SCONE-GAN: Semantic Contrastive learning-based Generative Adversarial Network for an end-to-end image translation

Iman Abbasnejad[1], Fabio Zambetta[1], Flora Salim[1,4], Timothy Wiley[1],
Jeffrey Chan[1], Russell Gallagher[3], Ehsan Abbasnejad[2]
[1]RMIT University, [2]Australian Institute for Machine Learning, [3] Rheinmetall Defence Australia,
[4]UNSW Sydney University
[*1]{iman.abbasnejad,fabio.zambetta,timothy.wiley,jeffrey.chan}@rmit.edu.au
[2]ehsan.abbasnejad@adelaide.edu.au,[3]russell.gallagher@gmail.com,[4]flora.salim@unsw.edu.au

Abstract

SCONE-GAN presents an end-to-end image translation, which is shown to be effective for learning to generate realistic and diverse scenery images. Most current image-to-image translation approaches are devised as two mappings: a translation from the source to target domain and another to represent its inverse. While successful in many applications, these approaches may suffer from generating trivial solutions with limited diversity. That is because these methods learn more frequent associations rather than the scene structures. To mitigate the problem, we propose SCONE-GAN that utilises graph convolutional networks to learn the objects dependencies, maintain the image structure and preserve its semantics while transferring images into the target domain. For more realistic and diverse image generation we introduce style reference image. We enforce the model to maximize the mutual information between the style image and output. The proposed method explicitly maximizes the mutual information between the related patches, thus encouraging the generator to produce more diverse images. We validate the proposed algorithm for image-to-image translation and stylizing outdoor images. Both qualitative and quantitative results demonstrate the effectiveness of our approach on four dataset.

1. Introduction

Generative Adversarial Networks (GANs) [12] are successful in generating high quality image samples from a random noise vector [21, 22]. However, generating scenery images with high-fidelity in complex domains with multiple factors of variation using a noise vector remains challenging. One such application is generating an outdoor scene under different weather conditions in either conditional or unconditional case. For example, image translation (*i.e.* conditional sampling) of an outdoor scene taken in summer (*i.e.* domain A) into a realistic image of the same scene in winter (*i.e.* domain B) [35, 49]. This image generation is useful for practical applications where it is necessarily to visualise scenery in different weather conditions, but it is not feasible or costly efficient to revisit and recapture the same location under multiple conditions. The majority of current state-of-the-art image-to-image translations try to compute a mapping function f_{AB} and an *estimate* of its inverse function f^{-1}_{BA} which is combined with a cycle-consistency loss for synthesising image in the target domain [17, 19, 29, 49].

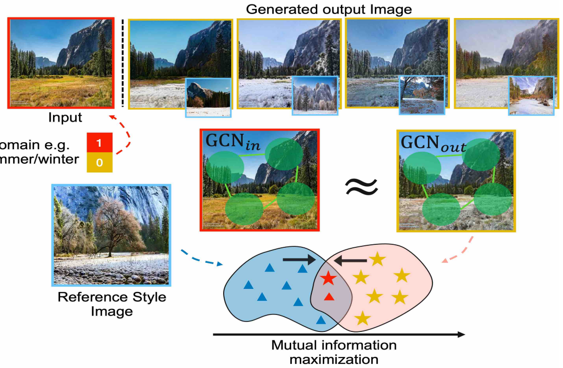


Figure 1. Overview of our proposed model. Given an input image (red rectangle), a reference style image (blue rectangle) and a domain (e.g. winter), our model can realistically generate images in the new domain (yellow rectangle). We establish a graph on the input and generated image for maintaining dependencies between objects (GCN$_{in}$ and GCN$_{out}$). We also maximize the mutual information between the reference and generated images for enhancing the diversity of generated images.

Despite the success of cycle-consistent loss, they have a major drawback. The reconstruction loss forces the generator to hide the information necessary to accurately re-

[*] Iman Abbasnejad worked on this paper when he was working at RMIT University and Russell Gallagher worked on this project when he was at Rheinmetall Defence Australia.

construct the input image due to a bias towards more frequent patterns rather than the structural consistencies [8]. The problem is particularly severe in high-frequency signals, such as outdoor scene synthesis, where the model must reconstruct too many features (including sky, clouds, mountains, etc.) and are unable to blend these elements together into a realistic image under the new domain. Therefore, cycle-consistent GANs can not be easily used to extract these information since all attributes such as objects, textures and colors are entangled. The lack of such control limits usage of image-to-image translation methods in many areas and may result in generating low quality with limited diversity images.

Various studies have been conducted to address these limitations. [17] used global structural consistency through pixel cycle-consistency and semantic losses to adapt representations at both the pixel-level and feature-level for scenery image translation. [19] decomposed the image representation into a content code, the information that should be preserved during translation, and a style code, the remaining variations that are not in the input image and should be mapped during translation, and generated images by combining content and style codes. [29] demonstrated by maximizing the distance between generated images with respect to their corresponding latent codes, generator produces more diverse images. [24] decomposed the input image into a domain-invariant content space and a domain specific attribute space for generating diverse images. [35] maximized the mutual information between input and output patches using contrastive learning for learning the commonalities between two domains.

Although some improvements have been made, the majority of previous techniques do not perform well on the complex outdoor scenery images. One common reason stems from the fact that, previous models did not fully represent an embedding space where the input image content is well preserved and fully mapped with the target domain information. To overcome these limitations, we propose a generative model that obtained state-of-the-art results on synthesizing outdoor scenery images given a domain and a style image. Unlike previous models that manipulate images in the target domain using a noise variable, our model generates more realistic images where a user can control the output style image using a reference image. Our model learns the semantic relationship between the objects in an end-to-end framework. To maintain the dependency between the objects we construct a graph convolutional network on the source and target images. For more realistic and diverse image creation, we encourage the network to maximize the mutual information between the reference and output images. We put weights on the mutual objects in the style image and output and penalize those that are different. We learn this through contrastive learning framework which has driven recent advances in learning representations [7, 13, 35]. Figure 1 illustrates an overview of our proposed model. In this paper we make the following contributions:

- We introduce SCONE-GAN which can synthesize an image in the highly complex natural scenery in an end-to-end framework using unpaired images.
- Maintaining the content and image structure between the input and output images is critical in a realistic image translation. We utilise graph convolutional networks to build the object dependencies to preserve and maintain the image structure during translation.
- We demonstrate the efficiency of contrastive learning for a diverse image-to-image translation. We leverage the power of contrastive learning by putting more weight on the objects that are perceptually similar and penalize those objects that are different for a realistic and diverse image-to-image translation.
- Controlling the style and content of generated images cannot be simply achieved by manipulating a noise vector. Therefore, we introduce a style-reference image that is used for stylizing the target image.

2. Related work

The image-to-image translation approaches can be classified into two categories of paired [20, 31, 36, 37] and unpaired training methods [2,17,19,24,27,35]. [20] introduced a General-purposed conditional GAN model [31] to learn a one-side mapping function from the input images to target images. [33] synthesised images using random noise and a corresponding class label. [36] learned to synthesise image given a segmentation mask and a style reference image. One common problem with the paired training approaches is that they only operate in a supervised setting when the paired training data is available.

Since providing a paired training sample is difficult, numerous methods have been introduced to tackle this limitation [4, 17, 19, 24, 27, 35, 43]. [27] considered coupled-GANs for learning a joint probability distribution of two unpaired examples to learn translation. [4] used a weight-sharing strategy to learn a common representation across domains. [49] used a cycle-consistency loss [48] to learn a mapping from translation and reconstruction of the input and output images. [46] enforced a smoothness regularization term over the CycleGAN network to preserve consistent mappings during the translation. They showed for a better translation, the inherent property of samples should be preserved. [43] considered a self-supervised module to preserve image content during translation.

While maintaining image content is crucial for image translation, a successful translator should be able to accurately transform the image appearance conditioned on the target domain information. It has been shown that [5, 16, 35, 42] maximizing mutual information between the input data and image representations, can improve the visual representation learning and reconstruction performance. [42] showed for image colorization, maximizing the similarity between the reference and target images can improve the results. [35] maximized the mutual information between input and output patches for one-sided image translation, and obtained better results.

In this work we build our unpaired image-to-image translator based on the assumption that a successful method should maintain the image content and be more reliant on the input image [43, 46]. To enforce the generator learn the input contents we build a graph on the semantic segments extracted from the image. This encourages generator to maintain the objects as well as their relationship during translation. For generating more diverse and higher quality images, we introduce style reference image and maximize the similarity between the output image and reference image. By maximizing the mutual information between the matched objects, we ensure the generator inherits the significant representations that are crucial for image creation. This enforces the generator to produce more diverse and realistic images. Finally, our model allows user control over the style of input image using style images.

3. SCONE-GAN

Problem Definition: Given the input domain $\mathcal{X}$, the output domain $\mathcal{Y}$, a reference style image $\mathbf{S}$, trainable parameters θ, and a set of unpaired examples $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ and $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$, the goal is to learn a mapping function $f(\mathbf{X}, \mathbf{Y}, \mathbf{S}; \theta) : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$ from $\mathcal{X}$ to $\mathcal{Y}$. This mapping function is an extension to a typical image-to-image translation as defined in [1, 3, 27, 46, 49] where $f(\mathbf{X}, \mathbf{Y}; \theta)$ maps $\mathbf{X}$ to a new domain $\mathbf{Y}$ using a Gaussian noise distribution. However, synthesizing a scene under new conditions cannot be easily controlled by using only a Gaussian noise vector, since more realistic and diverse image generation is required in the output domain. In our work, we introduce style image $\mathbf{S}$ for image manipulation, with the goal to synthesise images in the target domain using a reference style image, control the style and content of $\mathbf{Y} \in \mathcal{Y}$. For more realistic and diverse image generation we consider $\mathbf{S}$ having mutual properties as the input image (for example it is drawn from the same dataset as $\{\mathbf{X}, \mathbf{Y}\}$); however, we can ease this constraint and assume $\mathbf{S}$ is coming from any arbitrary distribution.

Overview of Approach: As explained earlier, a successful image translator should maintain the image structure in $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{Y}$. To encourage the generator to maintain the structural relationship between the objects, we build a graph on the semantic segments extracted from the input and output images. For generating more diverse and higher quality images, we introduce style reference image and maximize the mutual information between the matched objects in the output and the style reference image. This encourages the generator to inherit the significant features for stylizing images in the new domain. We generate style-vector from $\mathbf{S}$ to reflect the feature vector in the output images. There are two benefits in this setting: (i) during training the generator learns to output diverse images; (ii) it enables users to control the style of images in the target space. For example, not only can a user translate images between domains e.g. summer2winter, but she can also control the style of the output image (e.g., less snow on the mountain and more on the ground).

3.1. Network architecture

This section explains details regarding the modules that are used in this work.

Generator: Since the generator needs to learn style reference image characteristics as well as synthesising images using random noise, we feed the input image $\mathbf{x} \in \mathbf{X}$, style-vector $\mathbf{s} \in \mathbb{R}$ and the latent code $\mathbf{z} \in \mathbb{R}$ to the generator. The generator consists of four down-sampling blocks, four intermediate blocks, and four up-sampling blocks, which use pre-activation residual units [14]. We use the instance normalization (IN) [41] while down-sampling and utilize the adaptive instance normalization (AdaIN) [18] for up-sampling blocks. The style-vector and latent code are concatenated and injected into all AdaIN layers.

Style encoder: The encoder maps the reference style image to the style-vector $\mathbf{s} \in \mathbb{R}$. The encoder has a CNN layer and six pre-activation residual units [14] followed by two branches of fully connected layer size 64×2. We set the number of branches as two (binary translation).

Latent mapping network: The input of the mapping network is Gaussian noise vector and a domain. The output of the mapping encoder is a vector that feeds into the generator. The mapping network has eight fully-connected layers that takes the input vector and outputs a vector of size 64×2.

Discriminator: The discriminator has a CNN layer and six pre-activation residual units [14] and a fully-connected layer that is applied to the last residual block. The output of discriminator is a real/fake classifier for each branch.

3.2. Learning spatial dependencies

As mentioned earlier, a successful image-to-image translator should maintain the image structure while synthesizing the input image in the new domain [6, 46]. Therefore, learning the spatial relationship between the objects in the input image is vital for an accurate and realistic image generation. We learn the image structure using graph convo-

Figure 2. **a.** The graph convolutional network which learns the structures among the objects. We build two GCN on the input image and generated image. We use simCLR [7] to extract features from the segmented images., **b.** Our mutual information maximization, tries to learn the feature contents from the "Reference Style Image" in order to generate more realistic and diverse images. We segment the objects from both reference image and the output. We put weight on the objects that are similar and penalize the objects that are different.

lutional networks (GCN) which have been widely used in different computer vision problems [11, 32]. See [47] for a recent survey on models and applications of graph convolutional networks.

3.2.1 Graph Convolutional Networks

Typically graph convolutional networks (GCN), $\mathcal{G}(V, \mathbf{A})$ can be defined as a set of vertex nodes $V = \{v_1, ..., v_n\}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$, a symmetric (typically sparse) adjacency matrix. Each node v_i in the graph has a corresponding d-dimensional feature vector obtained from a linear operation on the $k-$th input $\mathbf{Q}^k = [\mathbf{q}_1^k, ..., \mathbf{q}_n^k] \in \mathbb{R}^{n \times d}$ where $\mathbf{Q}^0$ is the input and $k \in \{0, \ldots, L\}$ for each layer L. Given the input to a GCN layer $\mathbf{Q}^k \in \mathbb{R}^{n \times d}$, it produces $\mathbf{Q}^{k+1} \in \mathbb{R}^{n \times d_{k+1}}$ as the $k+1$-th output of the GCN(.) layer:

$$\mathbf{Q}^{k+1} = \rho(\mathbf{A}\mathbf{Q}^k\mathbf{\Theta}), \tag{1}$$

where ρ is a non-linearity function (Leaky ReLU [44] in our case) and $\mathbf{\Theta} \in \mathbb{R}^{d_k \times d_{k+1}}$ is a set of GCN parameters. For simplicity we consider adjacency operator learned as:

$$A_{i,j}^k = \phi(|\mathbf{q}_i^k - \mathbf{q}_j^k|), \tag{2}$$

where ϕ is a symmetric function. Here we consider a neural network that has 5 layers of convolutional layers and a fully connected layer which works on the absolute difference between two feature vectors. $A_{i,j}^k = 1$ if $i = j$ implies each vertex in the graph is self-connected. The final graph would be obtained by stacking Eq. 1 for L times:

$$\mathcal{G}(\mathbf{Q}; \mathbf{\Theta}) = \sigma(\text{GCN}_L(...\text{GCN}_1(\mathbf{Q}^0))), \tag{3}$$

where σ is a softmax function.

3.2.2 Graph convolution for spatial dependencies

As introduced in Section 3.2.1, a GCN can model the relations among different objects and learn powerful representations for object localization. Therefore, we use a GCN module for learning the spatial relations between the objects in the input and generated images.

We assume $\mathbf{Q}^0$ as the input signal to a GCN_1 module. We select n objects from the input image and pass them to a constractive learning module. We use a segmentation model to extract n objects from each image. Objects are selected as the most prominent objects per dataset. We use simCLR [7] which learns representations that are invariant under a set of augmentations through a contrastive loss. The intuition behind using a contractive learning framework is that, the output of an image-to-image translator must be a variation of the input image, therefore simCLR should consider the extracted segments from input and output as different views of same objects and should generate same $d-$dimensional features. If an object doesn't exist in the image a vector of zeros is put as the extracted features. We build two GCN on the input and output images (see Figure 2a. for an example) and minimize the following loss for learning the spatial dependencies between the objects:

$$\mathcal{L}_{spatio} = \|\mathcal{G}(\mathbf{Q}; \mathbf{\Theta}) - \mathcal{G}(\mathbf{Q}'; \mathbf{\Theta}')\|_2^2, \tag{4}$$

where $\mathbf{Q}$ and $\mathbf{Q}'$ are the features extracted from n segmented objects from input and output images respectively.

3.3. Mutual information maximization

As explained earlier, we intend to introduce a robust, realistic and diverse image generator. GCN can maintain the image structure between the input and output images, however it doesn't capture the necessary information (e.g. texture, color and etc.) for generating diverse images. In addition, we would like SCONE-GAN learns the style of reference image for image manipulation in the target domain. As introduced in [16] mutual information maximization between the reference image and generated output can improve the reconstruction quality. With mutual information maximization, the generator's prediction distribution will be balanced and better reflect the style of reference image in the output space. Inspired by [25] we use mutual information maximization for our image-to-image translation. Followed by [25] the mutual information maximization can be broken in two parts: $I(\mathbf{S}; \mathbf{Y}) = H(\mathbf{S}) - H(\mathbf{Y}|\mathbf{S})$, maximizing the info-entropy $H(\mathbf{Y})$ and minimizing the conditional entropy $H(\mathbf{Y}|\mathbf{S})$, where $\mathbf{Y} \in \mathcal{Y}$ is the generated image. This objective will enforce the generator to generate more diverse images in the target domain [10]. As is shown in [34] Noise-Contrastive Estimation (NCE) is a lower bound for maximizing mutual information. Therefore for maximizng the mutual information between the style image and the output we use the following loss function:

$$\mathcal{L}_{info}(\mathbf{Y}) = \mathbb{E}_{\mathbf{x}}\left[-\log\left(\frac{\sum_{j=1}^{S_p} e^{\tilde{\mathbf{s}}_j \frac{\tilde{\mathbf{y}}^+}{\eta}}}{\sum_{j=1}^{S_p} e^{\tilde{\mathbf{s}}_j \cdot \frac{\tilde{\mathbf{y}}^+}{\eta}} + \sum_{j=1}^{S_n} e^{\tilde{\mathbf{s}}_j \cdot \frac{\tilde{\mathbf{y}}^-}{\eta}}}\right)\right], \tag{5}$$

where η is a constant, S_p and S_n correspond to the set of positive and negative samples extracted from the encoded reference style image $\tilde{\mathrm{s}}$ and encoded generated image $\tilde{\mathbf{y}}$. Positive samples are defined as those objects that are same in the style image and output image, and negative ones are those that are different. In other words, we put weights on the objects that are mutual in the style reference image and the output and penalized the objects that are different. Figure 2b. visualizes the concept of our proposed mutual information maximization setting.

3.4. Final objective

In addition to the objectives we presented in Section 3.2.2 and Section 3.3, we use the *Adversarial loss, Cycle consistency* and *Style modelling loss*.

Adversarial loss: We use the adversarial loss [12] for training the generator and discriminator simultaneously:

$$\begin{aligned}\mathcal{L}_{adv}(G, D) = \mathbb{E}_{\mathbf{x},y}[\log D(\mathbf{x})]+ \\ \mathbb{E}_{\mathbf{x},y,\mathbf{z}}[\log(1 \quad - D_y(G(\mathbf{x}, \hat{\mathrm{s}})))],\end{aligned} \tag{6}$$

where G tries to minimize this objective against an adversarial D that tries to maximize it. In Eq. 6, $\hat{\mathrm{s}}$ is the output of the style encoder given the style image, $\mathbf{S}$, and target domain y. During training, the generator learns to predict images that are indistinguishable from the real images of domain y given the style image $\mathbf{S}$.

Cycle consistency loss: In image-to-image translation to learn the mapping between the unpaired input and output images a cycle-consistency loss is required [49]:

$$\mathcal{L}_{cycle} = \|\mathbf{x} - G(G(\mathbf{x}, \hat{\mathrm{s}}), \tilde{\mathrm{s}})\|_1, \tag{7}$$

where $\tilde{\mathrm{s}} = E(\mathbf{x})$ is the output of the encoder given the input image $\mathbf{x}$.

Style modelling: As mentioned in Section 3.3, the generator is encouraged to generate realistic and diverse images. However, style reference image and the output may have a very little mutual information. This may result very large values in Eq. 5 and promotes mode collapse for the generator. To reduce this behaviour we introduce style modelling loss as follows:

$$\mathcal{L}_{style} = \|\hat{\mathbf{s}} - E_y(G(\mathbf{x}, \hat{\mathbf{s}}))\|_1, \tag{8}$$

Final objective: Finally we optimize the weight summation of the losses in Section 3.2.1, Section 3.3 and Section 3.4. The entire loss function is defined as follows:

$$\begin{aligned}\min_{G,E,M} \max_{D} \lambda_{adv}\mathcal{L}_{adv} + \lambda_{spatio}\mathcal{L}_{spatio}+ \\ \lambda_{info}\mathcal{L}_{info} + \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{style}\mathcal{L}_{style}\end{aligned} \tag{9}$$

where λ are a set of hypo-parameters that are tuned manually during training.

4. Optimization and Inference

For network training similar to [12], instead of minimizing $\log(1 - D_y(G(\mathbf{x}, \hat{\mathbf{s}}))$ for training G, we maximize $\log D(\mathbf{x}, G(\mathbf{x}, \hat{\mathbf{s}}))$. We also use R_1 regularization [30] with $\gamma = 1$. Furthermore, we set $\lambda_{adv} = 1$, $\lambda_{spatio} = 1$, $\lambda_{info} = 2$, $\lambda_{cycle} = 2$ and $\lambda_{style} = 2$. We use the Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.99$. The learning rates for generator, discriminator, and the encoder, are set to 10^{-4}, and for the mapping network is set to 10^{-6}. We initialize all weights of the convolutional, fully-connected, and affine transform layers using $\mathcal{N}(0, 1)$. The biases and noise scaling factors are initialized to zero, except biases associated with the scaling vectors of AdaIN that are set to one. For our encoder, we use leaky ReLU with $\alpha = 0.2$ [28] and equalized learning rate [21] for all layers. We do not use batch normalization, spectral normalization, attention mechanisms, dropout, or pixelwise feature vector normalization [22].

5. Experiments

This section provides details regarding the datasets we used for evaluation, the evaluation metrics and the results.

To explore the performance of SCONE-GAN, we evaluate our method on a variety of unpaired datasets that contain several objects with different shapes:

Yosemite dataset: This dataset contains 854 winter images and 1273 summer images of Yosemite national park [49].

Monet2photo dataset: This dataset has 1072 images of painting and 6287 images of landscape [49].

Nordlandsbanen dataset: [39] This dataset consists videos from a train journey on the same railway track once in every season (spring, summer, autumn, winter). Each video features seasonal effects like snow, color changing foliage and different weather and lighting conditions. We select 5500 unpaired frames from the summer and winter videos for training and 280 paired images for testing.

Cityscape dataset: The dataset contains 2975 training and 500 testing images from the Cityscapes dataset [9].

5.1. Evaluation Metrics

We use the following evaluation metrics:

FID: To evaluate the quality of the generated images, we use FID [15]. FID measures the distance between the generated distribution and the real one through the extracted features by Inception Network [40]. Lower FID values indicate better quality of the generated images.

LPIPS: We use LPIPS [45] to evaluate the diversity of the generated images. LIPIS measures the average feature distances between generated samples. Higher LPIPS score indicates better diversity among the generated images.

NDB and **JSD:** Measure the similarity between the distribution between real images and generated samples [38] and

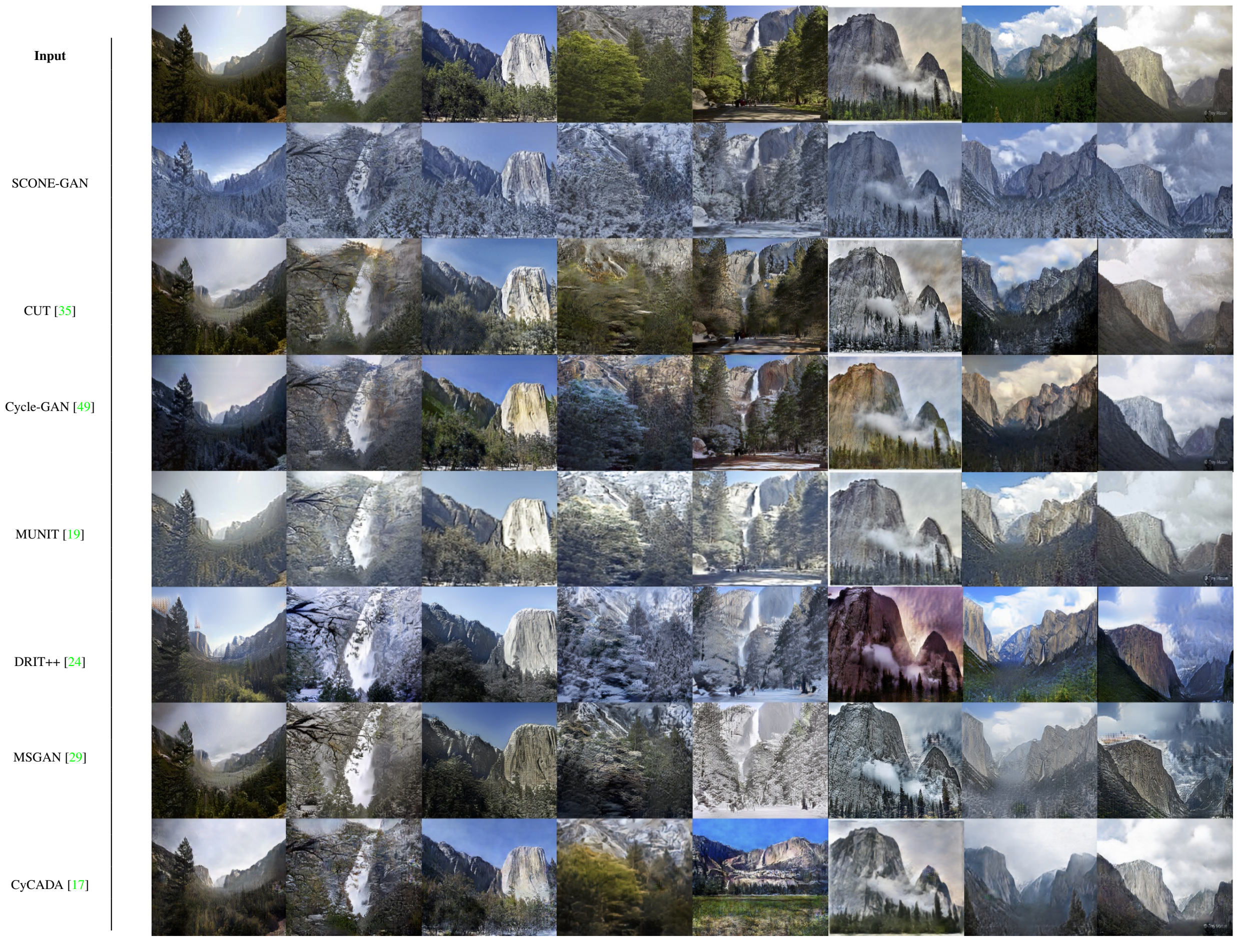


Figure 3. Examples of generated images using our proposed framework. The top row shows the original images in summer domain and the lower images shows the generated winter image using different methods.

	Yosemite [49]				Cityscape [9]				Monet2photo [49]			
Method	FID	LPIPS	NDB	JSD	FID	LPIPS	NDB	JSD	FID	LPIPS	NDB	JSD
CUT [35]	64.17	0.31	31.74	0.049	57.14	0.14	27.05	0.029	66.51	0.28	28.79	0.085
DRIT++ [24]	61.12	0.24	28.03	0.056	72.47	0.17	22.62	0.049	71.65	**0.53**	**21.14**	0.087
MUNIT [19]	66.64	0.21	33.61	0.062	84.53	0.22	34.54	0.065	72.42	0.51	23.47	0.091
CycleGAN [49]	71.20	0.13	45.73	0.057	76.30	0.13	29.67	0.058	77.85	0.35	26.78	0.098
MSGAN [29]	52.65	0.13	25.30	0.042	79.15	**0.26**	23.61	0.034	71.73	0.41	24.65	0.079
CyCADA [17]	54.69	0.33	25.18	0.034	69.31	0.20	21.15	0.028	68.41	0.42	27.46	0.075
SCONE-GAN	**51.70**	**0.40**	**22.51**	**0.031**	**56.37**	0.19	**20.54**	**0.021**	**63.24**	0.49	22.76	**0.074**

Table 1. Quantitative results for the Yosemite, Cityscape and Monet2photo datasets. We report FID (lower is better ↓), LPIPS (higher is better ↑) NDP (lower is better ↓), and JSD (lower is better ↓).

also evaluate the extent of mode missing of generative models. Following [38], the training samples are first clustered using K-means into different bins. Then each synthesized sample is assigned to the bin of its nearest neighbor. Then the bin-proportions of the training samples and the synthesized samples are calculated to evaluate the difference between the generated distribution and the training distribution. Lower NDB and JSD values mean the generated data distribution approaches the real data distribution better.

5.2. Setup

During training we use data augmentation. We flip images horizontally with a probability of 0.5. We resize all the images to 256 × 256, the batch size is set to 8 and all models are trained for 100K iterations. We use [23] method for segmenting the images.

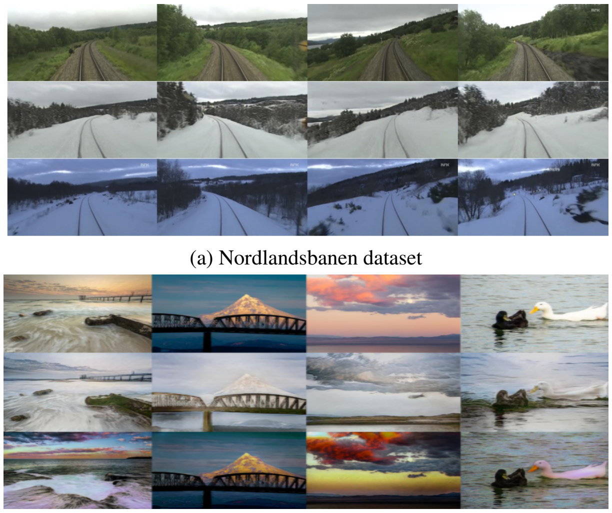
5.3. Results

Qualitatively: First we qualitatively compare our method with six baselines MUNIT [19], CycleGAN [49], DRIT++ [24], CUT [35], MSGAN [29] and CyCADA [17]. Since previous approaches mostly synthesised images in the output domain given a noise variable, in this experiment we do not use any reference image. Figure 3 compares SCONE-GAN with the introduced baselines for summer2winter evaluation. Our method encourages the generator to preserve the image structure while transforming images between domains. Therefore, during translation, the artifacts are minimum in the output images. In addition, we enforce the model to maximize the mutual information between the output and reference images for enhancing the diversity of generated images in the output space. As is shown in Figure 3, the generated images by SCONE-GAN are almost more meaningful and have less artifacts than other methods and better representing an image in winter domain (the trees, mountain and ground are better covered by snow). We also test SCONE-GAN on Nordlandsbanen dataset [39]. The results for this experiment are illustrated in Figure 4a. For this experiment we compare our results with the ground truth images which have been captured in winter. As is visualized in Figure 4a the generated results are realistic, and statistically similar to the real images. We have also compared our model on Monet2photo dataset [49]. The generated results for this experiment is shown in Figure 4b. For this evaluation we compare our model with ”CUT” [35] which have obtained state-of-the-art results on Monet2photo dastset [35]. As can be seen from the figure, SCONE-GAN preserves the image content and better translates the input image in the target domain.

Quantitative: As the quantitative results exhibited in Table 1, SCONE-GAN is outperforming the selected baseline in almost all metrics. We obtain the lowest FID [15] against the baseline, suggesting that the generated images are more realistic and have better image quality. We have also achieved improvements on LPIPS [45], NDB and JSD [38] values in comparison with the baseline confirming that our method can generate more diverse images. We observed that for Cityscape [9] and Monet2photo [49] datasets the segmentation module [23] sometimes fail to extract the accurate objects. In addition, simCLR [7] did not generate robust features on some classes because it was not trained on such images.

5.4. Amazon Mechanical Turk Experiment

We conducted a user study through Amazon Mechanical Turk where users were asked to measure the perceptual realism of the generated images. We show 180 images from Yosemite test dataset [49], 90 real and 90 fake, to 50 participants and asked to distinguish real from fake. We report average classification $\text{Accuracy}_{\text{time}}$ score, the minimum time in second that participants need to see an image and classify it as a real or fake image, in Table 2. In this experiment, we compare our method with CyCADA [17] and CUT [35]. In the gray columns we report the average classification accuracy of all participants. To have a better comparison and ensure that the participants have a better judgment, we choose the participants who have gained average classification accuracy of 80% or above on the real images. This experiment is shown in the cyan color column. As can be seen from this experiment, SCONE-GAN has obtained better results on human perceptual evaluation. In this table, higher classification accuracy on real images shows the participants were more successful on classifying real images and lower classification accuracy on fake images shows the method successfully fooled the participants.



(a) Nordlandsbanen dataset

(b) Monet painting dataset

Figure 4. **a.** Examples of generated images on the Nordlandsbanen dataset. The top row shows the input images. The middle row shows the generated images in the winter domain and the third row shows the ground truth (recorded by a camera in winter). **b.** Transferring images into Monet painting photos. First row is the original image, second row and third row show the transformed images using SCONE-GAN and CUT [35] respectively.

	$\text{Accuracy}_{20}\%$		$\text{Accuracy}_{\infty}\%$		$\text{Accuracy}_{20}\%$		$\text{Accuracy}_{\infty}\%$	
Method	Real	Fake	Real	Fake	Real	Fake	Real	Fake
SCONE-GAN	**95.82**	**28.26**	**93.54**	**41.94**	**60.13**	**47.58**	**61.04**	**47.28**
CUT [35]	95.85	40.78	93.25	45.01	59.78	51.46	60.64	50.25
CyCADA [17]	95.70	30.46	93.68	49.92	60.72	51.23	60.48	50.91

Table 2. AMT experiment. This table shows the average classification accuracy for SCONE-GAN, CUT [35] and CyCADA [17]. Higher classification accuracy on real images means the participants were more successful on classifying real images and lower classification accuracy on fake images shows the method were more successful on fooling the evaluators.

5.5. Reference-guided Image Synthesis

We can also use SCONE-GAN for synthesising images in a new domain conditioned on a style image. We first

	Yosemite dataset [49]				Cityscape dataset [9]				Monet2photo dataset [49]			
Method	FID	LPIPS	NDB	JSD	FID	LPIPS	NDB	JSD	FID	LPIPS	NDB	JSD
$\lambda_2 = 0, \lambda_{3,5} = 2$	63.96	0.25	39.74	0.049	68.46	0.21	24.13	0.038	69.17	0.39	25.14	0.084
$\lambda_2 = 1, \lambda_{3,5} = 0$	68.54	0.22	42.79	0.051	70.12	0.23	27.92	0.052	71.43	0.36	27.44	0.087
$\lambda_{2,3,5} = 0$	70.81	0.14	44.69	0.055	77.46	0.23	28.45	0.061	75.24	0.34	28.16	0.089

Table 3. Ablation experiments on components of the SCONE-GAN. $\lambda_{2,3,5}$ refers to $\lambda_{spatio,info,style}$. By setting $\lambda_2 = 0$ we are cancelling the GCN method and by putting $\lambda_3 = 0$ and $\lambda_5 = 0$ we are removing mutual information maximization block from SCONE-GAN.

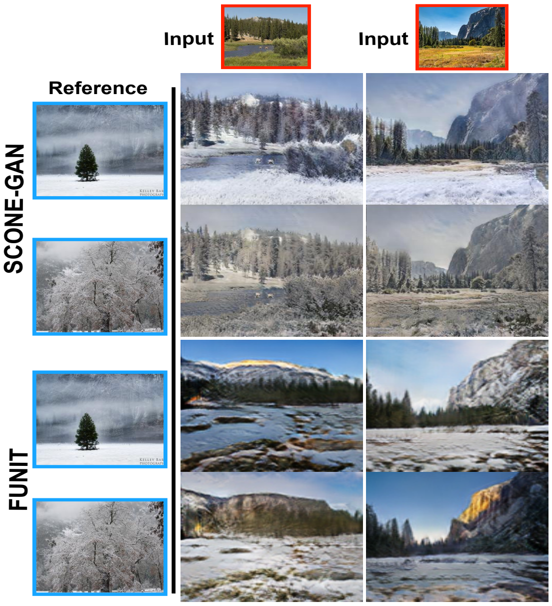


Figure 5. Reference-guided image synthesis. Given the input image and a reference style image, SCONE-GAN and FUNIT [26] synthesize summer2winter images in the target domain.

encode style image, **S**, to a reference style-vector and then feed the features into the generator for producing images in the new domain. By enforcing the generator to maximize the mutual information between the generated and reference images, our generator learns to produce more realistic and diverse images based on the information given in the reference image. Figure 5 shows examples from Yosemite dataset [49]. We also compare our results with FUNIT [26] method, as they have obtained state-of-the-art results on image translation using style image. As are shown in Figure 5, SCONE-GAN produces higher quality images with less artifacts. Also, trees, mountain and ground are better covered by snow and are higher correlated with the reference image.

5.6. Ablation study

We conduct abundant ablation experiments to analyze the components of SCONE-GAN. We follow the same process for training as we explain in Section 4 and Section 5.2. We first explore the effect of GCN 3.2.2 on SCONE-GAN by putting $\lambda_{spatio} = 0$ in Eq. 9. As can be seen from Table 3, the performance of SCONE-GAN drops significantly. Removing GCN block causes the model not to capture the precise dependency among the objects. In addition, we examine the effect of mutual information maximization block by setting $\lambda_{info} = 0, \lambda_{style} = 0$. In this experiment although the objects are preserved during the translation, however the generator doesn't learn to accurately synthesise images in the output space. Finally, we set $\lambda_{spatio} = 0, \lambda_{info} = 0, \lambda_{style} = 0$. In this experiment, we only use Adversarial loss, Eq. 6 and Cycle consistency loss, Eq. 7. Results from Table 3 for this experiment confirm that using $\mathcal{L}_{spatio}, \mathcal{L}_{style}$ and $\mathcal{L}_{info}$ can enormously improve the results across different metrics.

5.7. Limitations

SCONE-GAN relies on the contrastive learning and a segmentation frameworks. Therefore, the output quality may be deteriorated as these models failed on the input images. For example as is shown and discussed in Table 1, for Cityscape [9] and Monet2photo [49] experiments, segmentation [23] and simCLR [7] models failed on some classes. The reason stems from the fact that these models were not initially trained on these tasks (streets or painting photos).

6. Conclusion

We introduced SCONE-GAN for generating realistic and diverse scenery images. Our approach enforces the generator to learn the dependency between objects using graph convolutional network to maintain the structure of the images during translation. In order to extend the diversity of generated images we maximize the mutual information between the style image and the output. During training, the generator learns to utilize the necessary information for enhancing the image diversity. Moreover, this presentation enables users to manipulate scenery in different style conditions. Our qualitative and quantitative experiments on four datasets show that the generated images outperforms quality and diversity of the current state-of-the-art.

Acknowledgement

We would like to acknowledge our industry partner, Rheinmetall Defence Australia. None of this research presented in this paper would have been possible without their support in project DIMSIM (Deep Intelligence Machine SIMulator), part of the ACW Program.

References

[1] Ehsan Abbasnejad, Iman Abbasnejad, Qi Wu, Javen Shi, and Anton van den Hengel. Gold seeker: Information gain from policy distributions for goal-oriented vision-and-langauge reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13450–13459, 2020.

[2] Iman Abbasnejad, Sridha Sridharan, Simon Denman, Clinton Fookes, and Simon Lucey. From affine rank minimization solution to sparse modeling. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 501–509. IEEE, 2017.

[3] Iman Abbasnejad, Sridha Sridharan, Dung Nguyen, Simon Denman, Clinton Fookes, and Simon Lucey. Using synthetic data to improve facial expression analysis with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1609–1618, 2017.

[4] Yusuf Aytar, Lluis Castrejon, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Cross-modal scene networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2303–2314, 2017.

[5] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545, 2019.

[6] Dina Bashkirova, Ben Usman, and Kate Saenko. Adversarial self-defense for cycle-consistent gans. *arXiv preprint arXiv:1908.01517*, 2019.

[7] 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.

[8] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[10] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3950, 2020.

[11] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[13] 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.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[15] 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. *arXiv preprint arXiv:1706.08500*, 2017.

[16] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[18] 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.

[19] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

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

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[23] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2879–2888, 2020.

[24] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128(10):2402–2417, 2020.

[25] Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking distributional matching based domain adaptation. *arXiv preprint arXiv:2006.13352*, 2020.

[26] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 10551–10560, 2019.

[27] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[28] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.

[29] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1429–1437, 2019.

[30] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[31] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[32] Thien Nguyen and Ralph Grishman. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[33] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[35] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020.

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[37] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016.

[38] Eitan Richardson and Yair Weiss. On gans and gmms. *arXiv preprint arXiv:1805.12462*, 2018.

[39] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, page 2013. Citeseer, 2013.

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[41] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[42] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018.

[43] Xinpeng Xie, Jiawei Chen, Yuexiang Li, Linlin Shen, Kai Ma, and Yefeng Zheng. Self-supervised cyclegan for object-preserving image-to-image domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 498–513. Springer, 2020.

[44] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[46] Rui Zhang, Tomas Pfister, and Jia Li. Harmonic unpaired image-to-image translation. *arXiv preprint arXiv:1902.09727*, 2019.

[47] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[48] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 117–126, 2016.

[49] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.