Domain Gap Embeddings for Generative Dataset Augmentation

Supplementary Material

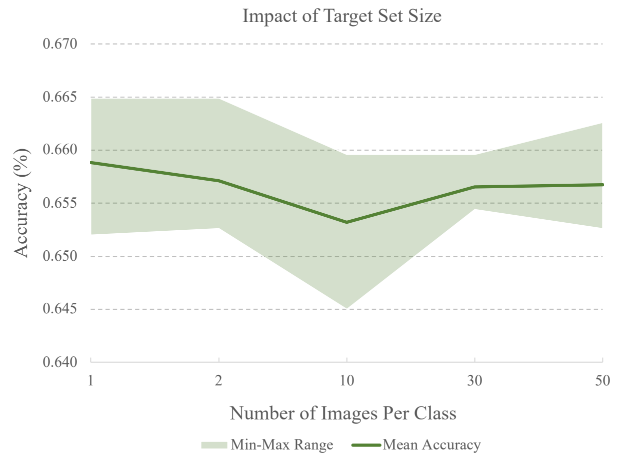

Figure 9. **Line plot of the impact of different target data sizes.** We evaluated the performance on our 20-class DomainNet (Real → Painting) classification task. The results illustrate that using as little as one image per class from the target distribution is sufficient for DoGE to generate effective training data.

A. Domain Gap Extraction

In this section, we explore more details in the domain gap extraction process, including different algorithms to capture the domain gap representation and the impact of sample sizes available for the extraction step.

A.1. Extraction Methods

In Sec. 3.1, we described two options for our domain gap extraction algorithms: the difference of means and the PCA-based method. For analysis, we qualitatively compare DomainNet (Real → Painting) generations using the two methods. We visualize the impact of different methods in Fig. 10. The results show that the difference of means yields better adaptation effectiveness, *i.e.* more aligned to target domains, than the PCA-based method. We adopt the difference of means as our domain gap representation for the following experiments.

A.2. Impact of Target Set Size

Besides the extraction algorithm, in our few-shot setting, the impact of different numbers of target samples available is also important to study. We evaluated the performances of our synthetic data generated with domain gap embeddings from different numbers of target samples. For analysis, we considered the first 20 classes in DomainNet (Real → Painting) and evaluated the performance on the 20-class classification task. We randomly sampled the same number of images per class from the source and target distributions, from 1, 2, 10, 30 to 50. Fig. 9 shows using as little as one image per class (20 images) from the target distribution is as effective as using 50 images per class (1000 images).

Algorithm 1 Confidence-Based Data Cleaning

```
Input: $G(x)$ - Our DoGE data augmenter
       $f(x)$ - The downstream task model to improve
       $(X, Y)$ - The source training set data and labels
       $t$ - Threshold for confidence-based filtering
for batch $b$ with label $y_b$ in $(X, Y)$ do
    $\hat{b} \leftarrow G(b)$   ▷ Augment source data to target domain
    for synthetic sample $\hat{x}$ with label $y$ in $\hat{b}$ do
        $\hat{p} \leftarrow \text{argmax}\, f(\hat{x})$   ▷ Model prediction
        $c \leftarrow \max f(\hat{x})$   ▷ Model confidence
        if $\hat{p} \neq y$ and $c \geq t$ then
            discard $\hat{x}$ from $\hat{b}$
    $f \leftarrow \text{AdamW}(\mathcal{L}(f(\hat{b})))$   ▷ Update model
Output: $f(x)$ - The adapted and improved model
```

B. Data Cleaning Algorithms

To further explain the training-time confidence-based data cleaning process in Sec. 3.3, we include Algorithm 1. Given a model $f(x)$ trained on original training data, we adapt it by fine-tuning on our synthetic dataset. During fine-tuning, for each batch of data in the original training set, we precomputed the image augmentations with DoGE and denote the corresponding augmented batch as $\hat{b}$. For each generation $\hat{x} \in \hat{b}$ with the original label y, we use the current model to predict the label $\hat{p} = f(\hat{x})$ and compute the confidence as the maximum softmax score among all classes $c = \max f(\hat{x})$. Then, we ignore $\hat{x}$ from this training batch if the prediction is wrong *i.e.* $\hat{p} \neq y$ with high confidence c over a certain threshold t. After we filter the entire batch as above, then we fine-tune the model on the cleaned batch.

C. Real-Synthetic Mixing Ratio

While the above data cleaning process filters out poor-quality samples and improves the usefulness of synthetic data, the effectiveness of our data is also dependent on how we leverage them to fine-tune downstream task models. One important decision is, when fine-tuning task models, how to take the most advantage of synthetic generations and the high-quality original training data. Hence we study the impact of various data mixing ratios during task model

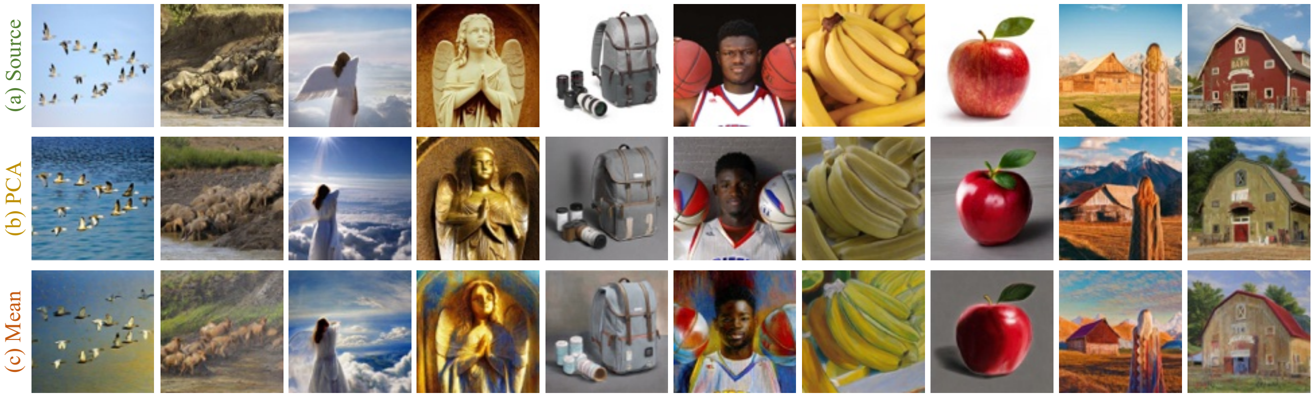

Figure 10. **Synthetic data from different domain gap extraction algorithms.** (a) Real source images from DomainNet were converted into Painting using (b) PCA and (c) the difference of means, shown accordingly. These examples illustrate that (c) is more effective than (b) in augmenting source (Real domain) images into the target distribution (Painting domain).

	UDA methods	Test Acc (%)			
		—	+ DA-Fusion [71]	+ DATUM [3]	+ DoGE
Real → **P**ainting	BSP [11]	46.76	46.78	41.89	**47.34**
	DANN [19]	47.01	48.83	42.73	**49.68**
	CDAN [45]	51.66	51.91	49.87	**52.11**
	MCD [65]	50.88	50.99	49.71	**52.14**
	MCC [35]	50.08	50.42	49.31	**52.95**
	MemSAC [36]	52.27	53.26	50.32	**54.16**
Real → **C**lipart	BSP [11]	46.78	45.11	39.43	**46.79**
	DANN [19]	**49.80**	47.82	42.70	48.11
	CDAN [45]	53.93	54.11	50.54	**54.53**
	MCD [65]	51.42	50.79	50.01	**54.02**
	MCC [35]	50.61	49.27	48.10	**51.99**
	MemSAC [36]	54.34	54.59	51.10	**55.35**
Real → **S**ketch	BSP [11]	36.47	36.81	28.38	**38.49**
	DANN [19]	38.72	38.45	36.13	**40.21**
	CDAN [45]	42.60	42.23	39.65	**43.00**
	MCD [65]	39.25	38.07	39.19	**42.78**
	MCC [35]	34.38	33.31	33.06	**37.23**
	MemSAC [36]	41.74	40.42	36.54	**43.23**

Table 8. **Test Accuracy of UDA methods on the DomainNet problem.** We evaluated existing UDA methods with and without synthetically supplemented training data; +DA-Fusion, +DATUM and +DoGE denote the methods used for the generation. This table shows that DoGE, while being compatible with and complementary to UDA methods, is also more effective than the competing methods.

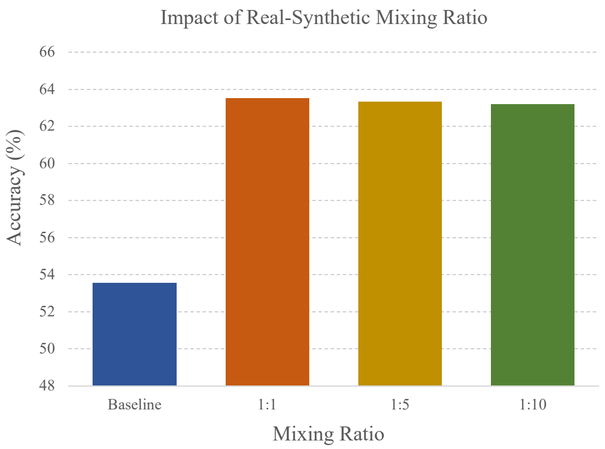

Figure 11. **Bar plot of the impact of different data mixing ratios.** We evaluated the performance on our 20-class DomainNet (Real → Painting) classification task. The results show that augmenting as little as 10% of the training data is sufficient to improve downstream task model performances.

fine-tuning. For analysis, we used the first 20 classes in DomainNet (Real → Painting) and evaluated the performance on the 20-class classification task. We changed the ratio of synthetic to real images in the training dataset from 1:1, 1:5 to 1:10. Fig. 11 shows that expanding the dataset by as little as 10% can be as effective as adding 100% more data.

D. Complete UDA-Based Comparison

This section extends the brief experiment (Tab. 5) in Sec. 4.3.1, which shows that DoGE is compatible and complementary to existing UDA methods. We show the full UDA-based evaluations in Tab. 8, where we also compare against other baselines DA-Fusion and DATUM on two more tasks. DoGE successfully improved and surpassed other baseline UDA evaluations in 17 out of 18 experiments.

E. More Method Ablation

To further isolate the effectiveness of and improvement from our domain gap embeddings, this section shows more ablation studies around CLIP and the generation pipeline.

	Acc↑
zero-shot CLIP	53.53
finetuned CLIP	72.77

	Acc↑	FID↓
Ours (noise)	38.97	39.36
Ours (DoGE)	**44.00**	**18.25**

Table 9. Evaluation on DomainNet (Real→Painting). (Left) We evaluated the zero-shot CLIP classifier and finetuned it on our synthetic dataset to show the effectiveness of our synthetic data. (Right) We compared embedding augmentation between noises and DoGE to demonstrate the performance gain.

E.1. Improvement on Top of CLIP

One foundation of the success of DoGE is the vast generalization capability and knowledge base in the CLIP latent space. However, in this section, we show that off-the-shelf CLIP is not sufficient against domain shifts. We focused on the DomainNet Real→Painting experiment setup in Sec. 4.3.1 and evaluated the zero-shot CLIP classifier against the CLIP classifier finetuned with our synthetic dataset in Tab. 9 left. We can see that our synthetic dataset can effectively improve the zero-shot CLIP classifier further in domain shifts.

E.2. Domain Gap Embeddings Isolation

This section demonstrates the effectiveness of DoGE by isolating the Domain Gap Embeddings from the rest of the generation pipeline. Specifically, in the same DomainNet Real→Painting experiment setup in Sec. 4.3.1, we generated and evaluated two sets of synthetic datasets. One dataset generation used the default DoGE pipeline and the other replaced the injected domain gap embeddings with small random noises while keeping the rest of generation pipeline the same. Then we evaluated these datasets in terms of FID and classification accuracies by finetuning. As shown in Tab. 9 right, using our domain gap embedding improves both FID and finetuning performance, demonstrating the effectiveness of DoGE.

F. Comparison to Style Transfer

Given the settings and method of DoGE, it may appear as a style transfer method. However, our goal, which is generative semantic data augmentation, is more than just style transfer. Similar to previous literature cited in the related works in Sec. 2, our method is designed for any kind of semantic data augmentation within CLIP's representation capacity rather than style transfer only. The first experiment in Sec. 4.2 shows our effectiveness in improving the subpopulation shift problem with object changes. Such augmentations (e.g., adding/removing eyeglasses in Fig. 3) are not regular style transfer tasks. Moreover, we can solve style transfer problems in a training-free and diverse fashion.

Nonetheless, existing style transfer methods [20, 48] are effective in many of our experiments and are important baselines to evaluate against. In Tab. 10, we evaluate against other style transfer methods on our GTA→CityScapes experiment in Sec. 4.3.2. The table shows that our generalized method is as performant as style transfer methods.

	[48]	[20]	Ours
mIoU	44.5	55.37	**57.30**

Table 10. Evaluation on GTA→CityScapes adaptation task.

G. More Visualizations

In this section, we present more samples of our generation that were briefly shown in Sec. 4. The generation setup is the same as mentioned in Sec. 4.1. For each task, we choose the difference of means as our domain gap representation. Except for synthetic CelebA data generation, we enable our ControlNet integration in every other task.

G.1. Imbalanced CelebA Classification

Fig. 12 presents more generated samples for our CelebA experiment in Sec. 4.2. Recall that in this subpopulation shift scenario, the source and target distribution differ by the semantic change of adding/removing eyeglasses in perceived female/male classes, as shown in the top-left corner of Fig. 12. The rest of the figure shows more synthetic data in both classes augmented by our pipeline, *i.e.* males without eyeglasses and females with eyeglasses.

G.2. DomainNet Domain Adaptation

This section presents more visual examples of our DomainNet synthetic data used in Sec. 4.3.1. Figs. 13 to 15 display more of our generation from Real domain to Painting, Sketch, and Infograph domains. These data were used to improve classification model performance in our evaluations. Along with the reference target image on the leftmost column, these figures demonstrate the quality and usefulness of our synthetic data.

G.3. FMoW Domain Adaptation

To extend the examples of synthetic FMoW data in Fig. 6, more samples are shown in Fig. 16. As described in Sec. 4.3.1, we generate recent satellite images from an older period. Fig. 16 lists 10 categories of land use and our generated data in each category. The figure illustrates our capability to generate high-quality satellite images.

G.4. GTA → CityScapes Segmentation

This section shows more generated data used in Sec. 4.3.2. The original training set is the GTA5 dataset and the target domain contains realistic driving scenes in CityScapes. Fig. 17 shows more synthetic examples from DoGE. Since the segmentation map is available, we also show the control maps leveraged during the generation with ControlNet enabled. As the figure shows, the generated image is able to maintain the same image structure honoring the edge and segmentation mask constraints.

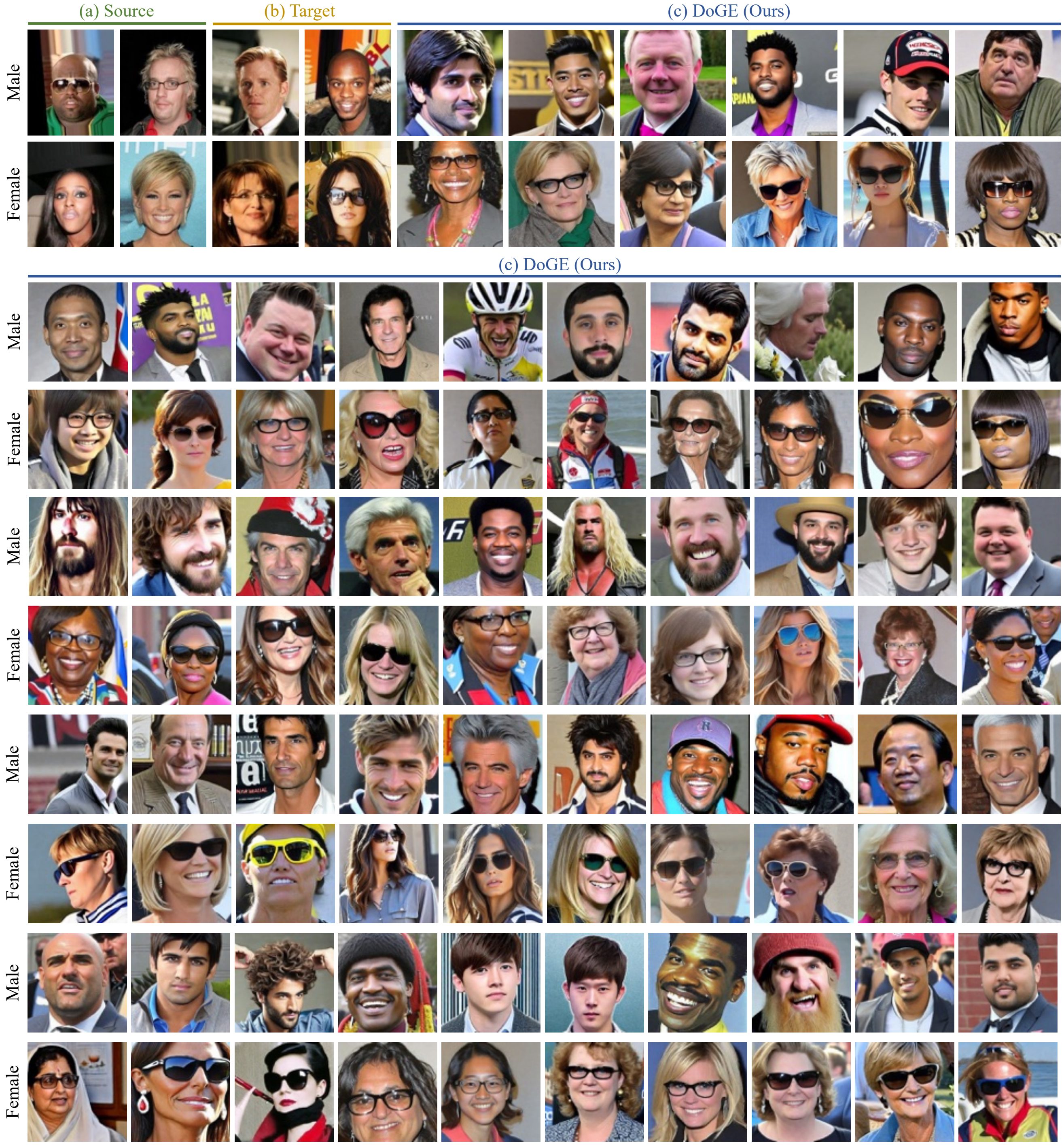

Figure 12. **More synthetic examples from the CelebA subpopulation shift experiment.** On the top-left, we show the (a) source and (b) target distribution as defined in our experiment setup. The rest of images (c) are synthetic data generated from DoGE. These examples illustrate our capability of capturing and applying semantic distribution gaps.

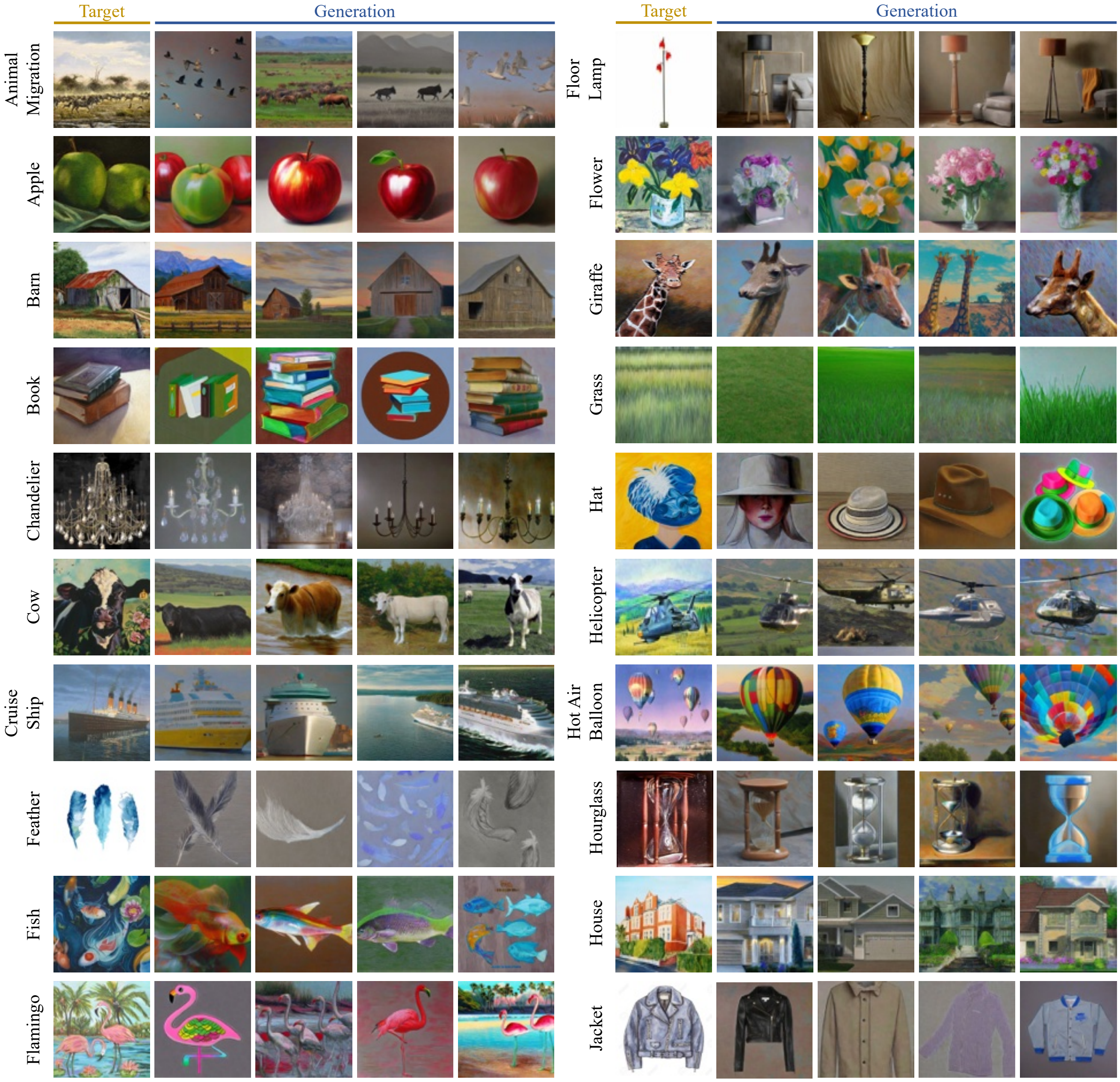

Figure 13. **More synthetic examples from the DomainNet Real → Painting generation.** We list more synthetic data generated in our Real→Painting UDA experiment in Sec. 4.3.1. We randomly select and show 20 classes from DomainNet. For each class, we present one image from the DomainNet Painting domain as a reference and four of our generations. These examples demonstrate our generation quality and capability to effectively augment real images into the Painting domain.

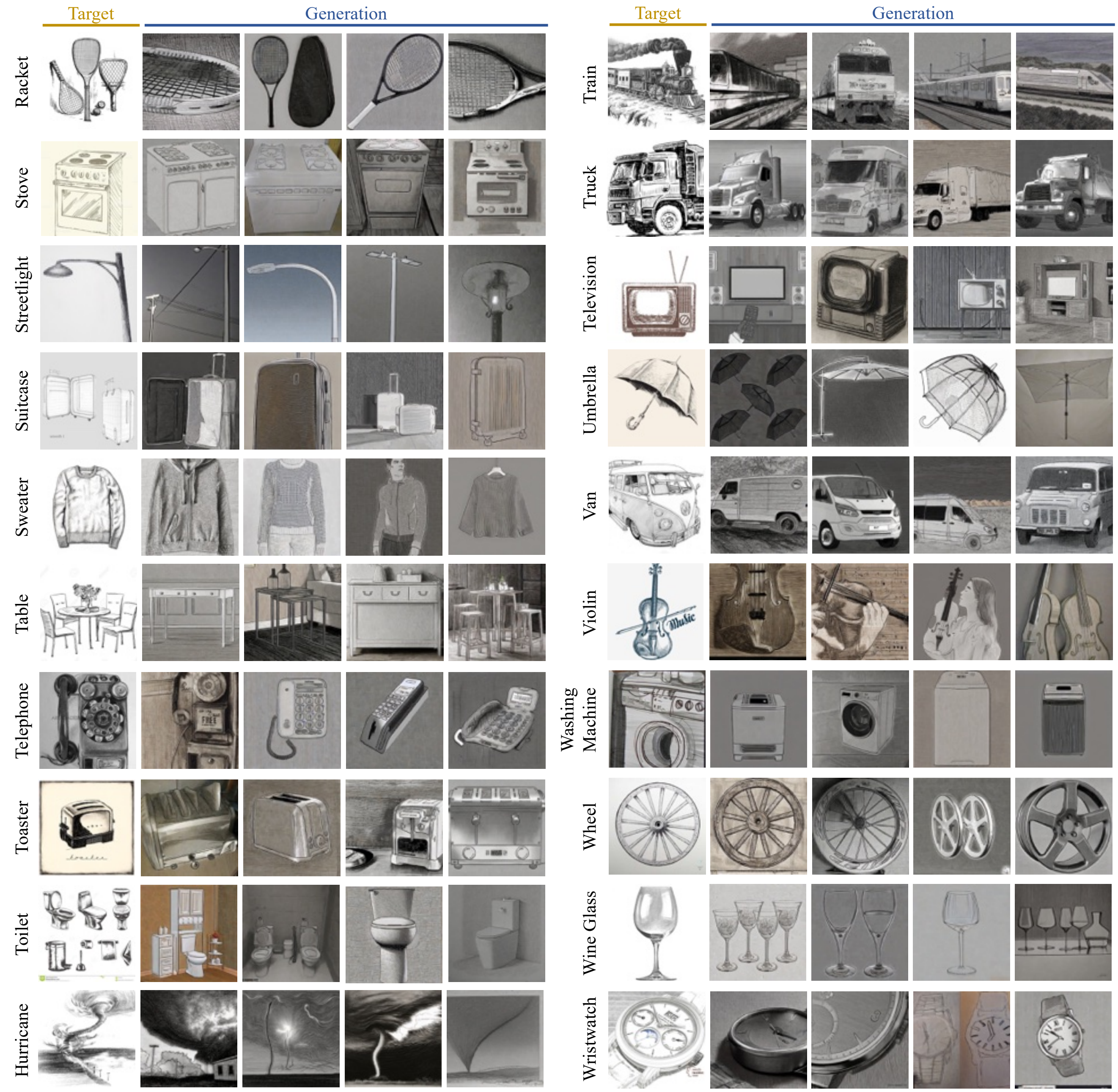

Figure 14. **More synthetic examples from the DomainNet Real → Sketch generation.** We list more synthetic data generated in our Real→Sketch UDA experiment in Sec. 4.3.1. We randomly select and show 20 classes from DomainNet. For each class, we present one image from the DomainNet Sketch domain as a reference and four of our generations. These examples demonstrate our generation quality and capability to effectively augment real images into the Sketch domain.

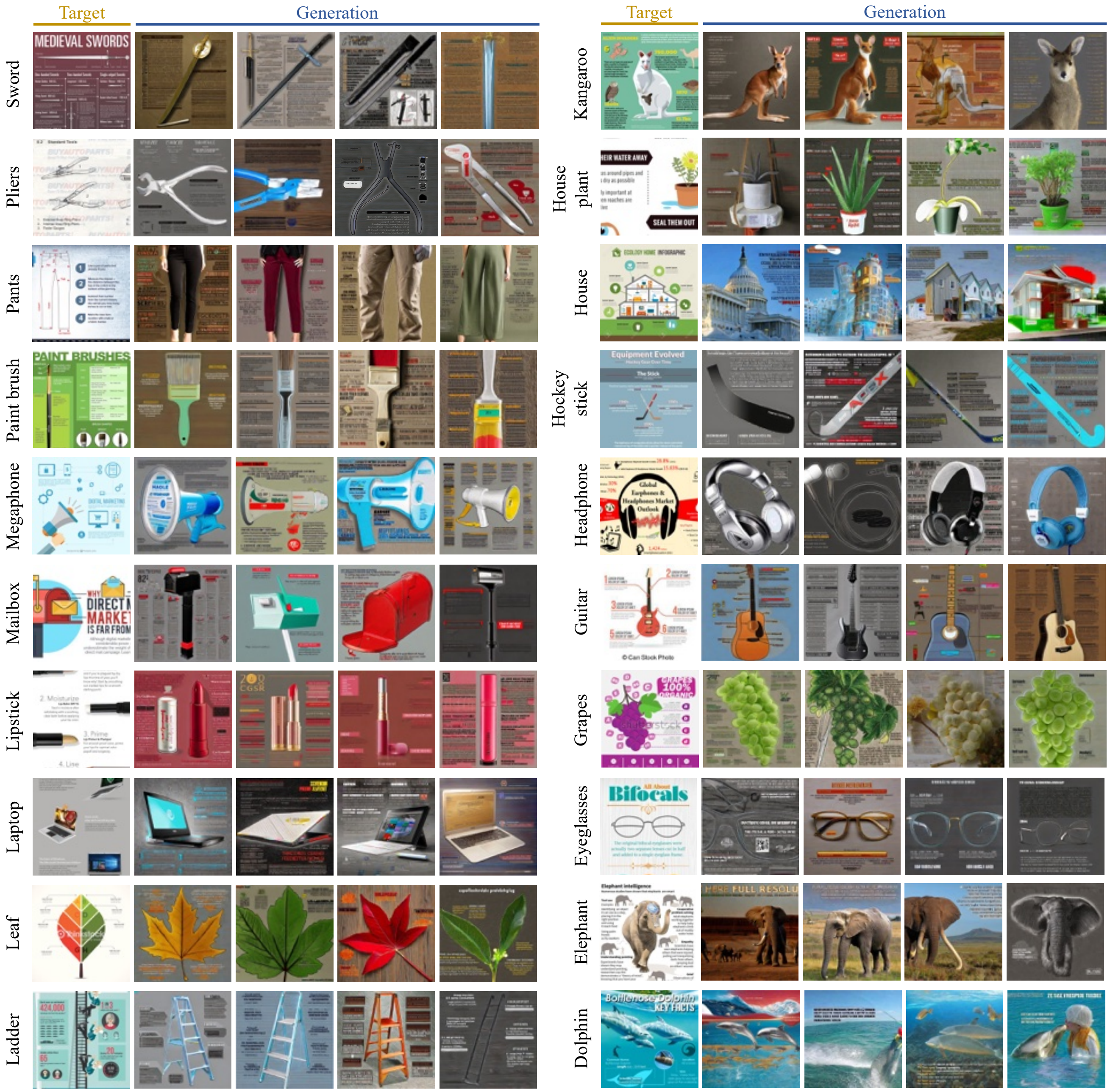

Figure 15. **More synthetic examples from the DomainNet Real → Infograph generation.** We list more synthetic data generated in our Real→Infograph UDA experiment in Sec. 4.3.1. We randomly select and show 20 classes from DomainNet. For each class, we present one image from the DomainNet Infograph domain as a reference and four of our generations. These examples demonstrate our generation quality and capability to effectively augment real images into the Infograph domain.

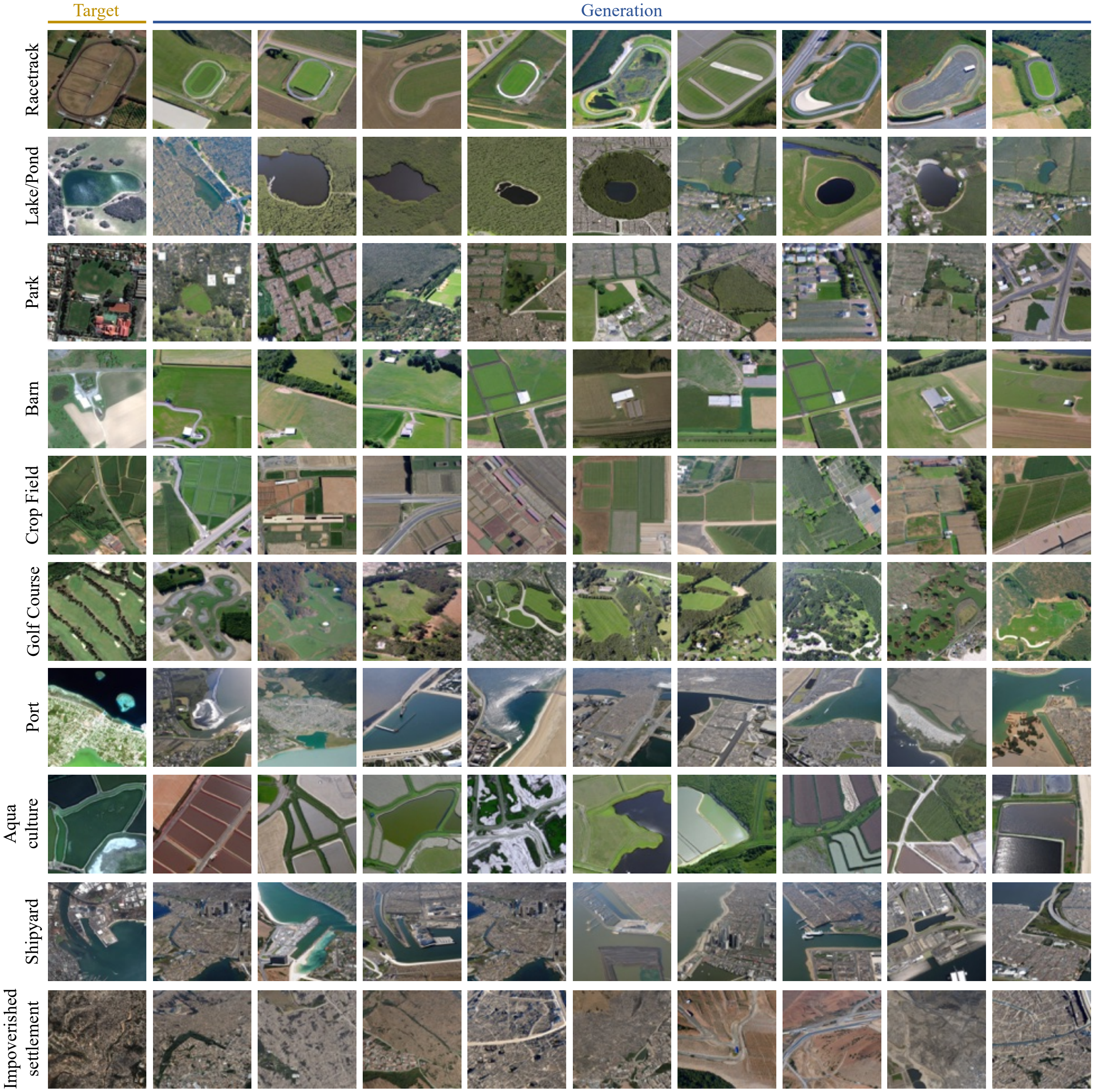

Figure 16. **More synthetic examples from the FMoW domain adaptation experiment.** Following our experiment setup, we augment satellite images from relatively older periods into more recent times. We randomly select and present 10 classes of land use. For each class (row), the leftmost column shows randomly selected references from the target domain; the remaining nine images are our synthetic data.

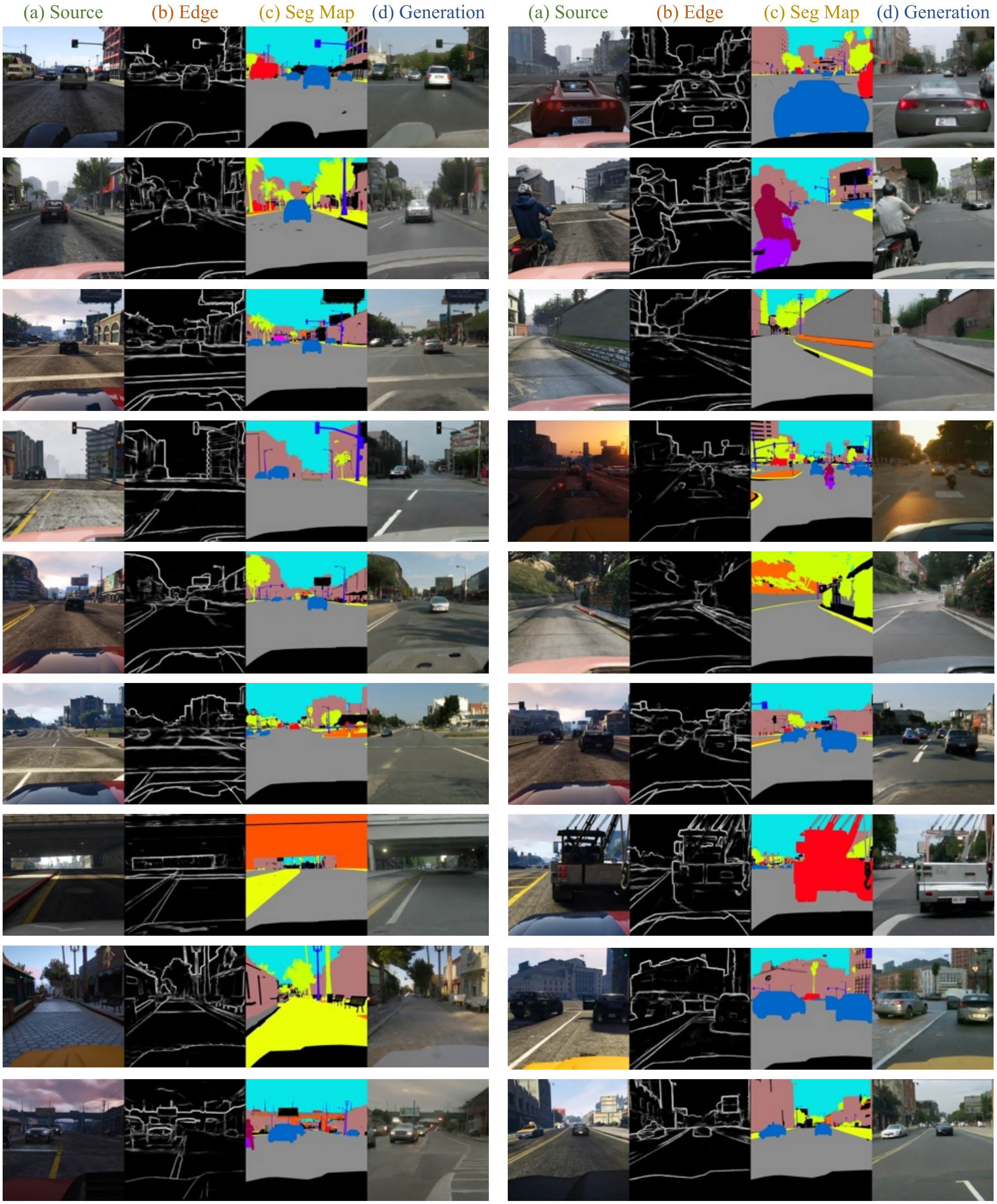

Figure 17. **More synthetic examples from the GTA5 → CityScapes segmentation experiment.** Given the (a) source data from GTA5, we extract (b) the corresponding edge map. Together with the provided (c) segmentation map, DoGE generated (d) synthetic images that are closer to the CityScapes data distribution. These examples showcase our generation capability for complex scenes.