Contrastive Learning for Unpaired Image-to-Image Translation

Taesung Park[1] Alexei A. Efros[1] Richard Zhang[2] Jun-Yan Zhu[2]

University of California, Berkeley[1] Adobe Research[2]

Appendix A Additional Image-to-Image Results

We first show additional, randomly selected results on datasets used in our main paper. We then show results on additional datasets.

A.1 Additional comparisons

In Figure 1, we show additional, randomly selected results for Horse→Zebra and Cat→Dog. This is an extension of Figure 3 in the main paper. We compare to baseline methods CycleGAN [27], MUNIT [11], DRIT [17], Self-Distance and DistanceGAN [1], and GcGAN [5].

A.2 Additional datasets

In Figure 2 and Figure 3, we show additional datasets, compared against baseline method CycleGAN [27]. Our method provides better or comparable results, demonstrating its flexibility across a variety of datasets.

- *Apple→Orange* contains 996 apple and 1,020 orange images from ImageNet and was introduced in CycleGAN [27].
- *Yosemite Summer→Winter* contains 1,273 summer and 854 winter images of Yosemite scraped using the FlickAPI was introduced in CycleGAN [27].
- *GTA→Cityscapes* GTA contains 24,966 images [22] and Cityscapes [4] contains 19,998 images of street scenes from German cities. The task was originally used in CyCADA [10].

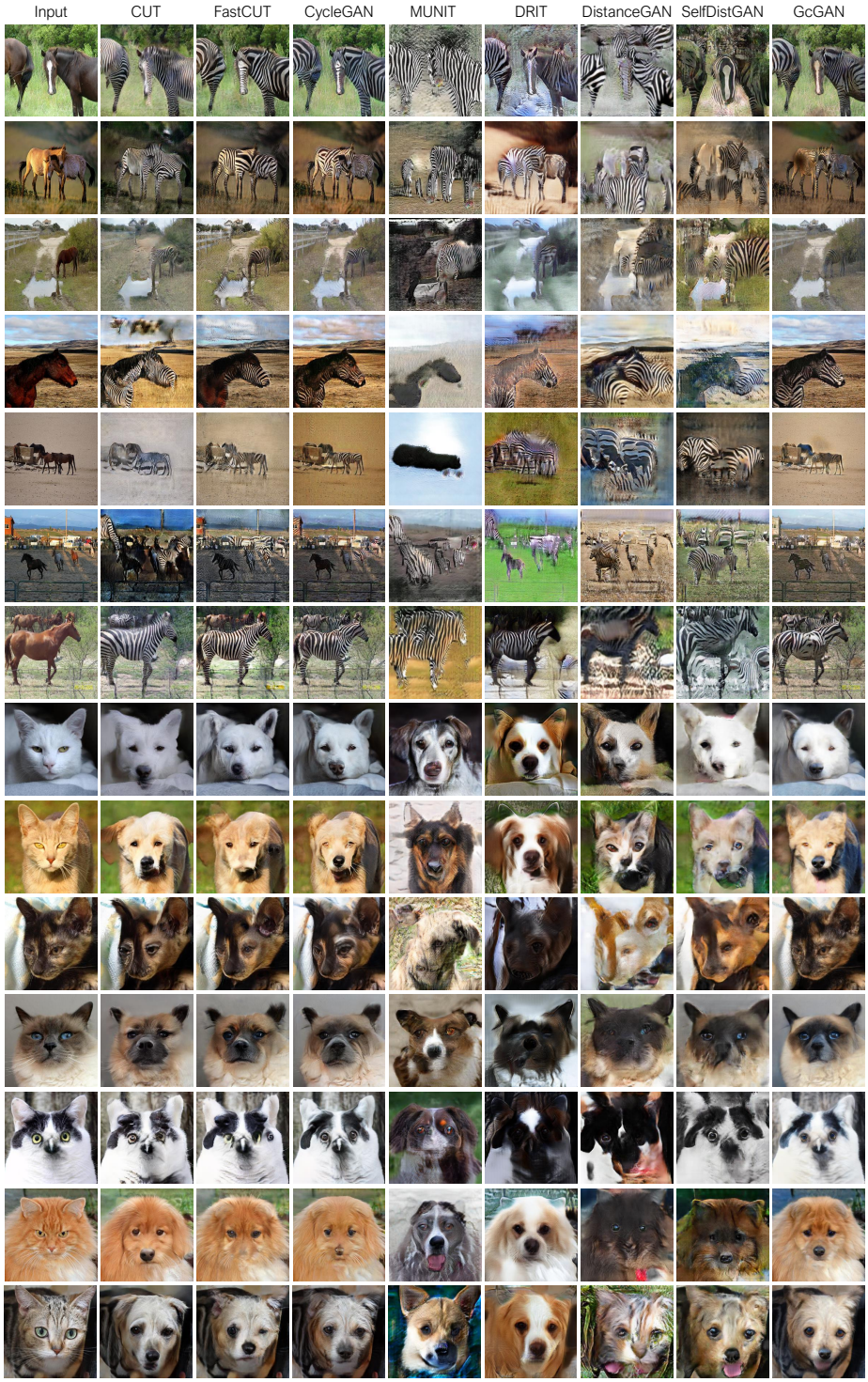

Fig. 1: **Randomly selected Horse→Zebra and Cat→Dog results**. This is an extension of Figure 3 in the main paper.

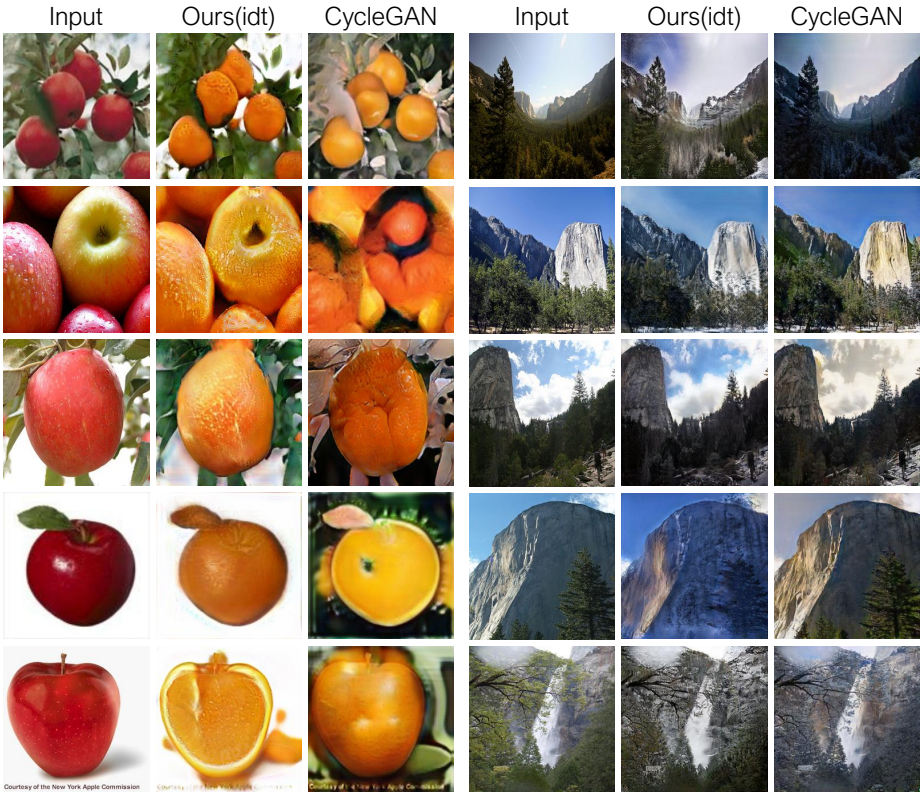

Fig. 2: **Apple→Orange** and **Summer→Winter Yosemite**. CycleGAN models were downloaded from the authors' public code repository. Apple→Orange shows that CycleGAN may suffer from color flipping issue.

Fig. 3: **GTA→Cityscapes** results at 1024 × 512 resolution. The model was trained on 512 × 512 crops.

.

Appendix B Additional Single Image Translation Results

We show additional results in Figure 4 and Figure 5, and describe training details below.

Training details. At each iteration, the input image is randomly scaled to a width between 384 to 1024, and we randomly sample 16 crops of size 128×128. To avoid overfitting, we divide crops into 64×64 tiles before passing them to the discriminator. At test time, since the generator network is fully convolutional, it takes the input image at full size.

We found that adopting the architecture of StyleGAN2 [14] instead of CycleGAN slightly improves the output quality, although the difference is marginal. Our StyleGAN2-based generator consists of one downsampling block of StyleGAN2 discriminator, 6 StyleGAN2 residual blocks, and one StyleGAN2 upsampling block. Our discriminator has the same architecture as StyleGAN2. Following StyleGAN2, we use non-saturating GAN loss [21] with R1 gradient penalty [20]. Since we do not use style code, the style modulation layer of StyleGAN2 was removed.

Single image results.

In Figure 4 and Figure 5, we show additional comparison results for our method, Gatys et al. [6], STROTSS [16], WCT^2 [25], and CycleGAN baseline [27]. Note that the CycleGAN baseline adopts the same augmentation techniques as well as the same generator/discriminator architectures as our method. The image resolution is at 1-2 Megapixels. Please zoom in to see more visual details.

Both figures demonstrate that our results look more photorealistic compared to CycleGAN baseline, Gatys et al [6], and WCT^2. The quality of our results is on par with results from STROTSS [16]. Note that STROTSS [16] compares to and outperforms recent style transfer methods (e.g., [7,19]).

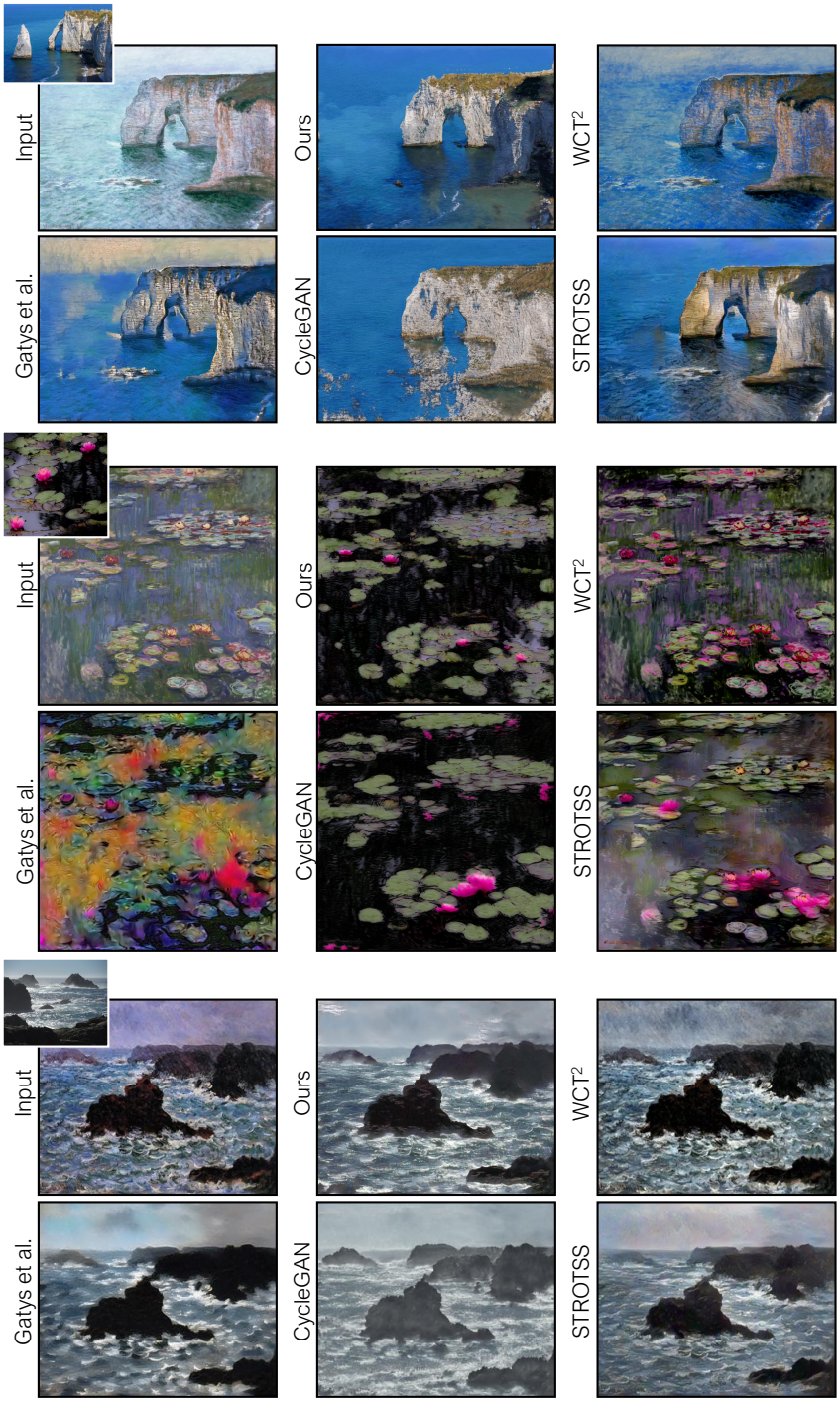

Fig. 4: **High-res painting to photo translation (I).** We transfer Monet's paintings to reference natural photos shown as insets at top-left corners. The training only requires a single image from each domain. We compare our results to recent style and photo transfer methods including Gatys et al. [6], WCT^2 [25], STROTSS [16], and our modified patch-based CycleGAN [27]. Our method can reproduce the texture of the reference photos while retaining structure of the input paintings. Our results are at 1k $\sim$ 1.5k resolution.

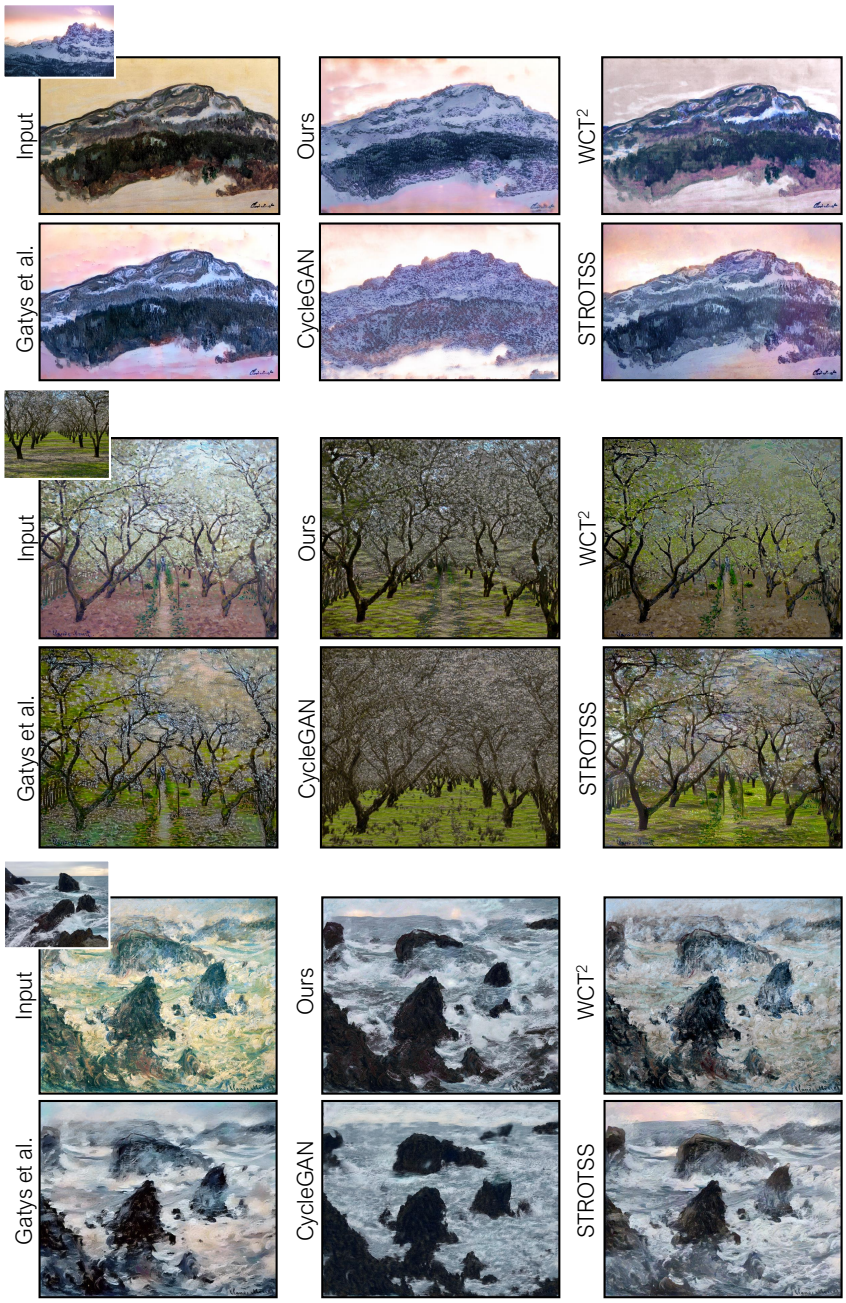

Fig. 5: **High-res painting to photo translation (II).** We transfer Monet's paintings to reference natural photos shown as insets at top-left corners. The training only requires a single image from each domain. We compare our results to recent style and photo transfer methods including Gatys et al. [6], WCT^2 [25], STROTSS [16], and our modified patch-based CycleGAN [27]. Our method can reproduce the texture of the reference photos while retaining structure of the input paintings. Our results are at 1k $\sim$ 1.5k resolution.

Appendix C Unpaired Translation Details and Analysis

C.1 Training details

To show the effect of the proposed patch-based contrastive loss, we intentionally match the architecture and hyperparameter settings of CycleGAN, except the loss function. This includes the ResNet-based generator [13] with 9 residual blocks, PatchGAN discriminator [12], Least Square GAN loss [18], batch size of 1, and Adam optimizer [15] with learning rate 0.002.

Our full model CUT is trained up to 400 epochs, while the fast variant FastCUT is trained up to 200 epochs, following CycleGAN. Moreover, inspired by GcGAN [5], FastCUT is trained with flip-equivariance augmentation, where the input image to the generator is horizontally flipped, and the output features are flipped back before computing the PatchNCE loss. Our encoder G_{enc} is the first half of the CycleGAN generator [27]. In order to calculate our multi-layer, patch-based contrastive loss, we extract features from 5 layers, which are RGB pixels, the first and second downsampling convolution, and the first and the fifth residual block. The layers we use correspond to receptive fields of sizes 1×1, 9×9, 15×15, 35×35, and 99×99. For each layer's features, we sample 256 random locations, and apply 2-layer MLP to acquire 256-dim final features. For our baseline model that uses MoCo-style memory bank [8], we follow the setting of MoCo, and used momentum value 0.999 with temperature 0.07. The size of the memory bank is 16384 per layer, and we enqueue 256 patches per image per iteration.

C.2 Evaluation details

We list the details of our evaluation protocol.

Fréchet Inception Distance (FID [9]) throughout this paper is computed by resizing the images to 299-by-299 using bilinear sampling of PyTorch framework, and then taking the activations of the last average pooling layer of a pretrained Inception V3 [23] using the weights provided by the TensorFlow framework. We use the default setting of https://github.com/mseitzer/pytorch-fid. All test set images are used for evaluation, unless noted otherwise.

Semantic segmentation metrics on the Cityscapes dataset are computed as follows. First, we trained a semantic segmentation network using the DRN-D-22 [26] architecture. We used the recommended setting from https://github.com/fyu/drn, with batch size 32 and learning rate 0.01, for 250 epochs at 256x128 resolution. The output images of the 500 validation labels are resized to 256x128 using bicubic downsampling, passed to the trained DRN network, and compared against the ground truth labels downsampled to the same size using nearest-neighbor sampling.

C.3 Pseudocode

Here we provide the pseudo-code of PatchNCE loss in the PyTorch style. Our code and models are available at our GitHub repo.

```
import torch
cross_entropy_loss = torch.nn.CrossEntropyLoss()

# Input: f_q (BxCxS) and sampled features from H(G_enc(x))
# Input: f_k (BxCxS) are sampled features from H(G_enc(G(x))
# Input: tau is the temperature used in NCE loss.
# Output: PatchNCE loss
def PatchNCELoss(f_q, f_k, tau=0.07):
    # batch size, channel size, and number of sample locations
    B, C, S = f_q.shape

    # calculate v * v+: BxSx1
    l_pos = (f_k * f_q).sum(dim=1)[:, :, None]

    # calculate v * v-: BxSxS
    l_neg = torch.bmm(f_q.transpose(1, 2), f_k)

    # The diagonal entries are not negatives. Remove them.
    identity_matrix = torch.eye(S)[None, :, :]
    l_neg.masked_fill_(identity_matrix, -float('inf'))

    # calculate logits: (B)x(S)x(S+1)
    logits = torch.cat((l_pos, l_neg), dim=2) / tau

    # return NCE loss
    predictions = logits.flatten(0, 1)
    targets = torch.zeros(B * S, dtype=torch.long)
    return cross_entropy_loss(predictions, targets)
```

C.4 Distribution matching

In Figure 6, we show an interesting phenomenon of our method, caused by the training set imbalance of the horse→zebra set. We use an off-the-shelf DeepLab model [3] trained on COCO-Stuff [2], to measure the percentage of pixels that belong to horses and zebras[1]. The training set exhibits dataset bias [24]. On average, zebras appear in more close-up pictures than horses and take up about twice the number of pixels (37% vs 18%). To perfectly satisfy the discriminator, a translation model should attempt to match the statistics of the training set. Our method allows the flexibility for the horses to change the size, and the percentage of output zebra pixels (31%) better matches the training distribution (37%) than the CycleGAN baseline (19%). On the other hand, our fast variant *FastCUT* uses a larger weight ($\lambda_X = 10$) on the Patch NCE loss and flip-equivariance augmentation, and hence behaves more conservatively and more similar to CycleGAN. The strong distribution matching capacity has pros and cons. For certain applications, it can create introduce undesired changes (e.g.,

[1] Pretrained model from https://github.com/kazuto1011/deeplab-pytorch

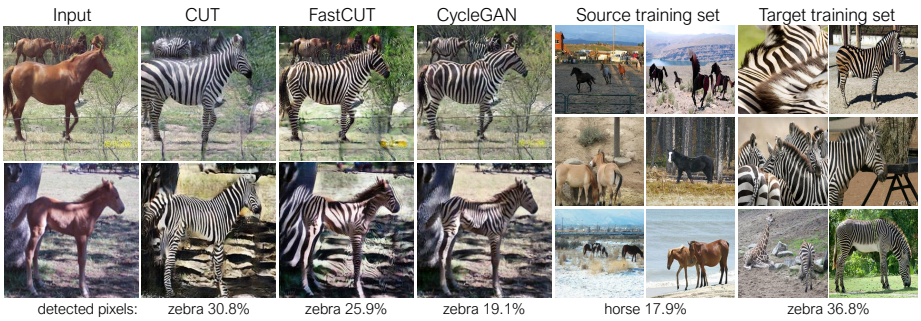

Fig. 6: **Distribution matching.** We measure the percentage of pixels belonging to the horse/zebra bodies, using a pre-trained semantic segmentation model. We find a distribution mismatch between sizes of horses and zebras images – zebras usually appear larger (36.8% vs. 17.9%). Our full method CUT has the flexibility to enlarge the horses, as a means of better matching of the training statistics than CycleGAN [27]. Our faster variant FastCUT, trained with a higher PatchNCE loss ($\lambda_X = 10$) and flip-equivariance augmentation, behaves more conservatively like CycleGAN.

zebra patterns on the background for horse→zebra). On the other hand, it can enable dramatic geometric changes for applications such as Cat→Dog.

C.5 Additional Ablation studies

In the paper, we mainly discussed the impact of loss functions and the number of patches on the final performance. Here we present additional ablation studies on more subtle design choices. We run all the variants on horse2zebra datasets [27]. The FID of our original model is **46.6**. We compare it to the following two variants of our model:

- Ours without weight sharing for the encoder G_{enc} and MLP projection network H: for this variant, when computing features $\{\boldsymbol{z}_l\}_L = \{H_l(G^l_{\text{enc}}(\boldsymbol{x}))\}_L$, we use two separate encoders and MLP networks for embedding input images (e.g., horse) and the generated images (e.g., zebras) to feature space. They do not share any weights. The FID of this variant is **50.5**, worse than our method. This shows that weight sharing helps stabilize training while reducing the number of parameters in our model.
- Ours without updating the decoder G_{dec} using *PatchNCE* loss: in this variant, we exclude the gradient propagation of the decoder G_{dec} regarding *PatchNCE* loss $\mathcal{L}_{\text{PatchNCE}}$. In other words, the decoder G_{dec} only gets updated through the adversarial loss $\mathcal{L}_{\text{GAN}}$. The FID of this variant is **444.2**, and the results contain severe artifacts. This shows that our $\mathcal{L}_{\text{PatchNCE}}$ not only helps learn the encoder G_{enc}, as done in previous unsupervised feature learning methods [8], but also learns a better decoder G_{dec} together with the GAN loss. Intuitively, if the generated result has many artifacts and is far from realistic, it would be difficult for the encoder to find correspondences between the input and output, producing a large *PatchNCE* loss.

References

1. Benaim, S., Wolf, L.: One-sided unsupervised domain mapping. In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 1
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 8
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **40**(4), 834–848 (2018) 8
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1
5. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Zhang, K., Tao, D.: Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 7
6. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 4, 5, 6
7. Gu, S., Chen, C., Liao, J., Yuan, L.: Arbitrary style transfer with deep feature reshuffle. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
8. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 7, 9
9. 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 (2017) 7
10. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning (ICML) (2018) 1
11. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. European Conference on Computer Vision (ECCV) (2018) 1
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 7
13. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (ECCV) (2016) 7
14. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015) 7
16. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4, 5, 6

17. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Diverse image-to-image translation via disentangled representation. In: European Conference on Computer Vision (ECCV) (2018) 1
18. Mao, X., Li, Q., Xie, H., Lau, Y.R., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: IEEE International Conference on Computer Vision (ICCV) (2017) 7
19. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. In: European Conference on Computer Vision (ECCV) (2018) 4
20. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International Conference on Machine Learning (ICML) (2018) 4
21. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2016) 4
22. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European Conference on Computer Vision (ECCV) (2016) 1
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 7
24. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 8
25. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.W.: Photorealistic style transfer via wavelet transforms. In: IEEE International Conference on Computer Vision (ICCV) (2019) 4, 5, 6
26. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 7
27. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (ICCV) (2017) 1, 4, 5, 6, 7, 9