Unsupervised Sketch-to-Photo Synthesis Supplementary Material

Runtao Liu[1]* Qian Yu[21]* Stella X. Yu[1]

UC Berkeley / ICSI[1] Beihang University[2]

In the main paper, we propose a model for the task of sketch-based photo synthesis, which can deliver sketch-faithful realistic photos. Our key insight is to decompose this task into two separate translations. Our two-stage model performs first geometrical shape translation in grayscale and then detailed content fill-in in color. Besides, at the first stage, a self-supervised learning objective along with noise sketch composition strategies and an attention module are brought up to handle abstraction and drawing style variations.

In this Supplementary, we first provide further implementation details in Section 1, including architectures of the proposed model, loss functions, and how we conducted the user study. Then in Section 2, we provide additional qualitative results to demonstrate the effectiveness of our model (see the caption of each figure for details). Additionally, we show results when applying our model in a *multi-class* setting.

1 Additional Implementation Details

Shape translation. The architecture of two generators, T and T', consists of nine residual blocks, two down-sampling, and two up-sampling layers. Instance normalization and ReLU is followed after each convolutional layer. The proposed *attention module* includes two convolutional layers. The Softmax activation function is used to produce the attention mask. We do not add a normalization layer after Conv layers in this module. The architecture of the *discriminators*, D_S and D_G, is composed of four convolutional layers, and each one is followed by instance normalization and LeakyReLU.

Content enrichment. The architecture of the *encoder* E consists of nine convolutional layers and two max-pooling layers, which shares the same structure of the first three blocks of VGG-19. The *decoder* D has twelve residual blocks and two up-sampling layers. When reference photos are available, E is used for feature extraction.

Loss function. For shape translation network, as indicated in the main paper, the loss function has five items:

$$L_{adv}(T, D_G; S, G) = (D_G(G))^2 + (1 - D_G(T(S)))^2 \tag{1}$$

$$L_{adv}(T', D_S; G, S)) = (D_S(S))^2 + (1 - D_S(T'(G)))^2 \tag{2}$$

[*] Equal contribution.

$$L_{cycle}(T,T';S,G) = \left\|S - T'\left(T(S)\right)\right\|_1 + \left\|G - T\left(T'(G)\right)\right\|_1 \tag{3}$$

$$L_{identity}(T,T';S,G) = \left\|S - T'(S)\right\|_1 + \left\|G - T(G)\right\|_1 \tag{4}$$

$$L_{ss}(T,T';S,S^{noise}) = \left\|S - T'\left(T(S^{noise})\right)\right\|_1 \tag{5}$$

For content enrichment network C, the objective has four items, they are:

$$L_{adv}(C,D_I;G,I) = \left(D_I(I)\right)^2 + \left(1 - D_I\left(C(G)\right)\right)^2 \tag{6}$$

$$L_{it}(C) = \left\|G - Gray\left(C\left(G\right)\right)\right\|_1 \tag{7}$$

$$L_{cont}(C;G,R) = \left\|E(D(t)) - t\right\|_1 \tag{8}$$

$$\begin{aligned} L_{style}(C;G,R) = &\sum_{i=1}^{K} \left\|\mu\left(\phi_i(D(t))\right) - \mu\left(\phi_i(R)\right)\right\|_2 \\ &+ \sum_{i=1}^{K} \left\|\sigma\left(\phi_i(D(t))\right) - \sigma\left(\phi_i(R)\right)\right\|_2 \end{aligned} \tag{9}$$

where

$$t = AdaIN(E(G),E(R)) \tag{10}$$

$Gray(.)$ represents the conversion from RGB to grayscale image. $\phi_i(.)$ denotes a layer of a pre-trained VGG-19 model. In implementation, we use $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$ layers with equal weights to compute style loss. The weights of these items, i.e., λ_1 to λ_7 in the main paper, are 1.0, 10.0, 0.5, 1.0, 10.0, 0.1 and 0.05 respectively.

About user study. One of the evaluation metrics we use is user study, i.e., *Quality* (Table 1 in our main paper), it reflects how the generated photos are agreed with human imagination given a sketch. Specifically, for each comparison, an input sketch and its corresponding generated photos from two methods (one is the proposed method, and the other is a baseline method) are shown to a user at the same time, and then the user needs to choose which one is closer to his/her expectation. The range of the value is $[1, 100]$ while the default value for our method is set to 50. It is the ratio of cases that users prefer for the compared method. When a value is less than 50, it means that the generated photos of a baseline method are *less* favored by volunteers compared with our proposed model; otherwise means people prefer the results of the baseline method. Note that all four subjects know nothing about our work.

Another option is showing the subjects the generated photos from all **five** methods and asking them to pick the one with the highest quality. However, this metric provides less information than the above metric. For example, when some baselines are far worse than the others, all these methods have score 0. So they are treated equally poor. In contrast, the above metric we used can reflect the relevant superiority between a baseline model and the proposed model.

Unsupervised sketch-based image retrieval. Our proposed model enables direct mapping between sketches and photos. There are two possible options: **1)** Translate gallery photos to sketches, and then find the nearest sketches to the query sketch; **2)** Translate a sketch to a photo and then find its nearest neighbors in the photo gallery. Specifically, for 1), after translating the candidate photos to sketches by the proposed model, we used a ResNet18 model [3] which is pretrained on the TU-Berlin dataset to extract features from the query and the translated sketches. And then we computed the Euclidean distance to find the nearest neighbors for the query sketch. For 2), we first use the proposed model to translate a query sketch to a photo, and then compute distances of features which are extracted from the translated and the candidate photos. An ImageNet pretrained ResNet18 model is used as the feature extractors for photos.

2 Additional Qualitative Results

In this section, we provide more qualitative results (Figure 1 to Figure 5) to show the effectiveness of our model.

References

1. Canny, J.: A computational approach to edge detection. TPAMI (6), 679–698 (1986)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
4. Li, M., Lin, Z., Mech, R., Yumer, E., Ramanan, D.: Photo-sketching: Inferring contour drawings from images. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (2019)
5. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV (2015)

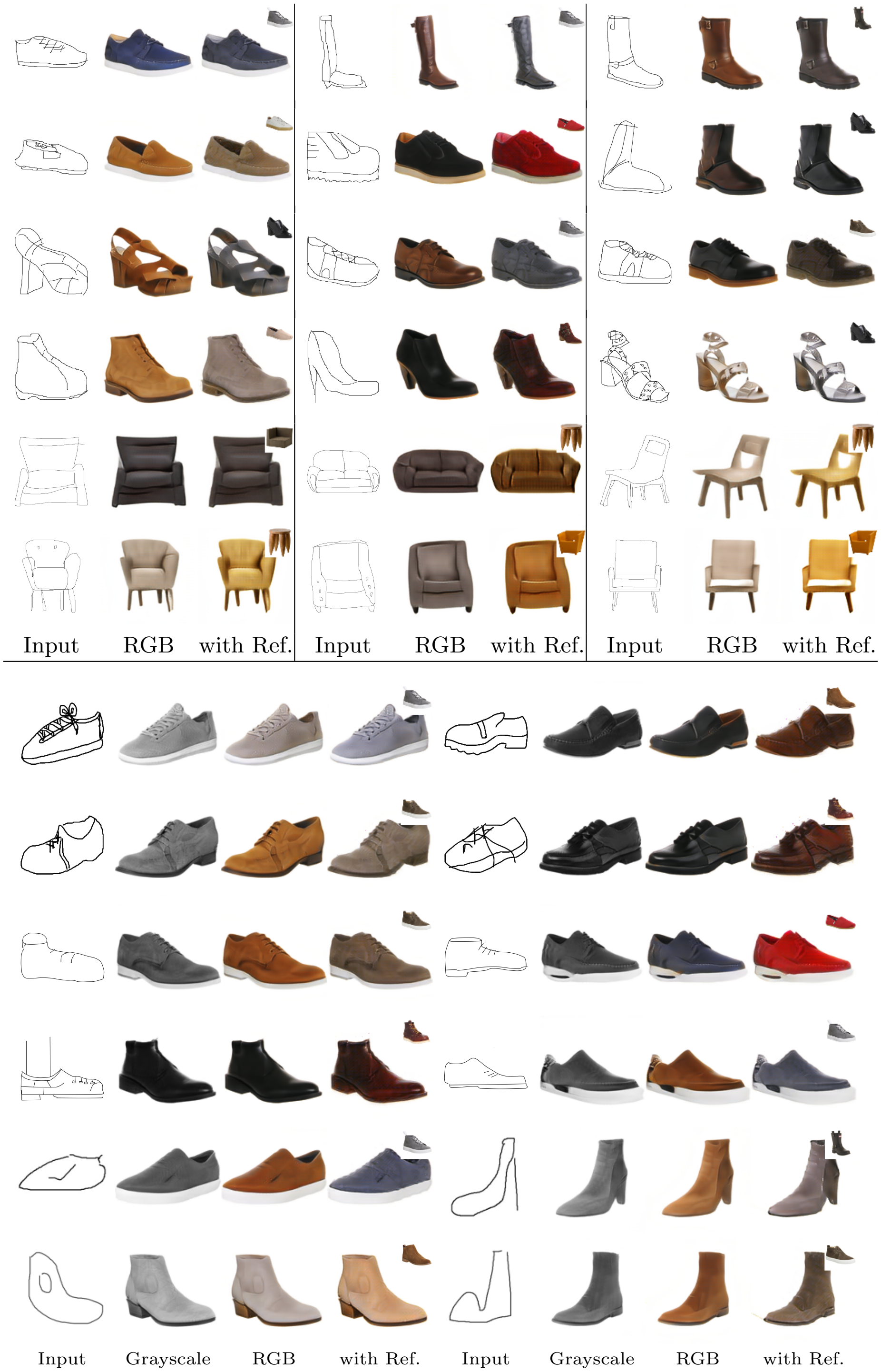

Fig. 1. Top: Synthesized results obtained by our model, with (the 3rd column) and without (the 2nd column) references. Reference images are shown at the top right corner. **Bottom:** Generalization across sketch datasets, including the *Sketchy* dataset (first two rows), the TU-Berlin dataset (middle two rows), and the *QuickDraw* dataset (last two rows). The images, from left to right, are input sketches, synthesized grayscale images, synthesized RGB photos, and RGB photos when reference images are available. Note that **1)** all these results are produced by our model trained on *ShoeV2* dataset; **2)** the model can handle sketches pointing to both sides as we applied flipping to data augmentation during training.

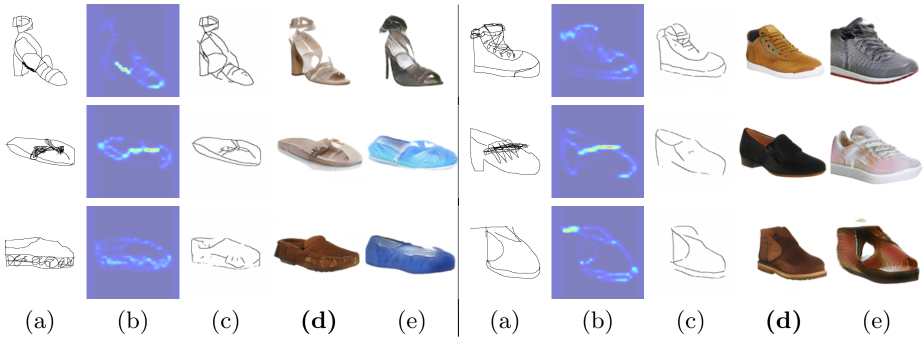

Fig. 2. Our model can deal with noise sketches. (a) are input sketches; (b)(c)(d) show learned attention masks, *reconstructed* sketches, and photos synthesized by our model. (e) are the results of UGATIT. It is clear to see that our model can handle noise sketches better than UGATIT. Besides, the disparity between (a) and (c) indicates what irrelevant noise strokes are *ignored* by our model.

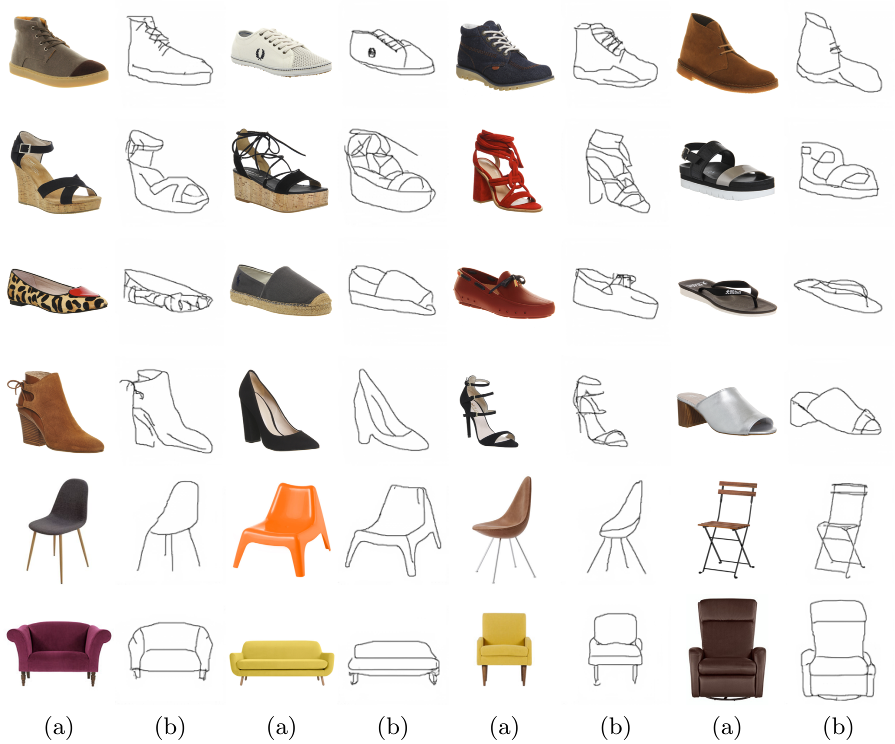

Fig. 3. Results of photo-based sketch synthesis. (a) Input photo, (b) synthesized sketch by our model. The synthesized sketches can not only reflect the distinguishing feature of original objects, but also mimic different drawing styles. For example, in the first row, *shoelace* are depicted in different styles.

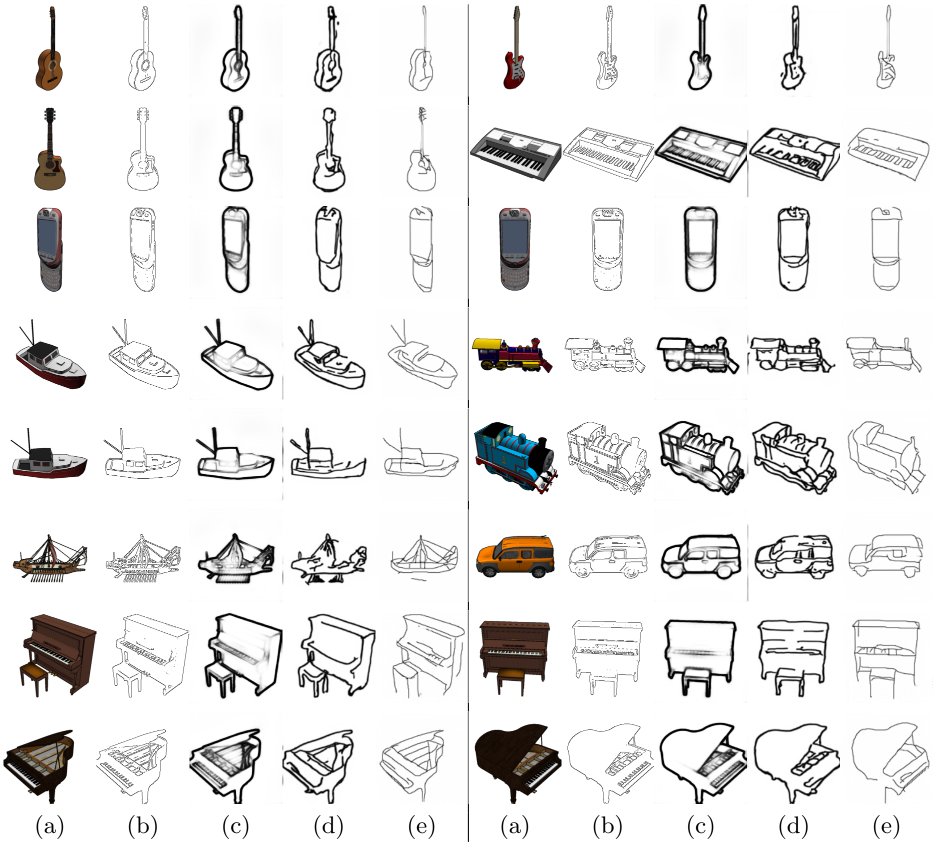

Fig. 4. Results obtained on ShapeNet [2]. (a) are input photos, (b) to (e) are lines derived by Canny [1], HED [5], Photo-Sketching [4], and our shoe model. Our model can generate lines with a hand-drawn effect, while HED and Canny detectors produce edgemaps faithful to original photos. Comparing with results of Photo-Sketching, ours are visually more similar with free-hand sketches.

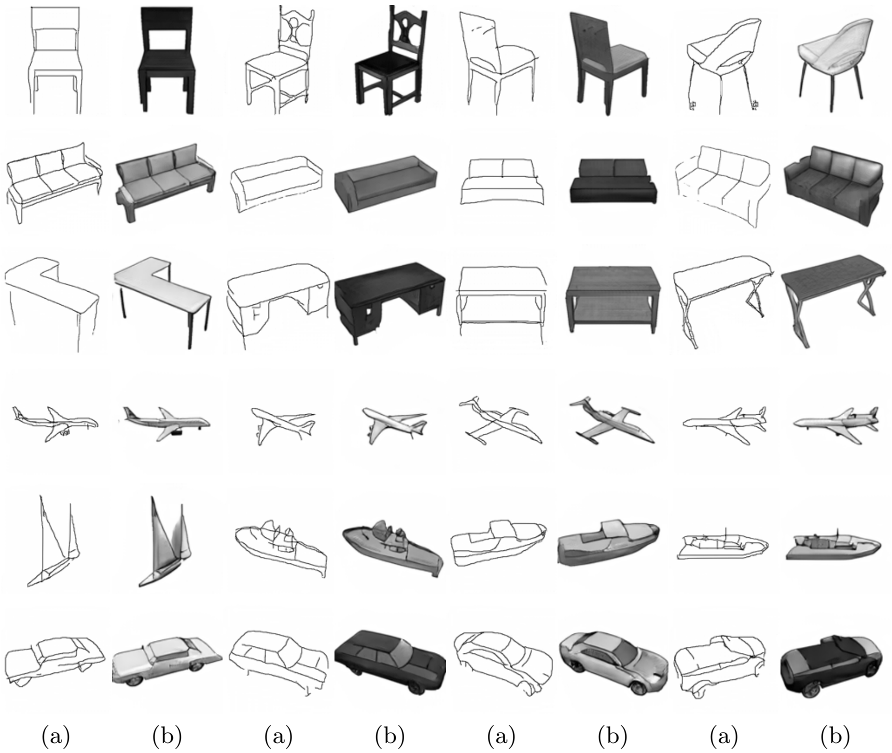

Fig. 5. Results of **multi-class sketch-to-photo synthesis** on ShapeNet dataset. Given performance achieved in the single-class setting, we wonder if our proposed model can work for multi-classes. We thus conduct experiments on ShapeNet. To be specific, we select 11 classes, each contains photos varying from 300 to 8000, and form training and testing set with 20,656 and 5,823 photos respectively. Then we generate fake *sketches* using our shoe model. Next, we train our *shape translation* network on the newly formed multi-class image set. All training settings are the same as training in a single class, and class information is **not** used during training. Resutls are displayed above. (a) are input sketches, and (b) are synthesized grayscale images. Examples in each row are from the same class. To our surprise, the model can generate photos for multiple classes, even **without** any class information. We assume that our model is capable of gaining semantic understanding during the class-agnostic training process.