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]

Abstract. In image-to-image translation, each patch in the output should reflect the *content* of the corresponding patch in the input, independent of domain. We propose a straightforward method for doing so – maximizing mutual information between the two, using a framework based on contrastive learning. The method encourages two elements (corresponding patches) to map to a similar point in a learned feature space, relative to other elements (other patches) in the dataset, referred to as negatives. We explore several critical design choices for making contrastive learning effective in the image synthesis setting. Notably, we use a multilayer, patch-based approach, rather than operate on entire images. Furthermore, we draw negatives from *within* the input image itself, rather than from the rest of the dataset. We demonstrate that our framework enables one-sided translation in the unpaired image-to-image translation setting, while improving quality and reducing training time. In addition, our method can even be extended to the training setting where each "domain" is only a single image.

Keywords: contrastive learning, noise contrastive estimation, mutual information, image generation

1 Introduction

Consider the image-to-image translation problem in Figure 1. We wish for the output to take on the *appearance* of the target domain (a zebra), while retaining the structure, or *content*, of the specific input horse. This is, fundamentally, a disentanglement problem: separating the content, which needs to be preserved across domains, from appearance, which must change. Typically, target appearance is enforced using an adversarial loss [21,31], while content is preserved using cycle-consistency [89,81,37]. However, cycle-consistency assumes that the relationship between the two domains is a bijection, which is often too restrictive. In this paper, we propose an alternative, rather straightforward way of maintaining correspondence in content but not appearance – by maximizing the mutual information between corresponding input and output patches.

In a successful result, given a specific patch on the output, for example, the generated zebra forehead highlighted in blue, one should have a good idea that it came from the horse forehead, and not the other parts of the horse or

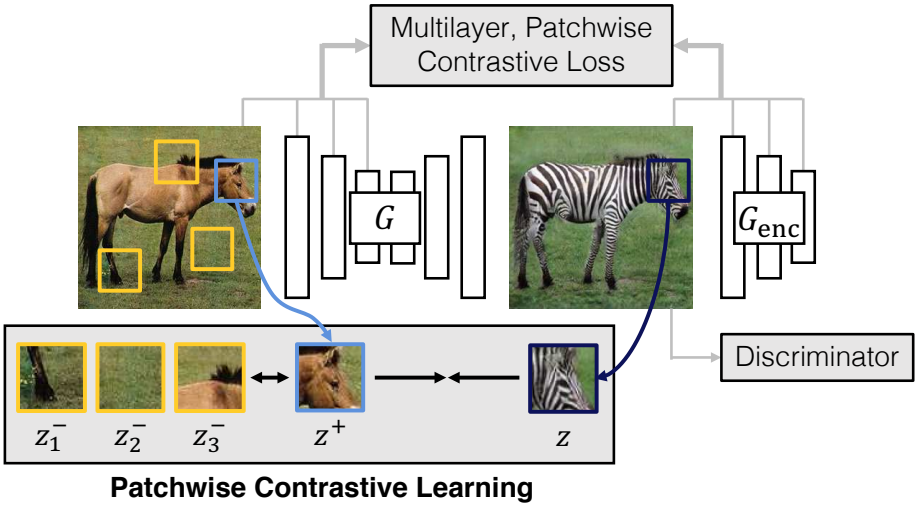

Fig. 1: **Patchwise Contrastive Learning for one-sided translation.** A generated **output patch** should appear closer to its **corresponding input patch**, in comparison to other **random patches**. We use a multilayer, patchwise contrastive loss, which maximizes *mutual information* between corresponding input and output patches. This enables one-sided translation in the unpaired setting.

the background vegetation. We achieve this by using a type of contrastive loss function, InfoNCE loss [57], which aims to learn an embedding or an encoder that *associates* corresponding patches to each other, while *disassociating* them from others. To do so, the encoder learns to pay attention to the commonalities between the two domains, such as object parts and shapes, while being invariant to the differences, such as the textures of the animals. The two networks, the generator and encoder, conspire together to generate an image such that patches can be easily traceable to the input.

Contrastive learning has been an effective tool in unsupervised visual representation learning [9,24,57,80]. In this work, we demonstrate its effectiveness in a conditional image synthesis setting and systematically study several key factors to make it successful. We find it pertinent to use it on a multilayer, patchwise fashion. In addition, we find that drawing negatives *internally* from within the input image, rather than externally from other images in the dataset, forces the patches to better preserve the content of the input. Our method requires neither memory bank [80,24] nor specialized architectures [25,3].

Extensive experiments show that our faster, lighter model outperforms both prior one-sided translation methods [4,18] and state-of-the-art models that rely on several auxiliary networks and multiple loss functions. Furthermore, since our contrastive representation is formulated within the same image, our method can even be trained on single images. Our code and models are available at GitHub.

2 Related Work

Image translation and cycle-consistency. Paired image-to-image translation [31] maps an image from input to output domain using an adversarial loss [21], in conjunction with a reconstruction loss between the result and target. In unpaired translation settings, corresponding examples from domains are not available. In such cases, *cycle-consistency* has become the de facto method for enforcing correspondence [89,81,37], which learns an inverse mapping from the output domain back to the input and checks if the input can be reconstructed. Alternatively, UNIT [44] and MUNIT [30] propose to learn a shared intermediate "content" latent space. Recent works further enable multiple domains and multimodal synthesis [10,90,1,41,45] and improve the quality of results [72,88,20,79,43]. In all of the above examples, cycle-consistency is used, often in multiple aspects, between (a) two image domains [37,81,89] (b) image to latent [10,30,41,44,90], or (c) latent to image [30,90]. While effective, the underlying bijective assumption behind cycle-consistency is sometimes too restrictive. Perfect reconstruction is difficult to achieve, especially when images from one domain have additional information compared to the other domain.

Relationship preservation. An interesting alternative approach is to encourage relationships present in the input be analogously reflected in the output. For example, perceptually similar patches *within* an input image should be similar in the output [88], output and input images share similar content regarding a pre-defined distance [5,68,71], vector arithmetic between input images is preserved using a margin-based triplet loss [2], distances *between* input images should be consistent in output images [4], the network should be equivariant to geometric transformations [18]. Among them, TraVeLGAN [2], DistanceGAN [4] and GcGAN [18] enable one-way translation and bypass cycle-consistency. However, they rely on relationship between entire images, or often with predefined distance functions. Here we seek to replace cycle-consistency by instead learning a cross-domain similarity function *between input and output patches* through information maximization, without relying on a pre-specified distance.

Emergent perceptual similarity in deep network embeddings. Defining a "perceptual" distance function between high-dimensional signals, e.g., images, has been a longstanding problem in computer vision and image processing. The majority of image translation work mentioned uses a per-pixel reconstruction metric, such as ℓ_1. Such metrics do not reflect human perceptual preferences and can lead to blurry results. Recently, the deep learning community has found that the VGG classification network [69] trained on ImageNet dataset [14] can be re-purposed as a "perceptual loss" [16,19,34,75,87,52], which can be used in paired image translation tasks [8,59,77], and was known to outperform traditional metrics such as SSIM [78] and FSIM [84] on human perceptual tests [87]. In particular, the Contextual Loss [52] boosts the perceptual quality of pretrained VGG features, validated by human perceptual judgments [51]. In these cases, the frozen network weights cannot adapt to the data on hand. Furthermore, the frozen similarity function may not be appropriate when comparing data *across*

two domains, depending on the pairing. By posing our constraint via mutual information, our method makes use of negative samples from the data, allowing the cross-domain similarity function to adapt to the particular input and output domains, and bypass using a pre-defined similarity function.

Contrastive representation learning. Traditional unsupervised learning has sought to learn a compressed code which can effectively reconstruct the input [27]. Data imputation – holding one subset of raw data to predict from another – has emerged as a more effective family of pretext tasks, including denoising [76], context prediction [15,60], colorization [40,85], cross-channel encoding [86], frame prediction [46,55], and multi-sensory prediction [56,58]. However, such methods suffer from the same issue as before — the need for a pre-specified, hand-designed loss function to measure predictive performance.

Recently, a family of methods based on *maximizing mutual information* has emerged to bypass the above issue [9,24,25,28,47,54,57,73,80]. These methods make use of noise contrastive estimation [23], learning an embedding where associated signals are brought together, in *contrast* to other samples in the dataset (note that similar ideas go back to classic work on metric learning with Siamese nets [12]). Associated signals can be an image with itself [49,67,17,24,80], an image with its downstream representation [28,47], neighboring patches within an image [33,25,57], or multiple views of the input image [73], and most successfully, an image with a set of transformed versions of itself [9,54]. The design choices of the InfoNCE loss, such as the number of negatives and how to sample them, hyperparameter settings, and data augmentations all play a critical role and need to be carefully studied. We are the first to use InfoNCE loss for the conditional image synthesis tasks. As such, we draw on these important insights, and find additional pertinent factors, unique to image synthesis.

3 Methods

We wish to translate images from input domain $\mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ to appear like an image from the output domain $\mathcal{Y} \subset \mathbb{R}^{H \times W \times 3}$. We are given a dataset of unpaired instances $X = \{\boldsymbol{x} \in \mathcal{X}\}, Y = \{\boldsymbol{y} \in \mathcal{Y}\}$. Our method can operate even when X and Y only contain a single image each.

Our method only requires learning the mapping in one direction and avoids using inverse auxiliary generators and discriminators. This can largely simplify the training procedure and reduce training time. We break up our generator function G into two components, an encoder G_{enc} followed by a decoder G_{dec}, which are applied sequentially to produce output image $\hat{\boldsymbol{y}} = G(\boldsymbol{z}) = G_{\text{dec}}(G_{\text{enc}}(\boldsymbol{x}))$.

Adversarial loss. We use an adversarial loss [21], to encourage the output to be visually similar to images from the target domain, as follows:

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) = \mathbb{E}_{\boldsymbol{y} \sim Y} \log D(\boldsymbol{y}) + \mathbb{E}_{\boldsymbol{x} \sim X} \log(1 - D(G(\boldsymbol{x}))). \tag{1}$$

Mutual information maximization. We use a noise contrastive estimation framework [57] to maximize mutual information between input and output. The

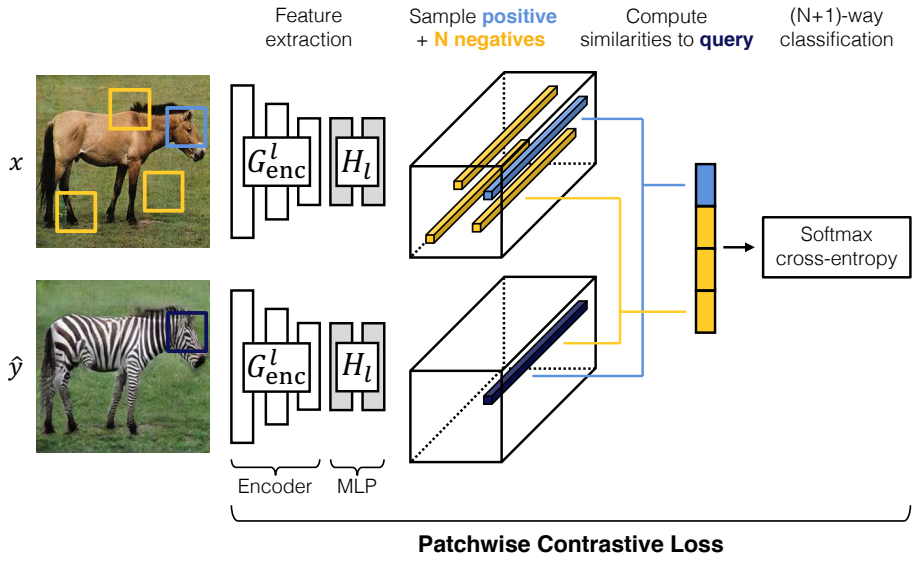

Fig. 2: **Patchwise Contrastive Loss.** Both images, $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$, are encoded into feature tensor. We sample a **query** patch from the output $\hat{\boldsymbol{y}}$ and compare it to the input patch at the same location. We set up an (N+1)-way classification problem, where N negative patches are sampled from the same input image at different locations. We reuse the encoder part G_{enc} of our generator and add a two-layer MLP network. This network learns to project both the input and output patch to a shared embedding space.

idea of contrastive learning is to associate two signals, a "query" and its "positive" example, in contrast to other points within the dataset, referred to as "negatives". The query, positive, and N negatives are mapped to K-dimensional vectors $\boldsymbol{v}, \boldsymbol{v}^+ \in \mathbb{R}^K$ and $\boldsymbol{v}^- \in \mathbb{R}^{N\times K}$, respectively. $\boldsymbol{v}_n^- \in \mathbb{R}^K$ denotes the n-th negative. We normalize vectors onto a unit sphere to prevent the space from collapsing or expanding. An $(N+1)$–way classification problem is set up, where the distances between the query and other examples are scaled by a temperature $\tau = 0.07$ and passed as logits [80,24]. The cross-entropy loss is calculated, representing the probability of the positive example being selected over negatives.

$$\ell(\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-) = -\log\left[\frac{\exp(\boldsymbol{v}\cdot\boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{v}\cdot\boldsymbol{v}^+/\tau) + \sum_{n=1}^{N}\exp(\boldsymbol{v}\cdot\boldsymbol{v}_n^-/\tau)}\right]. \tag{2}$$

Our goal is to associate the input and output data. In our context, query refers to an output. positive and negatives are corresponding and noncorresponding input. Below, we explore several important design choices, including how to map the images into vectors and how to sample the negatives.

Multilayer, patchwise contrastive learning. In the unsupervised learning setting, contrastive learning has been used both on an image and patch level [3,25]. For our application, we note that not only should the whole images share content,

but also corresponding patches between the input and output images. For example, given a patch showing the legs of an output zebra, one should be able to more strongly associate it to the corresponding legs of the input horse, more so than the other patches of the horse image. Even at the pixel level, the colors of a zebra body (black and white) can be more strongly associated to the color of a horse body than to the background shades of grass. Thus, we employ a *multilayer, patch-based* learning objective.

Since the encoder G_{enc} is computed to produce the image translation, its feature stack is readily available, and we take advantage. Each layer and spatial location within this feature stack represents a patch of the input image, with deeper layers corresponding to bigger patches. We select L layers of interest and pass the feature maps through a small two-layer MLP network H_l, as used in SimCLR [9], producing a stack of features $\{\boldsymbol{z}_l\}_L = \{H_l(G^l_{\text{enc}}(\boldsymbol{x}))\}_L$, where G^l_{enc} represents the output of the l-th chosen layer. We index into layers $l \in \{1, 2, ..., L\}$ and denote $s \in \{1, ..., S_l\}$, where S_l is the number of spatial locations in each layer. We refer to the corresponding feature as $\boldsymbol{z}^s_l \in \mathbb{R}^{C_l}$ and the other features as $\boldsymbol{z}^{S\setminus s}_l \in \mathbb{R}^{(S_l-1)\times C_l}$, where C_l is the number of channels at each layer. Similarly, we encode the output image $\hat{\boldsymbol{y}}$ into $\{\hat{\boldsymbol{z}}_l\}_L = \{H_l(G^l_{\text{enc}}(G(\boldsymbol{x})))\}_L$.

We aim to match corresponding input-output patches at a specific location. We can leverage the other patches *within* the input as negatives. For example, a zebra leg should be more closely associated with an input horse leg than the other patches of the same input, such as other horse parts or the background sky and vegetation. We name it as the *PatchNCE* loss, as illustrated in Figure 2. Appendix C.3 provides pseudocode.

$$\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{\boldsymbol{x}\sim X} \sum_{l=1}^{L} \sum_{s=1}^{S_l} \ell(\hat{\boldsymbol{z}}^s_l, \boldsymbol{z}^s_l, \boldsymbol{z}^{S\setminus s}_l). \tag{3}$$

Alternatively, we can also leverage image patches from the rest of the dataset. We encode a random negative image from the dataset $\tilde{\boldsymbol{x}}$ into $\{\tilde{\boldsymbol{z}}_l\}_L$, and use the following *external* NCE loss. In this variant, we maintain a large, consistent dictionary of negatives using an auxiliary moving-averaged encoder, following MoCo [24]. MoCo enables negatives to be sampled from a longer history, and performs more effective than end-to-end updates [57,25] and memory bank [80].

$$\mathcal{L}_{\text{external}}(G, H, X) = \mathbb{E}_{\boldsymbol{x}\sim X, \tilde{\boldsymbol{z}}\sim Z^-} \sum_{l=1}^{L} \sum_{s=1}^{S_l} \ell(\hat{\boldsymbol{z}}^s_l, \boldsymbol{z}^s_l, \tilde{\boldsymbol{z}}_l), \tag{4}$$

where dataset negatives $\tilde{\boldsymbol{z}}_l$ are sampled from an external dictionary Z^- from the source domain, whose data are computed using a moving-averaged encoder $\hat{H}_l$ and moving-averaged MLP $\hat{H}$. We refer our readers to the original work for more details [24].

In Section 4.1, we show that our encoder G_{enc} learns to capture domain-invariant concepts, such as animal body, grass, and sky for horse $\rightarrow$ zebra, while our decoder G_{dec} learns to synthesize domain-specific features such as zebra stripes. Interestingly, through systematic evaluations, we find that using internal

patches only outperforms using external patches. We hypothesize that by using internal statistics, our encoder does not need to model large intra-class variation such as white horse vs. brown horse, which is not necessary for generating output zebras. Single image internal statistics has been proven effective in many vision tasks such as segmentation [32], super-resolution, and denoising [91,66].

Final objective. Our final objective is as follows. The generated image should be realistic, while patches in the input and output images should share correspondence. Figure 1 illustrates our minimax learning objective. Additionally, we may utilize PatchNCE loss $\mathcal{L}_{\text{PatchNCE}}(G, H, Y)$ on images from domain $\mathcal{Y}$ to prevent the generator from making unnecessary changes. This loss is essentially a learnable, domain-specific version of the identity loss, commonly used by previous unpaired translation methods [71,89].

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y). \quad (5)$$

We choose $\lambda_X = 1$ when we jointly train with the identity loss $\lambda_Y = 1$, and choose a larger value $\lambda_X = 10$ without the identity loss ($\lambda_Y = 0$) to compensate for the absence of the regularizer. We find that the former configuration, named *Contrastive Unpaired Translation (CUT)* hereafter, achieves superior performance to existing methods, whereas the latter, named *FastCUT*, can be thought as a faster and lighter version of CycleGAN. Our model is relatively simple compared to recent methods that often use 5-10 losses and hyper-parameters.

Discussion. Li et al. [42] has shown that cycle-consistency loss is the upper bound of conditional entropy $\text{H}(X|Y)$ (and $\text{H}(Y|X)$). Therefore, minimizing cycle-consistency loss encourages the output $\hat{\boldsymbol{y}}$ to be more dependent on input $\boldsymbol{x}$. This is related to our objective of maximizing the mutual information $\text{I}(X, Y)$, as $\text{I}(X, Y) = \text{H}(X) - \text{H}(X|Y)$. As entropy $\text{H}(X)$ is a constant and independent of the generator G, maximizing mutual information is equivalent to minimizing the conditional entropy. Notably, using contrastive learning, we can achieve a similar goal without introducing inverse mapping networks and additional discriminators. In the unconditional modeling scenario, InfoGAN [7] shows that simple losses (e.g., L2 or cross-entropy) can serve as a lower bound for maximizing mutual information between an image and a low-dimensional code. In our setting, we maximize the mutual information between two high-dimensional image spaces, where simple losses are no longer effective. Liang et al. [43] proposes an adversarial loss based on Siamese networks that encourages the output to be closer to the target domain than to its source domain. The above method still builds on cycle-consistency and two-way translations. Different from the above work, we use contrastive learning to enforce content consistency, rather than to improve the adversarial loss itself. To measure the similarity between two distributions, the Contextual Loss [52] used softmax over cosine disntances of features extracted from pre-trained networks. In contrast, we learn the encoder with the NCE loss to associate the input and output patches at the same location.

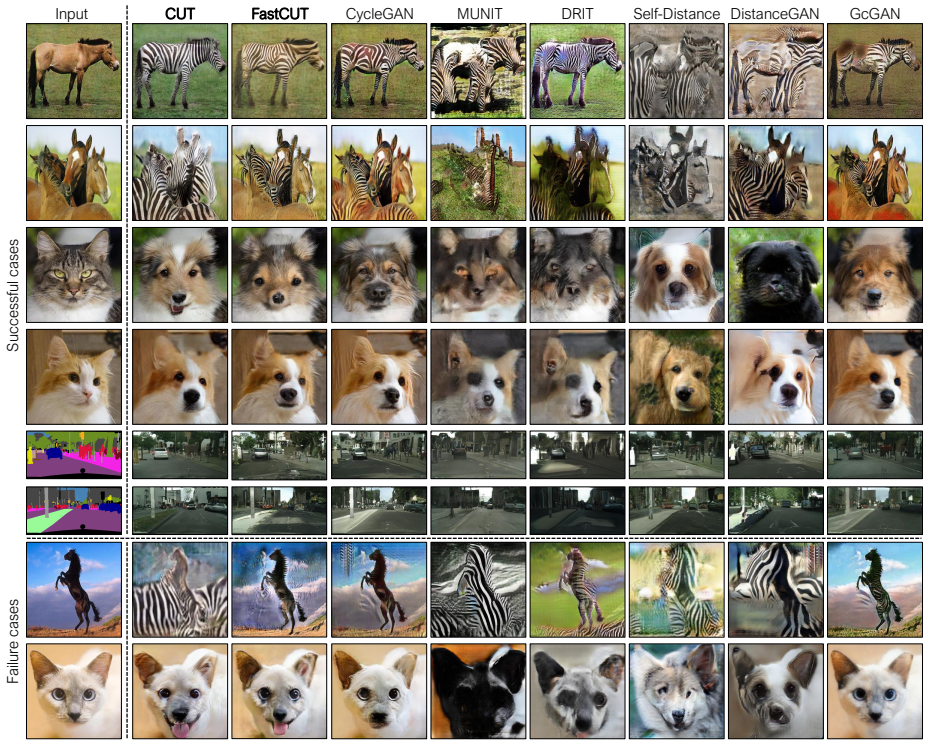

Fig. 3: **Results**. We compare our methods (CUT and FastCUT) with existing methods on the horse→zebra, cat→dog, and Cityscapes datasets. CycleGAN [89], MUNIT [44], and DRIT [41], are two-sided methods, while SelfDistance, DistanceGAN [4], and GcGAN [18] are one-sided. We show successful cases above the dotted lines. Our full version CUT is able to add the zebra texture to the horse bodies. Our fast variant FastCUT can also generate competitive results at the least computational cost of training. The final rows show failure cases. In the first, we are unable to identify the unfamiliar pose of the horse and instead add texture to the background. In the second, the method hallucinates a tongue.

4 Experiments

We test across several datasets. We first show that our method improves upon baselines in unpaired image translation. We then show that our method can extend to *single-image* training. Full results are available at our website.

Training details. We follow the setting of CycleGAN [89], except that the ℓ_1 cycle-consistency loss is replaced with our contrastive loss. In detail, we used LSGAN [50] and Resnet-based generator [34] with PatchGAN [31]. We define our encoder as the first half of the generator, and accordingly extract our multilayer features from five evenly distributed points of the encoder. For single image translation, we use a StyelGAN2-based generator [36]. To embed the encoder's

Method	Cityscapes				Cat→Dog	Horse→Zebra		
	mAP↑	pixAcc↑	classAcc↑	FID↓	FID↓	FID↓	sec/iter↓	Mem(GB)↓
CycleGAN [89]	20.4	55.9	25.4	76.3	85.9	77.2	0.40	4.81
MUNIT [44]	16.9	56.5	22.5	91.4	104.4	133.8	0.39	3.84
DRIT [41]	17.0	58.7	22.2	155.3	123.4	140.0	0.70	4.85
Distance [4]	8.4	42.2	12.6	81.8	155.3	72.0	**0.15**	2.72
SelfDistance [4]	15.3	56.9	20.6	78.8	144.4	80.8	0.16	2.72
GCGAN [18]	21.2	63.2	26.6	105.2	96.6	86.7	0.26	2.67
CUT	**24.7**	**68.8**	**30.7**	**56.4**	**76.2**	**45.5**	0.24	3.33
FastCUT	19.1	59.9	24.3	68.8	94.0	73.4	**0.15**	**2.25**

Table 1: **Comparison with baselines** We compare our methods across datasets on common evaluation metrics. CUT denotes our model trained with the identity loss ($\lambda_X = \lambda_Y = 1$), and FastCUT without it ($\lambda_X = 10, \lambda_Y = 0$). We show FID, a measure of image quality [26] (lower is better). For Cityscapes, we show the semantic segmentation scores (mAP, pixAcc, classAcc) to assess the discovered correspondence (higher is better for all metrics). Based on quantitative measures, CUT produces higher quality and more accurate generations with light footprint in terms of training speed (seconds per sample) and GPU memory usage. Our variant FastCUT also produces competitive results with even lighter computation cost of training.

features, we apply a two-layer MLP with 256 units at each layer. We normalize the vector by its L2 norm. See Appendix C.1 for more training details.

4.1 Unpaired image translation

Datasets We conduct experiments on the following datasets.

- *Cat→Dog* contains 5,000 training and 500 val images from AFHQ Dataset [11].
- *Horse→Zebra* contains 2,403 training and 260 zebra images from ImageNet [14] and was introduced in CycleGAN [89].
- *Cityscapes* [13] contains street scenes from German cities, with 2,975 training and 500 validation images. We train models at 256×256 resolution. Unlike previous datasets listed, this does have corresponding labels. We can leverage this to measure how well our unpaired algorithm discovers correspondences.

Evaluation protocol. We adopt the evaluation protocols from [26,89], aimed at assessing *visual quality* and *discovered correspondence*. For the first, we utilize the widely-used Fréchet Inception Distance (FID) metric, which empirically estimates the distribution of real and generated images in a deep network space and computes the divergence between them. Intuitively, if the generated images are realistic, they should have similar summary statistics as real images, in any feature space. For *Cityscapes* specifically, we have ground truth of paired label maps. If accurate correspondences are discovered, the algorithm should generate images that are recognizable as the correct class. Using an off-the-shelf network to test "semantic interpretability" of image translation results has been commonly used [85,31]. We use the pretrained semantic segmentation network DRN[83]. We train the DRN at 256x128 resolution, and compute mean average precision

Method	Training settings					Testing datasets		
	Id	Negs	Layers	Int	Ext	Horse→Zebra FID↓	Cityscapes FID↓	Cityscapes mAP↑
CUT (default)	✓	255	All	✓	✗	45.5	**56.4**	**24.7**
no id	✗	255	All	✓	✗	39.3	68.5	22.0
no id, 15 neg	✗	15	All	✓	✗	44.1	59.7	23.1
no id, 15 neg, last	✗	15	Last	✓	✗	**38.1**	114.1	16.0
last	✓	255	Last	✓	✗	441.7	141.1	14.9
int and ext	✓	255	All	✓	✓	56.4	64.4	20.0
ext only	✓	255	All	✗	✓	53.0	110.3	16.5
ext only, last	✓	255	Last	✗	✓	60.1	389.1	5.6

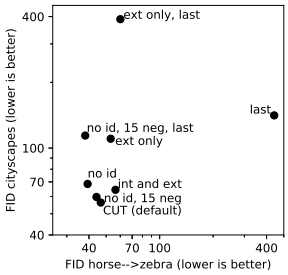

Fig. 4: **Ablations.** The PatchNCE loss is trained with negatives from each layer output of the same (internal) image, with the identity preservation regularization. *(Left)* We try removing the identity loss [**Id**], using less negatives [**Negs**], using only the last layer of the encoder [**Layers**], and varying where patches are sampled, internal [**Int**] vs external [**Ext**]. *(Right)* We plot the FIDs on horse→zebra and Cityscapes dataset. Removing the identity loss (`no id`) and reducing negatives (`no id, 15 neg`) still perform strongly. In fact, our variant FastCUT does not use the identity loss. However, reducing number of layers (`last`) or using external patches (`ext`) hurts performance.

(mAP), pixel-wise accuracy (pixAcc), and average class accuracy (classAcc). See Appendix C.2 for more evaluation details.

Comparison to baselines. In Table 1, we show quantitative measures of our and Figure 3, we compare our method to baselines. We present two settings of our method in Eqn. 5: CUT with the identity loss ($\lambda_X = \lambda_Y = 1$), and FastCUT without it ($\lambda_X = 10, \lambda_Y = 0$). On image quality metrics across datasets, our methods outperform baselines. We show qualitative results in Figure 3 and additional results in Appendix A. In addition, our Cityscapes semantic segmentation scores are higher, suggesting that our method is able to find correspondences between output and input.

Speed and memory. Since our model is one-sided, our method is memory-efficient and fast. For example, our method with the identity loss was 40% faster and 31% more memory-efficient than CycleGAN at training time, using the same architectures as CycleGAN (Table 1). Furthermore, our faster variant FastCUT is 63% faster and 53% lighter, while achieving superior metrics to CycleGAN. Table 1 contains the speed and memory usage of each method measured on NVIDIA GTX 1080Ti, and shows that FastCUT achieves competitive FIDs and segmentation scores with a lower time and memory requirement. Therefore, our method can serves as a practical, lighter alternative in scenarios, when an image translation model is jointly trained with other components [29,62].

4.2 Ablation study and analysis

We find that in the image synthesis setting, similarly to the unsupervised learning setting [25,24,9], implementation choices for contrastive loss are important. Here, try various settings and ablations of our method, summarized in Figure 4. By default, we use the ResNet-based generator used in CycleGAN [89], with patch-NCE using (a) negatives sampled from the input image, (b) multiple layers of the

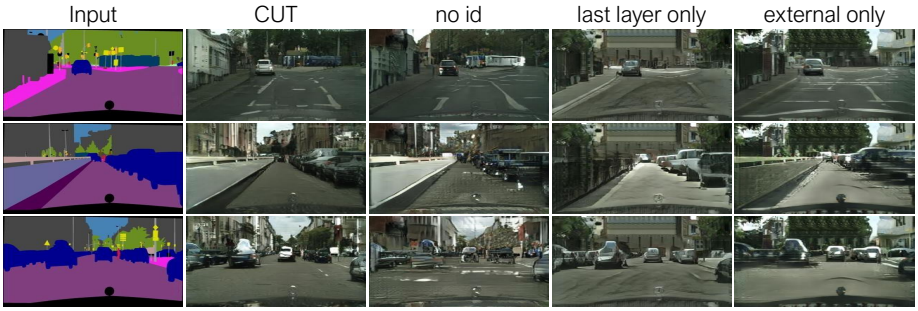

Fig. 5: **Qualitative ablation results** of our full method (CUT) are shown: *without* the identity loss $\mathcal{L}_{\text{PatchNCE}}(G, H, Y)$ on domain Y (`no id`), using only one layer of the encoder (`last layer only`), and using external instead of internal negatives (`external only`). The ablations cause noticeable drop in quality, including repeated building or vegetation textures when using only external negatives or the last layer output.

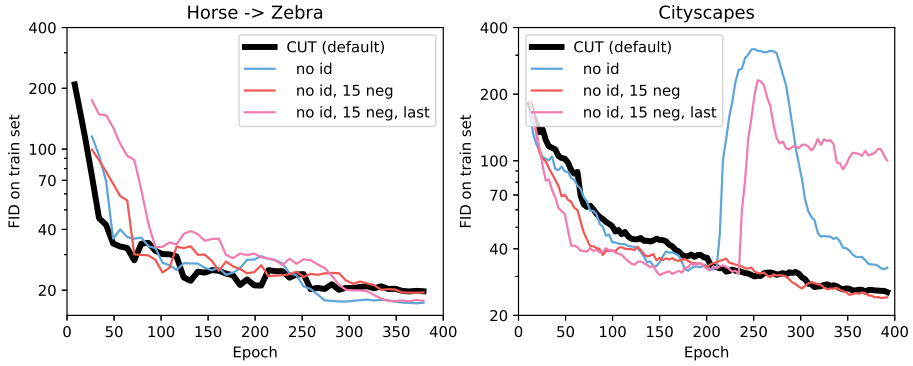

Fig. 6: **Identity loss $\mathcal{L}_{\text{PatchNCE}}(G, H, Y)$ on domain Y adds stability.** This regularizer encourages an image from the output domain $\boldsymbol{y}$ to be unchanged by the generator. Using it (shown in **bold, black** curves), we observe better stability in comparison to other variants. On the left, our variant without the regularizer, `no id`, achieves better FID. However, we see higher variance in the training curve. On the right, training without the regularizer can lead to collapse.

encoder, and (c) a PatchNCE loss $\mathcal{L}_{\text{PatchNCE}}(G, H, Y)$ on domain Y. In Figure 4, we show results using several variants and ablations, taken after training for 400 epochs. We show qualitative examples in Figure 5.

Internal negatives are more effective than external. By default, we sample negatives from *within* the same image (internal negatives). We also try adding negatives from other images, using a momentum encoder [24]. However, the external negatives, either as addition (`int and ext`) or replacement of internal negatives (`ext only`), hurts performance. In Figure 5, we see a loss of quality, such as repeated texture in the Cityscapes dataset, indicating that sampling negatives from the same image serves as a stronger signal for preserving content.

Importance of using multiple layers of encoder. Our method uses multiple layers of the encoder, every four layers from pixels to the 16^{th} layer. This

is consistent with the standard use of ℓ_1+VGG loss, which uses layers from the pixel level up to a deep convolutional layer. On the other hand, many contrastive learning-based unsupervised learning papers map the whole image into a single representation. To emulate this, we try only using the last layer of the encoder (`last`), and try a variant using external negatives only (`ext only, last`). Performance is drastically reduced in both cases. In unsupervised representation learning, the input images are fixed. For our application, the loss is being used as a signal for synthesizing an image. As such, this indicates that the dense supervision provided by using multiple layers of the encoder is important when performing image synthesis.

$\mathcal{L}_{\text{PatchNCE}}(G, H, Y)$ regularizer stabilizes training. Given an image from the output domain $\boldsymbol{y} \in \mathcal{Y}$, this regularizer encourages the generator to leave the image unchanged with our patch-based contrastive loss. We also experiment with a variant without this regularizer, `no id`. As shown in Figure 4, removing the regularizer improves results for the horse→zebra task, but decreases performance on Cityscapes. We further investigate by showing the training curves in Figure 6, across 400 epochs. In the Cityscapes results, the training can collapse without the regularizer (although it can recover). We observe that although the final FID is sometimes better without, the training is more stable with the regularizer.

Visualizing learned similarity by encoder G_{enc} To further understand why our encoder network G_{enc} has learned to perform horse→ zebra task, we study the output space of the 1st residual block for both horse and zebra features. As shown in Figure 7. Given an input and output image, we compute the distance between a query patch's feature vector $\boldsymbol{v}$ (highlighted as red or blue dot) to feature vectors $\boldsymbol{v}^-$ of all the patches in the input using $\exp(\boldsymbol{v} \cdot \boldsymbol{v}^-/\tau)$ (Eqn. 2). Additionally, we perform a PCA dimension reduction on feature vectors from both horse and zebra patches. In (d) and (e), we show the top three principal components, which looks similar before and after translation. This indicates that our encoder is able to bring the corresponding patches from two domains into a similar location in the feature embedding space.

Additional applications. Figure 8 shows additional results: Parisian street → Burano's brightly painted houses and Russian Blue cat → Grumpy cat.

4.3 High-resolution single image translation

Finally, we conduct experiments in the single image setting, where both the source and target domain only have one image each. Here, we transfer a Claude Monet's painting to a natural photograph. Recent methods [64,65] have explored training unconditional models on a single image. Bearing the additional challenge of respecting the structure of the input image, conditional image synthesis using only one image has not been explored by previous image-to-image translation methods. Our painting → photo task is also different from neural style transfer [19,34] (photo → painting) and photo style transfer [48,82] (photo → photo).

Since the whole image (at HD resolution) cannot fit on a commercial GPU, at each iteration we train on 16 random crops of size 128×128. We also randomly

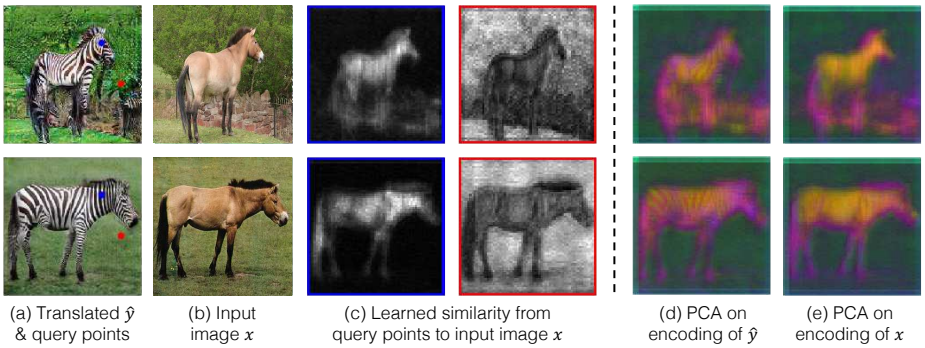

Fig. 7: **Visualizing the learned similarity by G_{enc}.** Given query points (blue or red) on an output image (a) and input (b), we visualize the learned similarity to patches on the input image by computing $\exp(\boldsymbol{v} \cdot \boldsymbol{v}^{-}/\tau)$ in (c). Here $\boldsymbol{v}$ is the query patch in the output and $\boldsymbol{v}^{-}$ denotes patches from the input. This suggests that our encoder may learn cross-domain correspondences implicitly. In (d) and (e), we visualize the top 3 PCA components of the shared embedding.

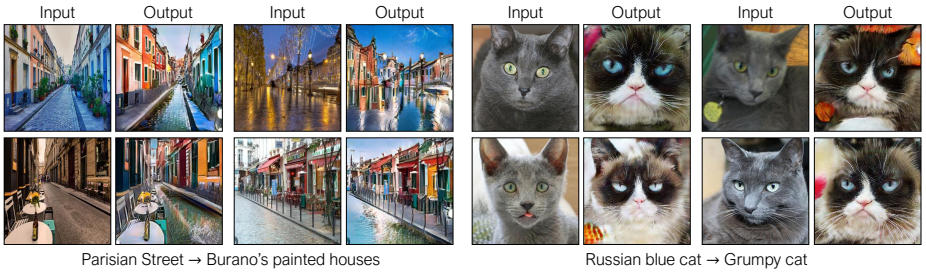

Fig. 8: **Additional applications** on Parisian street → Burano's colored houses and Russian Blue cat → Grumpy cat.

scale the image to prevent overfitting. Furthermore, we observe that limiting the receptive field of the discriminator is important for preserving the structure of the input image, as otherwise the GAN loss will force the output image to be identical to the target image. Therefore, the crops are further split into 64×64 patches before passed to the discriminator. Lastly, we find that using gradient penalty [53,35] stabilizes optimization. We call this variant SinCUT.

Figure 9 shows a qualitative comparison between our results and baseline methods including two neural style transfer methods (Gatys et al. [19] and STROTSS [39]), one leading photo style transfer method WCT^2 [82], and a CycleGAN baseline [89] that uses the ℓ_1 cycle-consistency loss instead of our contrastive loss at the patch level. The input paintings are high-res, ranging from 1k to 1.5k. Appendix B includes additional examples. We observe that Gatys et al. [19] fails to synthesize realistic textures. Existing photo style transfer methods such as WCT^2 can only modify the color of the input image. Our method SinCUT outperforms CycleGAN and is comparable to a leading style transfer method [39], which is based on optimal transport and self-similarity. Interestingly, our method

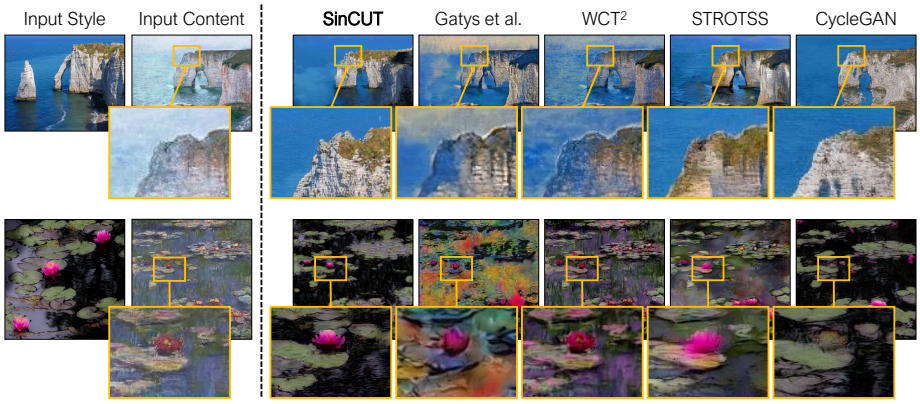

Fig. 9: **High-res painting to photo translation.** We transfer Claude Monet's paintings to reference natural photographs. The training only requires a single image from each domain. We compare our results (SinCUT) to recent style and photo transfer methods including Gatys et al. [19], WCT^2 [82], STROTSS [39], and patch-based CycleGAN [89]. Our method generates can reproduce the texture of the reference photo while retaining structure of input painting. Our generation is at 1k $\sim$ 1.5k resolution.

is not originally designed for this application. This result suggests the intriguing connection between image-to-image translation and neural style transfer.

5 Conclusion

We propose a straightforward method for encouraging content preservation in unpaired image translation problems – by maximizing the mutual information between input and output with contrastive learning. The objective learns an embedding to bringing together corresponding patches in input and output, while pushing away noncorresponding "negative" patches. We study several important design choices. Interestingly, drawing negatives from *within* the image itself, rather than other images, provides a stronger signal. Our method *learns a cross-domain similarity function* and is the first image translation algorithm, to our knowledge, to not use any pre-defined similarity function (such as ℓ_1 or perceptual loss). As our method does not rely on cycle-consistency, it can enable one-sided image translation, with better quality than established baselines. In addition, our method can be used for *single-image* unpaired translation.

Acknowledgments. We thank Allan Jabri and Phillip Isola for helpful discussion and feedback. Taesung Park is supported by a Samsung Scholarship and an Adobe Research Fellowship, and some of this work was done as an Adobe Research intern. This work was partially supported by NSF grant IIS-1633310, grant from SAP, and gifts from Berkeley DeepDrive and Adobe.

References

1. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. In: International Conference on Machine Learning (ICML) (2018) 3
2. Amodio, M., Krishnaswamy, S.: Travelgan: Image-to-image translation by transformation vector learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8983–8992 (2019) 3
3. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 2, 5
4. Benaim, S., Wolf, L.: One-sided unsupervised domain mapping. In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 2, 3, 8, 9, 20
5. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
6. 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) 27
7. 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) 7, 27
8. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: IEEE International Conference on Computer Vision (ICCV) (2017) 3
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (ICML) (2020) 2, 4, 6, 10
10. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 3
11. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 9
12. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 4
13. 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) 9, 20
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 3, 9
15. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: IEEE International Conference on Computer Vision (ICCV) (2015) 4
16. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in Neural Information Processing Systems (2016) 3

17. Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **38**(9), 1734–1747 (2015) 4
18. 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) 2, 3, 8, 9, 20, 26
19. 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) 3, 12, 13, 14, 23, 24, 25
20. Gokaslan, A., Ramanujan, V., Ritchie, D., In Kim, K., Tompkin, J.: Improving shape deformation in unsupervised image-to-image translation. In: European Conference on Computer Vision (ECCV) (2018) 3
21. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014) 1, 3, 4
22. 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) 23
23. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2010) 4
24. 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) 2, 4, 5, 6, 10, 11, 26, 28
25. Hénaff, O.J., Razavi, A., Doersch, C., Eslami, S., Oord, A.v.d.: Data-efficient image recognition with contrastive predictive coding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 4, 5, 6, 10
26. 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) 9, 26
27. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006) 4
28. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018) 4
29. 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) 10, 20
30. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. European Conference on Computer Vision (ECCV) (2018) 3, 20
31. 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) 1, 3, 8, 9, 26
32. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Crisp boundary detection using pointwise mutual information. In: European Conference on Computer Vision (ECCV) (2014) 7
33. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. arXiv preprint arXiv:1511.06811 (2015) 4

34. 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) 3, 8, 12, 26
35. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 13
36. 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) 8, 23
37. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: International Conference on Machine Learning (ICML) (2017) 1, 3
38. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015) 26
39. 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) 13, 14, 23, 24, 25
40. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6874–6883 (2017) 4
41. 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) 3, 8, 9, 20
42. Li, C., Liu, H., Chen, C., Pu, Y., Chen, L., Henao, R., Carin, L.: Alice: Towards understanding adversarial learning for joint distribution matching. In: Advances in Neural Information Processing Systems (2017) 7
43. Liang, X., Zhang, H., Lin, L., Xing, E.: Generative semantic manipulation with mask-contrasting gan. In: European Conference on Computer Vision (ECCV) (2018) 3, 7
44. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems (2017) 3, 8, 9
45. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: IEEE International Conference on Computer Vision (ICCV) (2019) 3
46. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104 (2016) 4
47. Löwe, S., O'Connor, P., Veeling, B.: Putting an end to end-to-end: Gradient-isolated learning of representations. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 4
48. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 12
49. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of Exemplar-SVMs for object detection and beyond. In: IEEE International Conference on Computer Vision (ICCV) (2011) 4
50. 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) 8, 26
51. Mechrez, R., Talmi, I., Shama, F., Zelnik-Manor, L.: Maintaining natural image statistics with the contextual loss. In: Asian Conference on Computer Vision (ACCV) (2018) 3

52. 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) 3, 7, 23
53. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International Conference on Machine Learning (ICML) (2018) 13, 23
54. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. arXiv preprint arXiv:1912.01991 (2019) 4
55. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: European Conference on Computer Vision. pp. 527–544. Springer (2016) 4
56. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: International Conference on Machine Learning (ICML) (2011) 4
57. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 2, 4, 6
58. Owens, A., Wu, J., McDermott, J.H., Freeman, W.T., Torralba, A.: Ambient sound provides supervision for visual learning. In: European Conference on Computer Vision (ECCV) (2016) 4
59. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
60. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016) 4
61. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2016) 23
62. Rao, K., Harris, C., Irpan, A., Levine, S., Ibarz, J., Khansari, M.: Rl-cyclegan: Reinforcement learning aware simulation-to-real. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 10
63. 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) 20
64. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: IEEE International Conference on Computer Vision (ICCV) (2019) 12
65. Shocher, A., Bagon, S., Isola, P., Irani, M.: Ingan: Capturing and remapping the" dna" of a natural image. In: IEEE International Conference on Computer Vision (ICCV) (2019) 12
66. Shocher, A., Cohen, N., Irani, M.: zero-shot super-resolution using deep internal learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 7
67. Shrivastava, A., Malisiewicz, T., Gupta, A., Efros, A.A.: Data-driven visual similarity for cross-domain image matching. ACM Transactions on Graphics (SIGGRAPH Asia) **30**(6) (2011) 4
68. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
69. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015) 3

70. 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) 26
71. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: International Conference on Learning Representations (ICLR) (2017) 3, 7
72. Tang, H., Xu, D., Sebe, N., Yan, Y.: Attention-guided generative adversarial networks for unsupervised image-to-image translation. In: International Joint Conference on Neural Networks (IJCNN) (2019) 3
73. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019) 4
74. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 27
75. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
76. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: International Conference on Machine Learning (ICML) (2008) 4
77. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 3
78. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) 3
79. Wu, W., Cao, K., Li, C., Qian, C., Loy, C.C.: Transgaga: Geometry-aware unsupervised image-to-image translation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
80. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2, 4, 5, 6
81. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. In: IEEE International Conference on Computer Vision (ICCV) (2017) 1, 3
82. 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) 12, 13, 14, 23, 24, 25
83. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 9, 26
84. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: A feature similarity index for image quality assessment. IEEE transactions on Image Processing **20**(8), 2378–2386 (2011) 3
85. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision (ECCV) (2016) 4, 9
86. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
87. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 586–595 (2018) 3
88. Zhang, R., Pfister, T., Li, J.: Harmonic unpaired image-to-image translation. In: International Conference on Learning Representations (ICLR) (2019) 3

89. 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, 3, 7, 8, 9, 10, 13, 14, 20, 23, 24, 25, 26, 28
90. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems (2017) 3
91. Zontak, M., Irani, M.: Internal statistics of a single natural image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 7

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 10, 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 [89], MUNIT [30], DRIT [41], Self-Distance and DistanceGAN [4], and GcGAN [18].

A.2 Additional datasets

In Figure 11 and Figure 12, we show additional datasets, compared against baseline method CycleGAN [89]. 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 [89].
- *Yosemite Summer→Winter* contains 1,273 summer and 854 winter images of Yosemite scraped using the FlickAPI was introduced in CycleGAN [89].
- *GTA→Cityscapes* GTA contains 24,966 images [63] and Cityscapes [13] contains 19,998 images of street scenes from German cities. The task was originally used in CyCADA [29].

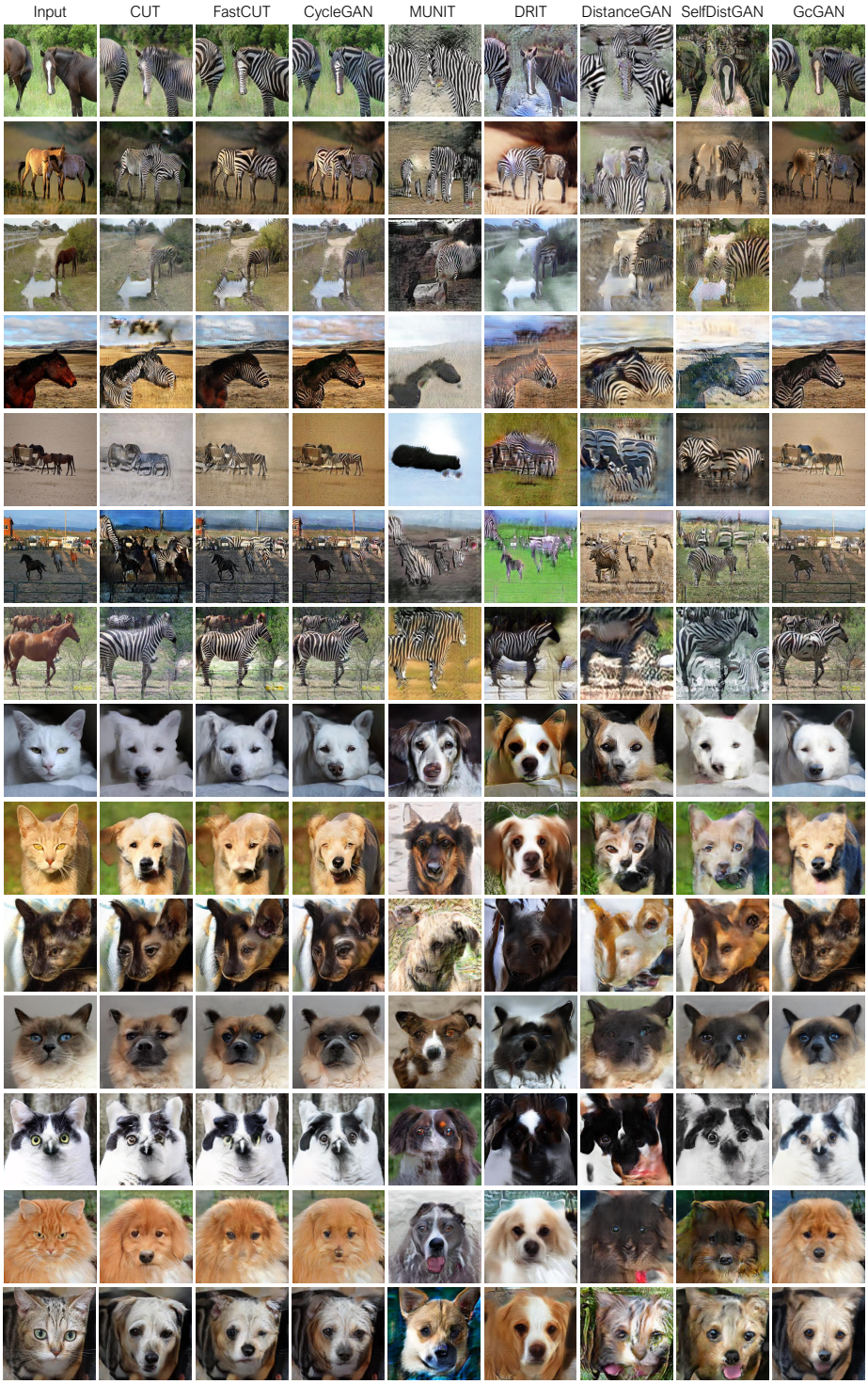

Fig. 10: **Randomly selected Horse→Zebra and Cat→Dog results**. This is an extension of Figure 3 in the main paper.

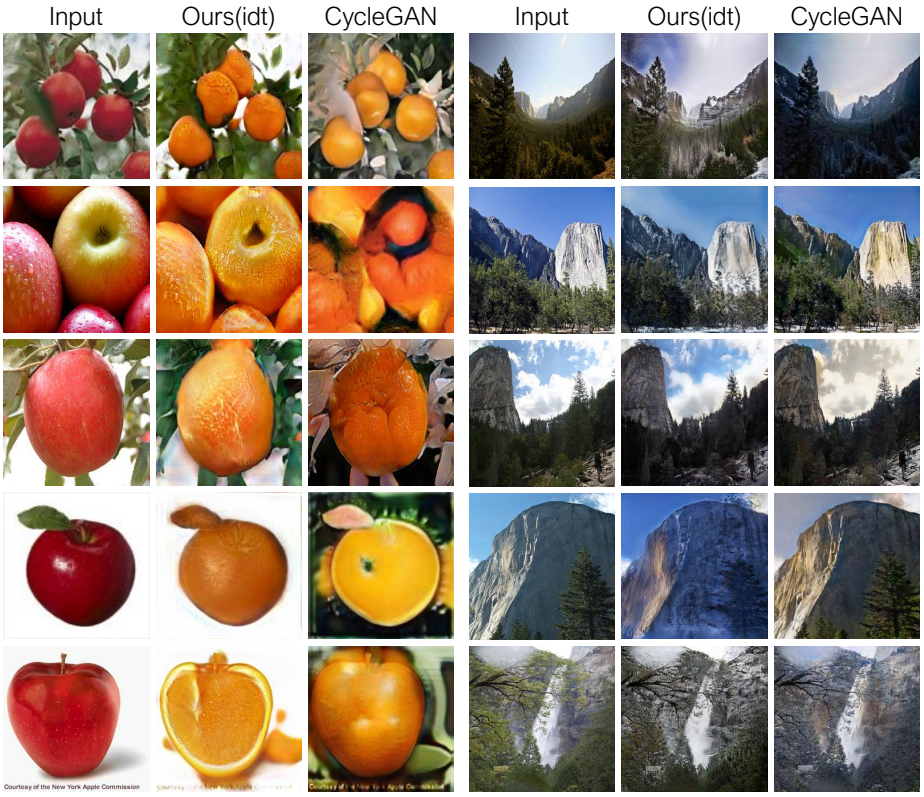

Fig. 11: **Apple→Orange** and **Summer→Winter Yosemite**. CycleGAN models were downloaded from the authors' public code repository. Apple→Orange shows that CycleGAN may suffer from color flipping issue.

Fig. 12: **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 13 and Figure 14, 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 [36] 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 [61] with R1 gradient penalty [53]. Since we do not use style code, the style modulation layer of StyleGAN2 was removed.

Single image results.

In Figure 13 and Figure 14, we show additional comparison results for our method, Gatys et al. [19], STROTSS [39], WCT^2 [82], and CycleGAN baseline [89]. 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 [19], and WCT^2. The quality of our results is on par with results from STROTSS [39]. Note that STROTSS [39] compares to and outperforms recent style transfer methods (e.g., [22,52]).

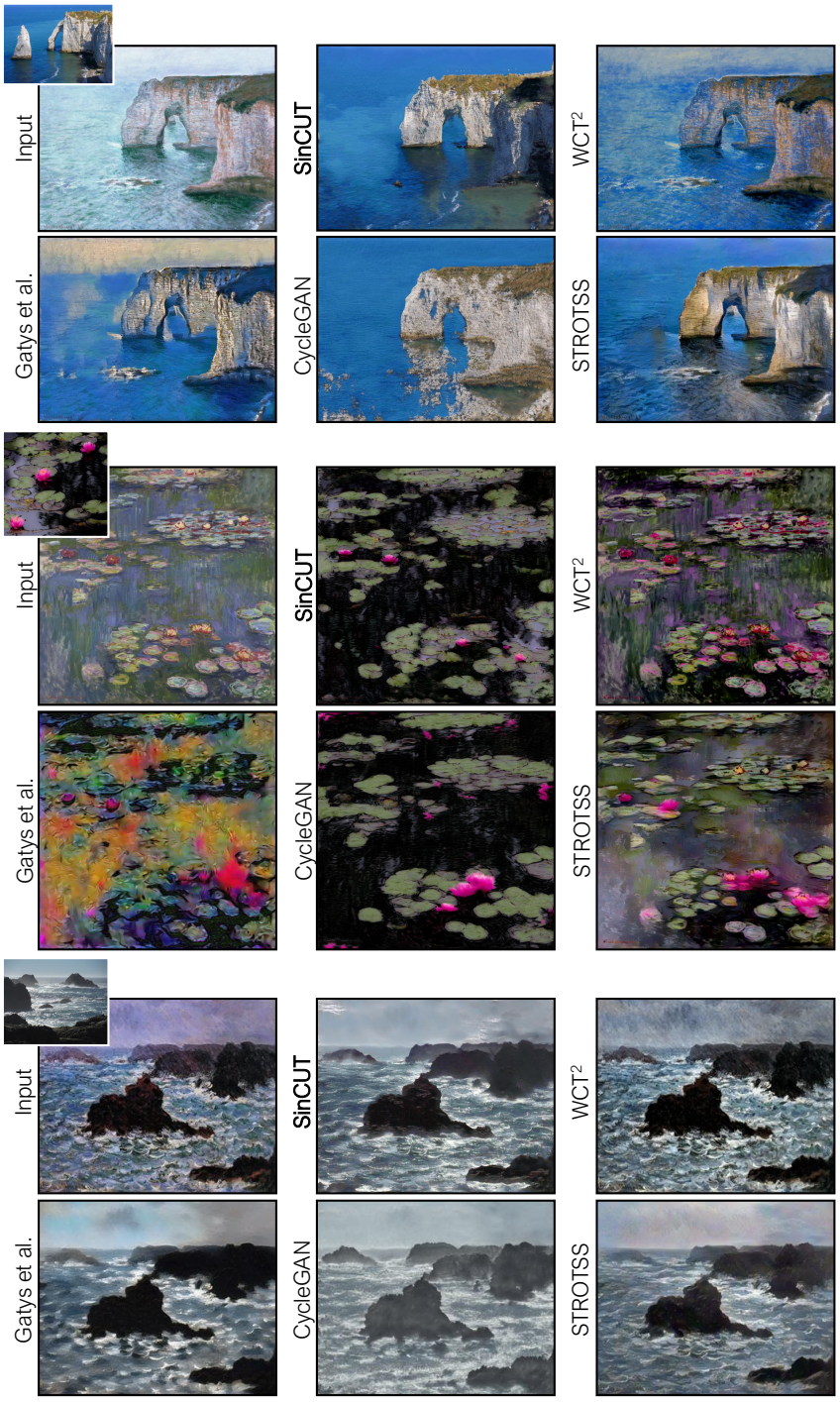

Fig. 13: **High-res painting to photo translation (I).** We transfer Monet's paintings to reference natural photos shown as insets at top-left corners. The training only requires a single image from each domain. We compare our results to recent style and photo transfer methods including Gatys et al. [19], WCT^2 [82], STROTSS [39], and our modified patch-based CycleGAN [89]. 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.

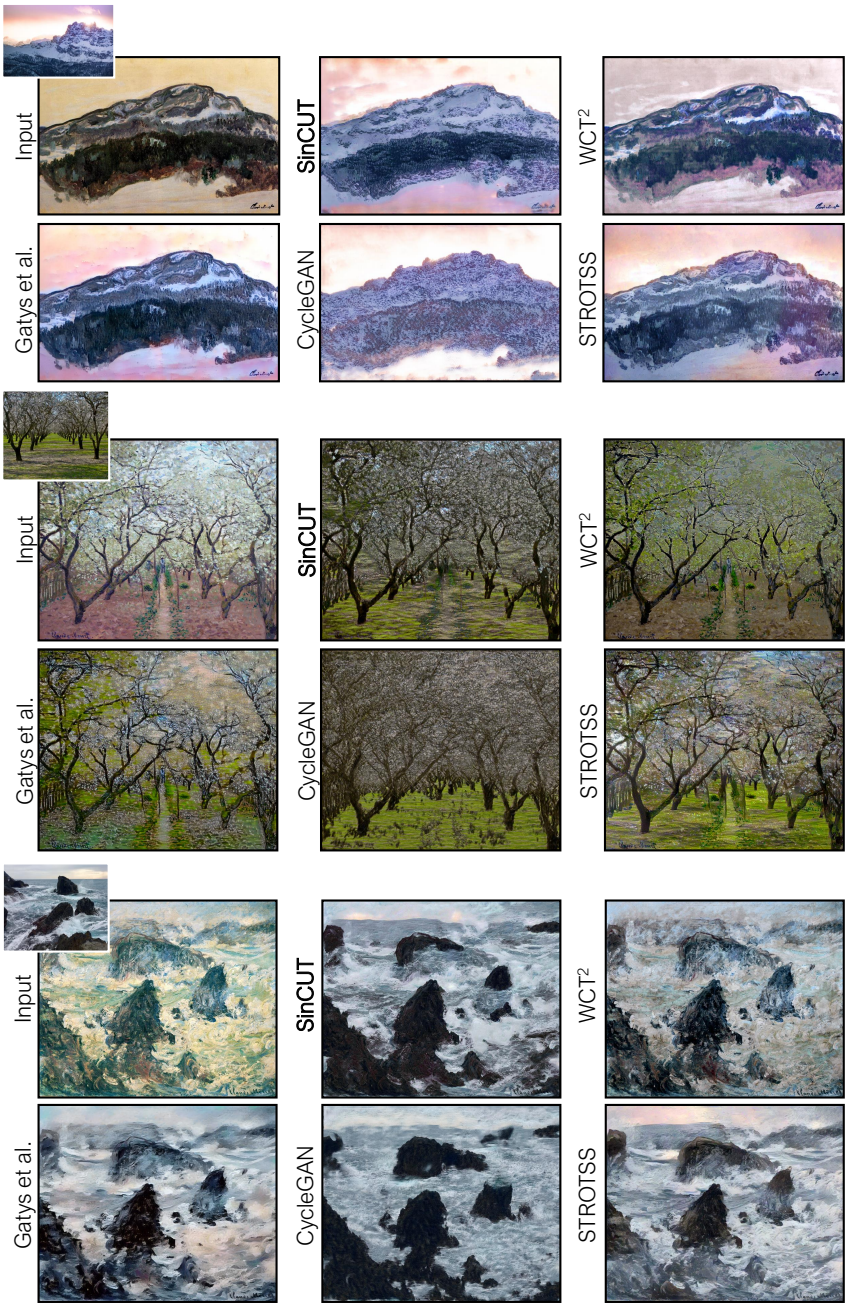

Fig. 14: **High-res painting to photo translation (II).** We transfer Monet's paintings to reference natural photos shown as insets at top-left corners. The training only requires a single image from each domain. We compare our results to recent style and photo transfer methods including Gatys et al. [19], WCT^2 [82], STROTSS [39], and our modified patch-based CycleGAN [89]. 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 [34] with 9 residual blocks, PatchGAN discriminator [31], Least Square GAN loss [50], batch size of 1, and Adam optimizer [38] 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 [18], 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 [89]. 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 [24], 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 [26]) 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 [70] 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 [83] 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 15, 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 [7] trained on COCO-Stuff [6], to measure the percentage of pixels that belong to horses and zebras[1]. The training set exhibits dataset bias [74]. 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

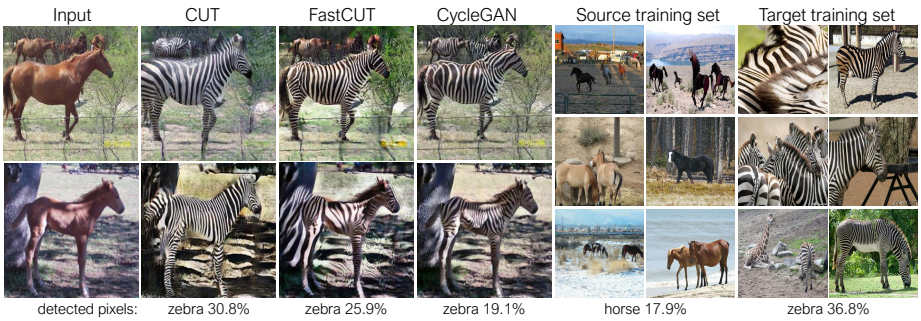

Fig. 15: **Distribution matching.** We measure the percentage of pixels belonging to the horse/zebra bodies, using a pre-trained semantic segmentation model. We find a distribution mismatch between sizes of horses and zebras images – zebras usually appear larger (36.8% vs. 17.9%). Our full method CUT has the flexibility to enlarge the horses, as a means of better matching of the training statistics than CycleGAN [89]. 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 [89]. 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 [24], 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.

Appendix D Changelog

v1 Initial preprint release (ECCV 2020)

v2 and v3 (1) Fix typos in Eqn. 3 and Eqn. 4. (2) Add additional related work.