Supplementary Material – Sym-parameterized Dynamic Inference for Mixed-Domain Image Translation

Simyung Chang[1,2], SeongUk Park[1], John Yang[1], Nojun Kwak[1]
[1]Seoul National University, Seoul, Korea
[2]Samsung Electronics, Suwon, Korea
{timelighter, swpark0703, yjohn, nojunk}@snu.ac.kr

A. Experiment Settings

A.1. 1D Toy Example

The polynomial $g(x)$ and the linear function $h(x)$ are defined such that the output is between 0 and 1 when the range of x is -1 to 1.

$$g(x) = x(x - 0.8)(x + 0.9) + 0.5 \tag{1}$$

$$h(x) = -0.1x + 0.5 \tag{2}$$

The sym-parametrized network $f(x, S)$ is an MLP network that has three hidden layers with the size of 64 and ReLU activations. It takes a tuple of (x, S) as an input and outputs a single value for either classification or regression. The mean squared error function is used for regression loss $\mathcal{L}_r$, and binary cross entropy is used for the classification loss $\mathcal{L}_c$. $\mathcal{L}_c$ is scaled down to 20% to balance the losses. In the training phase, the sym-parameter S is sampled from Dirichlet distribution with α valued (0.5, 0.5). We use ADAM with a batch size of 16, and learning rates of 0.01 during the first 200 epochs, 0.001 during the next 200 epochs and 0.0001 during the last 100 epochs.

A.2. Sym-Parameterized Generative Network

For three loss terms, $\mathcal{L}_{rec}$, $\mathcal{L}_{adv}$ and $\mathcal{L}_{per}$, we use L_1 norm for the reconstruction loss $\mathcal{L}_{rec}$, and have applied a technique of LSGAN [3] for the adversarial loss $\mathcal{L}_{adv}$ in which an MSE (mean squared error) loss is used. Additionally, the identity loss introduced by Taigman *et al.* [5] and used in CycleGAN [6] is used in $\mathcal{L}_{adv}$ for regularizing the generator, combined with the LSGAN loss. The implementation of the perceptual loss $\mathcal{L}_{per}$ follow the implementation of Johnson *et al.* [2], adopting both feature reconstruction loss and style reconstruction loss. At the training phase, all three loss terms were weighted by numbers sampled from Dirichlet distribution with all the α valued 0.5.

The architecture details of an SGN generator is provided in Table 1 of this supplementary. The discriminator is equivalently set as CycleGAN [6]. To regulate the imbalance between the losses, $\mathcal{L}_{rec}$ and $\mathcal{L}_{adv}$ are respectively weighted with 2 and 1. And for $\mathcal{L}_{adv}$, GAN loss and the identity loss are respectively weighted by 1 and 5. For the perceptual loss $\mathcal{L}_{per}$, we have used the pretrained Pytorch model of VGG16 [4] without batch-normalization layers. $\mathcal{L}_{per}$ is composed of a content loss and a style loss, computed at first 4 blocks of VGG16, which means it uses output features from Conv1-2, Conv2-2, Conv3-3, Conv4-3, and do not use the fifth block, following the work of Johnson *et al.* [2]. The content loss is weighted with 0.001 and the style losses at 4 layers are weighted as (0.1, 1.0, 10, 5.0)$\times$200 for the Model 1, (0.1, 1.0, 10, 5.0)$\times$100 for the other models. We use the ADAM optimizer with a batch size of 4. Then Model 1 and Model 2 are trained for 20 epochs and Model 3 is trained for 60 epochs. We keep the learning rate of 0.0002 for the first half of epochs and linearly decay it to zero for the remaining epochs. Source code and pre-trained networks are available in https://github.com/TimeLighter/pytorch-sym-parameter.

B. Additional Experimental Results

B.1. Continuous Translation

The video (https://youtu.be/i1XsGEUpfrs) is created by extracting images from the original and translating it through SGN. *ORIGINAL VIDEO* and *VIDEO WITH SGN* represent original video and SGN translated video, respectively.

Layer Configuration	Input Dimension	Layer Information	Output Dimension
Input convolution	$(h, w, 3)$	Convolution (K:7x7, S:1, P:3), IN, ReLU	$(h, w, 64)$
CCAM (Reduction Rate r: 4)			
2*Down-sampling	$(h, w, 64)$	Convolution (K:3x3, S:2, P:1), IN, ReLU	$(\frac{h}{2}, \frac{w}{2}, 128)$
	$(\frac{h}{2}, \frac{w}{2}, 128)$	Convolution (K:3x3, S:2, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
CCAM (Reduction rate r: 4)			
9*Residual Blocks	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
	$(\frac{h}{4}, \frac{w}{4}, 256)$	Convolution (K:3x3, S:1, P:1), IN, ReLU	$(\frac{h}{4}, \frac{w}{4}, 256)$
CCAM (Reduction Rate r: 4)			
2*Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256)$	Transposed Convolution (K:3x3, S:2, P:1), IN, ReLU	$(\frac{h}{2}, \frac{w}{2}, 128)$
	$(\frac{h}{2}, \frac{w}{2}, 128)$	Transposed Convolution (K:3x3, S:2, P:1), IN, ReLU	$(h, w, 64)$
Output Convolution	$(h, w, 64)$	Convolution (K:7x7, S:7, P:3), Tanh	$(h, w, 3)$

Table 1. **Generator network architecture.** In the Layer Information column, K: size of the filter, S: stride size, P: padding size, IN: Instance Normalization. CCAM has reduction rate r to reduce the amount of computation like SENet [1].

We use SGN models trained in the paper and translated images changing only the sym-parameters. The color bar above *VIDEO WITH SGN* represents the sym-parameter values for each domain. If the entire bar is red, then the sym-parameter S is (1,0,0). Since all the images are translated through SGN, reconstructed images may differ in some colors and appearance from the original ones.

B.2. More Results of CCAM

Figure 1 summarizes channel activation trends of CCAM when changing sym-parameters for a given test image. As can be seen in the plots, channels are activated differently for three cases of sym-parameter. First layer of CCAM is mainly responsible for scaling with no blocked channel. Considering the number of closed channels with zero activations are increased at deeper layers, CCAM selectively excludes channels and reduces influence of unnecessary channels to generate images in a mixed-domain conditioned by sym-parameters. This is the major difference of our CCAM and the CBN which utilizes bias in controlling each domain's influence. Figure 2 is additional images according to the sym-parameter injection method.

B.3. More Results of Image Translation

Figure 3, 4 and 5 shows additional images for models 1, 2, and 3 of the paper, respectively. Figure 6 is the results of the new SGN model not in the paper.

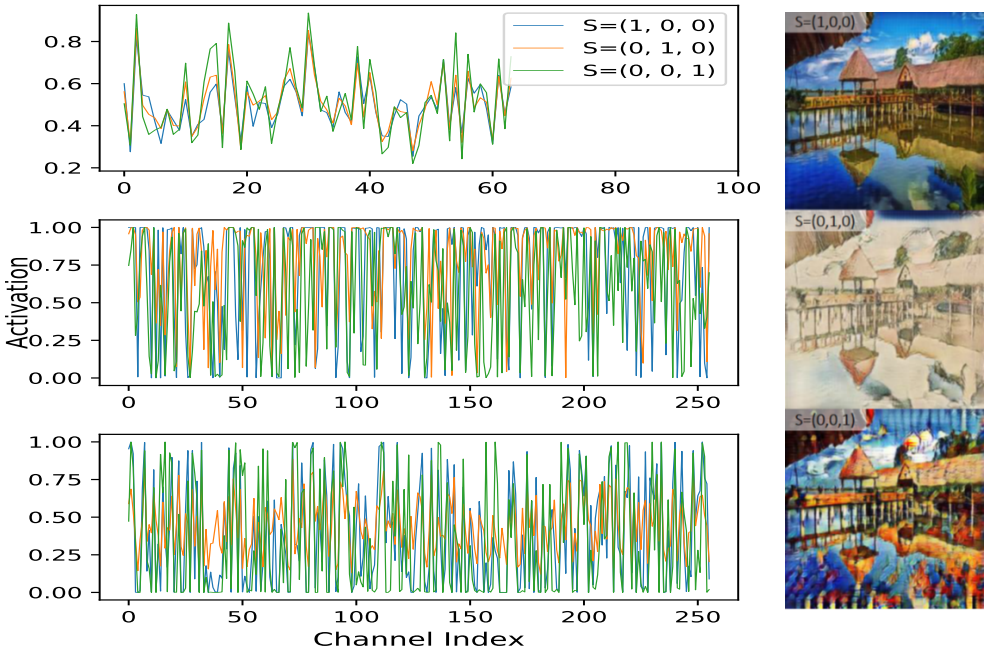

Figure 1. **Channel activation of CCAMs.** Each plot shows the channel activation results of the three CCAMs used in the SGN. The lower plot corresponds to the CCAM of deeper layer. This result indicates that the degree of activation is different for each channel when the sym-parameter is different for the same image.

Figure 2. **More results on injection methods** All settings except for the sym-parameter injection method are equivalently set up as Model 2. CONCAT does not show any difference according to the sym-parameter, CBN shows comparatively domain characteristic, but inter-domain interference is more than CCAM. For example, if we look at the image of $S = (0.5, 0.5, 0.0)$ in CBN, intense color appears at the top, which is not related to domains A and B at all. This result shows that CBN is hard to exclude the effect of domains not related to the input sym-parameter. The CCAM used for SGN has the most explicit domain-to-domain distinction and the least impact of irrelevant domains.

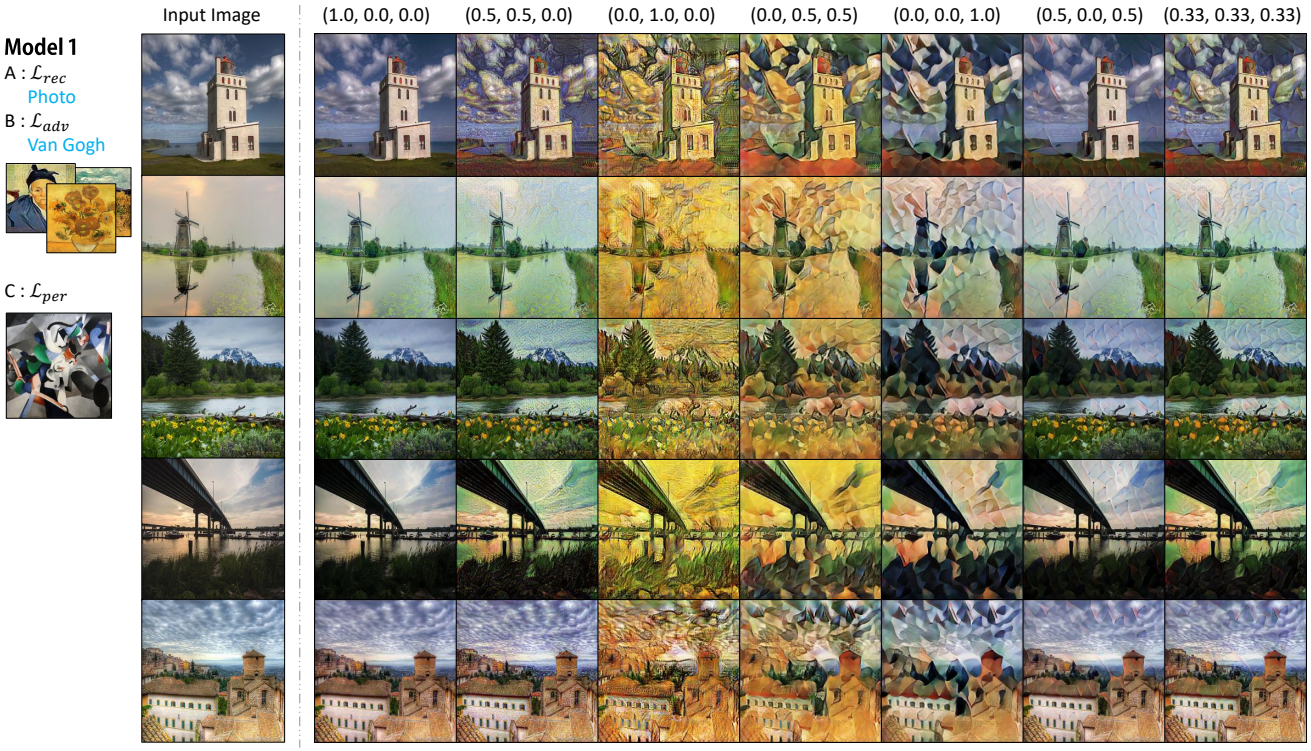

Figure 3. **More results of SGN Model 1.** The numbers in the parentheses are sym-parameters for each A, B, and C domain.

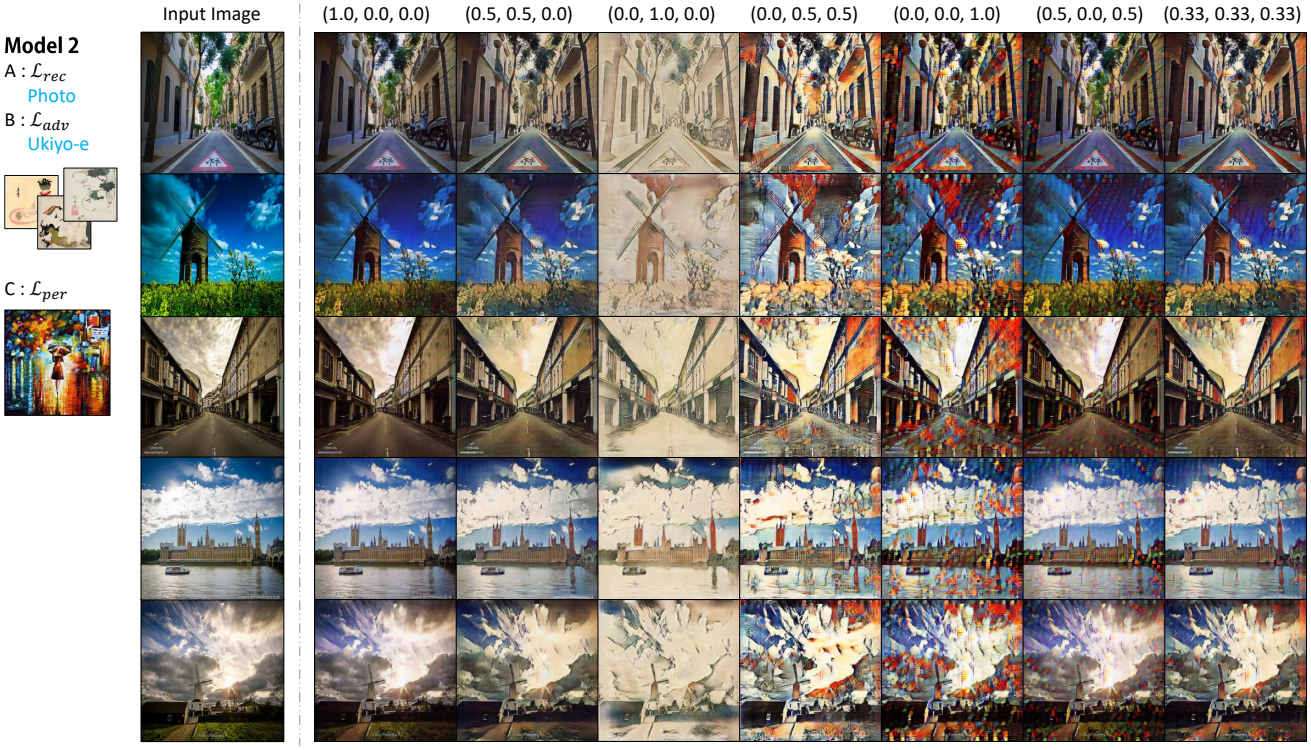

Figure 4. **More results of SGN Model 2.** The numbers in the parentheses are sym-parameters for each A, B, and C domain.

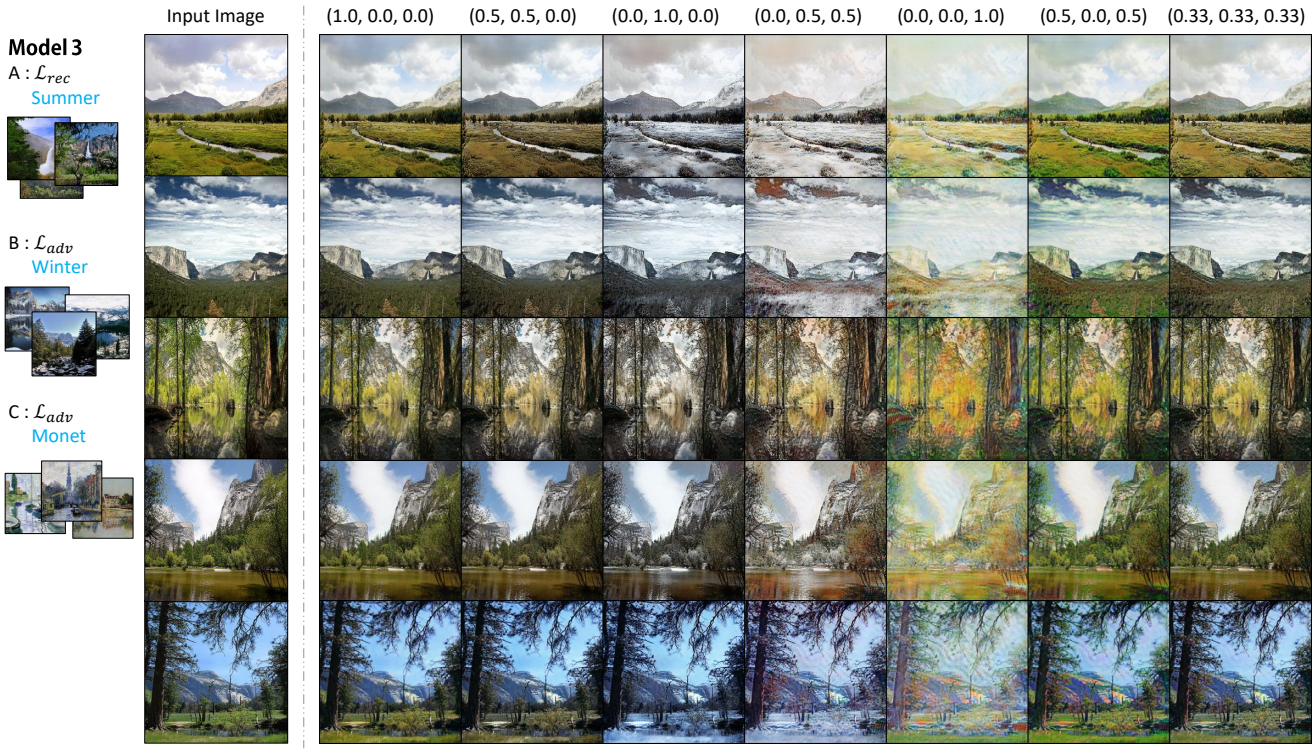

Figure 5. **More results of SGN Model 3.** The numbers in the parentheses are sym-parameters for each A, B, and C domain.

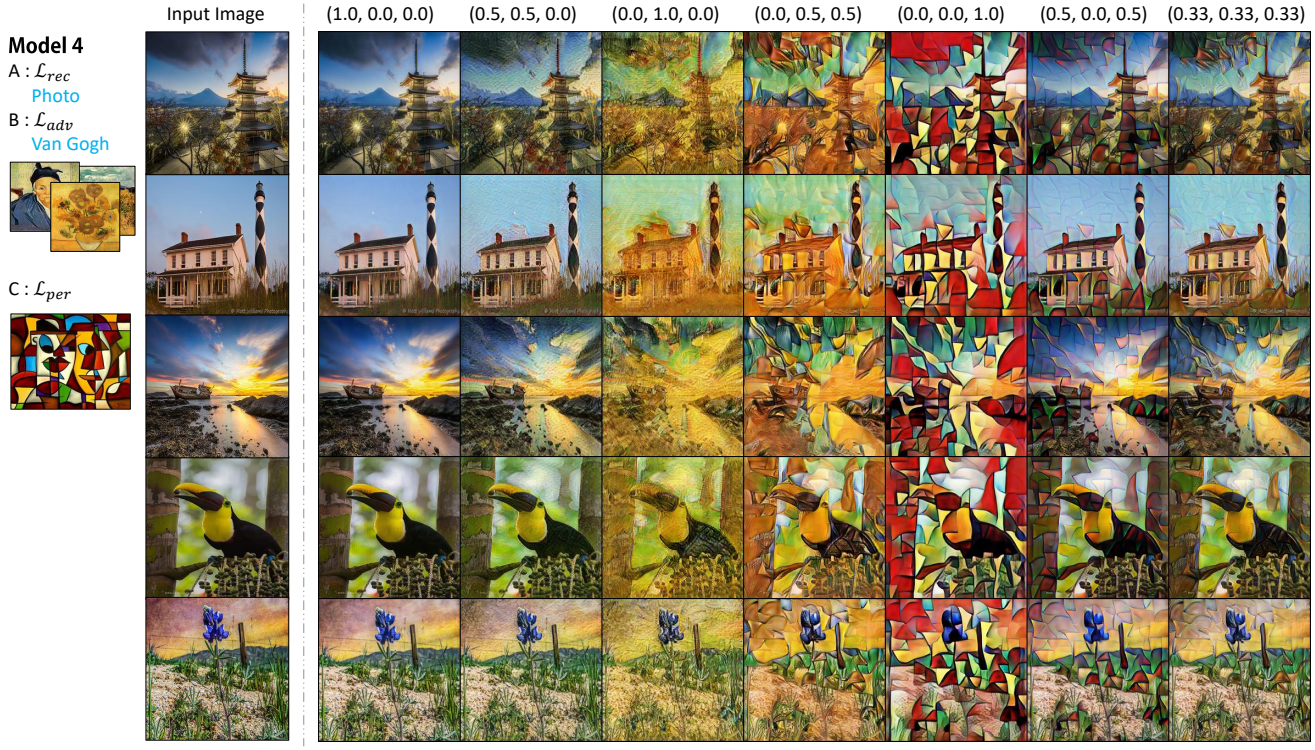

Figure 6. **Translation results of additional SGN model.** The numbers in the parentheses are sym-parameters for each A, B, and C domain.

References

[1] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. 2

[2] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 1

[3] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017. 1

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

[5] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 1

[6] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 1