

# Rethinking CycleGAN: Improving Quality of GANs for Unpaired Image-to-Image Translation

Dmitrii Torbunov, Yi Huang, Huan-Hsin Tseng, Haiwang Yu,  
Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, Yihui Ren  
Brookhaven National Laboratory, Upton, NY, USA

{dtorbunov, yhuang2, htseng, hyu, jhuang, sjyoo, mlin, bviren, yren}@bnl.gov

## Abstract

An unpaired image-to-image (I2I) translation technique seeks to find a mapping between two domains of data in a fully unsupervised manner. While the initial solutions to the I2I problem were provided by the generative adversarial neural networks (GANs), currently, diffusion models (DM) hold the state-of-the-art status on the I2I translation benchmarks in terms of FID. Yet, they suffer from some limitations, such as not using data from the source domain during the training, or maintaining consistency of the source and translated images only via simple pixel-wise errors. This work revisits the classic CycleGAN model and equips it with recent advancements in model architectures and model training procedures. The revised model is shown to significantly outperform other advanced GAN- and DM-based competitors on a variety of benchmarks. In the case of Male-to-Female translation of CelebA, the model achieves over 40% improvement in FID score compared to the state-of-the-art results. This work also demonstrates the ineffectiveness of the pixel-wise I2I translation faithfulness metrics and suggests their revision. The code and trained models are available at <https://github.com/LS4GAN/uvcgan2>.

## 1. Introduction

Image-to-image (I2I) translation models aim to find a mapping between two domains of images. When paired examples of images from two domains are available, such a mapping can be easily solved in a supervised manner. A more interesting case of I2I problems is the unpaired I2I translation, where examples of pairs are not available. The ability to perform an unpaired I2I translation is highly beneficial since obtaining paired datasets in the real world is often impossible, difficult, or time-consuming [4].

The advancement of unpaired I2I largely benefits from recent developments in deep generating models such as

(variational) Autoencoder [17, 26], generative adversarial networks (GANs), generating flows [36, 11, 12, 25]. One of the early successful unpaired I2I models is CycleGAN [49] that uses a cycle-consistency constraint, requiring that a cyclic back-and-forth translation between two domains results in the original image. Several succeeding models inspired by CycleGAN, such as STARGAN [7, 8], SEAN [50], U-GAT-IT [24], and CUT [34], are designed to further enhance the quality and diversity of the generated images. However, GAN-based I2I methods lag behind the general developments in the GAN architecture and training procedures [22, 39].

An alternative route to image generation is provided by diffusion models [18]. With a recent spike of interest in such models, several applications of DMs to unpaired I2I translation have been developed [6, 29, 45]. Despite being recent, the DM-based EGSDE [45] approach has demonstrated superior results on several benchmarks. However, the DM-based solutions may perform a suboptimal translation, since they do not use source images during the training [45]. Additionally, the DM-based methods rely on pixel-wise  $L_2$  distances to maintain the consistency of the source and translated images. Such a simple consistency measure is not guaranteed to preserve any semantically meaningful features and can restrict image transformations.

In the past, the approach of revisiting a classic neural architecture and improving it with a number of modern additions has led to large improvements in performance [28, 22, 2]. Based on this observation, we revisit one of the earliest GAN-based I2I models – the CycleGAN. Unlike the DM-based models, CycleGAN’s training procedure is able to effectively utilize images from the source and target domains simultaneously. Moreover, CycleGAN maintains an intrinsic consistency between the source and translated images (via the cycle-consistency constraint), a feature that cannot be achieved by simple pixel-wise consistency measures. In addition, recently, UVCGAN [40] work has shown that the CycleGAN performance can be significantly improved by modernizing its architecture.

Motivated by UVCGAN’s success we would like to re-think the classic CycleGAN architecture further. We take UVCGAN as a starting point, and redesign its generator, discriminator and training procedure to obtain a revised model – UVCGANv2.

**Our Contributions.** This work makes several technical improvements to the UVCGAN [40] architecture:

- We redesign the UVCGAN generator model and introduce style modulation to its decoding branch. We propose a style generation mechanism via a learnable Transformer token.
- We propose a modular discriminator architecture made of a traditional discriminator body and a special head, which prevents the problem of mode collapse [15]. We augment the discriminator with a cache of past discriminator encodings, allowing it to directly compare feature statistics between distributions of target and translated images.
- Combined with better training strategies, we demonstrate that the revised UVCGAN model is able to outperform the most advanced competitors by a large margin.
- We highlight the inconsistencies of the current unpaired I2I evaluation protocols and suggest a better faithfulness measure, based on deep image representations of Inception-v3.

## 2. Related Works

Problems related to unpaired I2I translation have been approached from multiple directions. There are two major classes of solutions: GAN-based and diffusion-based.

**GAN-based Methods.** Multiple GAN-based methods have been developed to tackle the problem of unpaired I2I translation. One distinct group of GAN-based methods involves methods that rely on cycle consistency, including CycleGAN [49], DualGAN [42], U-GAT-IT [24], and the recent UVCGAN [40]. This class of algorithms requires two generator networks that translate images in opposite directions. Its basis is a cycle-consistency constraint, requiring that a cyclically translated image should match the original. Cycle-consistent models can show remarkable performance [40], but there are concerns that the cycle-consistency condition might be too restrictive.

ACLGAN [46] attempts to relax the cycle-consistency constraint and replace it with a weaker adversarial one. Such relaxation allows the network to make larger changes to the source image, potentially achieving better translation quality. CouncilGAN [32] moves a step further and completely discards cycle consistency. Instead, it trains an ensemble of generators performing translation in a single direction, allowing for a larger diversity of generated images.

CUT [34] takes an alternative route and uses a contrastive loss to maximize the information between the source and the translated images. This approach removes the need to have multiple generators and allows CUT to

train faster. Using CUT as a basis, ITTR [48] improves its performance by modifying the generator architecture. In a similar fashion, LSeSim [47] designs a contrastive-based loss function that guides the image translation without the need for multiple generators.

**Diffusion-based Methods.** With the recent explosion of interest in diffusion models (DMs) multiple works have attempted to employ them for unpaired I2I translation. For instance, ILVR [6] achieves an unpaired image translation by modifying the standard Gaussian denoising process. It relies on a DM trained only on the target domain but guides it toward the source image during denoising.

SDEdit [29] introduces another viable approach for performing image translation. Instead of modifying the diffusion process itself, it simply changes the starting point of diffusion. SDEdit uses a source image perturbed by Gaussian noise as a seed image and runs the standard diffusion process on top of it.

Finally, recent EGSDE [45] work makes an observation that both ILVR and SDEdit are trained on the target domain data. As such, they may perform a suboptimal translation. EGSDE combines ILVR and SDEdit approaches and modifies both the starting point of the denoising process and the denoising process itself. To overcome the limitation of the DM being trained only on the target domain, it introduces a special energy function, pretrained on both domains. This energy function guides the denoising process, allowing it to achieve state-of-the-art results on several benchmarks.

## 3. Method

UVCGANv2 revisits the classic CycleGAN [49] architecture. UVCGANv2 inherits several advancements from UVCGAN [40] such as a hybrid U-Net-ViT generator architecture, self-supervised generator pre-training, and better training strategies. This section describes several improvements we make over UVCGAN, including the generator, discriminator and the training procedure.

### 3.1. Review of the original UVCGAN

UVCGAN follows the CycleGAN framework [49, 24] that interlaces two generator-discriminator pairs for unpaired I2I translation (Figure 1). Denote the two domains by  $A$  and  $B$ . Generator  $\mathcal{G}_{A \rightarrow B}$  translates images in  $A$  to resemble those from domain  $B$ . Discriminator  $\mathcal{D}_B$  distinguishes images in  $B$  from those translated from  $A$ . The same goes for the other translation direction,  $\mathcal{G}_{B \rightarrow A}$  and  $\mathcal{D}_A$ . Using the notations defined in Figure 1, the discriminators are updated by backpropagating loss in distinguishing real and translated (fake) images (called *GAN loss*):

$$\mathcal{L}_A^{\text{disc}} = \mathbb{E}_B \ell_{\text{gan}}(\mathcal{D}_A(a_f), 0) + \mathbb{E}_A \ell_{\text{gan}}(\mathcal{D}_A(a), 1), \quad (1)$$

$$\mathcal{L}_B^{\text{disc}} = \mathbb{E}_A \ell_{\text{gan}}(\mathcal{D}_B(b_f), 0) + \mathbb{E}_B \ell_{\text{gan}}(\mathcal{D}_B(b), 1) \quad (2)$$

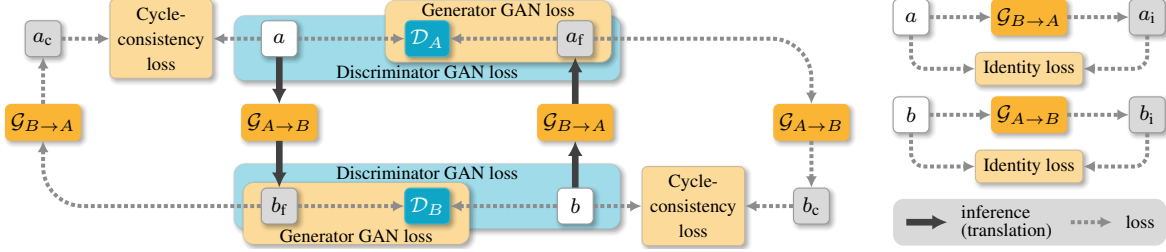


Figure 1. **CycleGAN framework.** The CycleGAN [49] consists of two pairs of GANs,  $(\mathcal{G}_{A \rightarrow B}, \mathcal{D}_B)$  and  $(\mathcal{G}_{B \rightarrow A}, \mathcal{D}_A)$ . The discriminators try to distinguish translations from real images, while the generators (or translators) seek to produce realistic translations that are also consistent with the input. The consistency is enforced by the cycle-consistency loss and (optional) identity loss. Here, we use  $a$  to denote an image from domain  $A$ ,  $b$  as an image from domain  $B$ ,  $(*)_f$  is a fake image (a translation),  $(*)_c$  notes a cyclic reconstruction, and  $(*)_i$  represents an identity reconstruction (when identity losses are used,  $\mathcal{G}_{A \rightarrow B}|_B$  and  $\mathcal{G}_{B \rightarrow A}|_A$  are encouraged to be identity maps).



Figure 2. Droplet-like artifacts produced by the original UVCGAN.

where  $b_f = \mathcal{G}_{A \rightarrow B}(a)$ ,  $a_f = \mathcal{G}_{B \rightarrow A}(b)$  (subscript f means fake), 0 is the label for fake images, 1 is the label for real images, and  $\ell_{\text{gan}}$  represents a classification loss function ( $L_2$ , cross-entropy, Wasserstein [1], etc.). The generators are updated by backpropagating loss from multiple sources: GAN loss for realistic translation, cycle-consistency loss and optionally identity loss for within-domain translation. Using domain  $A$  as an example, we have:

$$\mathcal{L}_A^{\text{gan}} = \mathbb{E}_A \ell_{\text{gan}}(\mathcal{D}_B(b_f), 1), \quad (3)$$

$$\mathcal{L}_A^{\text{cyc}} = \mathbb{E}_A \ell_{\text{reg}}(a_c, a), \quad \mathcal{L}_A^{\text{idt}} = \mathbb{E}_A \ell_{\text{reg}}(a_i, a) \quad (4)$$

where  $a_c = \mathcal{G}_{B \rightarrow A} \circ \mathcal{G}_{A \rightarrow B}(a)$ ,  $a_i = \mathcal{G}_{B \rightarrow A}(a)$ , and  $\ell_{\text{reg}}$  is any pixel-wise loss function ( $L_1$  or  $L_2$ , etc.). The generator loss is defined as a linear combination:

$$\mathcal{L}^{\text{gen}} = (\mathcal{L}_A^{\text{gan}} + \mathcal{L}_B^{\text{gan}}) + \lambda^c (\mathcal{L}_A^{\text{cyc}} + \mathcal{L}_B^{\text{cyc}}) + \lambda^i (\mathcal{L}_A^{\text{idt}} + \mathcal{L}_B^{\text{idt}}) \quad (5)$$

### 3.2. Source Driven Style Modulation

Upon carefully examining images generated by the reference UVCGAN implementation, we find the majority of them exhibit the characteristic droplet-like artifacts (Figure 2). These droplet artifacts are similar to those reported in StyleGANv2 [22]. We eliminate these artifacts by removing all instance normalization layers in the U-Net encoding branch and replacing those in decoding branch with learned style modulations.

Specifically, at the bottleneck of the generator, the image is encoded as a sequence of tokens to be fed to the Transformer network. We augment this sequence with an additional learnable style token  $S$ . The state of the  $S$  token at the output of the transformer serves as a latent image style. For each convolutional layer of the U-Net’s decoding branch we generate a specific style vector  $s_i$  from  $S$ , by trainable linear transformations.

The style modulation [22] effectively scales weights  $w_{i,j,x,y}$  of the convolutional operator by the supplied style vector  $s_i$ , yielding modulated weights:

$$w'_{i,j,x,y} = s_i \cdot w_{i,j,x,y} \quad (6)$$

where  $i, j$  refer to the input and output feature maps and  $x, y$  enumerate the spatial dimensions. To preserve the magnitude of the activations, the scaled weights  $w'_{i,j,x,y}$  need to be demodulated. The demodulation operation further renormalizes the convolution weights as follows:

$$w''_{i,j,x,y} = \frac{w'_{i,j,x,y}}{\sqrt{\sum_{i,x,y} (w'_{i,j,x,y})^2 + \epsilon}} \quad (7)$$

where  $\epsilon$  is a small number to prevent numerical instability.

Our approach is different from the StyleGANv2, which generates style vectors  $s_i$  for each convolutional operation by performing different affine transformations on the common vector  $w$ . This vector is obtained from a random non-learnable latent code by a multilayer perceptron network. The way how our  $S$  is processed is similar to the [class] token of the ViT [13], however their token are mainly used for classification task.

### 3.3. Batch Statistics Aware Discriminator

In our experiments, the PatchGAN discriminator architecture, used in the original UVCGAN, frequently causes a partial mode collapse [15]. To workaround this problem, we apply a variation of the Minibatch discrimination technique [37, 21].

Motivated by the neural architectures of the contrastive

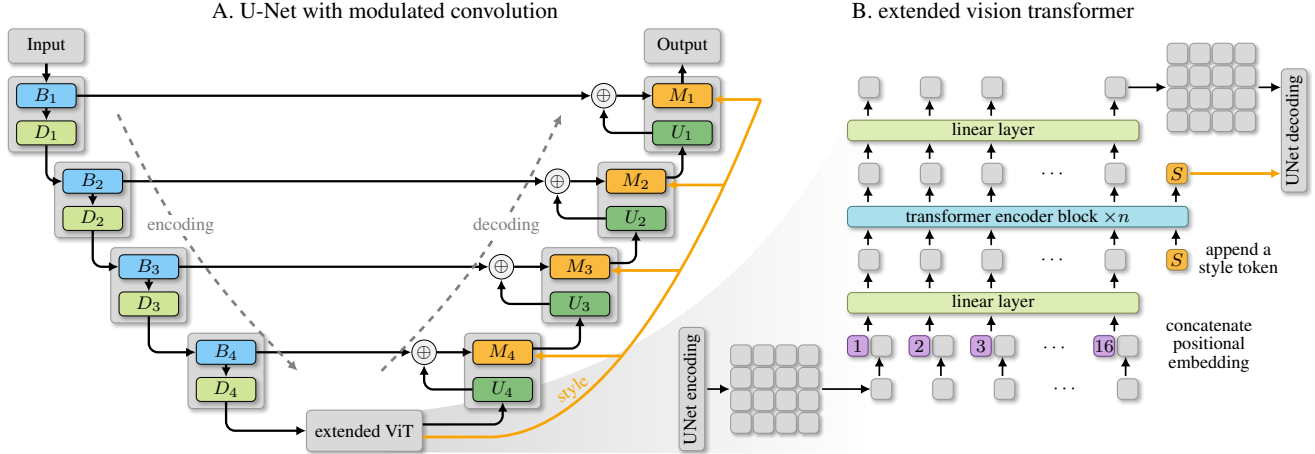


Figure 3. **UVCANv2 Generator.** The generator of UVCANv2 is a U-Net (Panel A) with an extended vision transformer bottleneck (eViT, Panel B). The eViT outputs a style token for the modulated convolution blocks [22] ( $M_i$ ,  $i = 1, 2, 3, 4$ ) in the decoding path of the U-Net. Refer to [40] for details about the input layer, output layer, basic block  $B_i$ , downsampling block  $D_i$ , and upsampling block  $U_i$  in the U-Net and the transformer encoder block in the eViT.

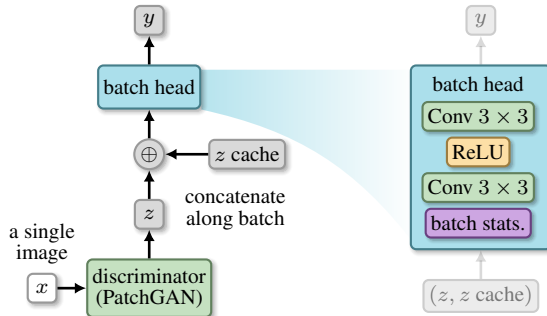


Figure 4. **UVCANv2 Discriminator with a batch head.** For the batch statistics block, we can use either a standard batch normalization layer or batch standard deviation [21].

methods [5] we design a composite discriminator made of a main *body* and a *batch head*. The body of the composite discriminator can be any common discriminator, but, for the purposes of this work, we use PatchGAN without the last layer.

The batch head is designed to equip the discriminator with a minibatch discrimination power and prevent the mode collapse. It is made of a layer that captures batch statistics, followed by two convolutional layers (Figure 4). Such a modular discriminator architecture allows one to easily swap different discriminator bodies, while still preserving the minibatch discrimination power of the batch head.

**Batch Statistics Layers.** Many neural layers can be used to capture batch statistics. In this work, we test two types of layers: batch standard deviation (BSD), introduced by ProGAN [21], and a simple Batch Normalization (BN), which has been found effective for preventing mode collapse in representation learning [14].

**Cache of Discriminator Features.** For the minibatch discrimination method to work, the model must be trained with batch sizes greater than 1. However, in our experiments, CycleGAN models achieve much better performance (for the same time/compute budget) when trained with a batch size of 1. To reconcile the batch size of 1 and the minibatch discrimination technique, we maintain a history (cache) of past inputs to the batch head.

During training, the batch feature statistics are stored in four separate caches: real images from domain  $a$ , real images from domain  $b$ , and fake images from both domains. All the caches have a fixed size and follow the first-in-first-out (FIFO) update policy.

The discriminator’s batch head receives a concatenation (along the batch dimension) of the discriminator body output for the current minibatch along with a history of the past outputs from a cache (Figure 4). The usage of feature caches allows disentangling the size of the minibatch from the size of the statistical sample of features provided to the batch head. It also synergizes with the composite discriminator architecture, allowing one to cache outputs of the discriminator body, which are expensive to recompute.

### 3.4. Pixel-wise Consistency Loss

To improve the consistency of the generated and source images, we experiment with the addition of an extra term  $\mathcal{L}_{\text{consist}}$  to generator loss (5). This term captures the  $L_1$  difference between the downsized versions of the source and translated images. For example, for images of domain  $A$

$$\mathcal{L}_{\text{consist},A} = \mathbb{E}_A \ell_1 (F(\mathcal{G}_{A \rightarrow B}(a)), F(a)) \quad (8)$$

where  $F$  is a resizing operator down to  $32 \times 32$  pixels (low-pass filter). We add this term to the generator loss (5) with a magnitude  $\lambda_{\text{consist}}$  for both domains.



### 3.5. Modern Training Techniques

UVCGAN and CycleGAN use outdated GAN training techniques. Hence, we revamp the training procedure with a few modern additions, which include adding exponential averaging of the generator weights [21], implementing spectral normalization of the discriminator weights [31], trying unequal learning rates for the generator and discriminator [16], and replacing the generic gradient penalty (GP) with an improved zero-centered GP version [39].

## 4. Experiments

### 4.1. Datasets

We study the performance of UVCGANv2 on two groups of datasets. The formerly widely used CelebA [27] and Anime [24] datasets and the modern, high-quality CelebA-HQ [21] and AFHQ [8] datasets. More details about these datasets can be found in Appendix A.

**CelebA and Anime.** The CelebA and Anime datasets have been commonly used to benchmark GAN-based unpaired I2I translation algorithms [40, 49, 32, 46, 24]. We study UVCGANv2 performance on three tasks related to the CelebA and Anime datasets: Male-to-Female translation on the CelebA dataset, Glasses Removal on the CelebA dataset, and Selfie-to-Anime translation on the Anime dataset. Because the CycleGAN setup learns translations in both directions simultaneously, we also get benchmarks in the opposite directions (Female-to-Male, Glasses Addition, and Anime-to-Selfie translations).

**CelebA-HQ and AFHQ.** To compare the performance of the UVCGANv2 against more recent unpaired I2I translation algorithms, such as EGSDE [45], we also consider the CelebA-HQ and AFHQ datasets. We investigate Male-to-Female translation on CelebA-HQ and three translations: Cat-to-Dog, Wild-to-Dog, and Wild-to-Cat on AFHQ.

**Preprocessing.** For a fair comparison with EGSDE [45], we downsize the CelebA-HQ and AFHQ images to  $256 \times 256$  pixels. To avoid Fréchet inception distance (FID) evaluation inconsistencies associated with a difference in the interpolation procedures between different frameworks [35], we use the Pillow [9] image manipulation library to perform the image resizing with Lanczos interpolation.

### 4.2. Training

When training our modified UVCGAN implementation, we seek to closely follow the original procedure [40]. It consists of two steps. First, pre-training of the generator in a self-supervised way on a task of image inpainting. The second step is the actual training of the unpaired I2I translation networks, starting from the pre-trained generators.

**Generator Pre-training.** For each dataset, the generators are pre-trained on image inpainting tasks. This task is set up in a fashion similar to the Bidirectional

Encoder Representations from Transformers (BERT) pre-training [10, 40]. For the inpainting task, input images of size  $256 \times 256$  pixels are tiled into a grid of patches at  $32 \times 32$  pixels. Then, each patch is masked with a probability of 40%. The masking is performed by zeroing out pixel values. The generator is tasked to recover the original unmasked image from a masked one. More details about this pre-training are available in Appendix B.

**Translation Training.** The unpaired image translation training is performed for 1 million iterations with the help of the Adam optimizer. Depending on the dataset, we use various data augmentations. For the preprocessed datasets, such as CelebA-HQ and AFHQ, only a random horizontal flip is applied. For the Anime and CelebA datasets, we use three augmentations: resize, followed by a random crop of size  $256 \times 256$ , followed by a random horizontal flip. Resizing for the Anime dataset is done from  $256 \times 256$  up to  $286 \times 286$ . For the CelebA dataset, the resizing is done from  $178 \times 218$  to  $256 \times 313$ .

**Hyperparameter Tuning.** For each translation, we perform a quick hyperparameter search, exploring the space of the cycle-consistency magnitude  $\lambda_{\text{cycle}}$ , magnitude of the zero-centered gradient penalty  $\lambda_{\text{GP}}$ , magnitude of the consistency loss  $\lambda_{\text{consist}}$ , learning rates of the generator and discriminator, and the choice of the batch head (BN versus BSD). We also explore turning the learning rate scheduler on and off. Refer to Appendix B for more details about the training procedure.

## 5. Results

### 5.1. Metrics

There are two dimensions along which the unpaired I2I style transfer models can be evaluated: *Faithfulness* and *Realism*. Faithfulness captures the degree of similarity between the source and its translated image at an individual level. Realism attempts to estimate the overlap of the distributions of the translated images and the ones in the target.

The quality of image translation, in terms of realism, is commonly judged according to the FID [16] and kernel inception distance (KID) [3] metrics. Both metrics measure the distance between the distributions of the latent Inception-v3 [38] features extracted from samples of the translated and target images. Smaller FID and KID values indicate more realistic images.

Early GAN-based works (e.g., [49, 32, 46, 24]) do not explicitly evaluate the faithfulness of the translation. To the best of our knowledge, there is no widely accepted faithfulness metric available. Some works [45] try to employ simple pixel-wise  $L_2$ , peak-signal-to-noise ratio (PSNR), or structural similarity index measure (SSIM) [41] scores to capture the agreement between the source and translation. Yet it is unclear how well these pixel-wise metrics relate to



Figure 5. **Sample translations for CelebA and Anime.** We show the translations produced by U-GAT-IT, UVCAN, and UVCANv2 for three tasks: Selfie-to-Anime, Male-to-Female, and Removing Glasses. The full grid with all benchmarking results and that for the three opposite translations can be found in [Appendix F](#).

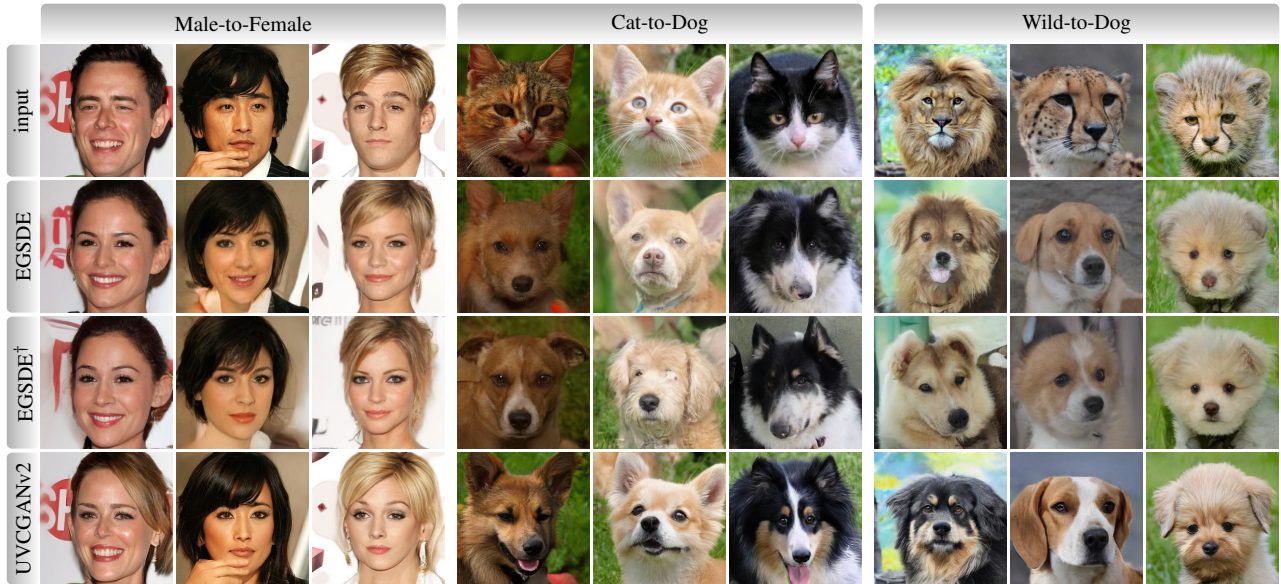


Figure 6. **Sample translations for CelebA-HQ and AFHQ.** We show translations for three tasks: Male-to-Female, Cat-to-Dog, and Wild-to-Dog. More translations for these three tasks and those for Wild-to-Cat can be found in [Appendix F](#).

the perceived image faithfulness.

## 5.2. Evaluation Protocol

Evaluation protocols differ drastically between different papers (see [Appendix C](#)). This makes the direct comparison of the translation quality metrics extremely challenging. For the fairness of comparisons with older works, we follow different evaluation protocols, depending on the dataset.

**CelebA and Anime.** When evaluating the quality of translation on the CelebA Male-to-Female, CelebA Glasses Removal, and Anime datasets, we use the evaluation protocol of UVCAN [40], which uniformized FID/KID evaluation across multiple datasets and models, allowing for a simple FID/KID comparison. For the actual FID/KID evaluation, we rely on `torch-fidelity` [33], which provides a validated implementation of these metrics.

The actual evaluation protocol for CelebA and Anime relies only on test splits to perform the FID/KID evaluation.

For the CelebA dataset, we use KID subset size of 1000. For the Anime dataset, we use the KID subset size of 50. We use unprocessed images of size  $256 \times 256$  when evaluating on the Anime dataset. For the CelebA dataset, we apply a simple pre-processing to both domains: resizing the smaller side to 256 pixels, then taking a center crop of size  $256 \times 256$ .

**CelebA-HQ and AFHQ.** EGSDE [45] has evaluated multiple models on the CelebA-HQ and AFHQ datasets under similar conditions. To compare our results to EGSDE, we replicate its evaluation protocol for CelebA-HQ and AFHQ. For the AFHQ dataset, we evaluate FID and KID scores between the translated images of size  $256 \times 256$  and the target images of size  $512 \times 512$  from the validation split. For the CelebA-HQ dataset, we evaluate the FID/KID scores between the translated images of size  $256 \times 256$  and the downsized target images of size  $256 \times 256$  from the train split. We perform the same channel standard-

ization as EGSDE with  $\mu = (0.485, 0.456, 0.406)$  and  $\sigma = (0.229, 0.224, 0.225)$ . To ensure full consistency, we use the reference evaluation code provided by EGSDE [44]. Appendix E provides results of an alternative evaluation protocol, which is uniform across all the datasets.

### 5.3. Quantitative Results

**CelebA and Anime.** Table 1 shows a comparison of the UVCGANv2 (trained without pixel-wise consistency loss) performance against ACLGAN [46], CouncilGAN [32], CycleGAN [49], U-GAT-IT [24], and UVCGAN [40]. The performance of the competitor models is obtained from the UVCGAN paper [40]. According to Table 1, UVCGANv2 outperforms all the competitor models in all translation directions, except Anime-to-Selfie. The degree of improvement ranges from about 5% in terms of FID on the Selfie-to-Anime translation, to around 51% on the Male-to-Female translation. Likewise, there is a significant improvement in the KID scores from about 13% on Anime-to-Selfie to 79% on Male-to-Female. Such a degree of improvement demonstrates the effectiveness of modern additions to the traditional CycleGAN architecture. Figure 5 provides a few translation samples. More samples can be found in Appendix E.

Table 1. FID and KID scores. Lower is better.

	Selfie to Anime		Anime to Selfie	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
ACLGAN	99.3	3.22 $\pm$ 0.26	128.6	3.49 $\pm$ 0.33
CouncilGAN	91.9	2.74 $\pm$ 0.26	126.0	2.57 $\pm$ 0.32
CycleGAN	92.1	2.72 $\pm$ 0.29	127.5	2.52 $\pm$ 0.34
U-GAT-IT	95.8	2.74 $\pm$ 0.31	<b>108.8</b>	<u>1.48 <math>\pm</math> 0.34</u>
UVCGAN	<u>79.0</u>	<u>1.35 <math>\pm</math> 0.20</u>	122.8	2.33 $\pm$ 0.38
UVCGANv2	<b>75.8</b>	<b>1.18 <math>\pm</math> 0.28</b>	<u>113.8</u>	<b>1.26 <math>\pm</math> 0.23</b>
	Male to Female		Female to Male	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
ACLGAN	<u>9.4</u>	0.58 $\pm$ 0.06	19.1	1.38 $\pm$ 0.09
CouncilGAN	10.4	0.74 $\pm$ 0.08	24.1	1.79 $\pm$ 0.10
CycleGAN	15.2	1.29 $\pm$ 0.11	22.2	1.74 $\pm$ 0.11
U-GAT-IT	24.1	2.20 $\pm$ 0.12	15.5	0.94 $\pm$ 0.07
UVCGAN	9.6	0.68 $\pm$ 0.07	<u>13.9</u>	<u>0.91 <math>\pm</math> 0.08</u>
UVCGANv2	<b>4.7</b>	<b>0.14 <math>\pm</math> 0.02</b>	<b>7.6</b>	<b>0.24 <math>\pm</math> 0.02</b>
	Remove Glasses		Add Glasses	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
ACLGAN	16.7	0.70 $\pm$ 0.06	20.1	1.35 $\pm$ 0.14
CouncilGAN	37.2	3.67 $\pm$ 0.22	19.5	1.33 $\pm$ 0.13
CycleGAN	24.2	1.87 $\pm$ 0.17	19.8	1.36 $\pm$ 0.12
U-GAT-IT	23.3	1.69 $\pm$ 0.14	19.0	1.08 $\pm$ 0.10
UVCGAN	<u>14.4</u>	<u>0.68 <math>\pm</math> 0.10</u>	<u>13.6</u>	<u>0.60 <math>\pm</math> 0.08</u>
UVCGANv2	<b>10.6</b>	<b>0.27 <math>\pm</math> 0.06</b>	<b>11.3</b>	<b>0.34 <math>\pm</math> 0.07</b>

**CelebA-HQ and AFHQ.** Table 2 compares the results of the UVCGANv2 evaluation against CUT [34],

Table 2. FID, PSNR, and SSIM scores.

	Male to Female		
	FID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
CUT	46.61	19.87	<b>0.74</b>
ILVR	46.12	18.59	0.510
SDEdit	49.43	20.03	0.572
EGSDE	41.93	<u>20.35</u>	0.574
EGSDE $\dagger$	30.61	18.32	0.510
UVCGANv2	<u>17.65</u>	19.44	<u>0.681</u>
UVCGANv2-C	<b>17.34</b>	<b>21.18</b>	<b>0.738</b>
	Cat to Dog		
	FID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
CUT	76.21	17.48	<u>0.601</u>
ILVR	74.37	17.77	0.363
SDEdit	74.17	<u>19.19</u>	0.423
EGSDE	65.82	<b>19.31</b>	0.415
EGSDE $\dagger$	<u>51.04</u>	17.17	0.361
UVCGANv2	<b>44.76</b>	15.55	0.562
UVCGANv2-C	52.48	18.30	<b>0.638</b>
	Wild to Dog		
	FID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
CUT	92.94	17.2	<u>0.592</u>
ILVR	75.33	16.85	0.287
SDEdit	68.51	17.98	0.343
EGSDE	59.75	<u>18.14</u>	0.343
EGSDE $\dagger$	<u>50.43</u>	16.40	0.300
UVCGANv2	<b>45.56</b>	15.59	0.551
UVCGANv2-C	55.61	<b>18.65</b>	<b>0.631</b>

ILVR [6], SDEdit [29], and two versions of the EGSDE [45]. In particular, this table compares two versions of the UVCGANv2: UVCGANv2 and UVCGANv2-C. UVCGANv2-C is a version that was trained with a pixel-wise consistency loss and  $\lambda_{\text{consist}} = 0.2$ . The performance of the competitor models is extracted from EGSDE [45].

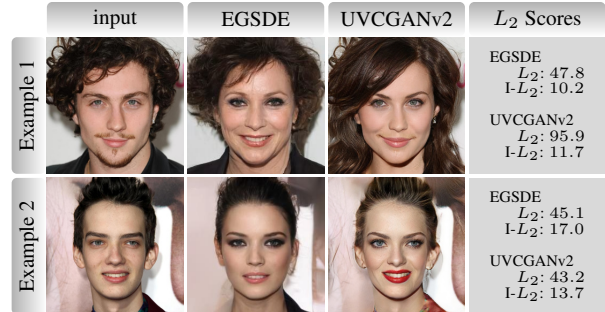
Table 2 shows that UVCGANv2 achieves the best translation quality, according to the FID scores, with improvements ranging from 10% on Wild-to-Dog translation to 43% on Male-to-Female translation. The addition of the consistency loss allows the UVCGANv2-C model to improve its pixel-wise PSNR and SSIM metrics, but at the expense of the FID score on AFHQ translation. UVCGANv2 and UVCGANv2-C achieve competitive SSIM scores but lose in terms of the PSNR ratio to the other models. However, as was pointed out before [43], pixel-wise measures PSNR and SSIM are not good metrics to judge perceptual image faithfulness. In the next subsection, we try to investigate better perceptual faithfulness metrics.

Overall, the gains in SSIM and PSNR scores, provided by the consistency loss to UVCGANv2-C, do not seem to outweigh the associated FID losses. Figure 6 demonstrates a few translation samples, with more samples available in Appendix E.



Table 3. **Measuring Faithfulness with  $L_2$ .** The pixel-wise  $L_2$  is labeled as  $L_2$  and the  $L_2$  between the latent Inception-v3 features is labeled as  $I-L_2$ . More examples, like the two on the right, can be found in [Appendix D](#).

	Male to Female		Cat to Dog		Wild to Dog	
	$L_2 \downarrow$	$I-L_2 \downarrow$	$L_2$	$I-L_2$	$L_2$	$I-L_2$
EGSDE	<b>42.04</b>	14.13	<b>47.22</b>	16.73	<b>54.34</b>	15.20
EGSDE <sup>†</sup>	53.44	15.37	62.06	16.82	66.52	15.44
UVCGANv2	64.19	<b>13.47</b>	77.72	<b>16.39</b>	81.57	<b>14.79</b>
UVCGANv2-C	<u>47.87</u>	<u>13.55</u>	<u>56.53</u>	<u>16.52</u>	<u>58.53</u>	<u>14.85</u>



#### 5.4. Toward Better Faithfulness Measures

The pixel-wise image similarity measures (such as  $L_2$ , PSNR, and SSIM) have been shown [43] to be weakly correlated with human perception of similarity. However, they are currently being used [45] as a faithfulness metric in the area of the unpaired I2I translation.

Following the discussions of perceptual content similarity measures [43, 19], we believe that a proper faithfulness measure should be based on deep image representations. Since the primary goal of this paper is to revisit the classic CycleGAN architecture, we are not going to perform an in-depth investigation of possible faithfulness metrics. Instead, we will briefly consider the usage of the  $L_2$  distance between the latent Inception-v3 [38] features as an alternative faithfulness measure.

In [Appendix D](#), we present a large sample of images, generated by EGSDE and UVCGANv2 models. It helps to investigate how well the pixel-wise  $L_2$  measure is correlated with a perception of image faithfulness. In this part of the paper, we provide two representative samples that demonstrate: 1. pixel-wise  $L_2$  faithfulness measure penalizes image changes that should be freely-modifiable during the translation (e.g. hairstyle); 2. pixel-wise  $L_2$  faithfulness measure fails to effectively penalize changes to features that are expected to be preserved (e.g. background, facial structure, etc).

The first row of images, to the right of [Table 3](#), provides a typical sample of Male-to-Female translations, comparing EGSDE and UVCGANv2 models. Both translation samples are in good agreement with the source image. However, the EGSDE translation has a pixel-wise  $L_2$  difference of 48 to the source, while the UVCGANv2 one has twice as high  $L_2 = 96$ . The high  $L_2$  difference is caused by a large amount of hair added by the UVCGANv2. Naturally, one may expect the hairstyle to be a free parameter of the Male-to-Female translation. Yet, the pixel-wise faithfulness metrics will highly penalize its changes.

The second row of images, to the right of [Table 3](#), demonstrates another sample of Male-to-Female translation, where pixel-wise  $L_2$  are similar between EGSDE ( $L_2 = 45$ ) and UVCGANv2 ( $L_2 = 43$ ). Yet, the im-

ages are rather different in structure. The one from UVCGANv2 preserves the details better: the image background, the bone structure of the face, the shape of the ear, forehead, nose, and even smile. The Inception-v3 scores also correlated with this observation, showing a preference for the UVCGANv2 image (13.7 versus 17.0). These observations suggest that the Inception-v3-based distance measure may be a better faithfulness metric than a simple pixel-wise one. However, we defer the proper investigation for future works. To conclude, we show the comparison of pixel-wise and Inception-v3  $L_2$  faithfulness metrics between EGSDE and UVCGANv2 models in [Table 3](#). This table shows that the UVCGANv2 model provides better faithfulness in terms of the Inception-v3 features, while EGSDE is more faithful if measured in pixel-wise distances between the source and translated images.

## 6. Conclusion

This work revisited the classic CycleGAN framework and built upon the more recent UVCGAN model. We demonstrated that source-driven style modulation and batch statistics aware discriminator are effective techniques to improve the model performance. Our UVCGANv2 was extensively bench-marked on four datasets on nine translation directions. Results show that our model can reach outstanding performance in terms of FID scores. At the same time, we note the absence of the proper faithfulness metric in the area of unpaired I2I translation, and general inconsistency of the evaluation procedures. We believe developing such a measure and uniformizing evaluation will be highly beneficial for future unpaired I2I methods development.

**Potential Negative Social Impact.** Deep generative models have potential to be misused for creating fakes [30]. However, since our models were trained on public datasets, fakes can be easily identified with modern tools [20]. Furthermore, we open-source our code and models, allowing researchers to develop countermeasures.

**Acknowledgement.** The LDRD Program at Brookhaven National Laboratory, sponsored by DOE’s Office of Science under Contract DE-SC0012704, supported this work.



## References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. **3**
- [2] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34:22614–22627, 2021. **1**
- [3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. **5**
- [4] M Boulanger, Jean-Claude Nunes, H Chourak, A Largent, S Tahri, O Acosta, R De Crevoisier, C Lafond, and A Barateau. Deep learning methods to generate synthetic CT from MRI in radiotherapy: A literature review. *Physica Medica*, 89:265–281, 2021. **1**
- [5] 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. **4**
- [6] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. **1, 2, 7**
- [7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. **1**
- [8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. **1, 5, 11, 12**
- [9] Alex Clark. Pillow, python imaging library (fork). **5, 14**
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. **5, 11**
- [11] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. **1**
- [12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. **1**
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. **3**
- [14] Abe Fetterman and Josh Albrecht. Understanding self-supervised and contrastive learning with “bootstrap your own latent” (byol). **4**
- [15] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. **2, 3**
- [16] 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. *Advances in neural information processing systems*, 30, 2017. **5**
- [17] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. **1**
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. **1**
- [19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. **8**
- [20] Felix Juefei-Xu, Run Wang, Yihao Huang, Qing Guo, Lei Ma, and Yang Liu. Countering malicious deepfakes: Survey, battleground, and horizon. *International Journal of Computer Vision*, pages 1–57, 2022. **8**
- [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. **3, 4, 5, 11**
- [22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. **1, 3, 4**
- [23] Junho Kim. GAN\_Metrics-Tensorflow, simple tensorflow implementation of metrics for gan evaluation. [https://github.com/taki0112/GAN\\_Metrics-Tensorflow](https://github.com/taki0112/GAN_Metrics-Tensorflow). **12**
- [24] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019. **1, 2, 5, 7, 11, 12**
- [25] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. **1**
- [26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. **1**
- [27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. **5, 11**
- [28] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. **1**
- [29] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided

- image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 2, 7
- [30] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021. 8
- [31] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5
- [32] Ori Nizan and Ayellet Tal. Breaking the cycle-colleagues are all you need. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7860–7869, 2020. 2, 5, 7, 11
- [33] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. 6, 14
- [34] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 319–345. Springer, 2020. 1, 2, 7, 12
- [35] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022. 5, 12, 14
- [36] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 1
- [37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 3
- [38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5, 8
- [39] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. *arXiv preprint arXiv:1902.03984*, 2019. 1, 5
- [40] Dmitrii Torbunov, Yi Huang, Haiwang Yu, Jin Huang, Shin-jae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren. Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 702–712, 2023. 1, 2, 4, 5, 6, 7, 11, 12
- [41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5
- [42] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017. 2
- [43] 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. 7, 8, 14
- [44] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. EGSDE energy-guided stochastic differential equations. <https://github.com/ML-GSAI/EGSDE>. 7
- [45] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *arXiv preprint arXiv:2207.06635*, 2022. 1, 2, 5, 6, 7, 8, 12, 14
- [46] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 800–815. Springer, 2020. 2, 5, 7
- [47] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16407–16417, 2021. 2
- [48] Wanfeng Zheng, Qiang Li, Guoxin Zhang, Pengfei Wan, and Zhongyuan Wang. Itrr: Unpaired image-to-image translation with transformers. *arXiv preprint arXiv:2203.16015*, 2022. 2
- [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. 1, 2, 3, 5, 7
- [50] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020. 1

## A. Datasets

In this section, we provide additional details about the datasets used in the main paper.

**CelebA [27].** The datasets for the Glasses Removal and Male-to-Female tasks are derived from the original CelebA dataset. For a fair comparison with older models [40], we used the pre-processed versions of Glasses Removal and Male-to-Female datasets provided by CouncilGAN [32]. The CelebA dataset is made of images of size  $178 \times 218$  pixels. The train split of the Glasses Removal task contains about 11K images with glasses and 152K without. The Male-to-Female dataset has about 68K males and 95K females. The test parts of the Glasses Removal and Male-to-Female datasets contain about 3K images with glasses and 37K images without glasses, and 16K males and 24K females respectively.

**Anime-to-Selfie [24].** The training split of the Anime-to-Selfie dataset contains 3400 Selfie images and 3400 Anime images. The test part contains just 100 samples from each domain. All images of this dataset have a size of  $256 \times 256$  pixels.

**CelebA-HQ [21].** The CelebA-HQ dataset has around 10K images of males and about 18K images of females in the train split. The test split contains 1000 male and 1000 female images. The size of a CelebA-HQ image is  $1024 \times 1024$  pixels.

**AFHQ [8].** The AFHQ dataset has around 5.2K cat, 4.7K dog, and 4.7K wildlife images in the train split, and 500 images of each in the test split. The images of the AFHQ dataset have a size of  $512 \times 512$  pixels. There are two versions of the AFHQ dataset provided by StarGANv2 [8]. We use version 1 to be consistent with previous models.

## B. Training Details

In this section, we expand on the generator pre-training, I2I translation training, HP optimization setup, the final model configurations, and the ablation study of UVC-GANv2.

**Generator Pre-Training.** The pre-training of the generators was performed in a BERT-like fashion [10] on an image inpainting pretext task.

To construct the image inpainting task, input images of size  $256 \times 256$  pixels are tiled into a grid of patches of  $32 \times 32$  pixels. Then, each patch is randomly masked with a probability of 40%. The masking is performed by zeroing out pixel values. The generator is tasked to recover the original unmasked image from a masked one.

For the pre-training, we use the AdamW optimizer together with a cosine learning rate annealing (with restarts). We set the initial learning rate to  $5 \times 10^{-3} \times (\text{batch size}/512)$ , and the weight decay factor to 0.05. The scheduler completes 5 annealing cycles during the pre-

training.

We apply several data augmentations, such as random rotation ( $\pm 10$  degree), random horizontal flip ( $p = 0.5$ ), and color jitter ( $\pm 0.2$  shift in brightness, contrast, saturation, and hue).

We pre-train the generators for 500 epochs for CelebA-HQ and AFHQ. Due to the small size of the Anime dataset, we run the pre-training for 2500 epochs. On the contrary, due to the large size of the CelebA dataset, we run the pre-training for 500 epochs, but limit the number of samples per epoch to 32,768. All the pre-trainings are performed with a batch size of 64.

**Image Translation Training.** We train the unpaired I2I translation models by closely following the procedure of UVC-GAN [40]. We use the Adam optimizer without weight decay. The training is performed for 1 million iterations. We experiment with using either a constant learning rate or applying a linear scheduler. If the linear scheduler is used, then the learning rate is maintained constant for the first 500K iterations, and then linearly annealed to zero during the subsequent 500K iterations.

We keep the batch size equal to 1 during the training. For consistency with ProGAN [21] we keep the sizes of the image caches at 3. This size effectively provides the batch head with 4 samples to estimate the batch statistics. To stabilize the generator, we apply an exponential weight averaging to the generator with a momentum of 0.9999.

**Hyperparameter Exploration.** When performing the final training, we explored the following grid of hyperparameters:

- Magnitude of the cycle-consistency loss  $\lambda_{\text{cyc}}$ :  $\{5, 10\}$ .
- Magnitude of the zero-centered gradient penalty  $\lambda_{\text{GP}}$ :  $\{0.001, 0.01, 1\}$ .
- Batch Head type: Batch Normalization (BN) vs Batch Standard Deviation (BSD).
- Generator’s and Discriminator’s learning rates:
  1. Equal learning rates of  $1 \times 10^{-4}$ .
  2. Unequal learning rates, with the learning rate of the discriminator of  $1 \times 10^{-4}$  and the learning rate of the generator of  $5 \times 10^{-5}$ .

The hyperparameter explorations were performed while keeping the magnitude of the consistency loss  $\lambda_{\text{consist}}$  equal to zero. The grid of hyperparameters above was suggested by the previous rough HP exploration.

For the AFHQ Cat-to-Dog, Wild-to-Dog, and CelebA-HQ Male-to-Female datasets, we have run a second hyperparameter exploration, studying the effect of the magnitude of the consistency loss  $\lambda_{\text{consist}}$  on the I2I performance. We have tried the following values of  $\lambda_{\text{consist}}$ :  $\{0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$ .

## B.1. Final Configurations

Table 4. Best Training Configurations.

Dataset	$lr_{gen}$	$\lambda_{GP}$	$\lambda_{cyc}$	B. Head
Anime to Selfie	$5 \times 10^{-5}$	0.01	10	BN
Male to Female	$1 \times 10^{-4}$	0.01	10	BSD
Glasses Removal	$5 \times 10^{-5}$	0.01	10	BSD
HQ Male to Female	$1 \times 10^{-4}$	1.0	5	BSD
Cat to Dog	$1 \times 10^{-4}$	1.0	5	BN
Wild to Dog	$1 \times 10^{-4}$	1.0	5	BN
Cat to Wild	$5 \times 10^{-5}$	1.0	5	BN

For all the translation tasks, UVCGANv2 achieves the best performance when the learning rate scheduler is not used. Table 4 summarizes the final hyperparameter configurations (generator’s learning rate, magnitudes of the gradient penalty and the cycle consistency loss, and the choice of batch head) that provide the best performance per translation task.

Generally, the high-quality datasets (CelebA-HQ and AFHQ) favor stronger values of the gradient penalty term  $\lambda_{GP} = 1$ , compared to the lower-resolution datasets (Anime-to-Selfie and CelebA), favoring  $\lambda_{GP} = 0.01$ . Other patterns of hyperparameters can be observed in the table. However, their impact on the model’s performance is relatively small compared to  $\lambda_{GP}$ .

We should note, that due to the instability associated with GANs training, some of the best values in Table 4 may be due to random fluctuations.

## B.2. Ablation Study

Table 5. Ablation Study of UVCGANv2 on CelebA. ”+” indicates that an option is added to the final UVCGANv2 configuration. ”-” indicates that an option is removed from UVCGANv2.

	Male to Female		Female to Male	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
UVCGAN	9.6	$0.68 \pm 0.07$	13.9	$0.91 \pm 0.08$
UVCGANv2	<b>4.7</b>	<b><math>0.14 \pm 0.02</math></b>	<b>7.6</b>	<b><math>0.24 \pm 0.02</math></b>
(a) - Style	8.1	$0.53 \pm 0.07$	11.1	$0.64 \pm 0.07$
(b) - B. Head	5.5	$0.21 \pm 0.03$	8.6	$0.31 \pm 0.03$
(c) - SN	4.7	$0.14 \pm 0.02$	7.7	$0.25 \pm 0.03$
(d) + Sched	6.3	$0.35 \pm 0.05$	9.5	$0.47 \pm 0.05$
(e) - Avg.	9.8	$0.71 \pm 0.05$	14.2	$0.91 \pm 0.07$

Table 5 summarizes UVCGANv2 ablation results on the Male-to-Female translation of CelebA. To produce this table we start with the final UVCGANv2 configuration and make one of following modifications separately: (a) disable style modulation in the generator; (b) disable batch head of the discriminator; (c) disable spectral normalization (SN); (d) add linear schedule; (e) remove exponential averaging of the generator weights.

According to Table 5, the generator modifications (a) account for the majority of the performance improvement. The removal of these modifications degrades the FID score of the Male-to-Female translation from 4.7 to 8.1. Likewise, the discriminator modifications (b) provide a significant but relatively smaller improvement in the I2I performance. The removal of the spectral normalization (c) decreases the model performance, but the effect is negligible.

Each of the items (d) and (e) of Table 5 may suggest that either the scheduler or exponential averaging of the generator weights is detrimental to the model’s performance. However, this is not the case, since the effects of (d) and (e) are entangled and mutually balancing. Individual modifications to (d) or (e) destroy the balance and produce large changes in the model’s performance. These changes are not indicative of the effects of the joint modifications.

## C. Remarks on Metric Evaluation Consistency

Inconsistency of unpaired I2I evaluation procedures is a widespread problem. For example, some works (e.g. [24]) roll out their own FID evaluation code [23] and report the so-called ”mean” FID and KID scores, where ”mean” means a weighted average of the actual FID/KID scores and various other metrics. Some other works [35] choose different image resizing algorithms, creating a noticeable discrepancy in the reported FID scores.

Another source of FID/KID score inconsistency is the difference in the evaluation protocols. For instance, works like [40] prefer to evaluate FID scores only on images of the test split, yet others [8] evaluate FID scores between translated test images and target images of the train split. Likewise, there is a difference in whether any pre-processing is used in the FID/KID evaluation. For example, one can evaluate FID scores between translated images with the pre-processing and target images without pre-processing [34], or one can apply the pre-processing step to both translated and target images [45]. Moreover, the pre-processing procedures may differ between different works.

To uniformize FID/KID evaluation procedures, we propose a consistent evaluation protocol in Appendix E.

## D. Metrics of Faithfulness to the Source

In this section, we provide more examples to illustrate that  $I-L_2$  (defined as the  $L_2$  distance between the latent Inception-v3 features) may be a more appropriate measurement of faithfulness to the source than pixel-level  $L_2$ . In Table 6, we select translations according to two types of criteria. Denote a translation produced by EGSDE as  $E$  and that by UVCGANv2 as  $U$ .

**Type 1:**  $I-L_2(E) \approx I-L_2(U)$  and  $L_2(U) - L_2(E) > 15$ .

**Type 2:**  $I-L_2(E) - I-L_2(U) > 3$ .



Table 6. Comparing EGSDE and UVCGANv2 translations with  $L_2$  and  $I-L_2$  scores.

		EGSDE	input	UVCGANv2			EGSDE	input	UVCGANv2	
Type 1		$L_2 = 35.7$ $I-L_2 = 12.6$ PIF = 7			$L_2 = 70.5$ $I-L_2 = 10.9$ PIF = 1, 2, 3, 4		$L_2 = 40.9$ $I-L_2 = 11.4$ PIF = 7			$L_2 = 59.0$ $I-L_2 = 11.5$ PIF = 2, 3, 6
		$L_2 = 37.6$ $I-L_2 = 13.2$ PIF =			$L_2 = 54.3$ $I-L_2 = 11.1$ PIF = 2, 3, 4		$L_2 = 48.9$ $I-L_2 = 14.8$ PIF =			$L_2 = 75.2$ $I-L_2 = 11.4$ PIF = 1, 3, 5, 6
		$L_2 = 39.9$ $I-L_2 = 10.6$ PIF = 7			$L_2 = 64.7$ $I-L_2 = 10.8$ PIF = 2, 3		$L_2 = 43.6$ $I-L_2 = 12.6$ PIF = 7, 8			$L_2 = 62.0$ $I-L_2 = 11.9$ PIF = 2, 3, 4
		$L_2 = 41.7$ $I-L_2 = 13.3$ PIF =			$L_2 = 71.2$ $I-L_2 = 10.5$ PIF = 2, 3, 4		$L_2 = 40.9$ $I-L_2 = 12.8$ PIF = 8			$L_2 = 69.0$ $I-L_2 = 11.9$ PIF = 1, 2, 4, 6
		$L_2 = 47.8$ $I-L_2 = 10.2$ PIF = 7, 8			$L_2 = 95.9$ $I-L_2 = 11.7$ PIF = 3, 4, 5, 6		$L_2 = 43.4$ $I-L_2 = 12.8$ PIF =			$L_2 = 80.9$ $I-L_2 = 10.5$ PIF = 1, 2, 3
		$L_2 = 41.0$ $I-L_2 = 11.9$ PIF =			$L_2 = 59.6$ $I-L_2 = 11.5$ PIF = 1, 2, 4, 6		$L_2 = 43.9$ $I-L_2 = 10.6$ PIF =			$L_2 = 71.9$ $I-L_2 = 10.6$ PIF = 2, 3, 4, 8
		$L_2 = 46.2$ $I-L_2 = 11.2$ PIF = 8			$L_2 = 81.5$ $I-L_2 = 10.3$ PIF = 1, 2, 4, 6		$L_2 = 46.2$ $I-L_2 = 11.0$ PIF =			$L_2 = 66.4$ $I-L_2 = 11.1$ PIF = 2, 3, 4, 8
Type 2		$L_2 = 43.6$ $I-L_2 = 13.3$ PIF = 7			$L_2 = 69.3$ $I-L_2 = 10.3$ PIF = 2, 3, 5		$L_2 = 45.6$ $I-L_2 = 18.0$ PIF = 7			$L_2 = 113.7$ $I-L_2 = 14.5$ PIF = 2, 3, 5, 6
		$L_2 = 40.5$ $I-L_2 = 19.2$ PIF =			$L_2 = 48.1$ $I-L_2 = 15.5$ PIF = 2, 3		$L_2 = 49.7$ $I-L_2 = 16.4$ PIF = 6			$L_2 = 38.2$ $I-L_2 = 11.6$ PIF = 1, 2, 3, 5
		$L_2 = 45.1$ $I-L_2 = 17.0$ PIF = 7			$L_2 = 43.2$ $I-L_2 = 13.7$ PIF = 1, 2, 3, 6		$L_2 = 40.0$ $I-L_2 = 14.9$ PIF = 7			$L_2 = 50.0$ $I-L_2 = 9.1$ PIF = 1, 2, 3, 6

Categories of perceived image faithfulness (PIF):

1. background 2. bone structure 3. expression 4. apparent age 5. eye color 6. eyebrows 7. hair color 8. hair texture

**Type 1** is designed to show what contributes to lower pixel-wise  $L_2$  while  $I-L_2$ -s are similar. **Type 2** selects examples with large  $I-L_2$  difference and helps readers to judge if  $I-L_2$  correlates with their own judgment on the similarity to the source (i.e. which pairs look more like siblings.)

We list eight categories of perceived image faithfulness (PIF) in the legend such as background, bone structure, facial expression, and so on. For each translated image, we indicate which categories it outperforms that generated by the other model. For example, for the input in Type 1 row 1 left,

EGSDE preserves the hairstyle and color (PIF=7) better than UVCGANv2, but the UVCGANv2 translation maintains a sharper background (1), preserves the bone structure (2) and expression (3) better, and exhibits more similarity in apparent age (4).

Type-1 examples suggest that pixel-level  $L_2$  is an inappropriate measurement for semantic consistency as UVCGANv2 translations manage to capture characterizing features (such as a bone structure) even with high pixel-level  $L_2$ . In fact, the high pixel-level  $L_2$  of UVCGANv2 translations is often a result of benign modifications such as the elongation of dark hair on a light background (e.g. Type 1 row 5 right) or a slight overall shift to a warmer hue (e.g. Type 1 row 2 right).

On the contrary, the Type-2 examples suggest that  $I-L_2$ , the  $L_2$  on Inception latent features, might be a better measurement of semantic consistency. While EGSDE fails to maintain features such as background, facial expression (neural v.s. smile), eye movement, and prominent bone structure and produces over-generalized translations, UVCGANv2 translations with significantly lower  $I-L_2$  manage to preserve those features and appear more individualized.

These examples illustrate that the pixel-wise  $L_2$  faithfulness metric may be in poor agreement with a human judgment of image faithfulness. They also point to a possibility that the  $I-L_2$  distances, based on deep features of Inception-v3, may better capture the perceived image faithfulness. Such observations mirror the conclusion of [43], about the effectiveness of deep features as perceptual metrics.

However, we stress again, that the main purpose of this paper is to improve the performance of the classic CycleGAN architecture, not the development of better faithfulness metrics. While this section points to a possibility of  $I-L_2$  being a better faithfulness metric, a full-scale investigation needs to be conducted to conclusively establish this. We leave such a study for future work.

## E. Toward Consistent FID Evaluation

The evaluation protocols used in the paper for CelebA-HQ and AFHQ are provided by EGSDE [45]. Being ad-hoc, these protocols lack consistency and differ significantly depending on the dataset. A variety of different evaluation protocols makes the evaluation of the unpaired I2I methods rather quirky and error-prone.

As a step toward consistent FID evaluation, we provide results of an alternative, but consistent evaluation protocol for UVCGANv2 in Table 7. The consistent evaluation protocol uses only test splits (or validation splits if the test ones are not available) of each dataset to assess the quality of image translation.

The evaluation protocol begins with pre-processing all the datasets in a consistent manner. The pre-processing step resizes images from their original size down to  $256 \times 256$

Table 7. Consistent FID and KID scores. Lower is better.

	Female to Male		Male to Female	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
UVCGANv2	29.7	$0.41 \pm 0.18$	24.2	$0.20 \pm 0.15$
	Dog to Cat		Cat to Dog	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
UVCGANv2	24.8	$0.23 \pm 0.13$	44.2	$0.76 \pm 0.23$
	Dog to Wild		Wild to Dog	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
UVCGANv2	18.7	$0.15 \pm 0.14$	44.7	$0.68 \pm 0.23$
	Cat to Wild		Wild to Cat	
	FID	KID ( $\times 100$ )	FID	KID ( $\times 100$ )
UVCGANv2	12.1	$0.01 \pm 0.09$	21.2	$0.20 \pm 0.13$

Table 8. Model Performance versus magnitude of the pixel-wise consistency loss

$\lambda_{\text{consist}}$	Male to Female		Cat to Dog	
	FID	$L_2$	FID	$L_2$
0	24.2	62.6	44.2	77.9
0.01	24.9	61.2	44.5	76.7
0.1	24.8	50.6	45.7	64.9
0.2	25.1	47.9	51.8	56.8
0.4	27.3	43.3	59.1	50.6
0.6	29.7	41.0	71.3	47.5
0.8	32.0	39.1	77.0	46.1
1.0	33.1	37.7	81.2	44.9

pixels (the same image size as is used for model training and inference). To avoid FID score inconsistencies created by aliasing artifacts [35] we rely on the Pillow library [9] and Lanczos interpolation method.

Once the data pre-processing and image translation are done, the actual evaluation can begin. To perform the FID/KID score computation we use a torch-fidelity package [33], which provides a validated implementation of these metrics. The KID evaluation procedure depends on a free parameter – the KID subset size. In this section, we choose the KID subset size of 100 for all the datasets.

The suggested evaluation protocol differs in a number of ways from the evaluation protocols of the AFHQ and CelebA-HQ datasets of EGSDE. It differs from the ad-hoc CelebA-HQ evaluation protocol [45] because the latter compares FID scores between samples of validation and train splits, while the consistent version only uses validation split. The consistent evaluation protocol is also different from the ad-hoc version of the AFHQ one, which performs FID evaluation between translated images of size  $256 \times 256$  and target images of size  $512 \times 512$ . The consistent protocol always uses pre-processed images of size  $256 \times 256$ .

## F. Additional Translation Samples

In this section, we provide additional translation samples to facilitate visual comparison of image quality. [Table 9](#) and [Table 10](#) demonstrate samples on the Anime dataset. [Table 11](#) and [Table 12](#) provide gender swap samples on the CelebA dataset. [Table 13](#) and [Table 14](#) show eyeglasses removal and addition samples on the CelebA dataset. Finally, [Table 15](#), [Table 16](#), and [Table 17](#) provide samples on the AFHQ dataset, and [Table 18](#), Male-to-Female samples on the CelebA-HQ dataset.

## G. Effects of the Pixel-Wise Consistency Loss

The main part of the paper compares models trained with two settings of the pixel-wise consistency loss. The UVCGANv2 model having  $\lambda_{\text{consist}} = 0$  and UVCGANv2-C model with  $\lambda_{\text{consist}} = 0.2$ . In this section, we show an ablation of the  $\lambda_{\text{consist}}$  values and their effect on the model performance.

[Table 8](#) demonstrates the effect of different values of  $\lambda_{\text{consist}}$  on the UVCGANv2 model realism (as measured by the FID scores) and pixel-wise faithfulness (as measured by the pixel-wise  $L_2$  distance).

As one might expect, the increase in  $\lambda_{\text{consist}}$  is accompanied by an improvement in pixel-wise image faithfulness and a decrease in image realism. Values of  $\lambda_{\text{consist}}$  below 0.2 allow one to significantly improve pixel-wise image faithfulness at the expense of a modest loss of image realism. Further increases in  $\lambda_{\text{consist}}$  produce larger improvements in pixel-wise faithfulness, but also lead to significant decreases in image realism.

Additionally, [Table 8](#) demonstrates that the trade-offs of image realism to pixel-wise faithfulness are not uniform across the datasets. High values of  $\lambda_{\text{consist}}$  allow one to achieve rather large improvements in pixel-wise faithfulness on the Male-to-Female task at a relatively small loss of image realism. On the other hand, Cat-to-Dog translation is subject to a catastrophic loss of image realism with the increase of  $\lambda_{\text{consist}}$ .



Table 9. Sample translations for Selfie-to-Anime on Anime.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
CycleGAN									
U-GAT-IT									
UVCGAN									
UVCGANv2									

Table 10. Sample translations for Anime-to-Selfie on Anime.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
CycleGAN									
U-GAT-IT									
UVCGAN									
UVCGANv2									



Table 11. Sample translations for Male-to-Female on CelebA.

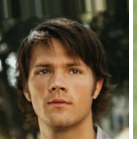

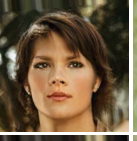
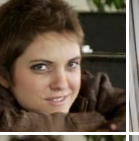
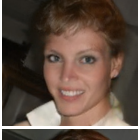
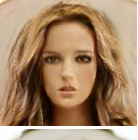
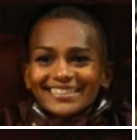
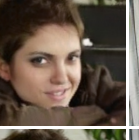
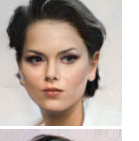


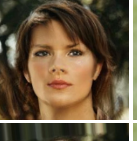


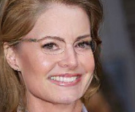

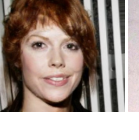
	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
CycleGAN									
U-GAT-IT									
UVCGAN									
UVCGANv2									

Table 12. Sample translations for Female-to-Male on CelebA.










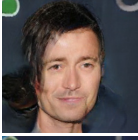



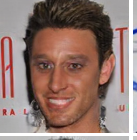



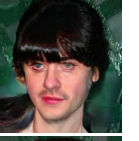
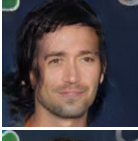



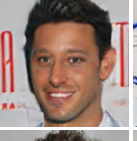



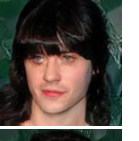




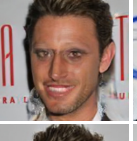



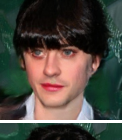
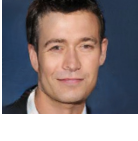



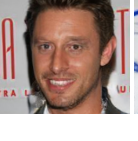

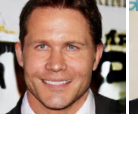
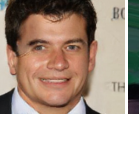
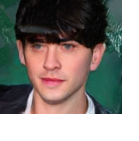
	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
CycleGAN									
U-GAT-IT									
UVCGAN									
UVCGANv2									



Table 13. Sample translations for Removing Glasses on CelebA.






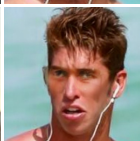
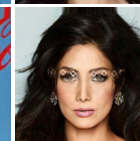
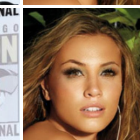
	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
CycleGAN									
U-GAT-IT									
UVCGAN									
UVCGANv2									

Table 14. Sample translations for Adding Glasses on CelebA.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
CycleGAN									
U-GAT-IT									
UVCGAN									
UVCGANv2									



Table 15. Sample translations for Cat-to-Dog on AFHQ.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
EGSDE									
EGSDE <sup>†</sup>									
UVCGAN <sup>v2</sup>									

Table 16. Sample translations for Wild-to-Dog on AFHQ.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
EGSDE									
EGSDE <sup>†</sup>									
UVCGAN <sup>v2</sup>									



Table 17. **Sample translations for Wild-to-Cat on AFHQ.** Since no benchmarking algorithms studied this task, we only show the input and UVCGANv2's translation.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
UVCGANv2									
	Example 10	Example 11	Example 12	Example 13	Example 14	Example 15	Example 16	Example 17	Example 18
input									
UVCGANv2									

Table 18. **Sample translations for Male-to-Female on CelebA-HQ.**

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9
input									
EGSDE									
EGSDE†									
UVCGANv2									