Modeling Artistic Workflows for Image Generation and Editing Supplementary Materials

Hung-Yu Tseng[1], Matthew Fisher[2], Jingwan Lu[2], Yijun Li[2], Vladimir Kim[2], Ming-Hsuan Yang[1]

[1]University of California, Merced [2]Adobe Research

1 Overview

In this supplementary material, we first present the implementation details for each component of the proposed framework. Second, we complement the experiment details. Third, we visualize the learning-based regularization. Fourth, we show visual examples illustrating the failure cases of the proposed method. Finally, we present more qualitative results to complement the paper.

2 Implementation Details

We implement our framework with PyTorch [12]. The details for each component are described as follows.

Workflow inference. The hyper-parameter λ^1 in Equation 5 of the paper is assigned to be 10. We use the Adam optimizer [8] with the learning rate of 2×10^{-4} and batch size of 8 for optimizing the model. We first train each network separately with $450,000$ iterations, then jointly train all the networks in the workflow inference module with $450,000$ iterations.

Artwork generation. We set the hyper-parameter λ^c in Equation 3 of the paper to be 1. Similar to the training for the workflow inference module, we use the Adam optimizer [8] with the learning rate of 2×10^{-4} and batch size of 8. We train each network separately with $1,200,000$ iterations, then jointly train all the networks in the artwork generation module with $600,000$ iterations. We adopt the objectives in the BicycleGAN [14] approach for training the artwork generation module, as described in Equation 3 in the paper. More specifically, the loss L_i^{bicycle} in Equation 3 is formulated as

$$L_i^{\text{bicycle}} = L_i^{\text{GAN}} + \lambda^1 L^1 + \lambda^{\text{latent}} L^{\text{latent}} + \lambda^{\text{KL}} L^{\text{KL}}, \tag{1}$$

where L_i^{GAN} is the hinge version of GAN loss [3], L^1 is the $\ell 1$ loss between the generated and ground-truth images, L^{latent} is the latent regression loss between the predicted and input latent representations, and L^{KL} is the KL divergence loss on the latent representations. Following the setting in the BicycleGAN scheme, we respectively assign the hyper-parameters λ^1, λ^{latent}, and λ^{KL} to be 10, 0.5,

Algorithm 1: Training overview of the learning-based regularization at i-th stage

Require: pre-trained generation model $\{E_i^G, G_i^G\}$, learning rate η, iterations T^{reg}, importance factor λ^{GAN}
$w_i = \mathbf{0.001} \in R^{1\times 8c}$
while $t = \{1, \ldots, T^{\text{reg}}\}$ **do**
  Sample (x_i, x_{i+1}) and x_i' from the dataset
  $z_i^{\text{Ada}} = E_i^G(x_i)$, $\delta_i^{\text{Ada}} = \mathbf{0} \in R^{1\times 8c}$
  // **Get reconstructed image before the AdaIN optimization**
  $\hat{x}_{i+1}^G = G_i^G(x_i, z_i^{\text{Ada}} + \delta_i^{\text{Ada}})$
  // **Optimize incremental term with the regularization function (AdaIN optimization)**
  $\tilde{\delta}_i^{\text{Ada}} = \delta_i^{\text{Ada}} - \alpha \left(\bigtriangledown_{\delta_i^{\text{Ada}}} L_{\text{Ada}}(\hat{x}_{i+1}^G, x_{i+1}) + w_i \delta_i^{\text{Ada}} \right)$
  // **Get the reconstructed image and editing results after the optimization**
  $\tilde{x}_{i+1}^G = G_i^G(x_i, z_i^{\text{Ada}} + \tilde{\delta}_i^{\text{Ada}})$
  $\tilde{x'}_{i+1}^G = G_i^G(x_i', z_i^{\text{Ada}} + \tilde{\delta}_i^{\text{Ada}})$
  // **Update the regularization function based on the reconstruction and editing results after the optimization**
  $L^{\text{L2R}} = L^{\text{Ada}}(\tilde{x}_{i+1}^G, x_{i+1}) + \lambda^{\text{GAN}} L^{\text{GAN}}(\tilde{x'}_{i+1}^G)$
  $w_i = w_i - \eta \bigtriangledown_{w_i} L^{\text{L2R}}$
end
Return: w_i

Table 1. FID scores of real images. We show the FID (↓) scores of the real images in the test set to supplement the results in Table 2 and Table 3 of the paper.

Datasets	Face	Anime	Chair
Real images	12.8	16.5	25.3

and 0.01. We use the network architecture proposed in the MUNIT [7] framework (involving AdaIN normalization layers [6]) rather than the U-Net structure in the BicycleGAN framework.

AdaIN optimization. In the editing scenario during the testing phase, we conduct the AdaIN optimization from the first to the last stages sequentially to refine the reconstructed image. For each stage, we set the hyper-parameters λ^p, α, T in Algorithm 1 in the paper to be 10, 0.1 and 150, respectively.

Learning-based regularization. We summarize the training of the proposed learning-based regularization in Figure 4 of the paper and Algorithm 1. The regularization function is trained separately for each creation stage. We respectively set the hyper-parameters η, T^{reg}, and λ^{GAN} to be 10^{-3}, 40000, and 1. We use the Adam optimizer [8] and the batch size of 1 for the training.

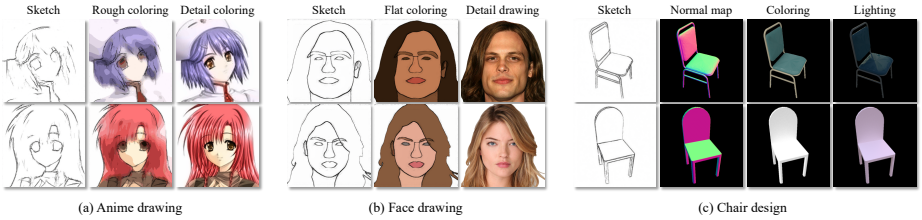

Fig. 1. Training examples in each dataset. For each dataset, we show the example training images at each creation stage.

3 Experiment Details

We illustrate how we process each dataset for evaluating the proposed framework. Example training images in each dataset are shown in Figure 1. In addition, we also describe how we compute FID [5] score.

Face drawing dataset. We collect the photo-realistic face images from the CelebAMask-HQ dataset [9]. We prepare three design stages for the face drawing dataset: sketch, flat coloring, and detail drawing. We use the ground-truth attribute segmentation mask to remove the background of the cropped RGB images in the CelebAMask-HQ dataset as the final-stage images. For the flat coloring, we assign pixels with the median color computed from the corresponding region according to the ground-truth attribute segmentation mask. Finally, we use the pencil sketch [10] model to generate simple sketch images from the flat coloring images.

Anime drawing dataset. We construct the dataset from the anime images in the EdgeConnect [11] dataset. Three stages are used in this dataset: sketch, rough coloring, detail coloring. For rough coloring, we first apply the SLIC [1] super-pixel approach to cluster the pixels in each anime image. For each cluster, We then compute the median color and assign to the pixels in that cluster. Finally, we adopt the median filter to smooth the rough coloring images. As for the sketch, we use the pencil sketch [10] scheme to extract the sketch image from the original anime image.

Chair design. We render the chair models in the ShapeNet dataset [4] via the photo-realistic renderer [2] for building the dataset. There are four stages presented in this dataset: sketch, normal map, coloring, and lighting. We sample two different camera viewpoints for each chair model. For each viewpoint, we randomly sample from 300 spherical environment maps of diverse indoor and outdoor scenes to render the last-stage image. For the coloring image, we use a default white lighting environment for the rendering. We configure the rendering tools to produce the corresponding depth map for each viewpoint and infer the normal map image from the depth map. Finally, we extract the sketch image from the normal map image using the pencil sketch model [10].

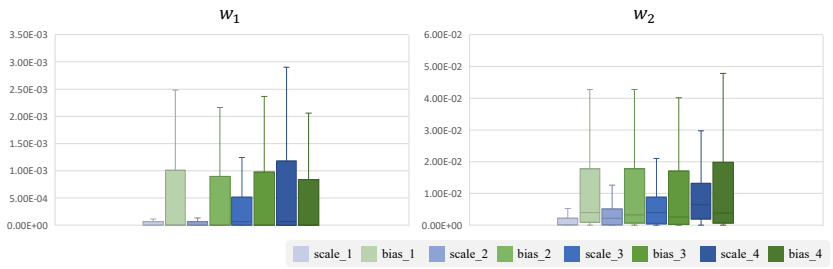

Fig. 2. Visualization of the proposed learning-based regularization. We show the quartile visualization of the hyper-parameter w_i for our learning-based regularization approach trained on the face drawing dataset. The learned function tends to have stronger regularization on the bias terms.

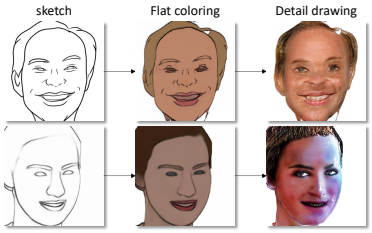

Fig. 3. Failure cases. Our framework fails to generate appealing results (*top*) if the style of the input sketch image is significantly different from that of training images, and (*bottom*) if we use extreme latent representations which are out of the prior distributions we sampled from during the training phase.

FID scores. We use the official implementation to compute the FID [5] scores.[1] For all experiments, we use the generated images from the whole test set as well as the real images in the training set. Since we need to re-sample the latent representations for the editing experiments presented in Table 3 in the paper, we conduct 5 trials for each experiment and report the average results. Moreover, we show the FID scores of real images in the test set in Table 1. The scores reported in this table can be considered as the lower-bound scores for each task.

4 Additional Experimental Results

Visualizing learning-based regularization. To better understand our learning-based regularization function, we visualize the learned hyper-parameter w_i of the weight decay regularization described in Section 3.3 in the paper and Algorithm 1. The value of the hyper-parameter indicates the strength of the regularization for the AdaIN optimization process. Figure 2 shows the visualization of

[1] https://github.com/bioinf-jku/TTUR

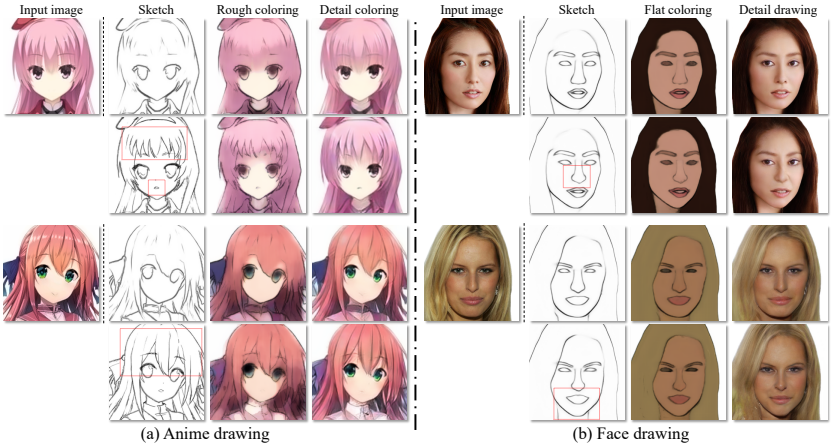

Fig. 4. Results of artistic editing. Given an input artwork image, we ask the artist to edit the inferred sketch image. The synthesis model then produces the corresponding edited artwork. The first row shows the input artwork and inferred images, and the red outlines indicate the edited regions.

the hyper-parameters trained on the face drawing dataset. In general, The regularization on the bias terms is stronger than that on the scaling terms. Since the goal is to minimize the appearance distance between the reconstructed and original input image, the AdaIN optimization tends to complement such discrepancy with the bias terms. However, as shown in Figure 8 in the paper, such optimization may lead the bias terms to extreme values and make the generation model sensitive to the change (i.e., editing) of the input image. The proposed learning-based regularization mitigates the problem by applying stronger regularization on the bias terms, thus encourage the optimization process to modify the scaling terms. Quantitative results shown in Section 4.3 in the paper validate that the proposed learning-based regularization improves the quality of the editing results.

Failure cases. We observe several failure cases of the proposed framework, which are presented in Figure 3. First, if the style of the input image is significantly different from the training data, the artwork generation module fails to produce appealing results. Similar to the Scribbler [13] approach, we argue that such a problem may be alleviated by diversifying the style of the training images. Second, during the creation process, the generation module synthesizes results with artifacts if we use extreme latent representations that are out of the prior distributions we sampled from during the training phase.

Qualitative results. We show more editing results conducted by the artists in Figure 4.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. TPAMI **34**(11), 2274–2282 (2012) 3
2. Adobe: Adobe dimension. https://www.adobe.com/products/dimension.html (2019) 3
3. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: ICLR (2019) 1
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015) 3
5. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS (2017) 3, 4
6. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV (2017) 2
7. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV (2018) 2
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 1, 2
9. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: CVPR (2020) 3
10. Li, Y., Fang, C., Hertzmann, A., Shechtman, E., Yang, M.H.: Im2pencil: Controllable pencil illustration from photographs. In: CVPR (2019) 3
11. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019) 3
12. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS workshop (2017) 1
13. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: CVPR (2017) 5
14. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: NIPS (2017) 1