Towards Fine-Grained Control over Latent Space for Unpaired Image-to-Image Translation

Lei Luo[1(✉)], William Hsu[1], and Shangxian Wang[2]

[1] Kansas State University, Manhattan, KS 66506, USA
{leiluoray,bhsu}@ksu.edu
[2] Johns Hopkins University, Baltimore, MD 21210, USA
wshangx1@jhu.edu

Abstract. We address the open problem of unpaired image-to-image (I2I) translation using a generative model with fine-grained control over the latent space. The goal is to learn the conditional distribution of translated images given images from a source domain without access to the joint distribution. Previous works, such as MUNIT and DRIT, which simply keep content latent codes and exchange the style latent codes, generate images of inferior quality. In this paper, we propose a new framework for unpaired I2I translation. Our framework first assumes that the latent space can be decomposed into content and style sub-spaces. Instead of naively exchanging style codes when translating, our framework uses an interpolator that guides the transformation and is able to produce intermediate results under different strengths of translation. Domain specific information, which might still exist in content codes, is excluded in our framework. Extensive experiments show that the translated images using our framework are superior than or comparable to state-of-the-art baselines. Code is available upon publication.

Keywords: Unpaired image-to-image translation · Latent space · Content codes · Style codes · Fine-grained control

1 Introduction

Image-to-image (I2I) translation refers to translating images from one domain to another with different properties. An example is the task of turning images of cartoon sketches into real-life graphs. Many tasks in computer vision can be posed as I2I translation, such as image inpainting [1], style and attribute transfer [2,3], and super-resolution [4]. Paired I2I transfer tasks require paired data sets that are costly to acquire, and such tasks are relatively easier to solve than their unpaired counterparts. Chen and Koltun translated paired images of semantic map to photographic images by taking a regression approach [5]. Isola et al. framed paired I2I translation tasks using conditional generative models [6]. Our work addresses the more challenging unpaired I2I task, where no paired data sets are available. Most of works on unpaired I2I translation draw inspiration

I. Farkaš et al. (Eds.): ICANN 2021, LNCS 12893, pp. 408–420, 2021.
https://doi.org/10.1007/978-3-030-86365-4_33

from CycleGANs using the cycle consistency constraint [7], and have achieved impressive results. These models, however, often have little control over the translation strength and can only provide a single translated image as output. Furthermore, they often disentangle latent space into domain-invariant (content codes) and domain-specific parts (style codes). When translating, content codes are kept while style codes are exchanged. Domain-specific information, however, might still exist in content codes, which leads to unnatural translation results if they are not removed [8].

In this work, we show a need for fine-grained control over latent space by demonstrating the inferior translation capability in previous works that solely depend on the cycle consistency constraint or translate images by simply exchanging style codes. Fine-grained control over latent space are manifested in three aspects: 1) latent codes can be decomposed into content and style, much like DRIT [9] and MUNIT [10]; 2) an interpolator, which is a neural network, is employed to guide the transformation of style codes instead of simply exchanging them; and 3) domain-specific information in content codes is removed before translation for better translation results. Similar to DRIT and MUNIT, our framework assumes that the latent space can be decomposed into content space and style space by the content encoder and the style encoder, respectively. Before decoding the latent codes to obtain translated results, redundant domain-specific information that exists in content codes is removed. Furthermore, another set of modules, which we call the interpolator, smoothly guide the transition of style codes and allow us to generate intermediate images under different degrees of transformation. In the end, our framework differentiates translated images by using a discriminator. Extensive experiments demonstrate that our method is superior than or comparable to state-of-the-art (SOAT) baselines in unpaired I2I translation.

2 Related Work

Generative Adversarial Networks. Ideally, generative models learn how data is distributed, thus allowing data synthesis from the learned distribution. Since the advent of GANs [11], generative models have achieved impressive results in various tasks like image editing [12] and style transfer [3]. GANs try to learn the data distribution by approximating the similarity of distributions between the training data and the fake data produced by the learned model. GANs usually comprise a generator and a discriminator. The entire model learns by playing a minimax game: the generator tries to fool the discriminator by gradually generating realistic data samples, and the discriminator, in turn, tries to distinguish real samples from fake ones. GANs have been improved in various ways. To produce more realistic samples, an architecture of stacked GANs have been proposed: the laplacian pyramid of GANs [13]; layered, recursive GANs [28]; and style-based GANs [2,3]. Several studies have attempted to solve the instability training of GANs using energy-based GANs [14], Wasserstein GANs [15], and boundary equilibrium GANs [16]. In this study, we use GANs with their

improved techniques to learn the distribution of image data and translate them among different domains.

Unpaired I2I Translation. Unpaired I2I translation translates images from one domain to another without paired data supervision. Much success in unpaired I2I translation is due to the cycle consistency constraint, proposed in three earlier works: CycleGANs [7], DiscoGANs [17], and DualGANs [18]. Recent systems such as MUNIT [10] and DRIT [9] were developed to perform multimodal I2I translation, which refers to producing images with the same content but different contexts. For example, a winter scene could be translated into many different summer scenes depending on weather or lighting. To translate more than two domains, StarGAN-V2 [19] and ModularGANs [20] were proposed. I2I translation methods using GANs that merely rely on cycle consistency constraints usually suffer from the issue of discreteness, which refers to inability to continuously control the transformation strength. In this study, we use an interpolator to guide the translation, which allows us to generate visually appealing intermediate translation results.

Our framework is closely related to MUNIT on that the latent space can be decomposed into a style sub-space and a content sub-space. Our framework, however, differs from MUNIT in four aspects: 1. Instead of having to train $n(n-1)$ sets of encoder-decoder for translating images between n domains, our framework consists of only one such set that works for multi-domains; 2. Our framework does not impose a Gaussian prior distribution for style codes, and instead learns the distributions during training; 3. Our framework removes redundant domain-specific information in content codes before translation, thus generating more natural-looking results; 4. Most unpaired I2I translation models that depend on the cycle-consistency loss cannot generate sequences of intermediate translation results. We employ an interpolator module that helps smoothly translate the latent codes of different domains and produces visually satisfying intermediate translation results.

3 Methods

3.1 Preliminaries

Let $x_m \in X_m$ and $x_n \in X_n$ be two images from domain X_m and domain X_n. Our goal is to estimate the conditional distributions $p(x_m|x_n)$ and $p(x_n|x_m)$ using the learned distribution $p(x_{n\to m}|x_n)$ and $p(x_{m\to n}|x_m)$, given the marginal distribution of $p(x_m)$ and $p(x_n)$ but without requiring access to the joint distribution of $p(x_m, x_n)$. Figure 1 shows an overview of our model. Our framework starts with an encoder $E = (E_s, E_c)$ that maps images from image space to latent space, where E_s is the style encoder and E_c is the content encoder. The latent codes consist of style latent codes (s_m, s_n) and content latent codes (c_m, c_n), where $(c_m, s_m) = (E_c(x_m), E_s(x_m))$ and $(c_n, s_n) = (E_c(x_n), E_s(x_n))$. After style codes are obtained, an interpolator I helps transform the style codes across different domains. The translated style codes $s_{m\to n}$ and $s_{n\to m}$ are obtained by calculating $s_m + \alpha * I_{mn}(s_n - s_m)$ and $s_n + \alpha * I_{nm}(s_m - s_n)$, where α is the transformation

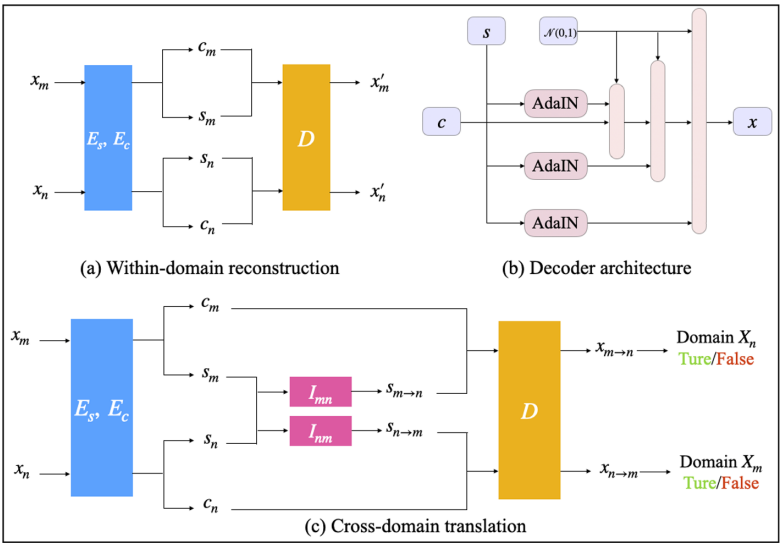


Fig. 1. The structure of our framework. (a) shows within-domain image reconstruction; (b) shows key components of the decoder. The number of convolutional layers are more than what the graph shows; (c) shows cross-domain translation.

strength. Style is injected into the decoder by AdaIN [21] operations. Before injecting the style of the target domain, we remove domain-specific information by injecting the negative style of the same domain, and the strength of the negative style is learned during training. Inspired by StyleGAN [2], we introduce stochastic variation into our model by injecting noise into the decoder. After the transformed style codes are obtained, the decoder D decodes the style and content codes back to image space, thus generating translated images x_{mn} and x_{nm}, where $x_{mn} = D(c_m, s_{m\to n})$ and $x_{nm} = D(c_n, s_{n\to m})$. Finally, the discriminator C tries to differentiate real images from fake ones.

3.2 Framework Architecture

In this section, we outline the architecture of different modules in our framework.

Encoder. Our encoder has two sub-encoders: the style encoder and the content encoder. The content encoder consists of three convolutional layers and four residual blocks [22]. All the layers use ReLU activation function and are followed by an instance normalization (IN) operation [23]. The style encoder starts with five convolutional layers, which are followed by an adaptive average pooling layer and a 1×1 convolutional layer. All layers in the style encoder use ReLU activation function except for the pooling layer.

Decoder. The decoder maps latent codes, which consist of style codes and content codes, to the original image space. The style codes go through a mapping network and are then injected into the decoder by AdaIN [21] operations. The

mapping network is a three layer multi-layer perceptron network. Each layer except for the last one is followed by ReLU. Before injecting the style of the target domain, we remove redundant domain-specific information by injecting the negative style of the source domain, and the strength of the negative style is learned via training. Taking transferring the image x_m from the domain X_m to the domain X_n as an example. We first remove domain specific information in content codes c_m by using AdaIN$(-\beta_m * s_m)$, where β_m is learned via training. Then, we inject the style codes s_n of image x_n by using AdaIN again, which is AdaIN(s_n). Inspired by StyleGAN [2], we introduce stochastic variation into our model by injecting Gaussian noise into the decoder. Our decoder consists of four residual blocks using AdaIN, and two sets of upsample and convolution layers. The last layer is a convolution layer with hyperbolic tangent activation function.

Interpolator. Our framework has an interpolator to guide style codes transiting from one domain to another. The interpolator has three convolutional blocks. The first two use ReLU activation function, and the last one does not have any activation function.

Discriminator. We use a multi-scale discriminator, whose architecture is similar to the one proposed by [29], to distinguish real images from fake ones. At each scale, images go through five convolutional layers before being downsampled. The losses at three scales are accumulated for calculating the final discriminator loss. The discriminator also works as a domain predictor, which consists of a three layer multi-layer perceptron with ReLU activation function except for the last one.

3.3 Loss Functions

In this section, we discuss the loss functions and the training algorithm of our framework.

Image Reconstruction Loss. After images are encoded to style and content codes, the decoder can map them back to the image space and reconstruct the image. Therefore, the image reconstruction loss of x_m is formulated as:

$$L_{recon}^{x_m} = \left\| D(E_c(x_m), E_s(x_m)) - x_m \right\|_1, \tag{1}$$

and $L_{recon}^{x_n}$ is expressed similarly. After images are translated from one domain to another, the images in the source domain can be reconstructed by inverting the process. For example, x_{mn} has the content of image x_m and the style from domain X_n. x_{mn} is obtained by evaluating $D(c_m, s_n)$. Encoding x_{mn} again produces (c'_m, s'_n), and by decoding $D(c'_m, s_m)$, we can reconstruct x_m, which is now denoted by x_{mnm}. Thus, we calculate $L_{recon}^{x_{mnm}}$ as follows:

$$L_{recon}^{x_{mnm}} = \left\| x_{mnm} - x_m \right\|_1 = \left\| D(E_c(x_{mn}), E_s(x_m)) - x_m \right\|_1. \tag{2}$$

Similarly, $L_{recon}^{x_{nmn}} = \left\| x_{nmn} - x_n \right\|_1$. The reconstructed images should be consistent with the semantics of the original images, so we penalize perceptual loss to minimize the semantic difference:

$$L_{perc}^{x_m} = \left\| \Phi_3(D(E_c(x_m), E_s(x_m))) - \Phi_3(x_m) \right\|^2, \tag{3}$$

where Φ_3 denotes the ReLU3_1 layer of a pretrained VGG network [27]. We can similarly calculate the perceptual loss for $L_{perc}^{x_n}$, $L_{perc}^{x_{mnm}}$, and $L_{perc}^{x_{nmn}}$.

Latent Code Reconstruction Loss. By encoding the translated images, we can obtain a new set of content and style codes. For example, encoding the translated image x_{mn} produces (c'_m, s'_n). We construct the latent code reconstruction loss as follows:

$$L_{recon}^{c} = \|c'_m - c_m\|_1 ; L_{recon}^{s} = \|s'_n - s_n\|_1 . \tag{4}$$

Interpolation Loss. Given latent codes of two domains one can interpolate latent codes in a linear fashion. For example, $s_m + \alpha * (s_n - s_m)$ translates s_m to s_n under translation strength α. This approach, however, does not guarantee smooth-looking results as the translation path might not be linear. We employ an interpolator to smoothly transit style codes of different domains, which is calculated as $s_m + \alpha * I_{mn}(s_n - s_m)$. α controls the translation strength and is a random value that is uniformly sampled from 0 to 1. Regarding to domain labels, however, we adopt a linear interpolation strategy. That is to say, we linearly interpolate the domain labels using the same α and use the interpolated domain label as ground truth. The intuition behind this is that linearly interpolated images are supposed to have linearly interpolated labels, but linearly interpolated images are not guaranteed to be smooth-looking. Therefore, an interpolator network is trained to guide the translation. The discriminator C is trained to produce realistic fake images and also to also predict domains of images, and we use the binary cross entropy (BCE) loss and adversarial loss jointly to train the interpolator. The BCE loss function for the interpolator I_{mn} is calculated as:

$$L_{I_{mn}} = \text{BCE}(C(x_{mn}), gt\_domain), \tag{5}$$

where x_{mn} is a translated image via $D(c_m, s_m + \alpha * I_{mn}(s_n - s_m))$ and gt_domain is the ground truth domain label, which is linearly interpolated via $\text{label}_m + \alpha * (\text{label}_n - \text{label}_m)$. $L_{I_{nm}}$ can be calculated similarly.

Regularizers on Style and Content Codes. To further encourage style codes being domain-variant and content codes being domain-invariant, we add regularizers on style and content codes. The style regularizer forces style codes of different domains to be different by minimizing L_{regu}^{s}, which is calculating as:

$$L_{regu}^{s} = - \|D(c_m, s_m) - D(c_m, s_n)\|_1 - \|D(c_n, s_m) - D(c_n, s_n)\|_1 . \tag{6}$$

The content regularizer encourages content codes of different domains to be similar by minimizing L_{regu}^{c}, which is formulated as:

$$L_{regu}^{c} = \|D(c_m, s_m) - D(c_n, s_m)\|_1 + \|D(c_m, s_n) - D(c_n, s_n)\|_1 . \tag{7}$$

Adversarial Loss. GANs are used to match the distribution of translated results to real image samples, so the discriminator finds real and fake samples indistinguishable. The loss for learning the discriminator C is formulated as:

$$L_C^{x_{mn}} = \mathop{\mathbb{E}}_{c_m \sim p(c_m), s_{m\to n} \sim p(s_n)} [log(1 - C(D(c_m, s_{m\to n})))] + \mathop{\mathbb{E}}_{x_n \sim p(X_n)} [logC(x_n)], \tag{8}$$

where the discriminator C tries to differentiate real images from X_n and translated images x_{mn}. $L_C^{x_{nm}}$ is obtained similarly.

Model Training. We alternately train our discriminator and the rest of modules, which are encoders, decoder, mapping network, and the interpolator. The training procedure of our framework is illustrated in Algorithm 1 using a convergence bound B that is empirically calibrated at 1,000,000.

Algorithm 1: Model training

Result: style encoder $\boldsymbol{E_s}$, content encoder $\boldsymbol{E_c}$, interpolators $\boldsymbol{I_{mn}}$, $\boldsymbol{I_{nm}}$, decoder $\boldsymbol{D}$, and $\beta_{\boldsymbol{m}}$, $\beta_{\boldsymbol{n}}$ that control the strength of negative style injected for removing domain dependent information.

$n = 0$;
while $n < B$ **do**
  Calculate $L_C^{x_{mn}}, L_C^{x_{nm}}$ according to (8);
  Update the discriminator $\boldsymbol{C}$;
  Calculate $L_{recon}^{x_m}, L_{recon}^{x_n}, L_{recon}^{x_{mnm}}, L_{recon}^{x_{nmn}}$ according to (1), (2);
  Calculate $L_{perc}^{x_m}, L_{perc}^{x_n}, L_{perc}^{x_{mnm}}, L_{perc}^{x_{nmn}}$ according to (3);
  Calculate $L_{recon}^{c}, L_{recon}^{s}$ according to (4);
  Calculate $L_{I_{mn}}, L_{I_{nm}}$ according to (5);
  Calculate $L_{regu}^{s}, L_{regu}^{c}$ according to (6), (7);
  Update the decoder $\boldsymbol{D}$, the style encoder $\boldsymbol{E_s}$, the content encoder $\boldsymbol{E_c}$, $\beta_{\boldsymbol{m}}$, $\beta_{\boldsymbol{n}}$, and the interpolator $\boldsymbol{I_{mn}}$, $\boldsymbol{I_{nm}}$;
  $n++$;
end

4 Experiments

In this section we talk about the data sets, baselines, and evaluation metrics that we use for testing our framework.

Data Sets. As in previous research [6,10,12], we use images of shoes and their edge map images, which are generated by [24]. There are 100,000 images of shoes $\leftrightarrow$ edges, and 400 images of them are used for testing; the rest are used as the training data set. The cats $\leftrightarrow$ dogs data set is provided in [10], which contains about 2,300 images of cats and dogs. We retain 100 images of cats and 100 images of dogs for testing with rest for training.

Baselines. We compare our framework against three baseline models developed in recent years. Our framework is closely related to DRIT and MUNIT, which we use as baseline models. StarGAN-V2 [19] was recently proposed and achieved SOTA results on unpaired I2I translation. Therefore, we use StarGAN-V2 as another baseline in our study.

Evaluation Metrics. We evaluate the visual quality using Frechét inception distance (FID) [25] and the diversity of translated images with learned perceptual image patch similarity (LPIPS) [26]. FID measures the discrepancy between two sets of images. For each test image in the source domain, we translate it into a target domain using 10 reference images randomly sampled from the test set of the target domain. We then calculate FID between the translated images and test images in the target domain. We calculate FID for every pair of image domains (e.g. cat $\leftrightarrow$ dog) and report the average value. LPIPS measures the diversity of generated images using the L_1 distance between features extracted from the pretrained AlexNet [30]. For each test image from a source domain, we generate 10 outputs of a target domain using 10 reference images randomly sampled from the test set of the target domain. Then, we compute the average of the pairwise distances among all outputs produced from the same input, which are 45 image pairs. Finally, we report the average of the LPIPS values over all test images. Lower FID values indicate that the two sets of images have similar distributions. Higher values of LPIPS indicate higher diversity of generated images.

Frechét inception distance (FID) [25] and the diversity of translated images with learned perceptual image patch similarity (LPIPS) [26] are commonly used for evaluating I2I translation performance. FID measures the distribution similarity between translation results and test set. LPIPS measures the diversity of generated images. Lower FID values indicate that the two sets of images have similar distributions. Higher values of LPIPS indicate higher diversity of generated images.

5 Results

In this section, we provide the qualitative and quantitative results of the experiments. Ablation study is also provided for evaluating the effectiveness of several key design choices.

Qualitative Results. We show several example translation results by different models in the graph (a) of Fig. 2. To evaluate visual quality of translation results, we utilize the Amazon Mechanical Turk (AMT) to compare our results against the baselines based on user preferences. Given a source image and a reference image, we instruct AMT workers to select the best transfer result among all models. We ask 60 questions for all ten workers. As shown in Table 1, our method slightly outperforms StarGAN-V2 [19] and exceed MUNIT [10] and DRIT [9] for a large margin. Unlike the baselines, which suffer from the issue of discreteness and can only produce one final translation image, our framework can generate sequences of intermediate translation results by interpolating style codes using different translation strengths. The graph (b) in Fig. 2 shows results of translating between the cat and dog domain under different strengths of translation.

Our framework uses $s_m + \alpha * I_{mn}(s_n - s_m)$ during interpolation, which generates smooth-looking intermediate results. Other baselines cannot produce intermediate translation results by default. If we interpolate the style codes linearly using $s_m + \alpha * (s_n - s_m)$, we can see that the translation results by StarGAN-V2 and MUNIT contain artifacts, and the results by DRIT only differ in lighting.

Quantitative Results. The qualitative observations above are confirmed via quantitative evaluations. As Table 2 shows, StarGAN-V2 achieves the lowset FID and highest LPIPS on the `cat2dog` data set among all models, but results by our model are comparable to StarGAN-V2. Translated images by our model on the `edges2shoes` have lower FID and higher LPIPS values than all the baselines.

Table 1. Votes from ATM workers for most preferred style transfer results.

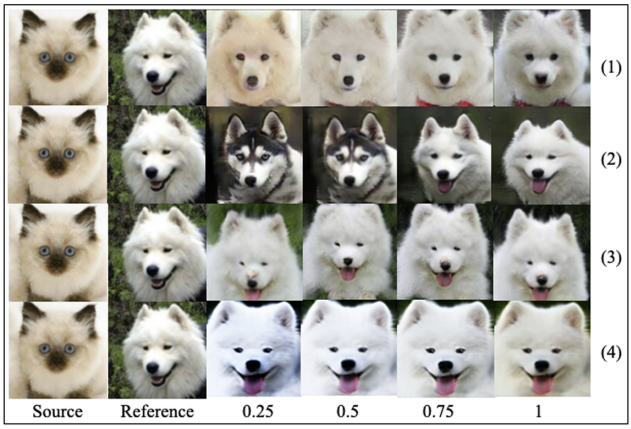
Models	Performance (↑)
MUNIT	13.22%
DRIT	15.06%
StarGAN-V2	35.11%
Ours	**36.61%**

Table 2. Quantitative evaluation of image translation using FID and LPIPS. Cat images are translated to dog images, and edges are translated to shoe images.

Metric	Data set	DRIT	MUNIT	StarGAN-V2	Ours
FID (↓)	cat2dog	148.87	122.04	**18.81**	21.53
FID (↓)	edge2shoes	273.93	274.11	63.78	**61.33**
LPIPS (↑)	cat2dog	0.251	0.263	**0.355**	0.341
LPIPS (↑)	edge2shoes	0.108	0.110	0.114	**0.126**

Ablation Studies. To further validate effects of key loss functions and design choices in our framework, we carry out ablation studies on the `cat2dog` data set. Let the model without domain-specific information removal ($\beta_{\boldsymbol{m}}$, $\beta_{\boldsymbol{n}}$), interpolators, latent codes regularizers, and noise injection be the naive model. We incrementally add modules to the naive model and calculate FID and LPIPS values. The quantitative evaluations are shown in Table 3, and qualitative results are in Fig. 3.

(a) Examples of unpaired I2I translation results from cats to dogs and edges to shoes. The first column are source images, and the second column are images of the target domain. The rest are results from various models.



(b) Examples of translating cats to dogs under different translation strengths by interpolating the style codes between two domains. The values from 0.25 to1 represent different strength levels. (1) is the results by our framework; (2) is by StarGAN-V2; (3) is by MUNIT; (4) is by DRIT.

Fig. 2. Examples of translating results by our framework. (a) compares translation results by different baselines; (b) shows examples of interpolation by all models.

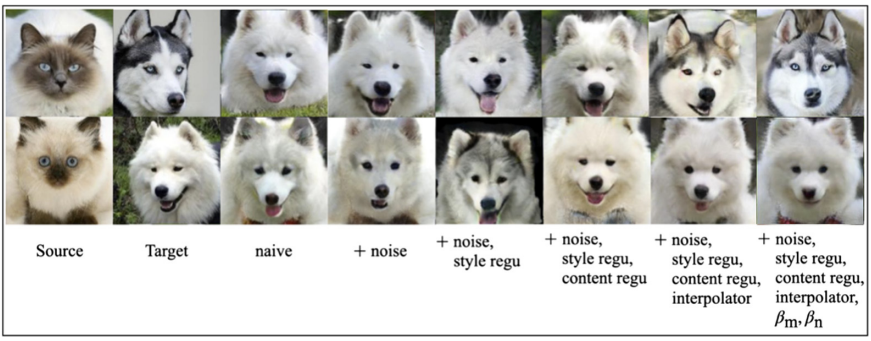


Fig. 3. Ablation study of our framework, which shows examples of translating cats to dogs by incrementally adding modules.

Table 3. FID and LPIPS results of incrementally adding modules to our framework. LPIPS values for the naive model are not reported as it is a deterministic model.

Modules	FID (↓)	LPIPS (↑)
Naive model	103.30	—
+ Noise injection	76.88	0.326
+ Style regularization	59.21	0.329
+ Content regularization	47.70	0.331
+ Interpolators	30.45	0.333
+ Domain-specific Information elimination	**21.53**	**0.341**

6 Conclusions

In this research, we have presented a new framework for unpaired I2I translation. Our framework proposes fine-grained control over latent codes for achieving better translation results. We show that removing redundant domain-specific information during cross-domain translation helps produce better results. We also show that rather than simply exchanging style codes, an interpolator can help guide the transformation to generate more visually appealing images, which also allows us to produce intermediate translation results. The qualitative results and quantitative evaluations show that our framework is superior than or comparable to the SOTA baselines in unpaired I2I translation.

References

1. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: Advances in Neural Information Processing Systems, Montréal, Canada, pp. 331–340. Curran Associates Inc (2018)

2. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, pp. 4401–4410. IEEE (2019)
3. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, pp. 8110–8119. IEEE (2020)
4. Wang, Z., Chen, J., Hoi, S.C.H.: Deep learning for image super-resolution: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 1 (2020)
5. Chen, Q.-F., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, pp. 1511–1520. IEEE (2017)
6. Isola, P., Zhu, J.-Y., Zhou, T.-H., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, pp. 5967–5976. IEEE (2017)
7. Zhu, J.-Y., Park, T., Isola, P., Efros A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 2242–2251. IEEE (2017)
8. Chang, H.-Y., Wang, Z., Chuang, Y.-Y.: Domain-specific mappings for generative adversarial style transfer. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 573–589. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58598-3_34
9. Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., Yang, M.-H.: Diverse image-to-image translation via disentangled representations. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 36–52. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_3
10. Huang, X., Liu, M.-Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 179–196. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_11
11. Goodfellow, I.J., et al.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, pp. 2672–2680. MIT Press, Montreal (2014)
12. Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 597–613. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_36
13. Denton, E.L., Chintala, S., Szlam, A., Fergus, B.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, pp. 1486–149. MIT Press (2015)
14. Zhao, T, Mathieu, M., LeCun, Y.: Energy-based generative adversarial networks. In: 5th International Conference on Learning Representations (ICLR), Toulon, France (2017). OpenReview.net
15. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML), Stockholm, Sweden, pp. 214–223. PMLR (2017)
16. Berthelot, D., Schumm, T., Metz, L.: BEGAN: boundary equilibrium generative adversarial networks. CoRR abs (1703.10717) (2017)

17. Kim, T., Cha, M., Kim, H., Lee, J.-K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, NSW, Australia, pp. 1857–1865. PMLR (2017)
18. Yi, Z.-L., Zhang, H., Tan, P., Gong, M.-L.: DualGAN: unsupervised dual learning for image-to-image translation. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 2868–2876. IEEE (2017)
19. Choi, Y., Uh, Y.-J., Yoo, J., Ha, J.-W.: StarGAN v2: diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, pp. 8185–8194. IEEE (2020)
20. Zhao, B., Chang, B., Jie, Z., Sigal, L.: Modular generative adversarial networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 157–173. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_10
21. Huang, X., Belongie, S.-J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 1510–1519. IEEE (2017)
22. He, K.-M., Zhang, X.-Y., Ren, S.-Q., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770–778. IEEE (2016)
23. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Improved texture networks: maximizing quality and diversity in feed-forward stylization and texture synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 4105–4113. IEEE (2017)
24. Xie, S.-N., Tu, Z.-W: Holistically-nested edge detection. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 1395–1403. IEEE (2015)
25. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, Long Beach, CA, pp. 6626–6637. Curran Associates Inc (2017)
26. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, pp. 586–595. IEEE (2018)
27. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
28. Yang, J.-W., Kannan, A., Batra, D., Parikh, D.: LR-GAN: layered recursive generative adversarial networks for image generation. In: 5th International Conference on Learning Representations (ICLR), Toulon, France (2017). OpenReview.net
29. Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, pp. 8798–8807. IEEE (2018)
30. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, pp. 1106–1114. MIT Press (2012)