risk extrapolation (REx) (Krueger et al., 2021), domain generalization by solving jigsaw puzzles (Carlucci et al., 2019).

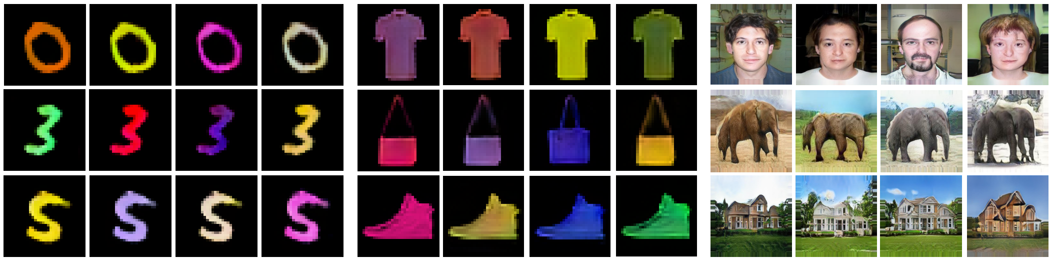
Evaluation Metric. For evaluating out-of-distribution generalization ability, the metric is the top-1 category classification accuracy. We use the metric Fréchet Inception Distance (FID) (Heusel et al., 2017) to evaluate the visual quality of synthesized data, which calculates the FID between 50,000 fake images and all the training images. Following the same setting in the work (Heusel et al., 2017), we use an official pre-trained Inception network to compute the FID.

Implementation details. For training the classifier on the Colored MNIST dataset, the backbone network of the baseline methods is a three-layer MLP. The number of training epoch is 500, the batch size is the whole training data. The optimizer is SGD with a learning rate of 0.01. The model trained was tested at the final epoch. The backbone network for the baselines on the PACS dataset is ResNet-18. We follow the same training, validation, and test split as in the work JiGen (Carlucci et al., 2019). The total training epoch is 100. The batch size is 64. Our framework was implemented with PyTorch 1.9.0 and CUDA 10.2. We conducted experiments on NVIDIA TITAN Xp. More implementation details can be found in the Appendix.

3.2 Results and Discussion

In this section, we evaluate and analyze the results of our method on four datasets: Full Colored MNIST, Fashion MNIST, PACS, which represent different aspects of distribution shifts.

Results on Full Colored MNIST. As shown in Table 1, we can see that the proposed method achieves consistent improvements over baselines. IRM combining with interpolated GAN method achieves better performance, even compared with advanced REx (Krueger et al., 2021) method. Our method further improves the performance in Colored MNIST by augmenting the biased training set with synthesized data. Noticing that all the four baselines show better OoD accuracy with increased



Full Colored MNIST **Colored Fashion MNIST** **PACS**

Figure 3: Visualization results of synthetic images on various datasets

volume of synthesized data. The superior performance confirms the possibility of improving OoD generalization ability via increasing the data diversity.

Results on Colored Fashion MNIST. Our method achieves much better performance, in terms of out-of-distribution accuracy on the fashion MNIST compared with ERM baseline (see Table 2). Specifically, ERM combining with our proposed augmentation method achieves 64.26% when synthetic data is $25,000$, which is much higher than the ERM baseline (55.93%). This may be because the synthetic data facility the classifier disregards spurious features that are correlated but not causal for prediction. The visualization of the generated data on fashion MNIST is shown in Figure 3.

Results on PACS. As shown in Table 3, our method achieves the state-of-the-art performance when using the ResNet-18 as the backbone network no matter whether the backbone is pre-trained or randomly initialized. This PACS dataset considers more realistic generalization scenarios with distribution shifts in styles. In our implementation, ERM with interpolateGAN achieves 83.24% average accuracy, compared with advanced OoD algorithms, such as MTL (Blanchard et al., 2017) (79.50%) and Mixup (Zhang et al., 2017) (80.09%). The poor performance of MTL and Mixup may be due to the biased training data and distribution shifts. Our proposed method has achieved SOTA performance on the challenging PACS benchmark. This demonstrates the superiority of our proposed method and its potential to be useful in practice.

3.3 Ablation Study

In this section, we compared our proposed method with advanced augmentation methods, such as Mixup (Zhang et al., 2017). This is to test whether directly applying typical augmentation methods to training classifiers can improve the out-of-distribution accuracy. We conduct an ablation study to investigate the importance of each component. We also conduct experiments on the different quantities of generated data with different baselines.

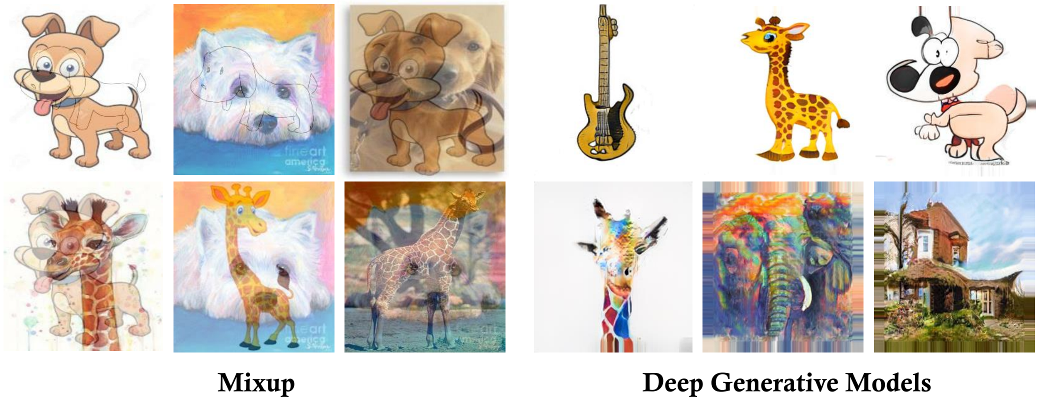
Experiments on the different quantities of generated data. For ablation study on the different quantities of generated data, we take the colored MNIST dataset for example. The results are shown in Table 1. We observe that without the synthetic data, IRM achieves 60.04% out of distribution accuracy that is much lower than training with $25,000$ synthetic data (68.04%). It can also be seen that training with a larger quantity of synthetic data achieves consistent better OoD accuracy than a small quantity of generated data on colored MNIST. The results confirm the effectiveness of increasing data diversity for improving OoD robustness.

Table 4: Synthesis Quality.

Sketch Domain	FID
Joint Photo and Sketch	101.6
Sketch	96.3
Cartoon to Sketch (ours)	**32.9**
Photo to Sketch (ours)	**20.2**

Table 5: Ablation Study.

StyleGAN2	Interpolation	Style-mixing	Accuracy
			46.82
✓			47.82
✓	✓		49.83
✓	✓	✓	**50.42**



Mixup **Deep Generative Models**

Figure 4: Comparison of the generated images from different data augmentation mechanisms. The samples generated by deep generative models are more realistic and preserves the class labels.

Table 6: Variants of the interpolated generative models.

ERM	Art	Cartoon	Sketch	Photos	Average
w/ StyleGAN2 (Karras et al., 2020)	40.68	50.24	43.06	57.31	47.82
w/ InterfaceGAN (Shen et al., 2020)	41.36	51.92	45.87	58.30	49.36
w/ **InterpolatedGAN (ours)**	**41.55**	**52.52**	**48.13**	**59.46**	**50.42**

Experiments on combining with different baselines. Our proposed method can be easily implemented on any OoD algorithms and GAN framework. As shown in Table 1, we change current ERM baseline to other advanced OoD algorithms, such as Mixup (Zhang et al., 2017), IRM (Arjovsky et al., 2019), and REx (Krueger et al., 2021). We observe that our proposed method achieves consistent improvement in terms of OoD accuracy on the four baselines. To be specific, we tried REx with augmented data on Colored MNIST, the result is 63.87% that is much higher than the baseline, which is 56.41%. This shows that the training data diversity is essential for increasing OoD robustness, and our proposed method is flexible to be inserted in any existing OoD algorithms.

Ablation study on different components. We conduct an ablation study on the network interpolation and style-mixing mechanism (see Table 5). The average accuracy on PACS without network interpolation and style-mixing is 47.82%. The average accuracy is improved after performing the network interpolation mechanism, which is 49.83% on the PACS dataset. The baseline algorithm is ERM, and we use StyleGAN2 (Karras et al., 2020) as the baseline generative model. This shows the effectiveness of network interpolation to facilitate the semantic augmentation process. A style-mixing mechanism is further proposed to increase the diversity of the generated data and achieves a higher average accuracy of 50.42%. This shows the effectiveness of the style-mixing mechanism.

Variants of the interpolated generative models. We change the current InterpolatedGAN (Shen et al., 2020) to InterfaceGAN for the data augmentation process, as shown in Table 6, which achieves 49.36% average accuracy, lower than the original InterpolatedGAN method. We also tried directly applying StyleGAN2 (Karras et al., 2020) for the multiple source domains on the PACS dataset. As shown in Table 6, the result is 47.82%. This may be because training conventional GAN models for data augmentation still preserves the correlations in the training data. This shows that the proposed interpolatedGAN framework is needed to improve OoD robustness via data augmentation.

3.4 Analysis of Generated OoD Data

Visualization of generated OoD data. We visualize the synthesized OoD samples on PACS in Figure 3. The quantitative results of the visual quality in terms of FID are shown in Table 4. The synthesis quality is substantially improved by our method. In particular, our method improves the FID for the sketch domain from 96.3 to 20.2 by fine-tuning from the photo domain to the sketch domain. This improvement supports one of our motivations that training GAN models directly on multiple source domains are not easy, and the pretraining and fine-tuning paradigm improves the visual quality of the synthesized OoD data.

4 RELATED WORK

Out-of-distribution generalization. Out-of-distribution generalization is a fundamental problem of deep learning models, where the test data come from another distribution. OoD-Bench (Ye et al., 2021) defines and measures the types of distribution shifts that are ubiquitous in various datasets. DomainBed (Gulrajani & Lopez-Paz, 2020) creates a living benchmark to facilitate reproducible domain generalization algorithms for robustness research. Multiple approaches have been proposed to improve the OoD generalization. IRM (Arjovsky et al., 2019) and its variants (Krueger et al., 2021; Ahuja et al., 2020) aims to find invariant representation from different training environments via an invariant risk regularization. GroupDRO (Sagawa et al., 2019) proposes to learn models that minimize the worst-case training loss over a set of pre-defined groups. MLDG (Li et al., 2018) introduces a meta-learning procedure, which simulates train and test domain shift during training. Jigsaw (Carlucci et al., 2019) proposes to learn the semantic labels in a supervised fashion, and jointly solve jigsaw puzzles on the same images. In this paper, we focus on improving OoD robustness in classifiers from a data augmentation perspective, which is the most straightforward and intuitive way to improve the generalization ability.

Generative Adversarial Networks. Generative adversarial networks can synthesize photo-realistic images (Goodfellow et al., 2014). Extensive efforts have been devoted to improving the quality of generated data (Karras et al., 2017; Brock et al., 2018; Yang et al., 2021a). Recent works observe layer-wise generative representations in GANs (Karras et al., 2019; 2020; Shen et al., 2020; Xu et al., 2021). Recently, some researchers take attempts to improve the fairness or acceptability of classifiers (Li & Xu, 2021; McDuff et al., 2019; Nguyen et al., 2017). The work of (Lang et al., 2021) proposes a training procedure, which incorporates the classifier model for a StyleGAN to learn a classifier-specific StyleSpace to explain a classifier. The work (Ramaswamy et al., 2021) introduces a GAN-based latent space de-biasing method to mitigate bias from data correlations for fair attribute classification. In this work, we propose to use generative models with layer-wise generative representations as a data source to perform semantic augmentation and increase the diversity of training data. However, GAN overfits easily and suffers from mode collapse training on discrete multiple source domains. Thus, it is highly non-trivial to extend existing GANs to improve OoD robustness.

Robustness from data augmentation perspective. Data augmentation mechanisms augment the training data with similar but different additional virtual examples lead to better generalization ability (Simard et al., 1998). Mixup (Zhang et al., 2017) presents a learning principle to generate virtual examples from a generic vicinal distribution, which trains a neural network on convex combinations of pairs of examples and labels. It has thereafter inspired some other advanced algorithms, such as Manifold Mixup (Verma et al., 2019), CutMix (Yun et al., 2019), and InterpCNN (Mao et al., 2019). DNI (Wang et al., 2019) applies linear interpolation in the parameter space of two or more correlated networks to achieve a smooth control of imagery effects. The work (Chai et al., 2021) using the different views with real-world variations generated by generative models to benefit image classification. The work (Jahanian et al., 2021) presents that the multiview data generated by generative models can naturally be used to identify positive pairs for contrastive methods. L2A-OT (Zhou et al., 2020) utilizes a data generator to synthesize pseudo-novel domains data to augment the source domains. However, they do not consider the OoD robustness of classifiers through interpolated generative models with layer-wise generative representations.

5 CONCLUSION

In this paper, we propose to use generative models as a data source to increase the diversity of the training domains to improve OoD robustness. We use the StyleGAN2 as the data source, which models multiple source domains in a pretraining and fine-tune manner. We apply linear interpolation of the multiple correlated networks in the parameter space to generate OoD samples. We further perform style-mixing mechanisms to control the semantic augmentation process. Our experiments show that InterpolatedGAN can explicitly generate diversified OoD samples and achieves consistent improvements over baselines on various OoD datasets.

Ethics Statement: The authors declare that they adhere to the ICLR Code of Ethics. This paper follows the general ethical principles: contribute to society and to human well-being, uphold high standards of scientific excellence, avoid harm, be honest, trustworthy and transparent, be fair and take action to avoid discrimination, respect the work required to produce new ideas and artefacts, respect privacy, and honour confidentiality.

Reproducibility Statement: The authors declare that all the experiments performed in studies are reproducible. We provide a complete description of implementation details and all the datasets used in Section 3.1 in the main text, and Section A.1 in the Appendix to support this reproducibility statement.

References

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *ICML*, 2020. 9

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv:1711.04340*, 2017. 1

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 2019. 2, 5, 6, 8, 9

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018. 1

Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *arXiv:1711.07910*, 2017. 7

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv:1809.11096*, 2018. 9

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019. 6, 9

Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In *CVPR*, 2021. 9

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. *NeurIPS*, 2001. 2, 4

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. 1

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv:1811.12231*, 2018. 2, 4

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020. 1

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 1, 9

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv:2007.01434*, 2020. 1, 9

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 6

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv:2106.05258*, 2021. 1, 9

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv:1710.10196*, 2017. 9

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 9

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv:2006.06676*, 2020. 2, 6, 8, 9

Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021. 1

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 1

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021. 6, 8, 9

Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. Explaining in style: Training a gan to explain a classifier in stylespace. *arXiv:2104.13369*, 2021. 9

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 6

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 9

Zhiheng Li and Chenliang Xu. Discover the unknown biased attribute of an image classifier. *arXiv:2104.14556*, 2021. 9

Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3d point cloud understanding. In *ICCV*, 2019. 9

Daniel McDuff, Shuang Ma, Yale Song, and Ashish Kapoor. Characterizing bias in classifiers using generative models. *arXiv:1906.11891*, 2019. 9

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv:2002.10964*, 2020. 2

Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*, 2017. 9

Vikram V Ramaswamy, Sunnie SY Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. In *CVPR*, 2021. 9

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv:1911.08731*, 2019. 9

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *T-PAMI*, 2020. 8, 9

Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, 1998. 2, 9

Shuhan Tan, Yujun Shen, and Bolei Zhou. Improving the fairness of deep generative models without retraining. *arXiv:2012.04842*, 2020. 2

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019. 9

Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition. In *CVPR*, 2019. 2, 4, 9

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017. 6

Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021. 9

Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. *arXiv:2106.04566*, 2021a. 9

Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *IJCV*, 2021b. 3

Nanyang Ye, Kaican Li, Lanqing Hong, Haoyue Bai, Yiting Chen, Fengwei Zhou, and Zhenguo Li. Ood-bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *arXiv:2106.03721*, 2021. 9

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3, 9

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv:1710.09412*, 2017. 2, 4, 6, 7, 8, 9

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 2019. 1

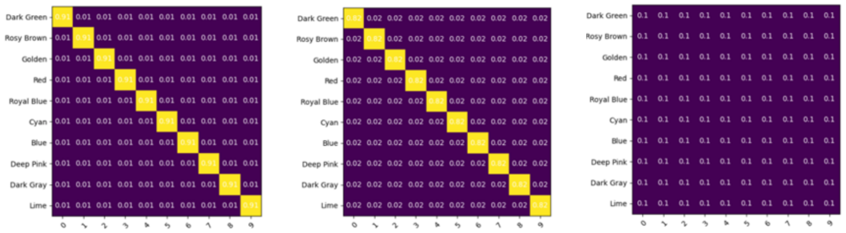
Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 9

A Appendix

A.1 Implementation Details

We conduct a fair comparison of our proposed InterpolatedGAN with various OoD generalization algorithms and SOAT GAN methods on challenging OoD datasets. For our proposed InterpolatedGAN method, the learning rate for the generation process is 0.0025. The optimizer for training the conditional generator is. For training the classifiers, we use the SGD optimizer with an initial learning rate of 0.01 for the Full Colored MNIST and the Colored Fashion MNIST. For the PACS dataset, the batch size is 64. The optimizer is SGD. We conduct hyper-parameter optimization (HPO) for all the baseline methods and compare our proposed method with their performance under the best hyper-parameters. The results for the Full Colored MNIST and Colored Fashion MNIST are averaged over 5 runs with the best set of hype parameters.

For the Full Colored MNIST, we use ten colors for all the data with 10 digits: Dark Green ([0, 100, 0]), Rosy Brown ([188, 143, 143]), Golden ([255, 215, 0]), Red ([255, 0, 0]), Royal Blue ([65, 105, 225]), Cyan ([0, 225, 225]), Blue ([0, 0, 255]), Deep Pink ([255, 20, 147]), Dark Gray ([160, 160, 160]), Lime ([0, 255, 0]). The images resolution is 32×32. The dataset size is 60000 images. The correlation is different during training and test time, as shown in Figure 5.



(a) $\rho = 0.9$ (b) $\rho = 0.8$ (c) **Target Domain**

Figure 5: Illustration of the Full Colored MNIST dataset. The 10 digits were colored with 10 colors based on different correlations with the labels to construct different environments, i.e., 80% and 90% for training environments and 10% for the test environments.

For the Colored Fashion MNIST with 10 colors and 10 categories (T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot). This is a more challenging task than the Colored MNIST. We color the objects based on different correlations to construct different environments. Similar to the Full Colored MNIST, We set a one-to-one digit-color relationship and set the bias coefficient to $\rho = 0.9$ and $\rho = 0.8$.

PACS is a widely used OoD dataset. This dataset contains 9991 images with 7 categories (dog, elephant, giraffe, guitar, horse, house, person) and 4 domains (photo, art painting, cartoon, sketch). The original images with 227×227 resolution were padding to 256×256 when training the generative models. The implementation details of the pretraining and fine-tuning stage for the pacs dataset are: 1) Target domain photo: pretraining on the cartoon domain, fine-tuning to sketch, and art painting. 2) Target domain art painting: pretraining on the photo domain, fine-tune to sketch, and cartoon. 3) Target domain sketch: pretraining on the photo domain, fine-tune to art painting, and cartoon. 4) Target domain cartoon: pretraining on the photo domain, fine-tune to art painting and sketch.

A.2 More Visualization Results

We also provide more visualization results of the synthesized images in Figure 6 and Figure 7.

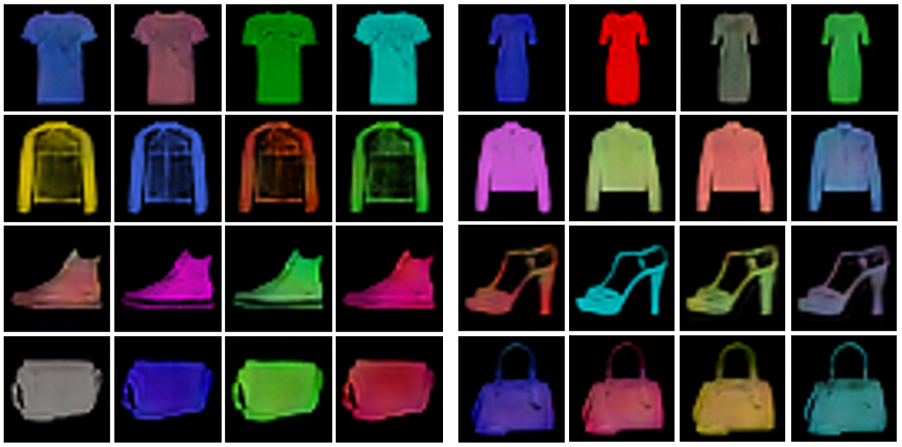


Figure 6: More visualization results of synthetic images on the Colored Fashion MNIST.

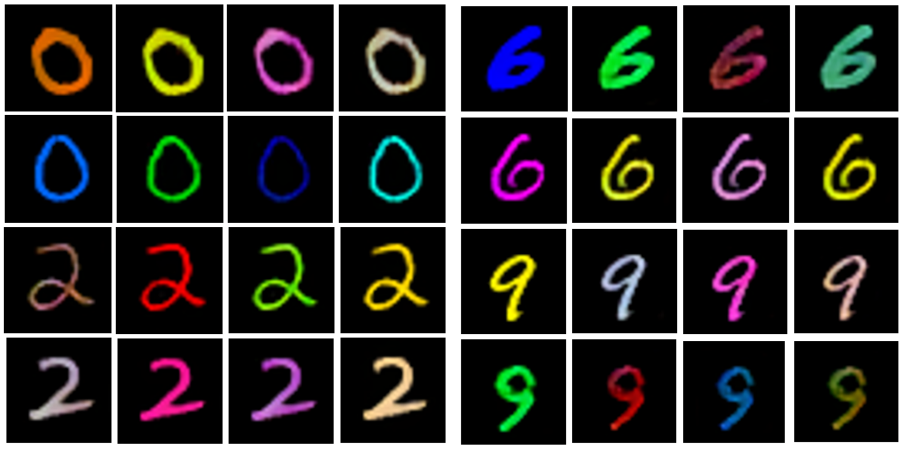


Figure 7: More visualization results of synthetic images on the Full Colored MNIST.