Supplementary Material: TransferI2I: Transfer Learning for Image-to-Image Translation from Small Datasets

Yaxing Wang[1,2], Héctor Laria[2] Joost van de Weijer[2], Laura Lopez-Fuentes[3], Bogdan Raducanu[2]
[1] PCALab, Nanjing University of Science and Technology, China
[2]Computer Vision Center, Universitat Autònoma de Barcelona, Spain
[3]Universitat de les Illes Balears, Spain
{yaxing,hlaria,joost,bogdan}@cvc.uab.es, l.lopez@uib.es

	Network	Optimizer	Lr	(β_1, β_1)	Bs	Is
Two-class I2I	G	Adam	$1e^{-5}$	(0.0,0.99)	16	256
	A, D	Adam	$1e^{-3}$	(0.0,0.99)	16	256
Multi-class I2I	G	Adam	$5e^{-5}$	(0.0,0.999)	16	128
	A, D	Adam	$2e^{-4}$	(0.0,0.999)	16	128

Table 1. The experiment configuration. Lr: learning rate, Bs: batch size, Is: image size.

A. Multi-class I2I translation

Here we introduce how to perform unpaired multi-class I2I translation. We consider two domains: source domain $\mathcal{X}_1 \subset \mathbb{R}^{H\times W\times 3}$ and target domain $\mathcal{X}_2 \subset \mathbb{R}^{H\times W\times 3}$ (it can trivially be extended to multiple classes). In this work, given limited training samples from both source and target domains, we aim to map a source image $\mathbf{x}_1 \in \mathcal{X}_1$ into a target sample $\mathbf{x}_{1\to 2} \in \mathcal{X}_2$ conditioned on the target domain label $\mathbf{c} \in \{1, \ldots, C\}$ and a random noise vector $\mathbf{z} \in \mathbb{R}^{\mathbf{Z}}$. Let image $\mathbf{x} \in \mathcal{X}_1 \bigcup \mathcal{X}_2$ is sampled from dataset.

As illustrated Figure 1, our framework is composed of three stages: *source-target initialization* (Figure 1(a)) aiming to obtain a satisfactory domain-specific GAN, which can then be used for I2I translation; *self-initialization of adaptor layer* (Figure 1(b)) which reduces the risk of overfitting of the adaptor layers when trained on limited data; and *transfer learning for I2I translation* (Figure 1(c)) which finetunes all networks, each of which is initialized according to the previous steps, on the few available source and target images.

Source-target initialization. we expect to study a excellent generative model utilizing the limited training data. Different to the model for two-class I2I translation, in this stage we train one generator and one discriminator on all images instead of class-specific generator and class-specific discriminator. The training objective is as following:

$$\begin{aligned}\mathcal{L}_{GAN} = &\mathbb{E}_{\mathbf{x}\sim\mathcal{X}_1\bigcup\mathcal{X}_2}\left[\log D\left(\mathbf{x}, c\right)\right] \\ &+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}),\mathbf{c}\sim p(\mathbf{c})}\left[\log\left(1-D\left(G\left(\mathbf{z}, c\right), c\right)\right)\right],\end{aligned} \tag{1}$$

where $\mathbf{p}\left(\mathbf{z}\right)$ follows the normal distribution, and $\mathbf{p}\left(\mathbf{c}\right)$ is the domain label distribution. Here the generative model is used to provide a better initialization for the I2I translation.

Self-initialization of adaptor layer. We expect to overcome the overfitting of the adaptor layers, as well as aligning the distribution of both the pretrained generator and the pretrained discriminator. As introduced in Section 3.1, we propose the *self-initialization* procedure, which leverages the previous pretrained model (Figure 1 (a)) to achieve this goal. Especially, both the noise $\mathbf{z}$ and the class embedding $\mathbf{c}$ are taken as input for the generator G, from which we extract the hierarchical representation $F_g(\mathbf{z}, \mathbf{c}) = \{G(\mathbf{z}, \mathbf{c})_l\}$ as well as the synthesized image $G(\mathbf{z}, \mathbf{c})$. Here $G(\mathbf{z}, \mathbf{c})_l$ is the $l_{th}(l = m, \ldots, n, (n > m))$ ResBlock [1] output of the generator G. We then take the generated image $G(\mathbf{z}, \mathbf{c})$ as input for the discriminator D, and similarly collect the hierarchical feature $F_d(\mathbf{z}) = \{D(G(\mathbf{z}, \mathbf{c}))_l\}$. The adaptor network A finally takes the output representation $\{D(G(\mathbf{z}, \mathbf{c}))_l\}$ as input, that is $A(F_d(\mathbf{z})) = \{A\}$. In this step, our loss is:

$$\mathcal{L}_{ali} = \sum_l \left\|F_g\left(\mathbf{z}\right) - A\left(D(G(\mathbf{z}, \mathbf{c}))\right)\right\|_1. \tag{2}$$

Transfer Learning for I2I translation. Figure 1(c) shows how to map the image from the source domain to target domain. In this stage, we propose an *auxiliary generator* $\widetilde{G}'$ which aims to improve the usage of the deep layers of the generator, largely due to the skip connections. It is relatively easy for the generator to use the information from the high-resolution skip connections (connecting to the upper layers of the generator), and ignore the deep layers of the generator, which require a more semantic understanding of the data, thus more difficult to train.

Our loss function for I2I translation is a multi-task objective comprising: (a) *conditional adversarial loss* which not only classifies the real image and the generated image,

[1] After each ResBlock the feature resolution is half of the previous one in both encoder and discriminator, and two times in generator

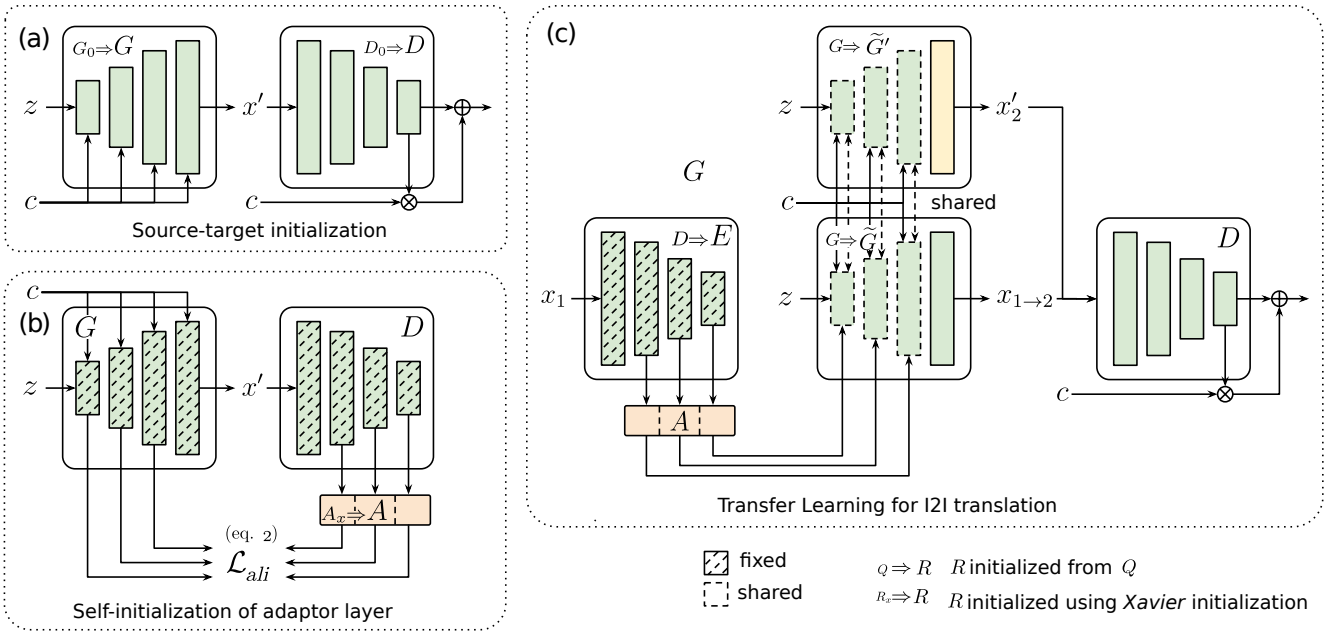

Figure 1. Conditional model architecture and training stages. Here modules come from the immediate previous stage unless otherwise indicated. A pretrained GAN (e.g., BigGAN [2]) is used as G_0 and D_0 to initialize the GAN. (a) *Source-target initialization* performs finetuning on all data to form a trained GAN model (i.e., the generator G and the discriminator D). (b) *Self-initialization* of adaptor layer to pretrain the adaptor A and align both the generator G and the discriminator D. We only update the adaptor layers A. (c) The I2I translation model is composed of five main parts: the encoder E, the adaptor layer A, the generator $\widetilde{G}$, the *auxiliary generator* $\widetilde{G}'$ and the discriminator D. Note the encoder E is initialized by the discriminator D. The portion of weights from G' that is not shared (in yellow), is initialized with G weights.

but encourages the networks $\{E, A, \widetilde{G}\}$ to generate class-specific images which correspondent to label $\mathbf{c}$. (b) *reconstruction loss* guarantees that both the input image $\mathbf{x}_1$ and the synthesized image $\mathbf{x}_{1\to 2} = \widetilde{G}(\mathbf{z}, \mathbf{c}, A(E(\mathbf{x}_1)))$ keep the similar structural information.

Conditional adversarial loss. We employ GAN [4] to optimize this problem as follows:

$$\begin{aligned}\mathcal{L}_{GAN} &= \mathbb{E}_{\mathbf{x_2}\sim\mathcal{X}_2,\mathbf{c}\sim p(\mathbf{c})}\left[\log D\left(\mathbf{x_2},\mathbf{c}\right)\right] \\ &+ \mathbb{E}_{\mathbf{x_1}\sim\mathcal{X}_1,\mathbf{z}\sim p(\mathbf{z}),\mathbf{c}\sim p(\mathbf{c})}\left[\log(1-D\left(\widetilde{G}\left(A\left(E\left(\mathbf{x_1}\right)\right),\mathbf{z},\mathbf{c}\right)\right)\right] \\ &+ \lambda_{aux}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}),\mathbf{c}\sim p(\mathbf{c})}\left[\log\left(1-D\left(\widetilde{G}'\left(\mathbf{z},\mathbf{c}\right)\right)\right)\right],\end{aligned} \tag{3}$$

The hyper-parameter λ_{aux} balances the importance of each terms. We set $\lambda_{aux} = 0.01$.

Reconstruction loss. We use reconstruction to preserve the structure of both the input image x_1 and the output image $x_{1\to 2}$. In the same fashion as results for photo-realistic image generation [5, 6, 9], we use the discriminator output to achieve this goal through the following loss:

$$\mathcal{L}_{rec} = \sum_l \alpha_l \left\| D\left(\mathbf{x_1}\right) - D\left(\mathbf{x_{1\to 2}}\right)\right\|_1, \tag{4}$$

where parameters α_l are scalars which balance the terms. Note we set $\alpha_l = 1$.

Full Objective. The full objective function of our model is:

$$\min_{E,A,\widetilde{G},\widetilde{G}'}\max_D \mathcal{L}_{GAN} + \lambda_{rec}\mathcal{L}_{rec} \tag{5}$$

where λ_{rec} is a hyper-parameter that balances the importance of each terms. We set $\lambda_{rec} = 1$.

The configure of the experiment is reported in Table 1

B. Adaptor

We use the adaptor A to connect the encoder E and the generator G, aiming to leverage both the structure and semantic information. We sum the output of the adaptor with the corresponding one of the generator, which is as following:

$$\hat{G}_l = G_l\left(\mathbf{x}_1, \mathbf{z}, \mathbf{c}\right) + w_l A_l\left(E_l\left(\mathbf{x}_1\right)\right) \tag{6}$$

where G_l is the output of the corresponding layer which has same resolution to A_l. The hyper-parameters w_l are used to balance the two terms (in this work we set w_l is 1 except for the feature (32*32 size) which is 0.1). Note for two-class I2I translation, we perform similar procedure.

C. Ablation study

We further qualitatively compare the generated images after *source and target initialization* on two-class I2I trans-

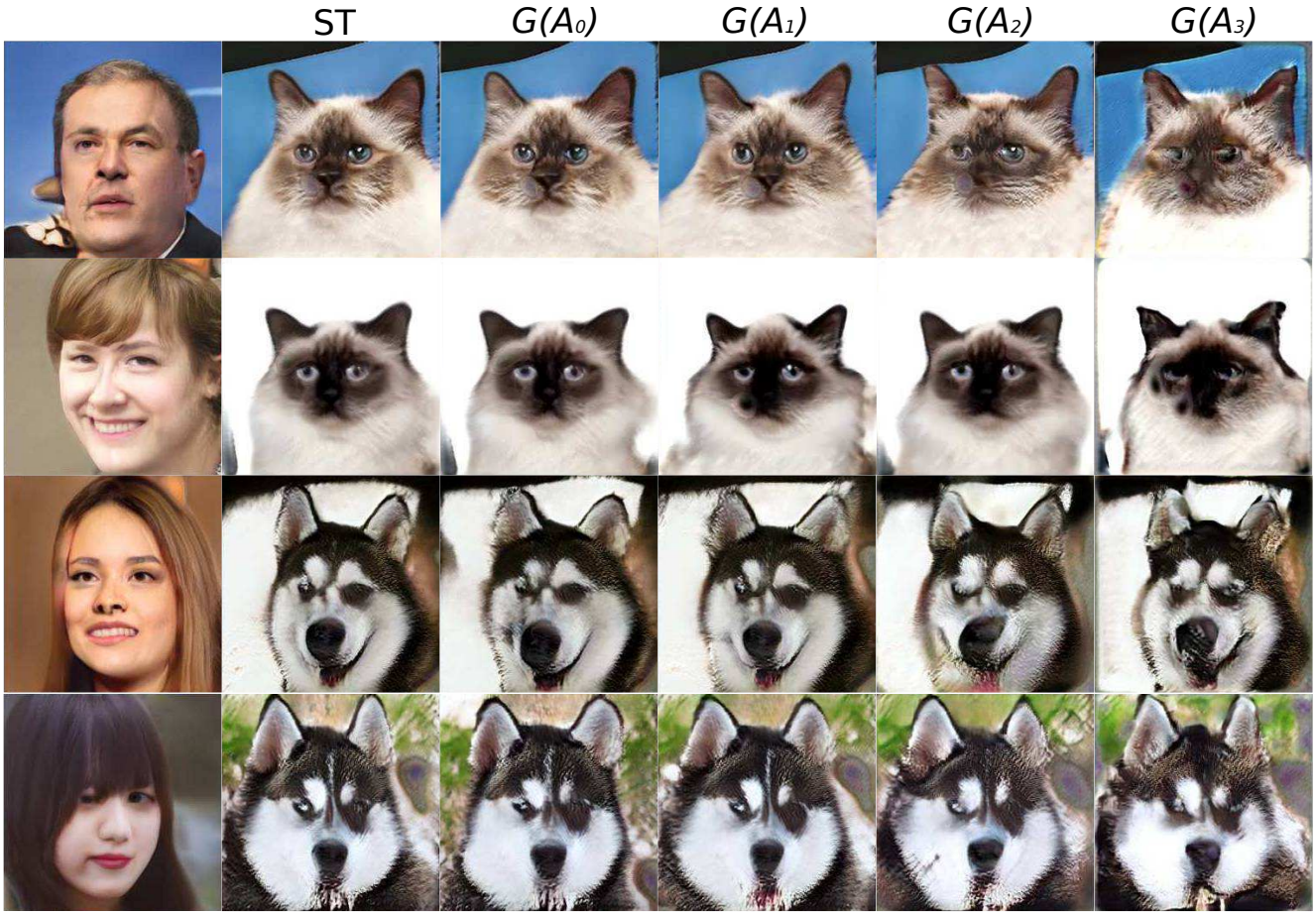

Figure 2. Examples generated by both *source and target initialization* and *self-initialization* of the adaptor on *cat2dog* dataset. The first two columns are the output of the StyleGAN and the generator after the *source and target initialization* respectively. The remaining columns ($G(A_i)(i = 0, 1, 2, 3)$) are the corresponding output of the generator G which only takes the corresponding output of the adaptor $A_i(i = 0, 1, 2, 3)$.

lation. The second column of Figure 2 shows the synthesized images after *source and target initialization*. We can see that the produced images are highly realistic and category-specific, indicating the effectiveness of this step. Next, we want to verify whether the *self-initialization* of the adaptor successfully aligns encoder and generator. Therefore, we take the noise as input for the generator, and obtain an image, which is further fed into the discriminator and then through the adaptor layer. The adaptor layer output is then used as the only input of the generator (now no noise input z is given). The results are provided in the third to last columns of Figure 2. The generator still produces high fidelity images when only inputting the output features from the adaptor. These results demonstrate that the distribution of the adaptor is aligned to the generator before performing the transfer learning for I2I translation (Figure 1(c)).

Method \ Dataset	(apple,orange):(100,100)				(face,moji):(100,100)	
	apple → orange		orange → apply		face → moji	
	FID↓	100 × KID ↓	FID↓	100 × KID ↓	FID↓	100 × KID ↓
NICEGAN [3]	193.2	9.98	233.1	15.2	139.4	10.7
CUT[8]	217.3	14.0	258.1	16.3	324.3	35.3
TransferI2I (ours)	**173.5**	**8.54**	**179.6**	**7.89**	**78.6**	**4.02**

Table 2. The metric results on apple2orange and face2moji datasets.

D. Two-class I2I translation

We also use two no-face datasets: apple2orange[60] and face2moji[34]. Each of them contains 100 images for training and 100 images for test. As depicted in Tab. 2, we outperform the other methods, on both metrics for both datasets.

E. Results

Figure 3 reports interpolation by freezing the input images while interpolating the class embedding between two classes. Our model still manages to generate high quality

Figure 3. Interpolation by keeping the input image fixed while interpolating between two class embeddings. The first column is the input images, while the remaining columns are the interpolated results. The interpolation results from *pug* to *mongoose*.

images even for never seen class embeddings. On the contrary, StarGANv2 with limited data shows unsatisfactory performance.

Our method obtains compelling performance for many cases, but suffers from some failures. Figure 4 shows a few failure cases. The input images are different from the normal animal faces. E.g. a side-view of a face, which is rare in the dataset, causes the model to produce unrealistic results.

We evaluate the proposed method on both *cat2lion* and *lion2cat* datasets, which has 100 images for each category. The qualitative results are shown in Figure 5.

We also show results translating an input image into all category on the *Animal faces*, *Foods*, and *Birds* in Figure 6, and 8.

F. T-SNE

We explore the latent space of the generated images. Given the target class $\mathbf{c}$ (e.g., *Rhodesian ridgeback*)), we take different noises $\mathbf{z}$ and the constant $\mathbf{c}$ as input for the networks $\{E, A, G\}$, and generate 1280 images. Thus we use Principle Component Analysis (PCA) [1] to extracted feature, following the T-SNE [7] to visualize the generated images in a two dimensional space. As shown in Figure 9, given the target class (Rhodesian ridgeback), TransferI2I correctly disentangles the pose information of the input classes. The T-SNE plot shows that input animals having similar pose are localized close to each other in the T-SNE plot. Furthermore, it shows TransferI2I has the ability of diversity. We also conduct T-SNE for 14,900 generated images across 149 categories (Figure 10).

Figure 4. Typical failure case of our method.

References

[1] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical Methods*, 6(9):2812–2831, 2014. 4

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 2

[3] Runfa Chen, Wenbing Huang, Binghui Huang, Fuchun Sun, and Bin Fang. Reusing discriminators for encoding towards unsupervised image-to-image translation. In *CVPR*, 2020. 3

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and

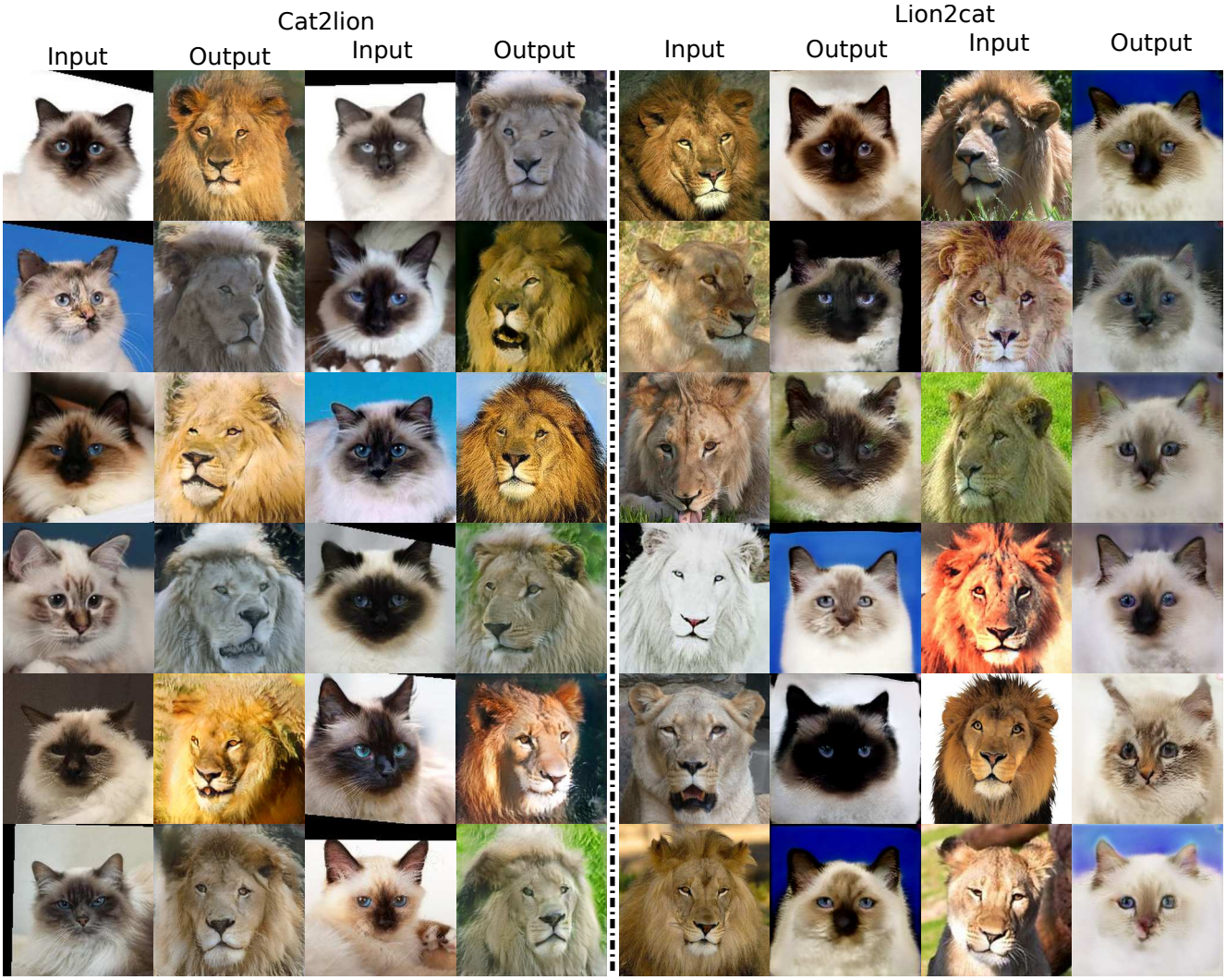

Figure 5. Qualitative results on both *cat2lion* and *lion2cat* dataset.

Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 2

[5] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, pages 4016–4027, 2018. 2

[6] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, 2018. 2

[7] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 4

[8] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for conditional image synthesis. In *ECCV*, 2020. 3

[9] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, pages 2107–2116, 2017. 2

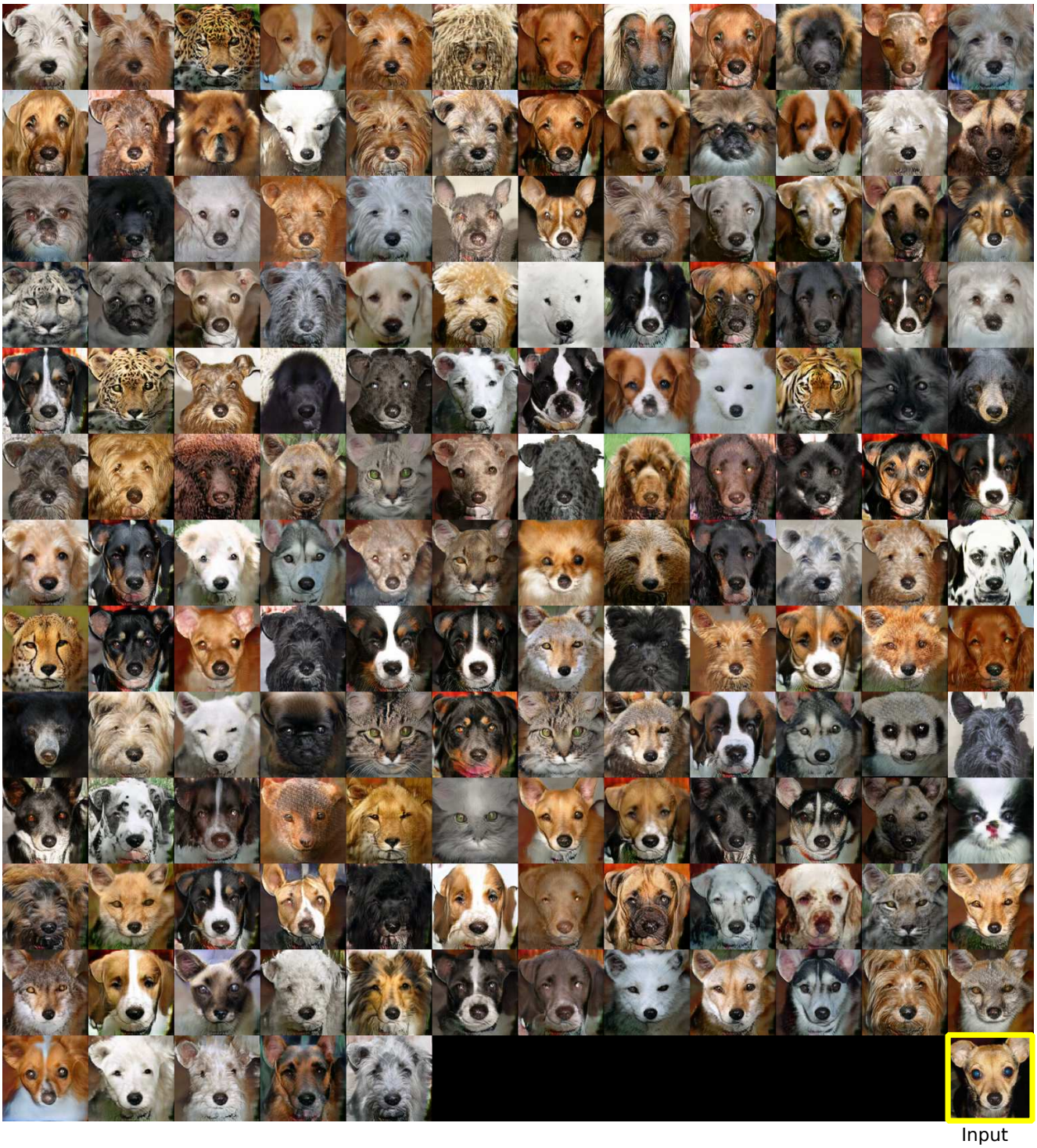

Figure 6. Qualitative results on the *Animal faces* dataset. We translate the input image (bottom right) into all 149 categories. Please zoom-in for details.

Figure 7. Qualitative results on the *Foods* dataset. We translate the input image (bottom right) into all 256 categories. Please zoom-in for details.

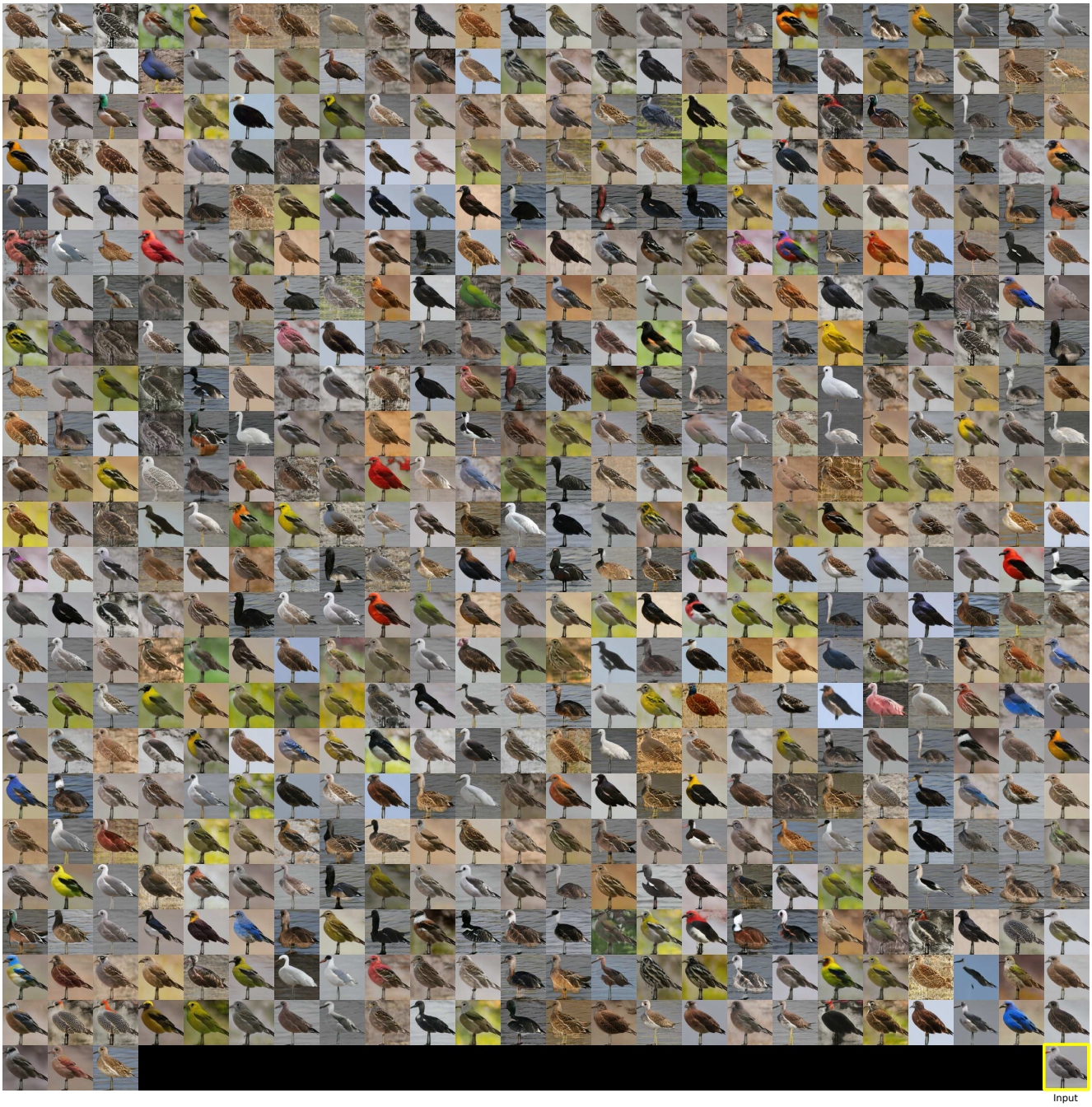

Figure 8. Qualitative results on the *Birds* dataset. We translate the input image (bottom right) into all 555 categories. Please zoom-in for details.

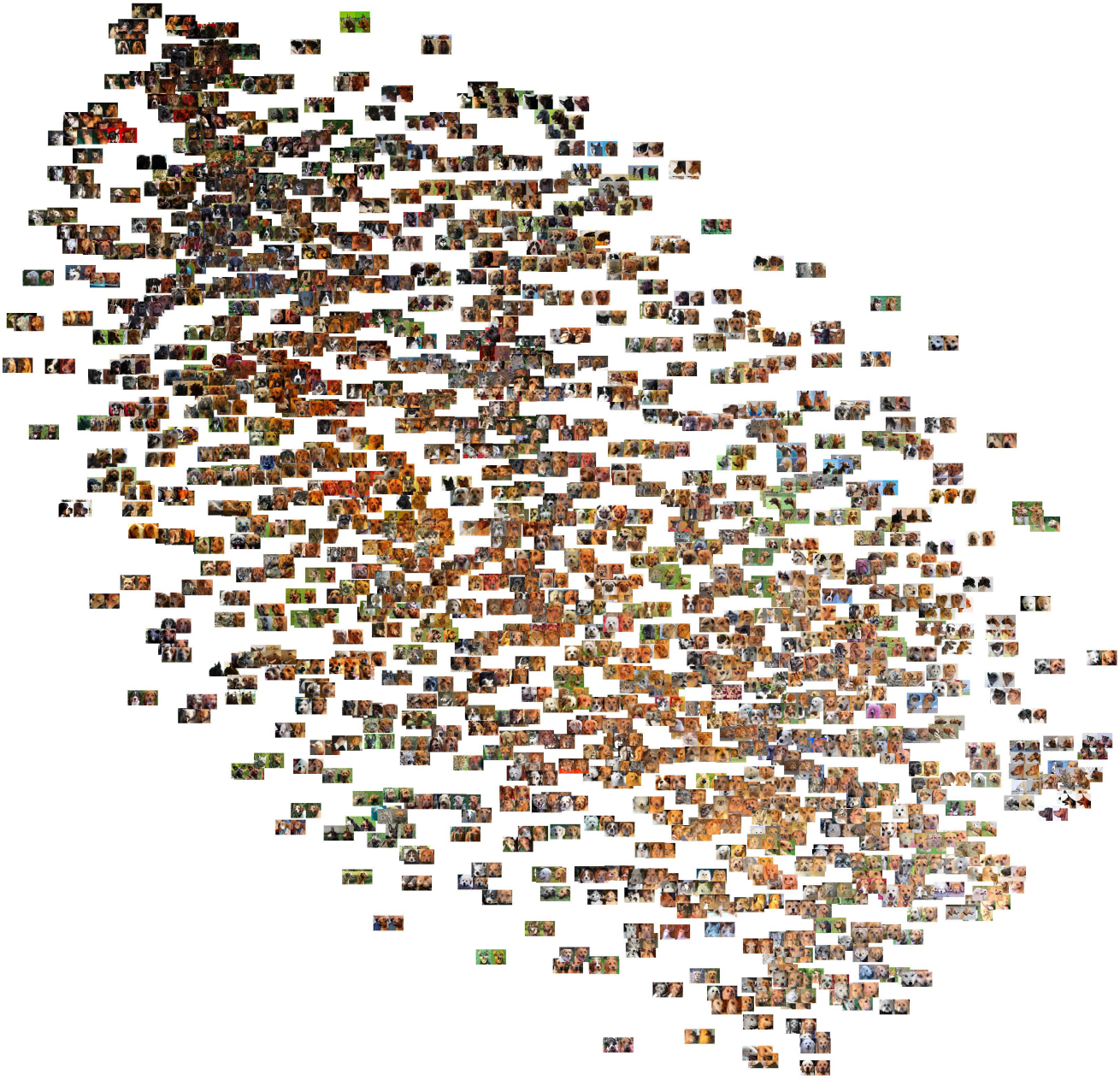

Figure 9. 2-D representation of the T-SNE for 1280 generated images, the target class is *Rhodesian ridgeback*. Note that for each pair image, the left is the input and the right is the output image. Please zoom-in for details.

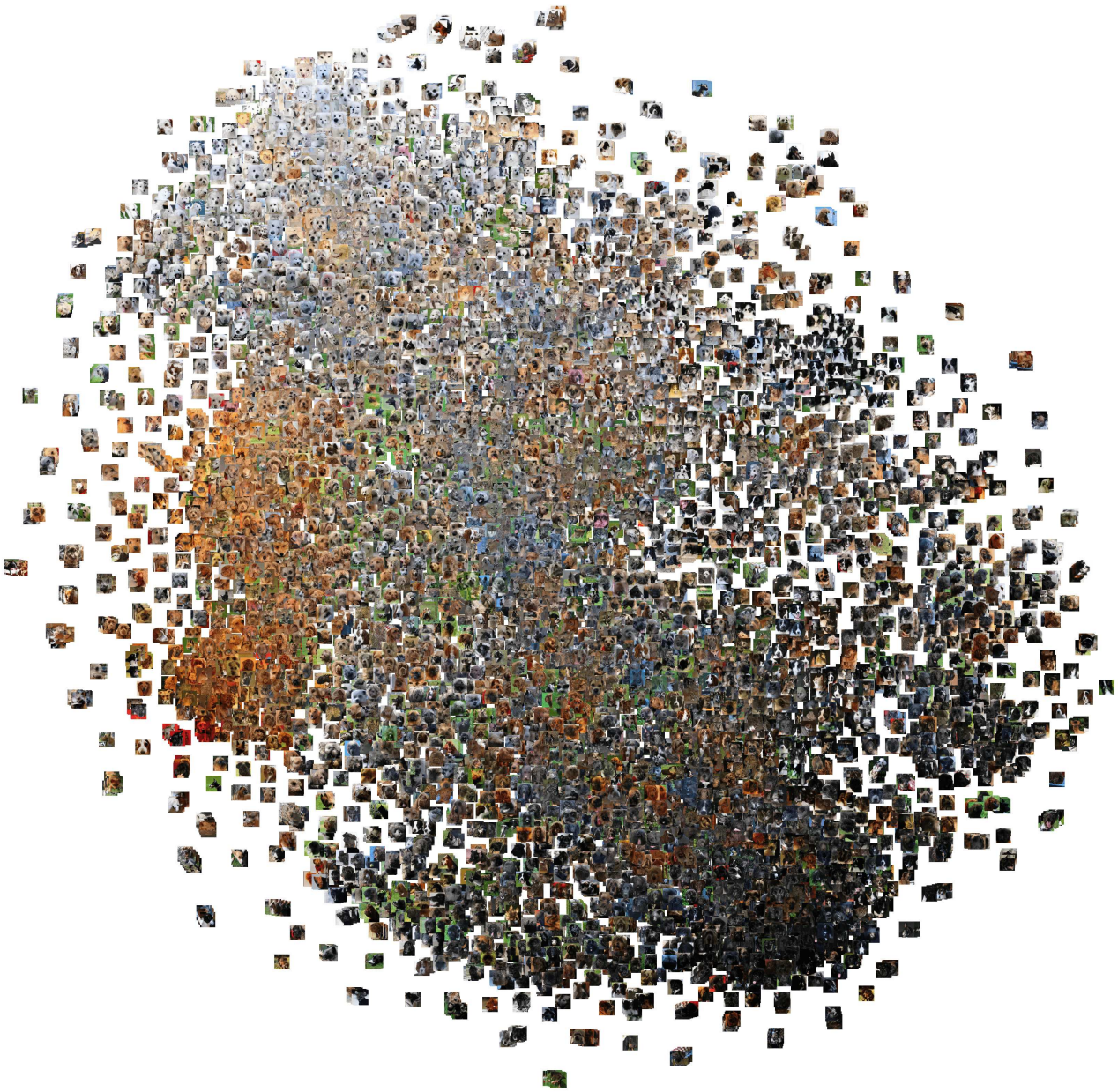

Figure 10. 2-D representation of the T-SNE for 14900 generated images across 149 classes. Please zoom-in for details.