Neural Transformation Network to Generate Diverse Views for Contrastive Learning

Anonymous CVPR submission

Paper ID 28

Abstract

Recent unsupervised representation learning methods rely heavily on various transformations to generate distinctive views of given samples. Transformations for these views are generally defined manually, requiring significant human effort to design detailed configurations and validate practical efficacy. Furthermore, the diversity of these views is quite limited in scope causing the network to be invariant to only a small set of data transformations. To address these problems, we introduce a neural transformation network that learns to generate diverse views. Our proposed framework consists of an encoder-decoder network architecture that encodes semantic information and then randomly stylizes it with style amplification. However, such generative processes tend to cause degradation compared to the original images, which can harm the quality of the learned representation. To remedy this issue and generate more diverse styles, we use a linear augmentation between the generated view and the original image. Finally, we apply geometric transformations to aid in contrastive learning of representations. We evaluate the learned representations on various downstream vision tasks. Results show highly competitive recognition performance compared to the state-of-the-art methods that use learned views or hand-crafted views for representation learning.

1. Introduction

Recently, unsupervised representation learning has garnered lot of attention in computer vision tasks because of its label-efficiency compared to traditional supervised learning. The common strategy for unsupervised representation learning has been to construct a self-supervised objective from the unlabeled data and use it to train the network. A common self-supervised objective is to predict the transformation type of an image. This includes predicting rotation types [14], predicting translation and scale types [36], solving jigsaw puzzles [35], etc. An alternative self-supervised methods utilize positive and negative samples to train semantic information from their similarity and dissimilarity [1, 6, 17], which is the main focus of our paper. These objectives tend to generate positive samples of a given sample through transformations and enforce to be similar while sometimes enforce the sample to be dissimilar to negative samples.

The choice of data transformations, augmentations or views, as we shall refer them interchangebly throughout the paper, is important for contrastive learning. Possible views for vision tasks can be of photometric type such as blurring, color jitter, etc. or of geometric type like cropping, rotation, etc. However, these view types have been human designed and therefore limited in their diversity. Consequently, constrastively learned representations would be invariant to only limited number of augmentation types and hence might be sub-optimal for downstream tasks. Thus, for contrastive learning frameworks, there is a need to generate more diverse views without altering semantic content of an image.

Only very few works have studied non-manual view generation beyond hand-designed views. Viewmaker [43] network learns to generate novel views through an additive noise map. This noise generator is adversarially trained against the contrastive loss. Since the Viewmaker network produces only low-level additive transformations, despite of its effectiveness on enhancing overall diversity of the dataset, it produces less diversified views for each given image (discussed in Section 4.3). Alternatively, Neutral AD [40] learns neural mask generator to diversify the masked views while preserving the semantic information of the given image. However, Neutral AD can only generate limited number of mask maps for each image, which suggests that the generated views likely have low diversity. Despite these promising studies, there still exists more non-manual views that are not fully studied so far. Since these unexplored views might be useful for contrastive learning of representations, our work proposes automatic novel view generation using unlabeled training data.

We focus on beating the limitation of these recent works

in two aspects. First, to explore beyond conventional augmented views, we transform images within the latent space while preserving the semantic information. Secondly, we pursue to generate large number of diverse views from a given image. To this end, we propose a neural transformation based method for generating non-manual diverse views. Our method consists of a two-stage procedure where the view generation network and the feature encoder are trained separately unlike [40, 43]. Our view generation network uses an encoder-decoder architecture with an adaptive instance normalization layer [22] in the latent feature space to modify the style of an input image. Since the view transformation occurs in the latent space, the novel views can possibly contain much more high-level contextual changes compared to [40, 43]. The stylization is further controlled by the features of another reference style map sampled from a standard distribution. This step is also quite different from that of [40, 43] which only uses single images for novel view generation. Hence, these networks are less globally-aware about the content and style distribution of the dataset. Furthermore, a discriminator is adversarially trained against the view generator which uses multiple reconstruction losses to constrain the semantic content of the novel view.

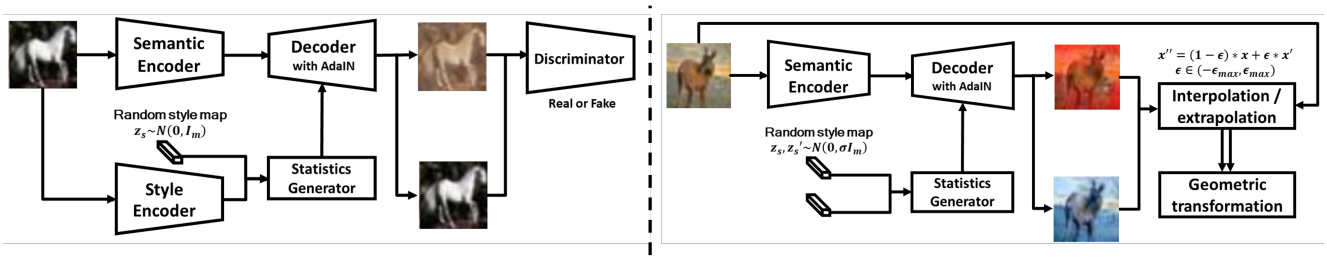
Once the view generator is trained, it is frozen and then the randomly sampled style map can be used to generate a large number of possible views. Thus, the diversity of the augmentations is much higher compared to [40, 43]. The stylized novel views are then combined with randomly generated geometric transformations and fed to the feature encoder to optimize the contrastive loss. Our method shows its advantages throughout, and the analysis in diversity metric also shows distinctive view generation capability compared to previous methods. To summarize, the contributions of the paper are as follows:

- We propose an encoder-decoder-based architecture that can produce diverse novel and stylized views beyond conventional noised, masked, or expert-designed views.
- During encoder training and downstream stages, we additionally modify the generated views through style distribution expansion and linear expansion to diversify the views from a given image.
- Compared to expert-designed views and existing view generation approaches, our generated views when combined with geometric transformations facilitate constrastively learned features that produces highly competitive recognition performance on downstream visual tasks.

2. Related Work

Unsupervised Representation Learning Learning representations from unlabeled data is a long standing problem in computer vision [3, 37]. Most modern unsupervised representation learning approaches use sample augmentations for constructing pretext tasks [12, 14, 33, 35, 57]. The pretext tasks include predicting patches [12], channels [57], rotations [14], and even order of a puzzle [35]. AET [27, 56] used image pairs, where the pretext task was estimating the transformation between them. Alternative unsupervised representation learning frameworks [1, 32] use augmentation invariance where the model representations are enforced to be invariant to certain geometric and photometric transformations. The idea of augmentation invariance has been mentioned in earlier works [2, 13, 16] but has recently garnered attention due to its applicability to multiple data-starved use cases. Contrastive learning realizes transformation invariance where representations of different input views are pulled together while views of a different input (i.e. negative pairs) are pushed apart [6–8, 17, 45]. Alternative methods exist [5, 15] that do not require presence of negative pairs to be pushed apart. All the above methods use handcrafted views while our method focuses on learning useful views for contrastive learning.

Useful Views for Representation Learning There has been significant research in obtaining useful transformations for learning representations. For supervised learning, there has been various works [19, 29, 38, 54, 55] studying effect of different handcrafted augmentation types. Automatic augmentation policies [10, 11, 20, 26, 41, 58] have also been learned to compose existing handcrafted data augmentations. Additionally, Tran et al. [47] formulated augmentations as missing variables within a Bayesian framework. Wong et al. [49] learned perturbation sets for adversarial robustness using a conditional variational auto-encoder. For self-supervised learning, some works [46, 50] consider views that maximize mutual information to be useful. Utility of views has been studied for transfer learning tasks [45] and for learning invariances [39]. Particularly, the authors of VTSS [36] hypothesize that instantiations of data transformations absent in the dataset are useful for unsupervised representation learning. Yang et al. [53] realized the VTSS hypothesis by adversarially learning the dataset's transformation distribution. Recently, authors of viewmaker [43] proposed to generate views for multiple modalities by adversarially learning residual perturbations applied to the input image. NeuTral AD [40] addresses self-supervised learning for anomaly detection by learning transformations on input samples that maintain semantics but also produce diversified augmentations. Both [44] and [40] only considered limited number of style and photometric transformations as learned augmentations while our method can produce much more diverse and effective transformations.



(a) View generation process during the proposed transformation network training

(b) View generation process during encoder pretraining

Figure 1. Overall framework of the proposed neural style transformation network. Our network consists of a semantic encoder, a style encoder, multi-layer perceptrons (MLPs) for generating normalization statistics, adaptive instance normalization (AdaIN) layers, and a decoder. (a) shows the pipeline for training the neural transformation network while (b) shows how the network generates novel views to be used in contrastive learning.

3. Method

Our method aims to generate novel diverse views by stylizing the input images with respect to the randomly sampled arbitrary style maps. In the first stage, we train a neural transformation network that learns to stylize images within a dataset. In the second stage, novel transformed images are generated to aid contrastive representation learning of a representation encoder. The details are described in the following sections while the overall framework is shown in Fig. 1.

3.1. Training neural style transformation network

Our goal is to explore a huge number of non-manual diverse views and finally achieve performance boost with contrastive learning. To generate such diverse views beyond conventional noised or masked views and manually transformed views, we design a neural style transformation network that consists of a content encoder E^c_θ, a style encoder E^s_θ, a decoder $G_{\theta,\gamma_1,\beta_1,...,\gamma_l,\beta_l}$ comprised of multiple adaptive instance normalization (AdaIN) layers [23] with AdaIN statistics $\gamma_1, \beta_1, ..., \gamma_l, \beta_l$, a statistics generator MLP_θ for AdaIN statistics computation from a given style map s, and trainable parameters θ for the whole aforementioned encoder-decoder architecture. The content encoder E^c_θ and style encoder E^s_θ aim to disentangle semantic information and style information from given images. On the other hand, the decoder $G_{\theta,\gamma_1,\beta_1,...,\gamma_l,\beta_l}$ reconstructs an image corresponding to a given semantic information and AdaIN statistics. Here, each ith AdaIN layer adjusts the semantic feature c by:

$$AdaIN(c, \gamma_i, \beta_i) = \gamma_i \Big(\frac{c - \mu(c)}{\sigma(c)} \Big) + \beta_i, \qquad (1)$$

where $\mu(c)$ and $\sigma(c)$ denotes the mean and standard deviation of the feature c. Such AdaIN statistics can be derived by MLP_θ from a given style map. Our method trains style reference from style of each training samples. During this training stage, our network learns to encode content map and style map through content and style encoders, reconstruct an input image from its content and style maps, and transform the input image to the style of other training samples.

Training objectives We enforce adversarial learning, style map reconstruction, semantic map reconstruction, and image reconstruction loss functions to train the neural style transformation network. Specifically, suppose we sampled an image $x \in \mathcal{X}$ from a training dataset $\mathcal{D}$. The semantic encoder E^c_θ and style encoder E^s_θ disentangle semantic maps $E^c_\theta(x)$ and style maps $E^s_\theta(x)$ from the given image x. Then, MLP_θ computes proper AdaIN statistics with respect to the given style map $E^s_\theta(x)$ by:

$$\gamma_1^{E^s_\theta(x)}, \beta_1^{E^s_\theta(x)}, ..., \gamma_l^{E^s_\theta(x)}, \beta_l^{E^s_\theta(x)} = MLP_\theta(E^s_\theta(x)). \qquad (2)$$

We update the decoder $G_{\theta,\gamma_1^{E^s_\theta(x)},\beta_1^{E^s_\theta(x)},...,\gamma_l^{E^s_\theta(x)},\beta_l^{E^s_\theta(x)}}$ with the computed statistics. In this state, we can reconstruct the input image from the corresponding semantic map $E^c_\theta(x)$ and style map $E^s_\theta(x)$ by:

$$\hat{x} = G_{\theta,\gamma_1^{E^s_\theta(x)},\beta_1^{E^s_\theta(x)},...,\gamma_l^{E^s_\theta(x)},\beta_l^{E^s_\theta(x)}}(E^c_\theta(x)), \qquad (3)$$

which should be identical to x. Thus, we minimize image reconstruction loss L_{img}, an L1 error between the input image x and the reconstructed image $\hat{x}$ as follows:

$$L_{img} = \mathbb{E}_{x \sim p(x)} \big[||x - G_{\theta,\gamma_1^{E^s_\theta(x)},\beta_1^{E^s_\theta(x)},...,} \gamma_l^{E^s_\theta(x)}, \beta_l^{E^s_\theta(x)}(E^c_\theta(x))||_1 \big]. \qquad (4)$$

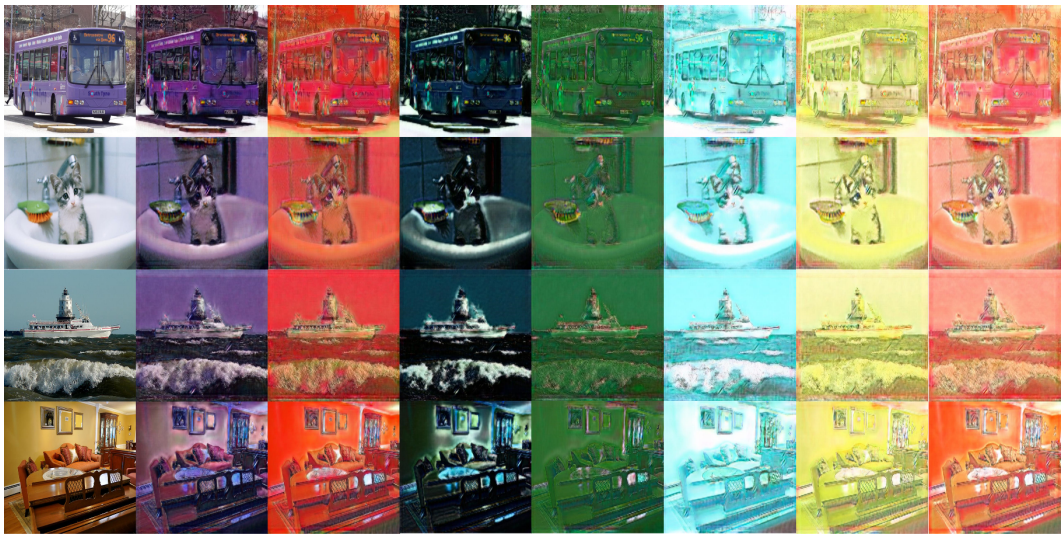


Figure 2. Qualitative results of the neural style transformation network for varying input images and style maps. The first column shows the input images, and each column represents the transformed results with each style map. We randomly sampled style maps from $N(0, 8\mathbf{I_m})$. The input images are sampled from the validation split of the MSCOCO dataset.

Due to the generation process of $\hat{x}$, the reconstructed content map $E^c_\theta(\hat{x})$ and style map $E^s_\theta(\hat{x})$ should be identical to $E^c_\theta(x)$ and $E^s_\theta(x)$, respectively.

On the other hand, the decoder should also have capability of properly generating new views from given input images and arbitrary style maps. Thus, we define a discriminator D_ϕ with trainable parameters ϕ, and we minimize the following adversarial loss L_{adv}:

$$L_{adv} = \mathbb{E}_{x\sim p(x), s\sim N(0,\mathbf{I_m})}\big[\log(1 - D_\phi(G_{\theta, \gamma_1^{E^s_\theta(x)}, \beta_1^{E^s_\theta(x)}, \ldots, \gamma_l^{E^s_\theta(x)}, \beta_l^{E^s_\theta(x)}}(E^c_\theta(x)))\big] + \mathbb{E}_{x\sim p(x)}\big[\log(D_\phi(x))\big], \tag{5}$$

where s denotes an m-dimensional style vector sampled from $N(0, \mathbf{I_m})$. Moreover, since we generate the novel view from the semantic map $E^c_\theta(x)$ and the style map s, it should also be disentangled to $E^c_\theta(x)$ and s. Thus, we define a semantic map reconstruction loss L_c and a style map reconstruction loss L_s by:

$$\begin{aligned} L_c &= \mathbb{E}_{x\sim p(x)}\big[||E^c_\theta(x_i) - E^c_\theta(G_{\theta, \gamma_1^{E^s_\theta(x)}, \beta_1^{E^s_\theta(x)}, \ldots, \gamma_l^{E^s_\theta(x)}, \beta_l^{E^s_\theta(x)}}(E^c_\theta(x)))||_1\big], \\ L_s &= \mathbb{E}_{x\sim p(x)}\big[||E^s_\theta(x_i) - E^s_\theta(G_{\theta, \gamma_1^{E^s_\theta(x)}, \beta_1^{E^s_\theta(x)}, \ldots, \gamma_l^{E^s_\theta(x)}, \beta_l^{E^s_\theta(x)}}(E^c_\theta(x)))||_1\big]. \end{aligned} \tag{6}$$

Therefore, the final objective for neural style transformation training is:

$$\min_\theta \max_\phi L_{total} = \min_\theta \max_\phi \lambda_{img} L_{img} + \lambda_c L_c + \lambda_s L_s + \lambda_{adv} L_{adv}, \tag{7}$$

where λ_{img}, λ_c, λ_s, and λ_{adv} are 20.0, 2.0, 2.0, and 2.0, respectively. The detailed training process is shown in Algorithm 1 in the supplementary material.

3.2. View generation for representation learning

After training the neural style transformation network, we utilize the overall encoder-decoder framework as a transformation function. However, we observed that directly using the generated views does not sufficiently utilize the capability of our framework to produce highly diversified views. In this section, we introduce some schemes that can boost the effectiveness of contrastive learning.

Random style distribution expansion While the previous training stage enables the encoder-decoder framework to address arbitrary style maps sampled from $N(0, \mathbf{I_m})$, we generate the novel views from expanded style map distribution $N(0, \sigma\mathbf{I_m})$, $\sigma > 1$. By controlling the magnitude of the random style vector with a standard deviation σ, we expect the network to generate more drastic changes on the given images. We analyze the effects of the expanded style distribution in Section 4.3.1. Suppose an input image is sampled as $x \in \mathcal{X}$, this scheme can be formulated by the following

	Method	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	Avg.
SimCLR [6]	Expert-designed view	86.2	49.9	**32.5**	30.4	97.1	88.3	11.2	53.3	**96.6**	60.6
	Viewmaker	84.5	50.4	31.7	28.8	**98.7**	**91.5**	8.7	53.6	94.9	60.3
	Viewmaker + Geo.	83.0	49.4	29.7	27.9	95.2	90.1	10.1	56.8	94.1	59.6
	Ours	**86.9**	**52.3**	29.7	**32.7**	96.6	89.2	**15.4**	**61.1**	93.5	**63.1**
InstDisc [51]	Expert-designed view	82.4	48.6	**37.7**	29.8	98.7	89.2	13.7	61.5	**98.9**	62.3
	Viewmaker	80.1	50.2	33.5	29.8	**98.9**	**91.4**	9.4	54.8	94.3	60.3
	Viewmaker + Geo.	82.6	51.0	29.5	29.3	96.2	88.2	13.3	64.1	85.1	59.9
	Ours	**83.2**	**54.5**	34.7	**33.3**	97.5	89.7	**18.6**	**66.7**	96.6	**63.9**

Table 1. Quantitative comparison between manual transformations, Viewmaker [43], Viewmaker with geometric transformations, and our method. We conducted linear evaluation on the CIFAR-10 dataset and transferred the pretrained encoder to (a) CIFAR-10, (b) MSCOCO, (c) Aircraft, (d) DTD, (e) MNIST, (f) FaMNIST, (g) CUBirds, (h) VGGFlower, and (i) TrafficSign. We used ResNet-18 [18] as a backbone network.

equation:

$$\gamma_1^s, \beta_1^s, ..., \gamma_l^s, \beta_l^s = MLP_\theta(s), s \sim N(0, \sigma \mathbf{I_m}) \\ x' = G_{\theta, \gamma_1^s, \beta_1^s, ..., \gamma_l^s, \beta_l^s}(E_\theta^c(x)). \tag{8}$$

Linear augmentation. Subsequent to the style expansion, we linearly augment the views with the input images through random interpolation and extrapolation. The purpose of the linear augmentation is to continuously diversify novel views along the input images and the generated views and to recover the lost semantic information of the input image during our neural style transformation. We control the degree of the linear augmentation with a magnitude ϵ since too much wide range of the linear combination ratio may generate redundant or ineffective views. The process can be simply represented as follows:

$$x'' = (1 - \epsilon)x + \epsilon x', \quad \epsilon \in (-\epsilon_{max}, \epsilon_{max}), \tag{9}$$

where x' denotes a generated view derived in Eqn. 8.

Combining with geometric transformations. We apply random geometric transformations upon the adjusted views x'' to enhance distinctiveness among the generated views. Suppose we have a distribution of geometric transformation $\mathcal{T}_{geo}$, then this scheme can be written by:

$$x''' \leftarrow t_{geo}(x''), t_{geo} \sim \mathcal{T}_{geo}, \tag{10}$$

where $t_{geo} \in \mathcal{T}_{geo}$ is a randomly sampled geometric transformation.

3.3. Encoder training and transfer learning

During encoder pretraining, we replace the conventional manual transformation with our modified neural transformation framework. For the trainable encoder E_ψ with its parameters ψ and the contrastive learning loss L_{cont}, the training objective for the encoder training is:

$$\min_\psi L_{enc} = L_{cont}(l_\psi(E_\psi(x_1'''), E_\psi(x_2'''))) \tag{11}$$

where x_1''' and x_2''' denote for two randomly transformed views of x.

Now we can transfer the pretrained encoder to various downstream tasks including classification, detection, and segmentation. During the transfer learning, we fix the encoder and only update the task-specific prediction layer (e.g. classification layer of the classification task) through supervision with labels. Suppose we have a trainable task-specific prediction layer l_ξ and an input data $x \in \mathcal{X}$ with corresponding ground-truth $y \in \mathcal{Y}$. Then, the training objective for the transfer learning can be represented as follows:

$$\min_\xi L_{trans} = L_{task}(l_\xi(E_\psi(x''')), y) \tag{12}$$

where L_{task} denote the task-specific objective function and x''' represents for the adjusted views in Eqn. 10.

4. Experiments

4.1. Datasets

We evaluated our framework using the following image classification datasets: (a) CIFAR-10 [24], (b) MSCOCO [28], (c) Aircraft [30], (d) DTD [9], (e) MNIST [25], (f) FaMNIST [52], (g) CUBirds [48], (h) VGGFlower [34], and (i) TrafficSign [21]. These datasets contain a large variety of natural and human-made categories and they are popularly used for evaluating self-supervised methods. All details of the datasets are described in the supplementary material.

Experimental setup. For the neural style transformation network, our encoders consist of several strided convolution layers and four subsequent residual blocks. The style encoders additionally have a global average pooling layer followed by a fully connected layer since we defined a style map by an m-dimensional vector. Here, we set m as 8. The AdaIN-based decoder consists of four residual blocks with

σ	ϵ_{max} 0.1	0.5	1.0	2.0
1	57.4	67.1	69.1	69.6
2	63.6	71.5	73.7	75.4
4	66.3	74.6	77.3	76.8
8	66.6	75.1	**78.2**	77.6
16	66.7	75.7	78.1	77.2
32	67.6	76.0	77.9	77.2

(a) Encoder pretraining performance

σ	ϵ_{max} 0.1	0.5	1.0	2.0
1	76.4	81.5	83.1	83.4
2	79.7	84.1	85.2	85.8
4	80.8	85.5	86.7	86.1
8	80.9	85.9	**86.9**	86.4
16	81.8	86.3	**86.9**	86.4
32	82.0	86.4	86.8	86.8

(b) Linear evaluation performance

Table 2. Performance comparison varying hyperparameters σ and ϵ_{max} iteration number of neural transformation network training. The performances are evaluated on the CIFAR-10 dataset.

eight AdaIN layers and several convolution layers with upsampling. Our multi-layer perceptrons (MLPs) consists of three layers and output concatenation of eight gamma and beta value for AdaIN layers. We used the LSGAN discriminator [31] with four convolutional layers. The encoder-decoder framework is trained for 200k iterations on the CIFAR-10 dataset with batch size 64.

For the encoder pretraining step in learning representations, we followed the configurations and evaluation protocol of Viewmaker [43] for fair comparison. We adopt SimCLR [6] and InstDisc [51] to verify the compatibility on contrastive learning approaches. The SimCLR method used a temperature of 0.07 and the InstDisc method used 4096 negative samples from the memory bank with update rate 0.5. We pretrain ResNet-18 for 200 epochs with a batch size of 256 on CIFAR-10 using SGD optimizer with learning rate 0.03, momentum 0.9, and weight decay 1×10^{-4}.

For the linear evaluation, we fixed the encoder and train the prediction layer with supervised loss on various datasets using SGD optimizer with learning rate 0.01, momentum 0.9, weight decay 0 for 100 epochs with a batch size of 128. The learning rate is reduced by a factor of 10 on 60 and 80 epochs. Once the network has been trained, we evaluate it on the test split of the corresponding dataset and report the recognition performance.

We provided detailed information of the neural style transformation network in the supplementary material.

4.2. Performance comparisons

4.2.1 Qualitative results of the view generation

We visualized the augmentation results of the neural style transformation network varying input images and style maps to validate the distinctive appearances across the randomly generated views. As shown in Fig. 2, our transformation network can generate distinctive transformation attributes for each of the style maps. Moreover, each style map maintains its own transformation attribute despite varying inputs. On the other hand, the neural network preserves the semantic information of the input image well so that we can assign similarity between the transformed samples.

4.2.2 Performance comparisons on classification tasks

Having verified the capability of our framework to generate diverse augmentations, we pretrain the target encoder and conduct linear evaluation by transferring the fixed encoder to various classification tasks. Then, we compared with models, where each has been trained by expert-designed views, viewmaker-based views, and views generated by our method. Since Viewmaker does not utilize geometric transformations only among the three type of generated views, we additionally evaluate for Viewmaker-based views with geometric transformations for fair comparison.

Table 1 shows linear evaluation results on nine datasets including CIFAR-10. Here, left and right parts of the table show the experimental results with the SimCLR [6] and InstDisc [51] methods, respectively. For the SimCLR approach, our method outperformed models learnt by other types of views on average. Specifically, our method tends to be effective on the datasets with diverse attributes such as CIFAR-10, MSCOCO, DTD, CUBirds, and VGGFlower datasets. On the other hand, our method is relatively less effective on the datasets with stereotyped backgrounds or appearances, such as the Aircraft dataset with blue skies, the MNIST dataset with black background and white digits, the FashionMNIST dataset with black backgrounds, and the TrafficSign dataset with standardized instances. Another noticeable observation is that the Viewmaker was not effective when complementary geometric transformations were added. This observation supports the extendibility of our method on various geometric transformations.

4.3. Analysis

4.3.1 Quantitative ablation study on hyperparameters σ and ϵ_{max}

We compared encoder pretraining and linear evaluation recognition performance on the CIFAR-10 dataset to find

Figure 3. Qualitative results of view generation varying hyperparameter σ and the training iteration numbers of the neural style transformation network. Each row represents for the transformed results using $N(0, \sigma \mathbf{I_m})$ with $\sigma = 1, 2, 4, 8$ from top to bottom. Each column represents for the results transformed by a model trained by 50k, 100k, 150k, 200k, 250k, 300k, 350k iterations from left to right. The input image is sampled from the validation split of the MSCOCO dataset.

optimal values for the hyperparameters σ and ϵ_{max}. We conducted experiments for $\sigma = 1, 2, 4, 8, 16, 32$ and $\epsilon_{max} = 0.1, 0.5, 1.0, 2.0$, and used the SimCLR approach to pretrain encoders for the encoder pretraining stage.

Table 2 shows the ablation study results for varying σ and ϵ_{max}. As shown in the Table 2 (a) and (b), increasing ϵ_{max} causes performance improvements within a certain range by continuously expanding the variety of the novel views between the input images and the generated views, but such range appears to be tighter when σ is larger. The optimal value for ϵ_{max} clearly reveals to be 1.0 since all the results tend to peak at 1.0 across various σ values.

For the hyperparameter σ, both encoder pretraining and linear evaluation performances tend to increase as σ increases within a certain range, and such increase becomes tighter as ϵ_{max} increases. These results indicate that expanding the distribution of the style maps cause far distinctive views advantageous for contrastive learning. We set σ and ϵ_{max} as 8 and 1.0 for all the subsequent experiments in this section.

4.3.2 Qualitative ablation study on hyperparameters σ and iteration number

To visually compare the influence of the hyperparameter σ and the number of neural transformation training iterations on generated views, we visualized qualitative results of our neural style transformation network varying σ from 1 to 8 with a factor of 2 and the iteration number from 50k to 350k with 50k intervals. As shown in Fig. 3, the generated views tend to converge to the input image regardless of the σ values. This is because the image reconstruction loss and the semantic reconstruction loss enforce the network to preserve the quality and semantic information of the input images as much as possible, respectively. Thus, a proper iteration number is necessary for training the neural transformation framework optimally. Besides, larger σ can effectively diversify the generated views from the given image. These results are aligned with the quantitative performance tendency for σ, which corroborate the effectiveness of style map distribution expansion on view generation.

4.3.3 Ablation study on each part of the method

We validated the effect of each view transformation step in our method. We mainly verified style distribution expansion, linear augmentation, and combination with geometric

(a)	(b)	(c)	Encoder pretraining	Linear evaluation
			51.0	73.1
✓			53.4	71.5
	✓		62.8	77.2
✓	✓		66.5	79.5
	✓	✓	69.1	82.8
✓	✓	✓	**78.2**	**86.9**

Table 3. Ablation study on (a) style distribution expansion, (b) Linear augmentation, and (c) geometric transformation of our neural transformation framework. We compared encoder training and linear evaluation performance for various view generation setup. In the table. The performances are evaluated on the CIFAR-10 dataset.

transformation. All the experiments are conducted on the CIFAR-10 dataset and with $\epsilon_{max} = 1.0$ whenever the linear augmentation is applied. As shown in Table 3, all the proposed schemes have a positive effect on both encoder pretraining and linear evaluation performance, except for when only the $\sigma = 8$ is applied during the linear evaluation. Moreover, the performance between each configuration has a considerable gap, meaning that each transformation step of our framework contributes significantly to better the recognition performance. In addition, our neural transformation method can work in complementary with geometric transformation to boost recognition performance.

4.3.4 Quantitative diversity comparison.

As a visual comparison, it is not possible to thoroughly compare the diversity of views generated by each method. Thus, we evaluate two evaluation metrics called the Conditional Inception Score (CIS) [4] and the Inception Score (IS) [42]. IS has been adopted to measure the quality of a generated image. Though both metrics measure diversity of the generated views, IS tends to measure the overall diversity of all the generated views while CIS focuses more on diversity of outputs conditioned on a single input image. These properties can also be explained from the following definitions:

$$IS = \exp(\mathbb{E}_{x\sim p(x)}[\mathbb{E}_{x'\sim p(x'|x)}[KL(p(y|x')||p(y))]])$$
$$CIS = \exp(\mathbb{E}_{x\sim p(x)}[\mathbb{E}_{x'\sim p(x'|x)}[KL(p(y|x')||p(y|x))]]) \quad (13)$$

where x' represents views generated from x. $KL(\cdot)$ is the Kullback–Leibler divergence while $p(y|x)$ and $p(y)$ represent the conditional label and marginal label distributions respectively.

Table 4 shows the CIS and IS for the expert-designed views, Viewmaker-based views [43], and our method. Note that we only considered the style transformation for each method. Since any sample in the input dataset does not have any other view except itself, the CIS of the input dataset should be 1.000 regardless of its high inception score. On the other hand, Expert-designed views did not show any notable score on both metrics despite frequent use. Viewmaker-based views shows highest IS after the input dataset, but its CIS is also lowest after the input dataset. And with increasing additive weight, the IS decreases while the CIS increases. It is also to be noted that IS and CIS also considers generation quality of the generative model. Hence, we can interpret this observation that the noise maps of Viewmaker decreased the the quality of the generation process while they differentiate the generated views better.

Method	Diversity metrics	
	CIS	IS
Expert-designed views	1.283	1.999
Viewmaker (weight=0.05)	1.044	5.458
Viewmaker (weight=0.1)	1.108	5.146
Ours ($\sigma = 8$)	**2.241**	4.608

Table 4. Diversity comparison among manually transformed views, Viewmaker-based views [43], and our views. We measure conditional inception score (CIS) [4] and inception score (IS) [42] on the CIFAR-10 dataset.

Besides, our method achieved the best CIS among all the comparisons, which implies that our neural style transformation network is most specialized in generating diverse views from a given image. Considering that distinctive transformations are key aspects for successful contrastive learning, our method can produce highly transferable representations for downstream tasks. Moreover, our method can also obtain better IS compared to the conventional manual transformations.

5. Conclusion

In this paper, we proposed a novel neural network architecture along with a two-stage training scheme that learns to generate diverse views for contrastive learning. Novel views are generated by disentangling an input image into its style and content map and then mixing randomly generated styles. The novel views are then used to learn representations with self-supervised instance discrimination tasks. The learned representations are evaluated on downstream classification tasks on which our proposed framework produces highly competitive recognition performance and more diverse views compared to existing view generation methods. Furthermore, we carried out extensive ablation studies and analyses and qualitatively and quantitatively confirmed the optimal design choices and hyperparameter configurations. In the future, we would like to validate our novel view generation method on additional self-supervised learning based vision tasks.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 1, 2

[2] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, 1992. 2

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[4] Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129(5):1712–1731, 2021. 8

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2, 5, 6

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 2

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5

[10] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 2

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[13] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 2

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 1, 2

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[19] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 2

[20] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019. 2

[21] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. Ieee, 2013. 5

[22] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[26] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32:6665–6675, 2019. 2

[27] Feng Lin, Haohang Xu, Houqiang Li, Hongkai Xiong, and Guo-Jun Qi. Aetv2: Autoencoding transformations for self-supervised representation learning by minimizing geodesic distances in lie groups. *arXiv preprint arXiv:1911.07004*, 2019. 2

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[29] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. 2

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[31] Xudong Mao, Qing Li, Haoran Xie, Raymond Lau, Wang Zhen, and Stephen Smolley. Least squares generative adversarial networks. pages 2813–2821, 10 2017. 6

[32] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 2

[33] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 2

[34] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 5

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 1, 2

[36] Dipan K Pal, Sreena Nallamothu, and Marios Savvides. Towards a hypothesis on visual transformation based self-supervision. *British Machine Vision Conference*, 2020. 1, 2

[37] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 2

[38] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. 2

[39] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020. 2

[40] Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pages 8703–8714, 2021. 1, 2

[41] Alexander J Ratner, Henry R Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*, 30:3239, 2017. 2

[42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 8

[43] Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. In *International Conference on Learning Representations*, 2021. 1, 2, 5, 6, 8

[44] Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. In *International Conference on Learning Representations*, 2021. 2

[45] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2

[46] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 2

[47] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. A bayesian data augmentation approach for learning deep models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2794–2803, 2017. 2

[48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[49] Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020. 2

[50] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*, 2020. 2

[51] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*, 2018. 5, 6

[52] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[53] Seunghan Yang, Debasmit Das, Simyung Chang, Sungrack Yun, and Fatih Porikli. Distribution estimation to automate transformation policies for self-supervision. *NeurIPS Workshops*, 2021. 2

[54] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019. 2

[55] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[56] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2547–2555, 2019. 2

[57] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017. 2

[58] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *International Conference on Learning Representations*, 2019. 2