# Exploring Patch-wise Semantic Relation for Contrastive Learning in Image-to-Image Translation Tasks
# (Supplementary Material)

## A. Experimental Details

### A.1. Single-modal Translation

For implementing single-modal image translation, we modified the official source code of CUT [6][1]. Specifically, instead of PatchNCE in CUT, we use our proposed loss $L_{semantic}$ which includes Decoupled infoNCE with hard negative $L_{hDCE}$ and semantic relation consistency loss $L_{SRC}$. The model architecture is identical for fair comparison.

We introduce the loss functions to train the networks. Firstly, our GAN loss is as follows:

$$L_{GAN} = \mathbb{E}_{y \sim p_Y}\left[\log D(y)\right] + \mathbb{E}_{x \sim p_X}\left[\log(1 - D(G(x)))\right],$$

where $G$ is the generator and $D$ is the discriminator, and $p_X$ and $p_Y$ are the data distributions for source and target domain, respectively.

Then, we calculate the proposed loss using the embedded features for $L$ intermediate layers. Specifically, for each $l$-th layer, the intermediate features of input $x$ and translation output $y_{fake}$ (i.e. $y_{fake} = G(x)$) is projected by the projection head $F^l$. Then, we sample 256 embedding vectors $z_k^l$ and $w_k^l$(i.e. $K = 256$) from the projected features $z^l$ and $w^l$, which are formalized as:

$$\begin{aligned} \left\{z_k^l\right\}_{k=1}^K &\sim z^l = F^l(G_{enc}^l(x)) \\ \left\{w_k^l\right\}_{k=1}^K &\sim w^l = F^l(G_{enc}^l(y_{fake})), \end{aligned}$$

where $G_{enc}^l(x)$ refers the $l$-th layer feature of the encoder, which is a part of the generator. $G_{enc}^l(y_{fake})$ is the feature of the output. Then, the loss of the proposed method is given by

$$\begin{aligned} &L_{semantic, X \to Y}(\lambda_{hDCE}, \lambda_{SRC}, \gamma, \tau) \\ &\quad = \sum_{l=1}^{L} \lambda_{SRC} L_{SRC}(\left\{z_k^l\right\}_{k=1}^K, \left\{w_k^l\right\}_{k=1}^K) \\ &\quad + \lambda_{hDCE} L_{hDCE}(\left\{z_k^l\right\}_{k=1}^K \left\{w_k^l\right\}_{k=1}^K; \gamma, \tau), \end{aligned}$$

| Dataset | $\lambda_{SRC}$ | $\lambda_{hDCE}$ | $\tau$ | $\Gamma_o$ |
|---|---|---|---|---|
| **H→Z** | 0.05 | 0.1 | 0.07 | 50 |
| **city** | 0.1 | 0.1 | 0.07 | 50 |

Table 1. Hyperparameters for each dataset. **H→Z** refers to Horse→Zebra dataset. **city** refers to Cityscapes dataset.

where each loss is calculated as suggested in the main paper.

Similarly, we calculate the loss for the real image $y_{real}$ from target domain and the identity image $y_{idt} = G(y_{real})$ by the generator $G$. Then, we obtain the embedding vectors, which are formalized as:

$$\begin{aligned} \left\{u_k^l\right\}_{k=1}^K &\sim u^l = F^l(G_{enc}^l(y_{real})) \\ \left\{v_k^l\right\}_{k=1}^K &\sim v^l = F^l(G_{enc}^l(y_{idt})) \end{aligned}$$

Then, we calculate the loss using the vectors from $L$ intermediate layers, which is as follows:

$$\begin{aligned} &L_{semantic, Y \to Y}(\lambda_{hDCE}, \lambda_{SRC}, \gamma, \tau) \\ &\quad = \sum_{l=1}^{L} \lambda_{SRC} L_{SRC}(\left\{u_k^l\right\}_{k=1}^K, \left\{v_k^l\right\}_{k=1}^K) \\ &\quad + \lambda_{hDCE} L_{hDCE}(\left\{u_k^l\right\}_{k=1}^K, \left\{v_k^l\right\}_{k=1}^K; \gamma, \tau). \end{aligned}$$

Finally, the total loss is given as:

$$\begin{aligned} L_{total} = L_{GAN} &+ L_{semantic, X \to Y}(\lambda_{hDCE}, \lambda_{SRC}, \gamma, \tau) \\ &+ L_{semantic, Y \to Y}(\lambda_{hDCE}, \lambda_{SRC}, \gamma, \tau) \end{aligned}$$

The hyperparameters for each dataset is shown in the Table 1. Other training settings are same with the CUT [6], for a fair comparison with the baseline.

For the curriculum learning with hard negative mining, we gradually change the parameter $\gamma$ as follows:

$$\gamma(t) = 1/\Gamma(t)$$

and

$$\Gamma(t) = \Gamma_o + (\Gamma_{final} - \Gamma_o) \cdot \frac{t}{T}$$

[1] https://github.com/taesungp/contrastive-unpaired-translation

where $t$ is training step, and $T$ is total number of epoch. For Horse→Zebra datast, we use $\Gamma_o$ as 50 and linearly decrease into $\Gamma_{final} = 10$ at the last training step. For cityscapes dataset, we set the initial value $\Gamma_o$ as 50 and progressively decreases into 30 until 200 epoch, then we fixed $\Gamma(t) = 1$ until final training step.

## A.2. Multi-modal Translation

In the training procedure for the multi-modal translation, all of the images have a resolution of 256×256. The total training step is $T = 100,000$, and we use the models with best quantitative score during training.

For implementing multi-modal image translation, we modified the official source code of StarGANv2 [2][2]. Specifically, the basic StarGANv2 consists of 4 different networks: a discriminator $D$, generator $G$, style encoder $E$, and latent mapping network $M$. Since we experimented our method on the dataset which mostly consists of scenery images, we used the modified version of generator model which has two downsample layers (until resolution 64×64) and 4 bottleneck layers, 4 bottleneck layers with AdaIN, and two upsample layers with AdaIN.

For feature embedding, we select the intermediate features of bottleneck layers without AdaIN (i.e. $L = 4$). Similar to the single-modal image translation, we project each $l$-th layer feature using the projection head $F^l$. We sample 256 vectors (i.e. $K = 256$) for the features of input $x$ and translated output $G(x, \hat{s})$, where $\hat{s}$ is the style code generated from a random vector. Then, the embedded vectors are formalized as:

$$\begin{aligned}\{z_k^l\}_{k=1}^K &\sim z^l = F^l(G_{enc}^l(x))\\ \{w_k^l\}_{k=1}^K &\sim w^l = F^l(G_{enc}^l(G(x, \hat{s})),\end{aligned}$$

where $G_{enc}^l$ represents $l$-th feature of the encoder part in the generator model.

Now, we introduce the loss functions to train the networks. First, to generate realistic images, we use GAN loss which is as follows:

$$L_{adv} = \mathbb{E}_{x,y}[\log D_y(x)] + \mathbb{E}_{x,\hat{y},z}[\log(1 - D_{\hat{y}}(G(x, \hat{s}))],$$

where $y$ is an one-hot label for the domain of the input $x$, and $\hat{y}$ is a target domain one-hot label. The style code $\hat{s} = M_{\hat{y}}(z)$ is a style code for the target domain $\hat{y}$, which is generated by $M$ from a random vector $z$ Then, the cyclic losses for the style code and the output image are added as following:

$$\begin{aligned}L_{sty} &= \mathbb{E}_{x,\hat{y},z}[||\hat{s} - E_{\hat{y}}(G(x, \hat{s}))||_1]\\ L_{cyc} &= \mathbb{E}_{x,y,\hat{y},z}[||x - G(G(x, \hat{s}), \tilde{s})||_1]\end{aligned}$$

where $\tilde{s} = E_y(x)$ is predicted style code for the input $x$. Furthermore, the diversity sensitive loss is imposed to generate images with diverse styles.

$$L_{ds} = \mathbb{E}_{x,\hat{y},z_1,z_2}[||G(x, \hat{s}_1) - G(x, \hat{s}_2)||_1]$$

where $z_1$ and $z_2$ are two independently sampled random vectors, and the style codes are generated as $\hat{s}_1 = M_{\hat{y}}(z_1)$ and $\hat{s}_2 = M_{\hat{y}}(z_2)$.

Finally, we add our proposed loss function to utilize the patch-wise semantic relation. As in the single-modal image translation, the proposed loss $L_{semantic}$ is as follows:

$$\begin{aligned}&L_{semantic}(\lambda_{hDCE}, \lambda_{SRC}, \gamma, \tau)\\ &\quad= \sum_{l=1}^{L} \lambda_{SRC} L_{SRC}(\{z_k^l\}_{k=1}^K, \{w_k^l\}_{k=1}^K)\\ &\quad\quad + \lambda_{hDCE} L_{hDCE}(\{z_k^l\}_{k=1}^K, \{w_k^l\}_{k=1}^K; \gamma, \tau)\end{aligned}$$

Hence, the final loss function is,

$$\begin{aligned}L_{total} = &\lambda_{adv} L_{adv} + \lambda_{sty} L_{sty} + \lambda_{cyc} L_{cyc}\\ &- \lambda_{ds} L_{ds} + L_{semantic}(\lambda_{SRC}, \lambda_{hDCE}, \tau, \gamma)\end{aligned}$$

Similar to the single-modal I2I framework, $\tau$ is fixed as 0.07. Also, we gradually increase the $\gamma$ with reciprocal function given as,

$$\gamma(t) = 1/\Gamma(t)$$

and

$$\Gamma(t) = \Gamma_o + (\Gamma_{final} - \Gamma_o) \cdot \frac{t}{T}$$

where $t$ is training step, and $T$ is total number of epoch. For both datasets, we set $\Gamma_o$ as 50, and $\Gamma_{final}$ as 1. Other training settings are identical to the baseline [2] for both datasets.

The detailed hyperparameter settings are in Table 2.

| **Dataset** | $\lambda_{adv}$ | $\lambda_{sty}$ | $\lambda_{cyc}$ | $\lambda_{ds}$ | $\lambda_{SRC}$ | $\lambda_{hDCE}$ |
|---|---|---|---|---|---|---|
| Seasons | 1 | 1 | 1 | 2 | 0.1 | 1 |
| Weather | 1 | 1 | 1 | 2 | 0.1 | 0.1 |

Table 2. Hyperparameters for multimodal image translation.

## A.3. GAN compression

We used the Fast GAN compression framework [5] along with our loss functions for hDCE and SRC. The training stage consists of two steps: the first step is training the student model, and the second step is searching the optimal channel configuration by evolution search algorithm. We transferred the semantic relational knowledge in the first step of the training.

[2] https://github.com/clovaai/stargan-v2

We introduce the loss functions to train the student model. As introduced in [5], pretrained teacher discriminator $D$ is used to guide the student model $G_s$. The GAN loss function for the student $G_s$ is as following:

$$L_{GAN} = \mathbb{E}_{y \sim p_Y}\left[\log D(y)\right] + \mathbb{E}_{x \sim p_X}\left[\log(1 - D(G_s(x)))\right]$$

where $p_X$ is the distribution for the input domain, and $p_Y$ is for the target domain. Then, the distillation loss imposes the layer-wise feature matching between the teacher $G_t$ and the student $G_s$. The channel configuration is different between the student model and the teacher. Hence, for each $l$-th layer, a network $f_l$ is used to map the student's feature to have the same number of channel with the corresponding teacher's feature. The distillation loss is calculated as:

$$L_{distill} = \sum_{l=1}^{L} ||f_l(G_s^l(x)) - G_t^l(x)||_2$$

where $G_s^l(x)$ refers the intermediate feature from $l$-th layer of the student model for input $x$. $G_t^l(x)$ represents the feature from the teacher model. $L$ is the total number of the intermediate layers for the distillation. Additionally, the outputs of the teacher and the student are matched by the reconstruction loss formulated as:

$$L_{recon} = ||G_s(x) - G_t(x)||_1$$

Lastly, we transfer the patch-wise semantic relation formed by the hard negative DCE loss, as our method proposes. We applied our method for all $L$ layers. For the $l$-th layer features of the teacher and the student, we project them with the shared projection head $F^l$. Then, we sample 256 embedding vectors $z_k^l$ and $w_k^l$ (i.e. $K = 256$) from the projected features $z^l$ and $w^l$, which is formalized as:

$$\{z_k^l\}_{k=1}^K \sim z^l = F^l(G_t^l(x))$$
$$\{w_k^l\}_{k=1}^K \sim w^l = F^l(G_s^l(x)).$$

Then the proposed loss is given as:

$$\begin{aligned} L_{semantic} = &\sum_{l=1}^{L} \lambda_{SRC} L_{SRC}(\{z_k^l\}_{k=1}^K, \{w_k^l\}_{k=1}^K) \\ &+ \lambda_{hDCE} L_{hDCE}(\{z_k^l\}_{k=1}^K, \{w_k^l\}_{k=1}^K; \tau, \gamma) \end{aligned}$$

The $\gamma$ is gradually increased in reciprocal function for the curriculum learning. Specifically, $\gamma$ is set as:

$$\gamma(t) = 1/\Gamma(t)$$

with

$$\Gamma(t) = \Gamma_o + (\Gamma_{final} - \Gamma_o) \cdot \frac{t}{T}$$

where $t$ is the training step and $T$ is the total epoch. $\Gamma_o$ and $\Gamma_{final}$ are set as 50 and 10 for all three datasets.

| Dataset | Model | $\lambda_{recon}$ | $\lambda_{distill}$ | $\lambda_{SRC}$ | $\lambda_{hDCE}$ |
|---|---|---|---|---|---|
| **H→Z** | CyleGAN | 10 | 0.01 | 10 | 0.1 |
| **M→S** | Pix2pix | 10 | 0.01 | 10 | 0.1 |
| **city** | Pix2pix | 100 | 1 | 10 | 0.1 |

Table 3. Hyperparameters for each dataset. **H→Z** refers to Horse→Zebra dataset. **M→S** refers to Map→Satellite dataset. **city** refers to Cityscapes dataset.

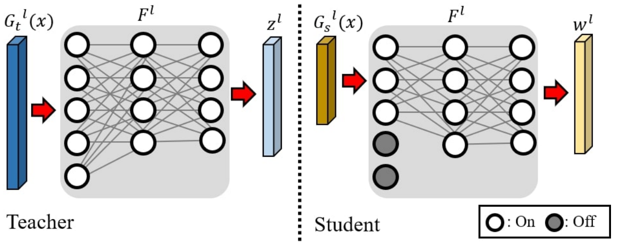


Figure 1. Projection head $F^l$ with elastic width for the GAN compression task. $l$-th layer features $G_s^l(x)$ and $G_t^l(x)$ are projected into embedding vector $z^l$ and $w^l$ with same size for the hDCE loss. For the student, only the subset of weight is used.

In summary, the final loss function to train the student model is,

$$\begin{aligned} L_{total} = &L_{GAN} + \lambda_{distill} L_{distill} + \lambda_{recon} L_{recon} \\ &+ L_{semantic}(\lambda_{SRC}, \lambda_{hDCE}; \gamma, \tau) \end{aligned}$$

where $\tau$ is fixed as 0.07. By the proposed loss, the student model additionally leverages the relational information and shows improved performance.

As shown in Table 3, the training settings for baseline loss functions are identical to the original work [5] for a fair comparison.

**Retrained teacher and student:** Since our retrained teacher models performed slightly different from the results reported in the original work [5], we compressed the retrained teacher models for a fair comparison. Hence, we also retrained all the corresponding student models by the baseline method and compared with our proposed method.

**Projection heads for model compression task:**
For the single-modal and multi-model image translation tasks, shared projection heads are used to project the input feature and the output feature. However, in case of model compression task, the number of channels for intermediate features are different for the teacher model and the student model.

Hence, inspired by the once-for-all network [1], we implement the projection head consists of two linear layers with elastic width and the ReLU activation between the layers. Specifically, only a subset of the weight parameters are

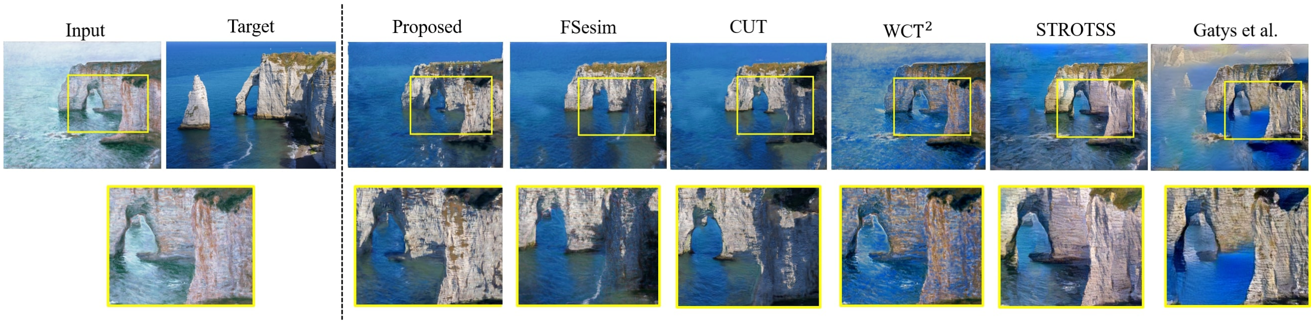


Figure 2. Qualitative comparison on high resolution single image translaton. Our method shows photorealistic output with less distortion in content information.

used when the student feature is input to the network, as shown in the Fig. 1.

## B. High-resolution single image translation

We evaluate the proposed method for the high resolution single image translation, which translates a single high-resolution painting image to a natural photograph. We followed the experimental protocols in the previous works [6, 9] for a fair comparison.

**Experiment Settings:** The input image is a Claude Monet's painting with 1200×966 size, and the target image is natural photograph with 1024×768. Following the settings in the [6, 9], randomly cropped image patches with 64×64 size are used for the training.

The implementation is based on the offcial code of SinCUT [6][3]. We replaced the patchNCE loss with our proposed hDCE loss and SRC loss. Following SinCUT, the gradient penalty term $L_{R1}$ is added to the loss function for the single-modal image translation task. $L_{R1}$ penalize the gradient of the discriminator $D$, which stabilize the training [6]. $L_{R1}$ is given as,

$$L_{R1} = \mathbb{E}_{y\sim p_Y}\left[||\nabla D(y)||^2\right]$$

where $y$ refers the image of the target domain. Hence, the total loss function is given as

$$\begin{aligned} L_{total} = & L_{GAN}(G,D,X,Y) + \lambda_{R1}L_{R1} \\ & + L_{semantic,X\to Y} + L_{semantic,Y\to Y} \end{aligned}$$

where the terms except $L_{R1}$ are same with the loss functions of single-modal image translation. The hyperparameters are defined as following: $\lambda_{R1}$=1, $\lambda_{hDCE}$=4 and $\lambda_{SRC}$ = 10.

The $\gamma$ is reciprocally increase for each training epoch $t$ as following:

$$\gamma(t) = 1/\Gamma(t)$$

with

$$\Gamma(t) = \Gamma_o + (\Gamma_{final} - \Gamma_o)\cdot\frac{t}{T}$$

where $\Gamma_o$ = 200 and $\Gamma_{final}$ = 40. $T$ is total training epoch.

We compare our method with the related works, which are FSeSim [9], CUT [6], STROTSS [4], WCT$^2$ [8] and Gatys et al. [3].

**Results:** The Fig. 2 show the qualitative comparison between the proposed method and the previous methods. The proposed method outputs more photorealistic image compared to the Gatys et al., STROTSS and WCT$^2$ . Also, compared with CUT and F-Sesim, our method shows better correspondence with the input image, showing less distortion in the shape of objects of the input image. Specifically, the shape of the cliff is preserved in our method, whereas CUT and F-Sesim show severe distortion. The results confirm that the proposed method is effective to preserve the content information, regularizing the consistency of semantic relation between the input and output.

| Number of Neg. | H→Z | Cityscapes | | | |
|---|---|---|---|---|---|
| | FID↓ | mAP↑ | pAcc↑ | cAcc↑ | FID↓ |
| 64 | 39.4 | 27.0 | 72.4 | 33.6 | 50.1 |
| 128 | 35.4 | 27.8 | 73.3 | 34.4 | 48.9 |
| 512 | 38.0 | 28.6 | 73.4 | 34.3 | 50.4 |
| 256 | **34.4** | **29.0** | **73.5** | **35.6** | **46.4** |

Table 4. Additional ablation studies on single-modal image translation on the number of negative samples.

## C. Additional ablation study

### C.1. Loss ablation

**Single-modal image translation:** We conduct the experiments to check the effect of the number of negative samples. As shown in Table 4, the results are not dramatically affected by the number of samples, but we observed a slight performance drop when different number of negative samples are used.

[3] https://github.com/taesungp/contrastive-unpaired-translation

| Settings | | | | Latent | | Reference | |
|---|---|---|---|---|---|---|---|
| InfoNCE Loss | DCE Loss | SRC Loss | Hard Neg Mining | FID↓ | LPIPS↑ | FID↓ | LPIPS↑ |
| × | × | × | × | 63.06 | 0.413 | 61.19 | 0.346 |
| ✓ | × | × | × | 59.02 | 0.437 | 61.62 | 0.300 |
| × | ✓ | × | × | 58.43 | 0.496 | 59.19 | 0.317 |
| ✓ | × | ✓ | × | 57.15 | 0.483 | 57.29 | 0.304 |
| × | ✓ | × | ✓ | 58.18 | 0.495 | 59.34 | 0.317 |
| × | ✓ | ✓ | × | 57.86 | 0.484 | 58.99 | 0.325 |
| ✓ | × | ✓ | ✓ | 55.91 | 0.471 | 55.37 | 0.347 |
| × | ✓ | × | ✓ | 58.18 | 0.495 | 59.34 | 0.317 |
| × | ✓ | ✓ | ✓ | **54.70** | **0.496** | **54.23** | **0.365** |

Table 5. Quantitative results of ablation studies on multimodal image translation tasks. All models are trained on *Seasons* dataset.

| Settings | | | H→Z (CycleGAN) | | |
|---|---|---|---|---|---|
| DCE Loss | SRC Loss | Hard Neg Mining | #Param↓ | MACs↓ | FID↓ |
| Teacher | | | 11.378M | 56.80G | 59.46 |
| Baseline Student | | | 0.412M | 2.962G | 74.39 |
| ✓ | × | × | 0.459M | 2.942G | 67.28 |
| × | ✓ | × | 0.412M | **2.862**G | 66.20 |
| ✓ | ✓ | × | 0.412M | 2.962G | 65.91 |
| ✓ | × | ✓ | 0.450M | 2.960G | 65.49 |
| ✓ | ✓ | ✓ | 0.412M | 2.962G | **64.64** |
| Seperate Projection Head | | | 0.412M | 2.962G | 78.63 |

Table 6. Quantitative results of ablation studies on GAN compression. H→Z represents Horse→Zebra.

**Multimodal image translation:** We also present an ablation study on the loss functions for the multi-modal image translation task. Due to the limited resources, the ablation study focuses on the *Seasons* dataset.

Table 5 shows the results of the ablation study on our proposed loss components. We obtain the best results in both of latent-guided and reference-guided synthesis when all of the proposed components are used. For further specification, when we only use the contrastive loss, the results are slightly improved compared to the baseline model. With an addition of the SRC loss, the model shows more improved result in FID scores. Finally, when all the components are used, we obtain the best quantitative scores in both of FID and LPIPS. Furthermore, in overall experiments, the models trained with DCE loss shows better quantitative results than the models trained with basic InfoNCE losses.

**GAN compression:** To verify our proposed components on GAN compression framework, we experimented Fast GAN compression with various settings. As a representative model and dataset, we used CycleGAN trained with the Horse→Zebra dataset for the ablation study. Table 6 is the quantitative results of our ablation study. Each loss component contribute to the improvement of the performance, showing the best FID score when all components are used. Although several models show better compression rate especially for MACs, but the difference is negligible. Considering both compression rate and image quality scores, we obtain the best results when all of the components are used together.

## C.2. Separated projection for GAN compression

The intermediate features of the teacher and the student have different channel configuration. In the Section A.3, shared projection head with elastic width is suggested as a remedy. In this section, we conduct the additional experiment using two seperate projection heads, one is for the student and the other is for the teacher. Then, the embedding vectors $z_k^l$ and $w_k^l$ from the teacher and the student is formalized as,

$$
\begin{aligned}
\{z_k^l\}_{k=1}^K \sim z^l &= F_t^l(G_t^l(x)) \\
\{w_k^l\}_{k=1}^K \sim w^l &= F_s^l(G_s^l(x))
\end{aligned}
$$

where the $F_t^l$ and $F_s^l$ are the projection heads for the $l$-th layer features from the teacher model and the student model. The channel width of the projection heads are fixed, as shown in the Fig. 3.

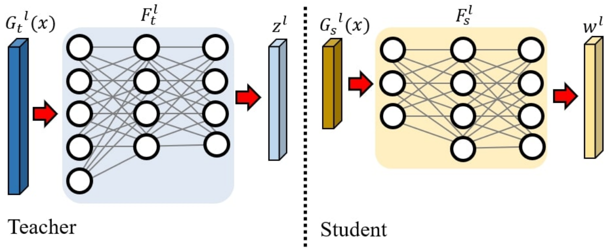
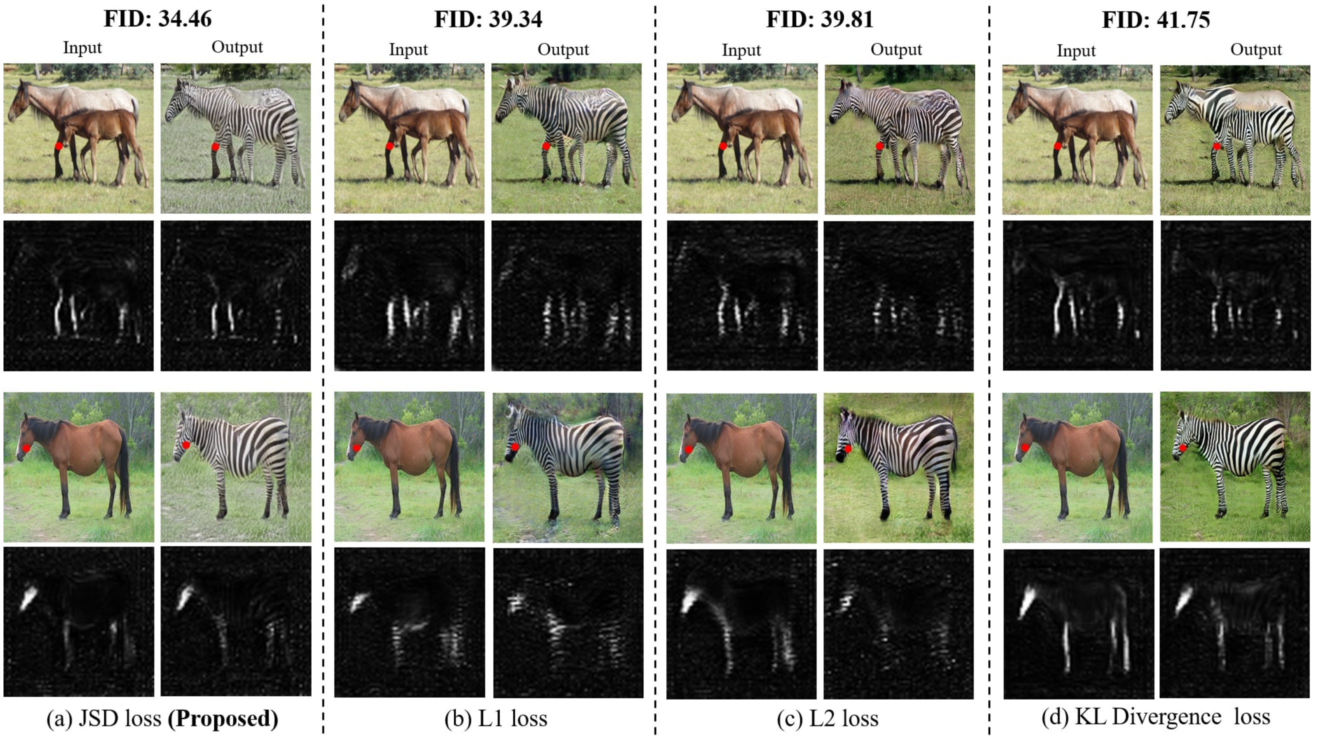


Figure 3. Projection head $F_t^l$ and $F_s^l$ with fixed width for the teacher and the student. The parameters of the projection heads are not shared between the teacher and the student. (i.e. $F_t^l$ and $F_s^l$ are updated independently.)

As shown in the Table 6, the separate projection heads do not improve the performance compared to the baseline. Since different projection heads are used for the teacher and the student, the features from each model are projected into different embedding spaces. Hence, the consistency of the relational knowledge or the contrastive loss do not give any benefits to the performance.

## C.3. SRC loss

Consistency regularization can be imposed by various form of functions. In the main script, we used the Jensen-Shannon divergence (JSD) loss for the consistency regularization. In this section, we conduct the ablation study on the loss, replacing the JSD function with other functions such as L1 loss, L2 loss and Kullback-Leibler (KL) divergence function. The ablation study is proceeded on the single-modal image translation task using the



(a) JSD loss **(Proposed)**

(b) L1 loss

(c) L2 loss

(d) KL Divergence loss

Figure 4. Ablation study for the function of SRC loss. Query points are marked with the red dots in the images. The query points are at the front leg in the first image, and the head in the second image. JSD loss outperforms other functions with an improved FID score, along with the enhanced consistency of the similarity relation.

Horse→Zebra dataset. For a fair comparison, the training settings are identical.

Now, we introduce the loss functions for the ablation study. We denote the $\{z_k\}_{k=1}^K$ and $\{w_k\}_{k=1}^K$ as a set of the $K$ embedded vectors for the input and translation output. Then, the SRC loss by L1 loss function is as follows,

$$L_{SRC,L1} = \sum_{k=1}^{K}\sum_{j\neq k} ||z_k^\top z_j - w_k^\top w_j||_1$$

Similarly, SRC loss with L2 loss function is given as,

$$L_{SRC,L2} = \sum_{k=1}^{K}\sum_{j\neq k} ||z_k^\top z_j - w_k^\top w_j||_2^2$$

Lastly, SRC loss with KL divergence function is as,

$$L_{SRC,KL} = \sum_{k=1}^{K} KL(P_k||Q_k)$$

where $P_k$ and $Q_k$ are the distribution of similarity relation for the input embedding vector $\{z_k\}_{k=1}^K$ and the output vector $\{w_k\}_{k=1}^K$, as introduced in the main script.

Fig. 4 shows the similarity relation between the query point and the other locations. We observe that the losses contribute to the consistency of the similarity relation, with showing the similar appearance between the similarity maps of the input and the output.

| Settings | H→Z | Cityscapes | | | |
|---|---|---|---|---|---|
| | FID↓ | mAP↑ | pAcc↑ | cAcc↑ | FID↓ |
| $\gamma = 0.02$ | 38.5 | 28.0 | 72.5 | 34.1 | 48.9 |
| $\gamma = 0.1$ | 37.1 | 27.2 | 71.7 | 33.3 | 51.49 |
| $\gamma = 1$ | 37.5 | 18.0 | 57.0 | 24.9 | 102.3 |
| Curriculum | **34.4** | **29.0** | **73.5** | **35.6** | **46.4** |

Table 7. Ablation studies on parameter $\gamma$.

The JSD loss, in particular, improves the consistency the most compared to the other functions. The difference between the maps is much less for the JSD loss, compared to the other loss functions which obviously shows some discrepancy between the input similarity map and the output similarity map. The FID scores also support the superiority of JSD function against other losses.

## C.4. Hyperparameter $\gamma$

Our method explicitly controls the hardness of the negative mining using $\gamma$, compared to the NEGCUT [7] which implicitly controls the hardness using the negative generator. In this section, we investigate the effect of $\gamma$ to the performance, and claim the risk of implicit control of the hardness for the negative mining.

As shown in the Table 7, the hardness largely affect the performance. The performance is best when the curriculum learning is applied. Also, it is notable that the failure of the training is observed when the negative mining is too hard (i.e. large $\gamma$). Hence, the implicit control of the hardness may induce the degradation of the performance by imposing inappropriate hardness for the negative mining.

## D. Additional Results

We provide additional results for the three tasks introduced in the main script: single-modal, multi-modal image translation tasks and GAN model compression.

For single-modal image translation, we show the additional results for the Horse→Zebra dataset in Fig. 5, and for the Cityscapes dataset in the Fig. 6. For multi-modal image translation, we show the results in Fig. 7, 8 for *Seasons* dataset, and Fig.9,10 for *Weather* dataset. The input images are translated into multiple domain classes, considering the style of reference images. Fig. 11 shows the additional results for the GAN compression tasks on the Horse→Zebra, Map→Satellite and Cityscapse datasets.

## E. Additional Datasets

We further provide results for the single-modal image translation tasks on the additional datasets with true label: GTA→Cityscapes, Map→Satellite. For Map→Satellite, the hyperparameters are identical with the case of Horse→Zebra dataset. For GTA→Cityscapes, the hyperparameters are identical to the Cityscapes dataset case, and we set the total training epoch as 10, and followed the experimental settings in [6].

Our method outputs better quality of images along with the improved metric scores. For Map→Satellite dataset, the FID score is improved to 45.15, compared to the baseline [6] with 53.12. For GTA→Cityscapes dataset, our method improve the pixel accuracy score as 58.9, compared to the baseline [6] with 54.9. We evaluated the methods using the pretrained DRN model for the segmentation of Cityscapes dataset. The quality of the output images supports the quantitative evaluation, as shown in Fig.12, 13.

## F. Limitation and Potential negative impact

Even though the hard negative mining in the proposed method showed its effectiveness, an excessive strength of the hardness in the early training stage may cause the instability of the training procedure. Hence the strength of the hardness should be carefully selected. Likewise, the curriculum learning which controls the strength of the hardness should be carefully planned, considering the characteristic of the dataset and the network architecture. Finally, the approximation for the $q_{NPC}$ in the Section 3.3 only works when the algorithm correctly converges, so that the approximation should be used carefully only to reveal the important of the patch-wise semantic relationship rather than deriving a new algorithm.

Regarding on the social impact, as most of image generation methods shares, the generator in the proposed work may produce a social disinformation by creating realistic fake images. Also, the generative model has potential risk of adversarial attack.

## References

[1] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. 3

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[4] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[5] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. Gan compression: Efficient architectures for interactive conditional gans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3

[6] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 319–345, Cham, 2020. Springer International Publishing. 1, 4, 7

[7] Weilun Wang, Wengang Zhou, Jianmin Bao, Dong Chen, and Houqiang Li. Instance-wise hard negative example generation for contrastive learning in unpaired image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14020–14029, October 2021. 6

[8] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 4

[9] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16407–16417, June 2021. 4

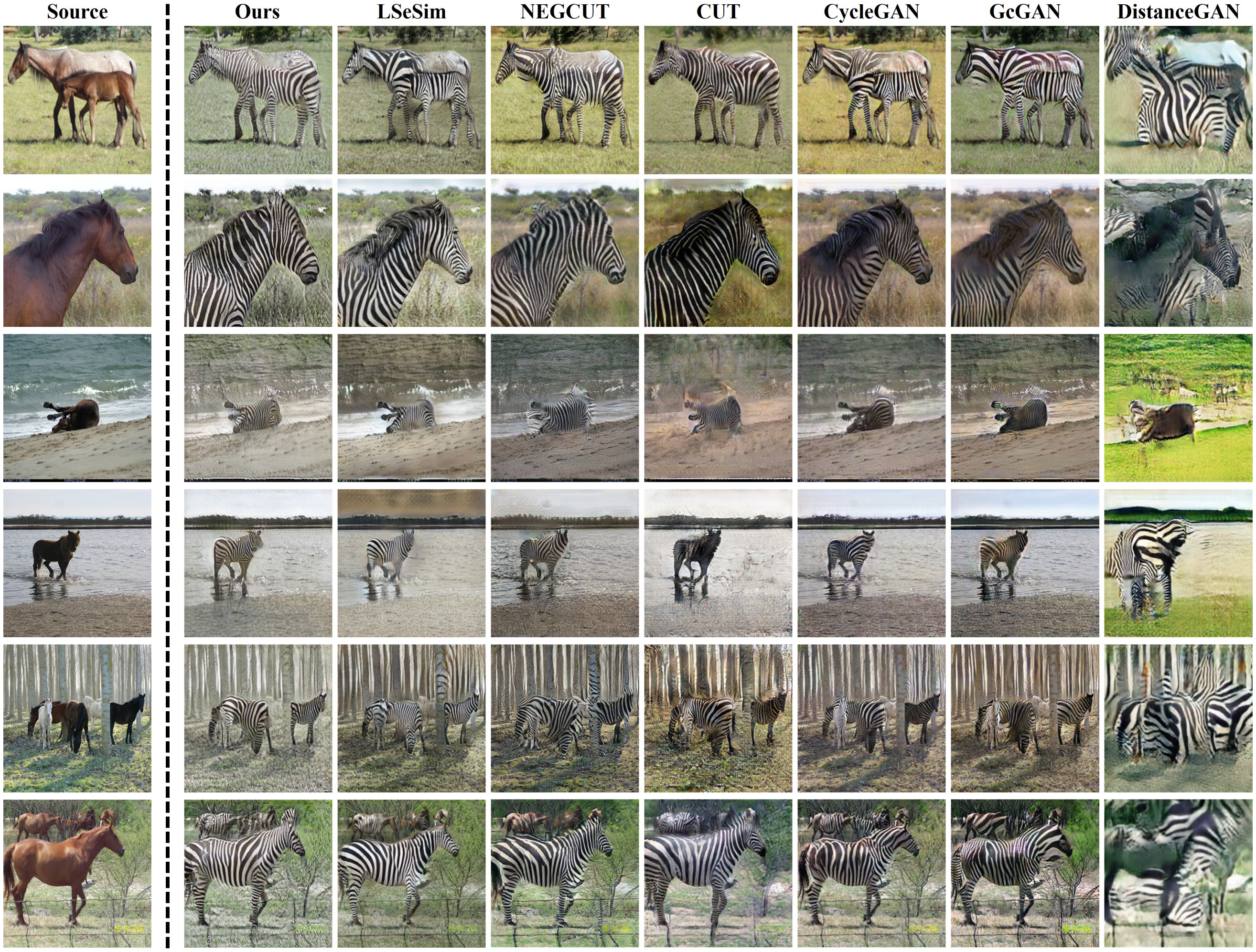


Figure 5. Additional results for single-modal image translation model trained on Horse→Zebra dataset.

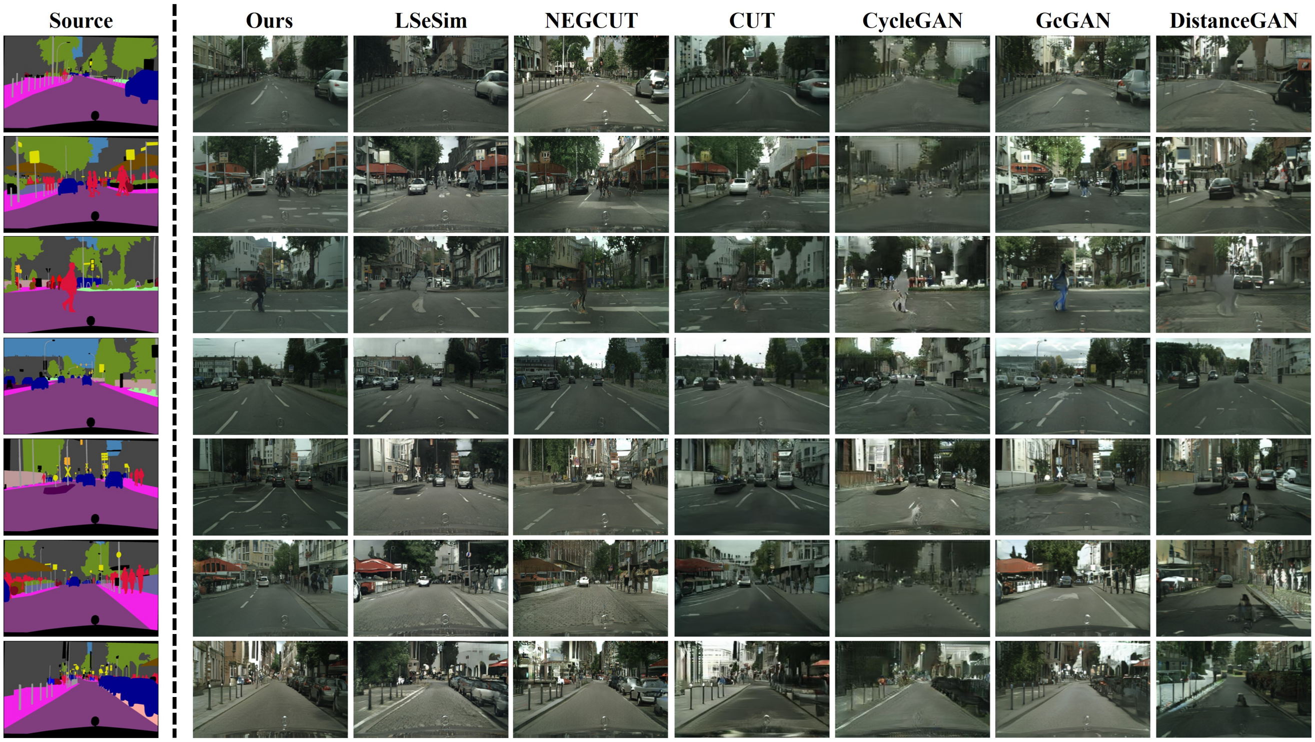


Figure 6. Additional results for single-modal image translation model trained on cityscapes dataset.

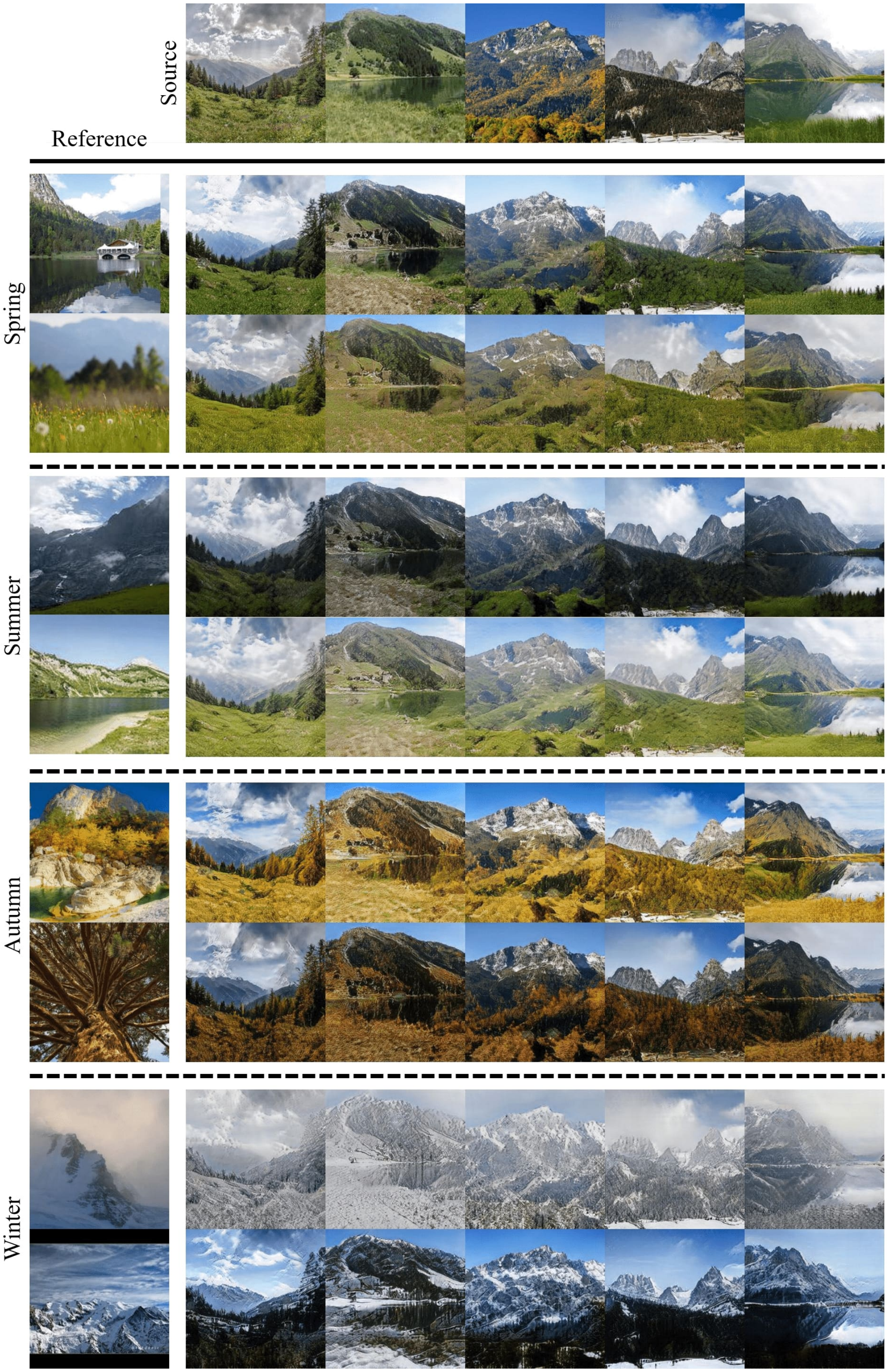


Figure 7. Additional results for multi-modal image translation model trained on *Seasons* dataset. Our method can translate the input images into multiple domain outputs with reflecting the reference style images.

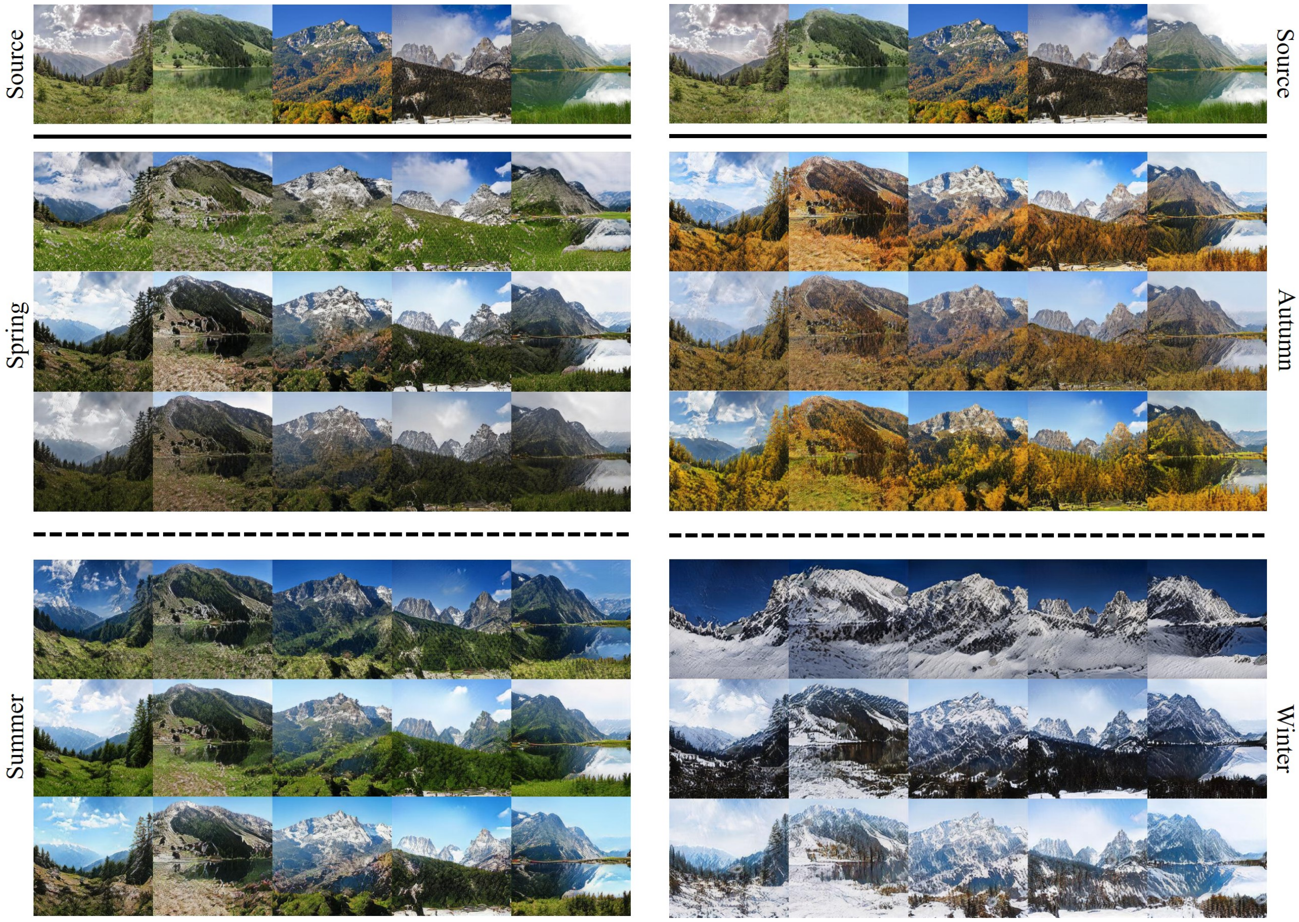


Figure 8. Additional results for multi-modal image translation model trained on *Seasons* dataset. Our method can translate the input images into multiple domain outputs with random styles.

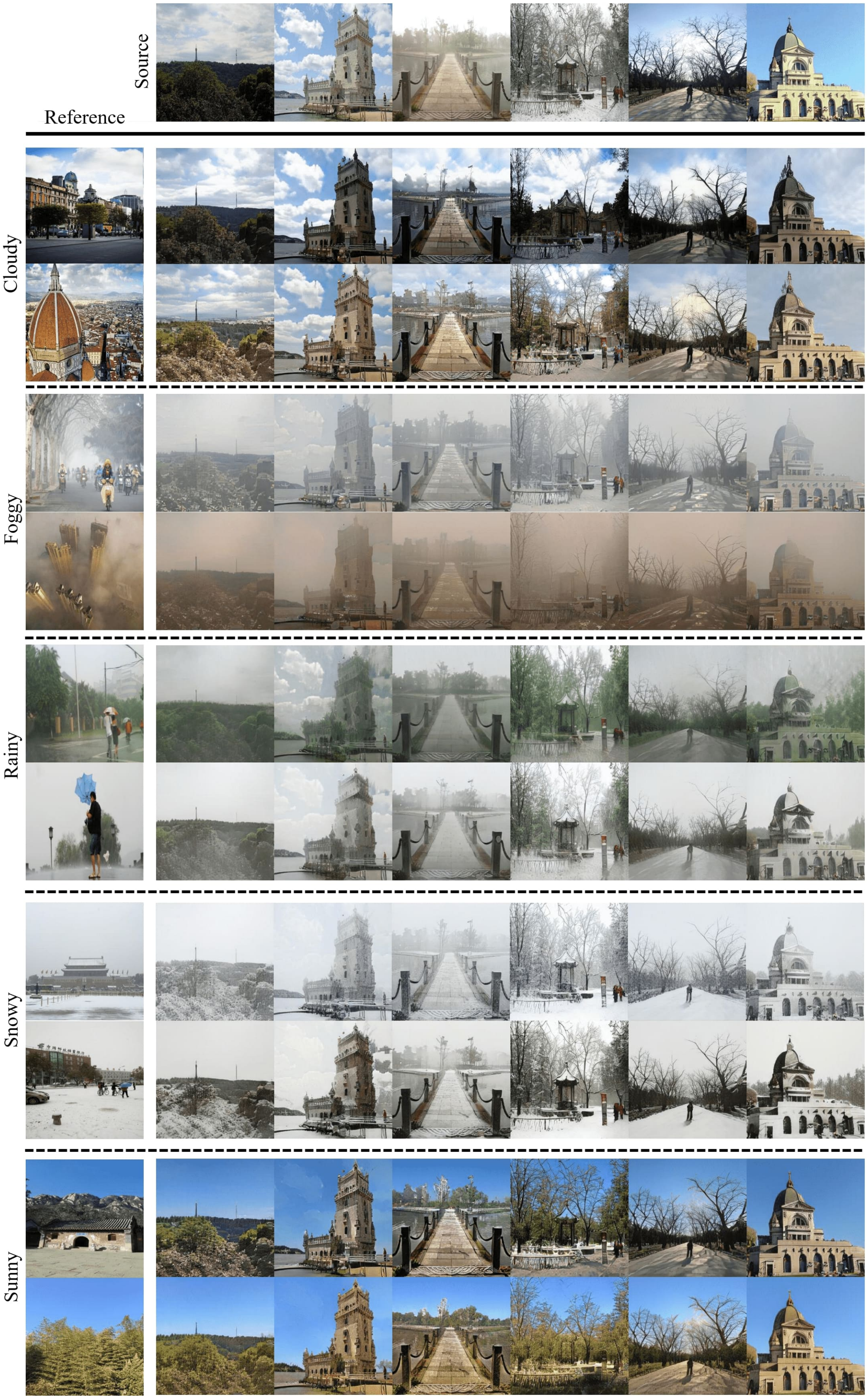


Figure 9. Additional results for multimodal image translation model trained on *Weather* dataset. Our method can translate the input images into multiple domain outputs with reflecting the reference style images.

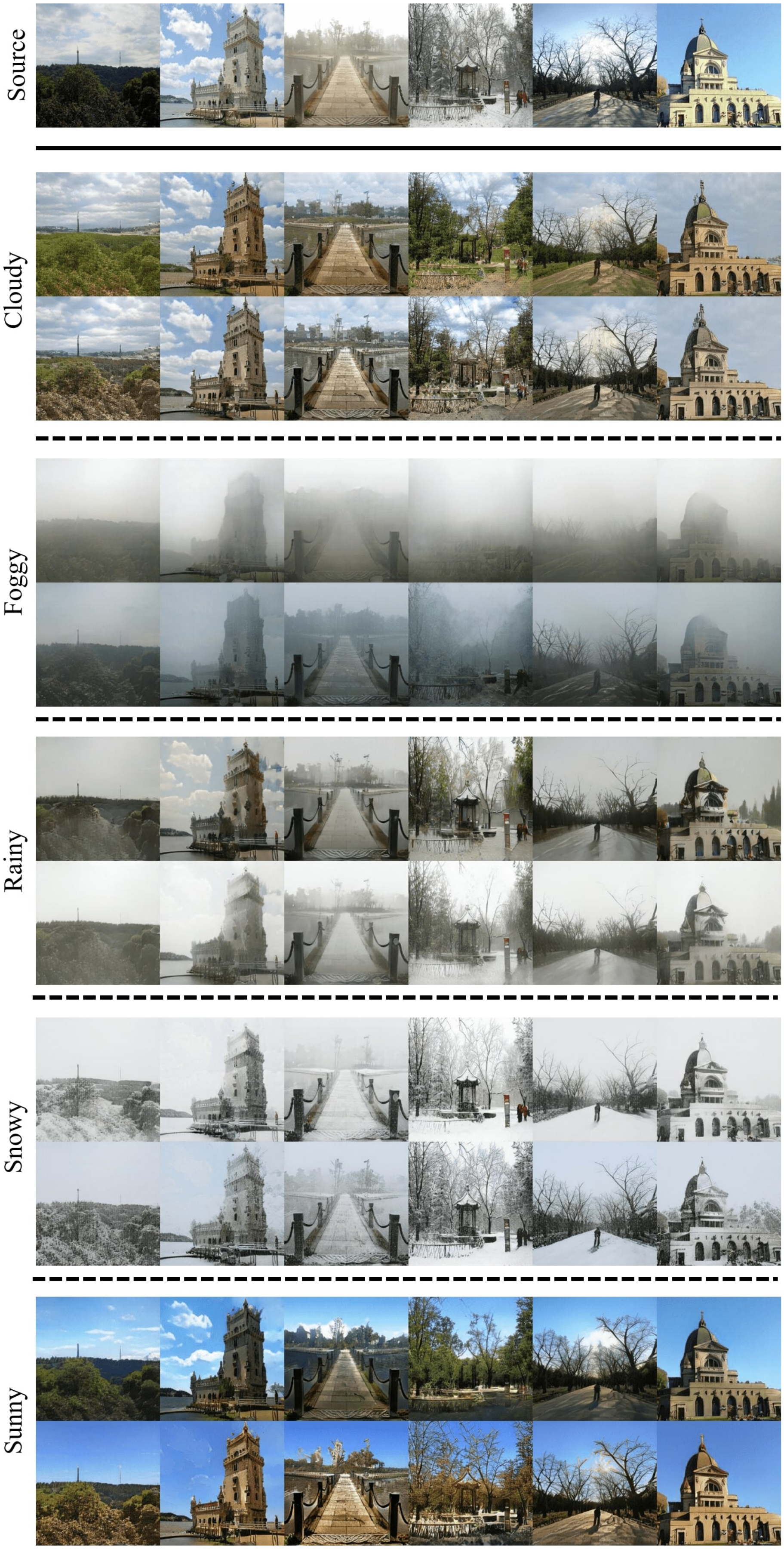


Figure 10. Additional results for multimodal image translation model trained on *Weather* dataset. Our method can translate the input images into diverse multiple domain outputs with random styles conditioned on each season.

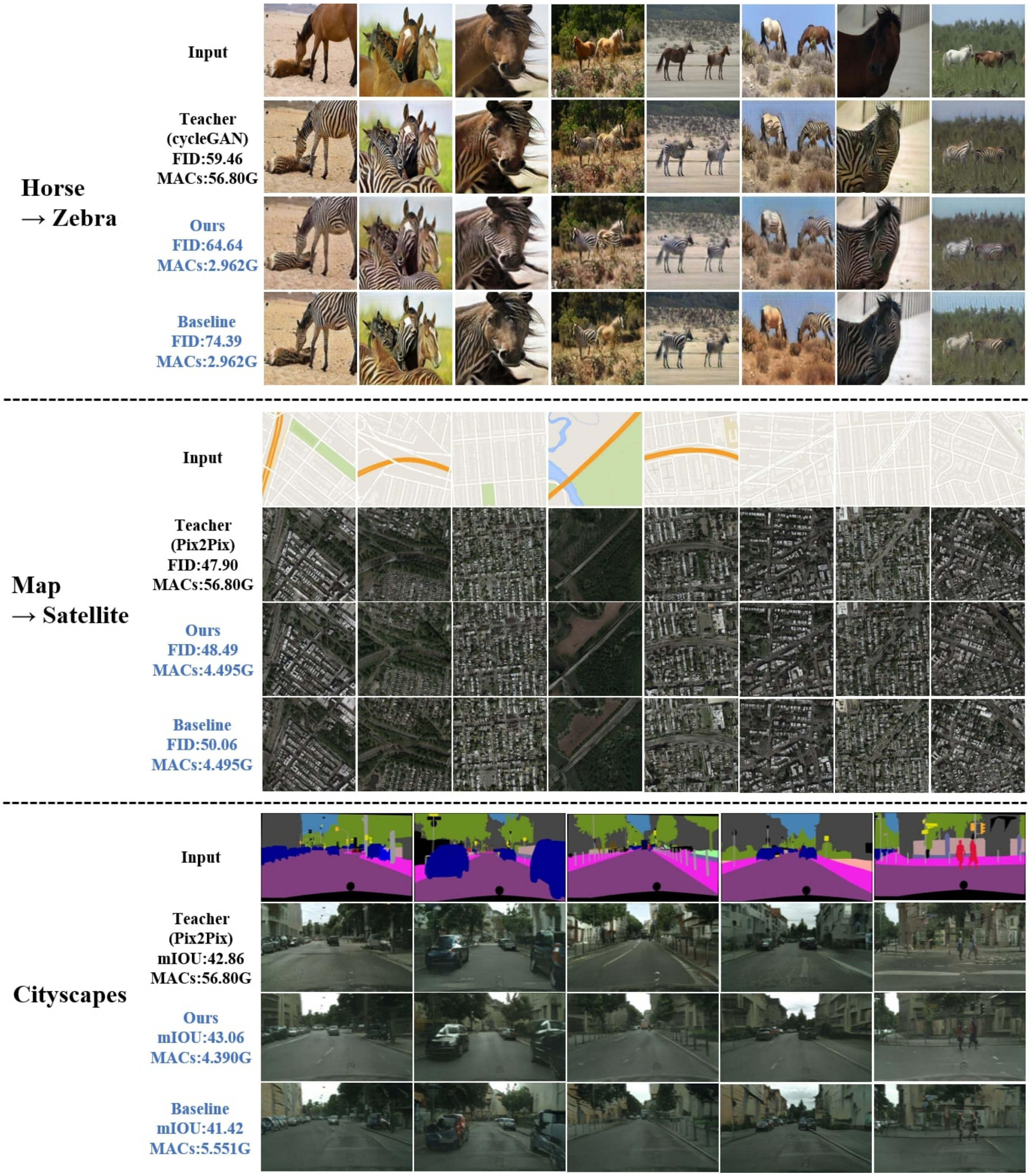


Figure 11. Additional results for GAN model compression for image translation tasks. Our method outputs better quality of images along with the improved metric scores (FID, mIOU) and the model size.

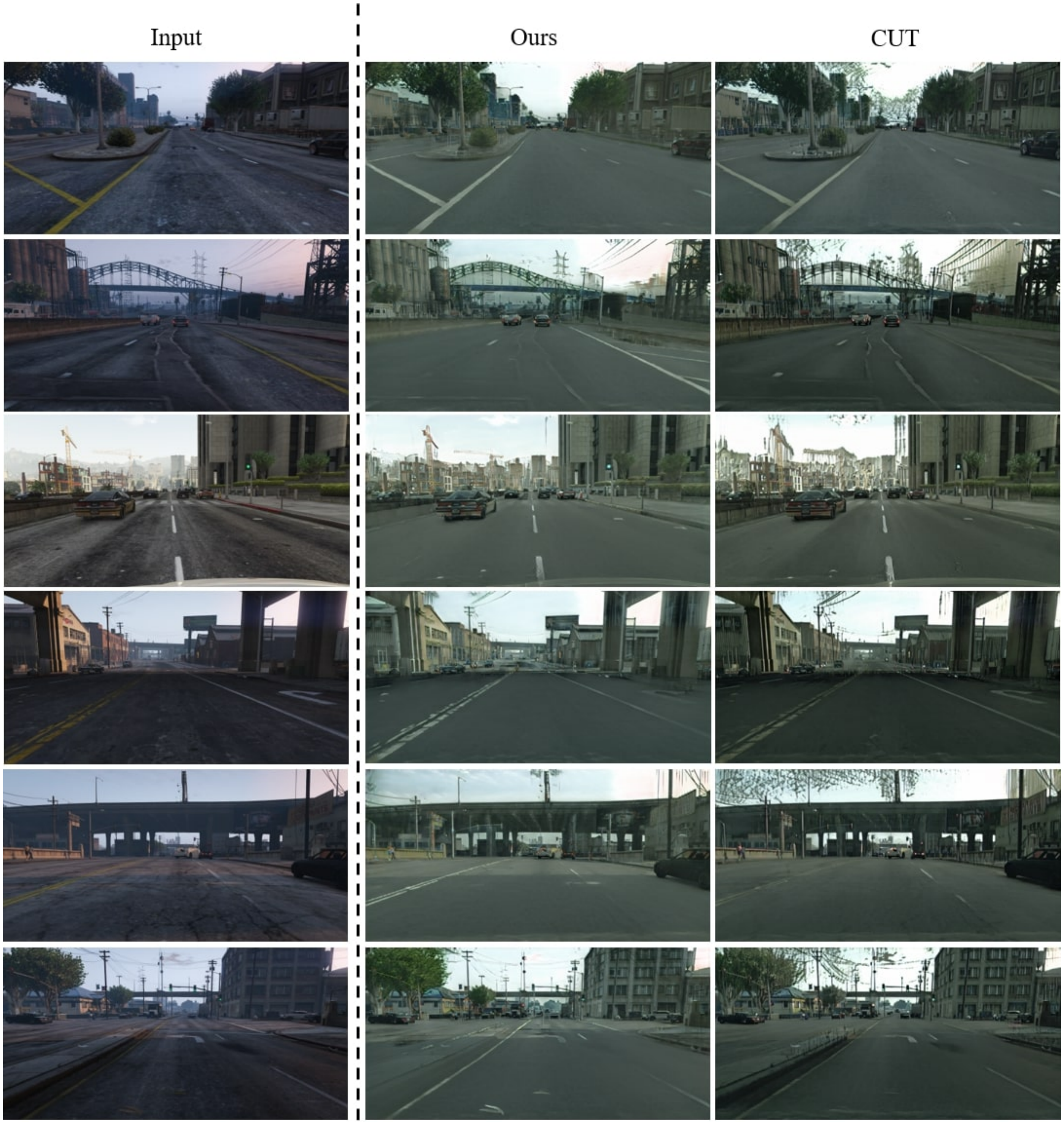


Figure 12. Additional results for single-modal image translation model trained on GTA → Cityscapes dataset.

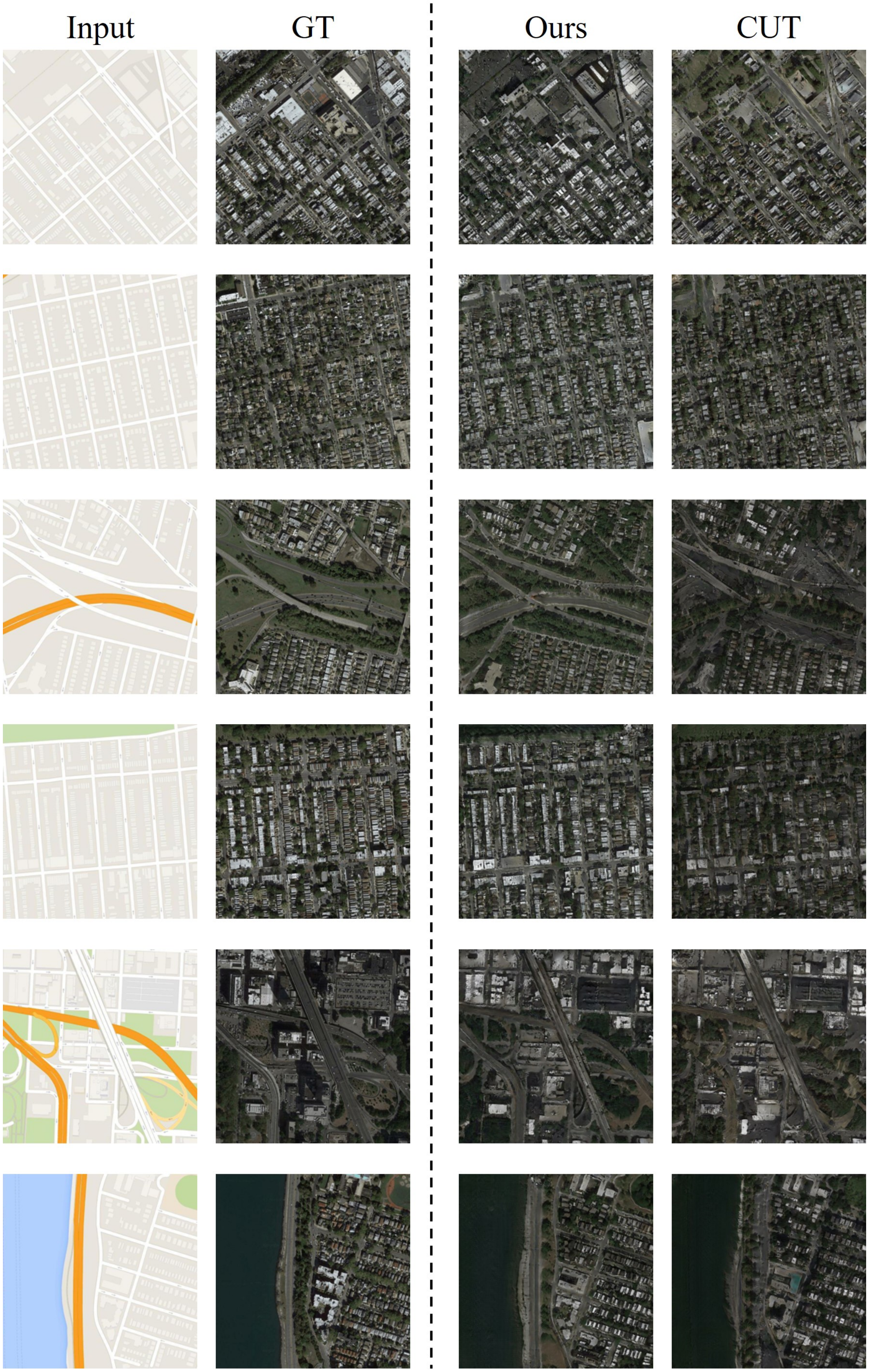


Figure 13. Additional results for single-modal image translation model trained on Map → Satellite dataset.