SketchInverter: Multi-Class Sketch-Based Image Generation via GAN Inversion – Supplementary Material –

Zirui An*
Beihang University

Jingbo Yu*
Beihang University

Runtao Liu
Johns Hopkins University

Chuang Wang
Beihang University

Qian Yu†
Beihang University

1. Architecture Details

Conditional Encoder *E*. In Fig. 1, we show the architecture of our conditional encoder ***E***. The conditional encoder ***E*** takes a sketch ***s*** with a class label ***y*** as the input and outputs a latent code ***z***. It consists of five residual blocks with bottleneck layers [3], one convolutional layer, one down-sampling (max pooling) layer, and one linear projection layer. We use a shared class embedding as the condition. As in [1], the condition vector of each block is linearly projected to produce per-sample gains and biases for the Batch-Norm layers. The bias projections are centered at zero, and the gain projections are centered at one. In each ResBlock, conditional batch normalization and ReLU are followed after each convolutional layer (as illustrated in Fig. 1(right)). Table 1 provides further details of the conditional encoder.

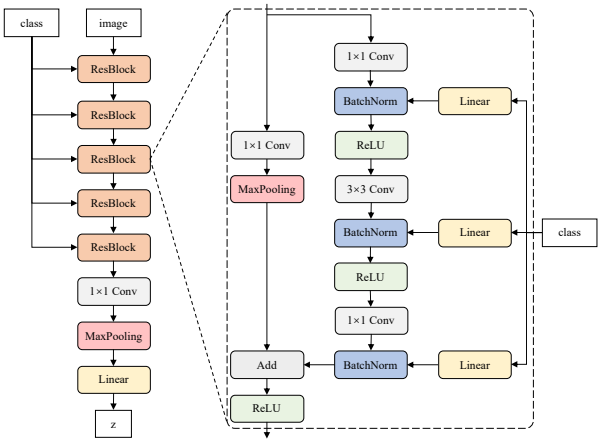

Figure 1. The detailed architectural for our conditional encoder ***E*** (left). The architecture of a residual block [3] (ResBlock down) in ***E*** (right).

Photo-to-Sketch Network *S*. Figure 2 illustrates the architecture of the photo-to-sketch network ***S***. It consists of twelve residual blocks, including six blocks for down-sampling (ResBlock-down) and six for up-sampling (ResBlock-up), and one convolutional layer. As shown in Fig. 2(a), in each ResBlock-down, the main connection con-

Table 1. Architecture of the conditional encoder ***E***. The input is a sketch image sketch $s \in \mathbb{R}^{128\times128\times3}$, while the output is a latent code $z \in \mathbb{R}^{120}$.

Index	Layer	Output size
(1)	Input sketch	$128 \times 128 \times 3$
(2)	ResBlock down	$64 \times 64 \times 32$
(3)	ResBlock down	$32 \times 32 \times 64$
(4)	ResBlock down	$16 \times 16 \times 128$
(5)	ResBlock down	$8 \times 8 \times 256$
(6)	ResBlock down	$4 \times 4 \times 512$
(7)	Conv(1×1)	$4 \times 4 \times 512$
(8)	Max pooling	$2 \times 2 \times 512$
(9)	Flatten	2048
(10)	Linear	120

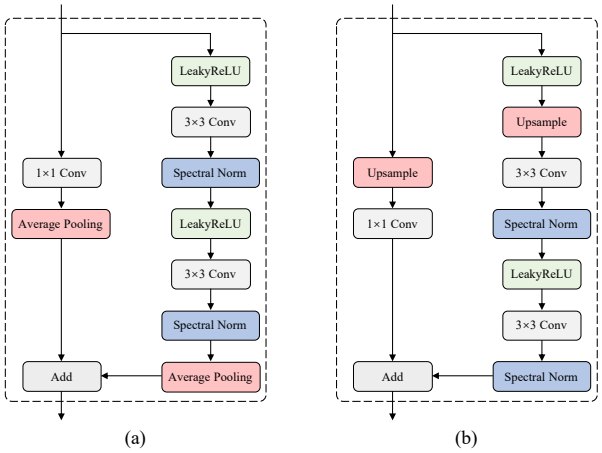

Figure 2. Architectures of two types of Residual blocks [3], ResBlock down and ResBlock up, in Image-to-sketch network ***S***. (a) ResBlock down in ***S***.(b) ResBlock up in ***S***.

*These authors contributed equally to this work.
†Corresponding author.

Table 2. Image-to-sketch network $\boldsymbol{S}$ architecture for image $x \in \mathbb{R}^{128\times128\times3}$. The output is sketch $s \in \mathbb{R}^{128\times128\times3}$.

Index	Layer	Output size
(1)	Input image	$128 \times 128 \times 3$
(2)	ResBlock down	$64 \times 64 \times 128$
(3)	ResBlock down	$32 \times 32 \times 128$
(4)	ResBlock down	$16 \times 16 \times 256$
(5)	ResBlock down	$8 \times 8 \times 256$
(6)	ResBlock down	$4 \times 4 \times 512$
(7)	ResBlock down	$2 \times 2 \times 512$
(8)	ResBlock up	$4 \times 4 \times 512$
(9)	Concatenate (6) and (8)	$4 \times 4 \times 1024$
(10)	ResBlock up	$8 \times 8 \times 256$
(11)	Concatenate (5) and (10)	$8 \times 8 \times 512$
(12)	ResBlock up	$16 \times 16 \times 256$
(13)	Concatenate (4) and (12)	$16 \times 16 \times 512$
(14)	ResBlock up	$32 \times 32 \times 128$
(15)	Concatenate (3) and (14)	$32 \times 32 \times 256$
(16)	ResBlock up	$64 \times 64 \times 128$
(17)	Concatenate (2) and (16)	$64 \times 64 \times 256$
(18)	ResBlock up	$128 \times 128 \times 64$
(19)	Conv(1×1)	$128 \times 128 \times 3$

sists of two 3×3 convolutional layers, which are followed by a spectral normalization layer. Before feeding into the 3×3 convolutional layer, the input will first go through a LeakyReLU layer. The residual connection consists of one 1×1 convolutional layer and one down-sampling (average pooling) layer. The output of the main and the residual connection will pass to an average pooling layer and then are fused by addition. The architecture of the ResBlock-up is similar to that of the ResBlock-down, the only difference is that each ResBlock-up uses the up-sampling layer instead of the down-sampling (average pooling) layer before convolutional layers. Table. 2 provides further details of the network $\boldsymbol{S}$.

2. Additional Qualitative Results

In this section, we provide more qualitative results to demonstrate the effectiveness of our proposed method. Figure 3 and Fig. 4 show more results generated by our model on the Sketchy Database and ablation studies, respectively. Figure 5 shows representative results of our proposed method, *SketchInverter*, and four baseline methods.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[2] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5174–5183, 2020.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. In *SIGGRAPH*, 2016.

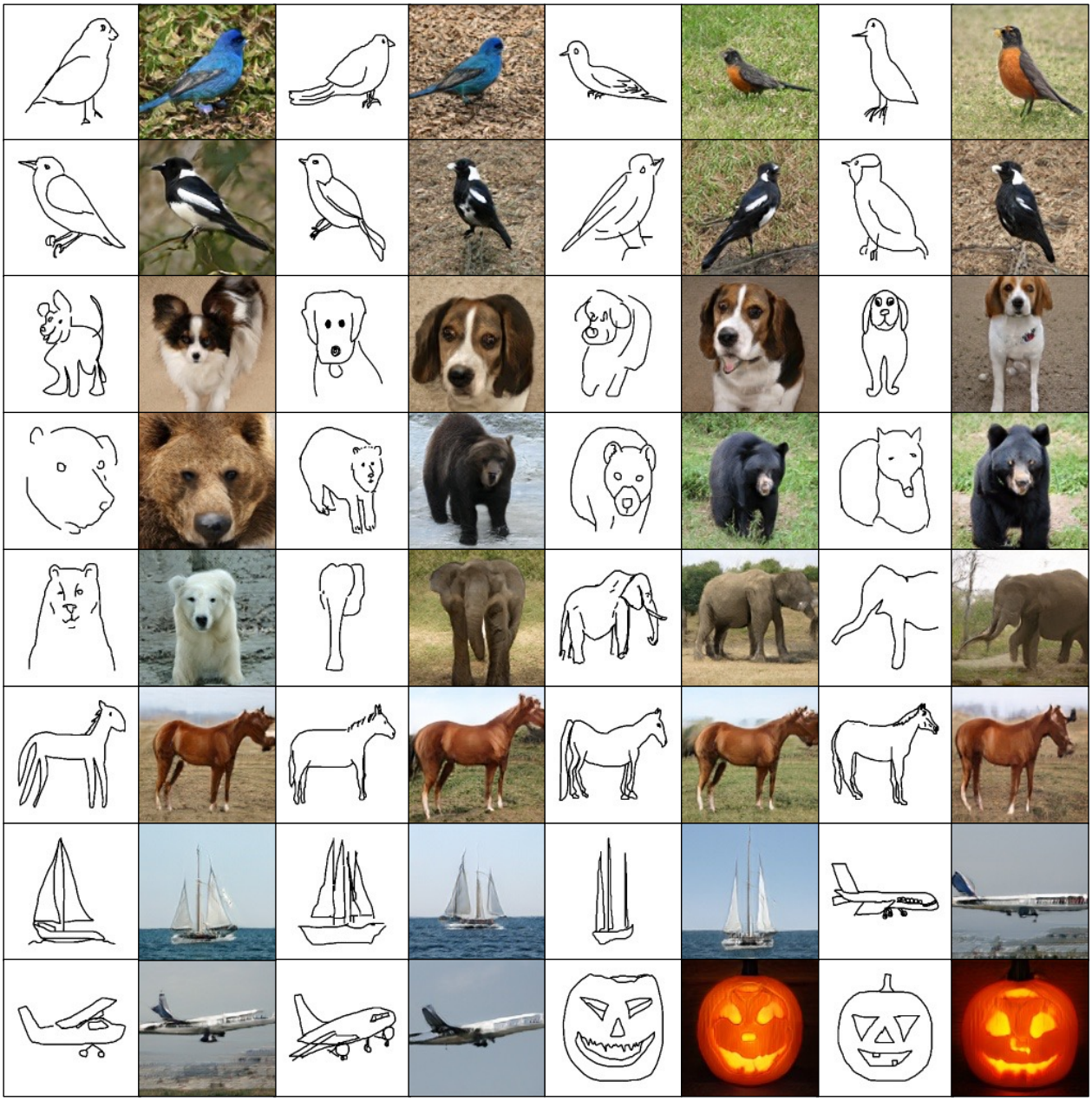

Figure 3. More visualization results of our full model on the Sketchy Database [4] and SketchyCOCO dataset [2]

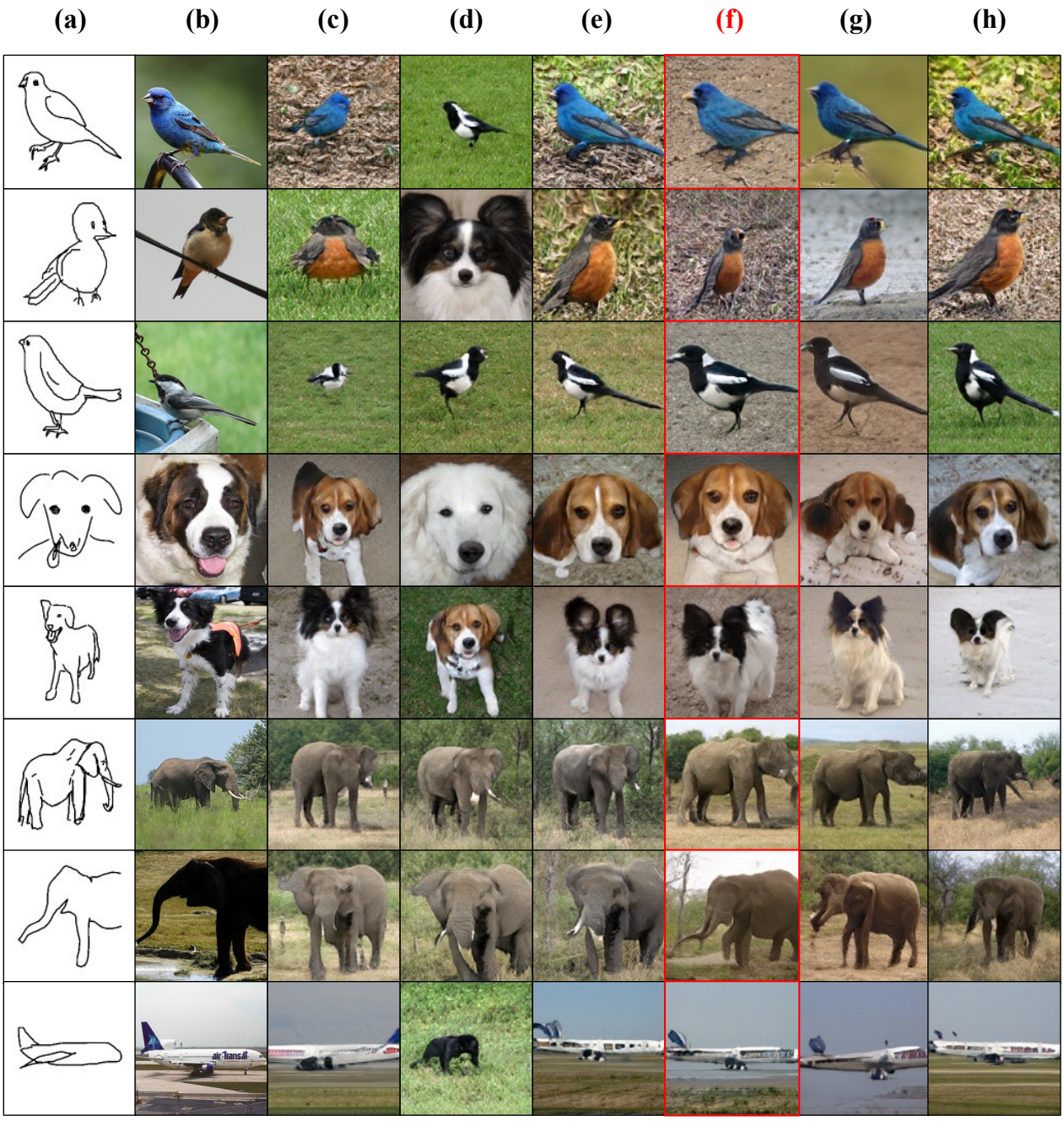

Figure 4. More visualization results of ablation study. (a) sketch; (b) ground-truth; (c) using one non-conditional encoder; (d) using two non-conditional encoders; (e) training without shape loss; (f) our full model; (g) our full model fine-tuned on the Sketchy database; (h) our full model trained on the mix of our synthetic dataset and the Sketchy database. The models of (c)(d)(e)(f) are trained under the same setting that training on our synthetic dataset and testing on the Sketchy database.

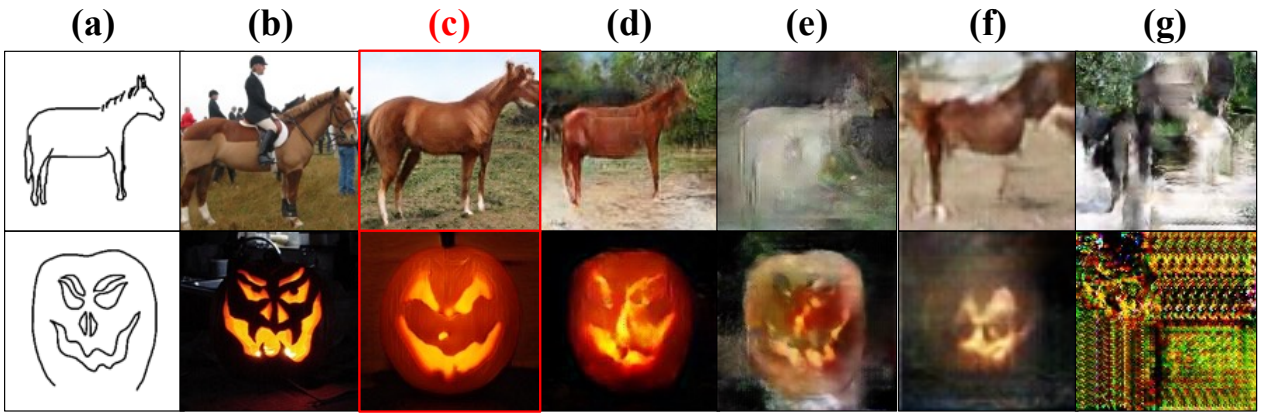

Figure 5. Visualization results tested on sketches from Sketchy database. (a) Sketch; (b) Ground-truth; (c) Our method; (d) Pix2pix-Sep: one model per class; (e) Pix2pix-Mix: a single model for all classes; (f) EdgeGAN-S; (g) AODA. (c)(d)(e)(f)(g) are all trained on our synthetic dataset and tested on the Sketchy database.