Real-Time Data-Efficient Portrait Stylization via Geometric Alignment, Supplementary Materials

Xinrui Wang, Zhuoru Li, Xuanyu Yin, Xiao Zhou, Yusuke Iwasawa, Yutaka Matsuo and Jiaxian Guo

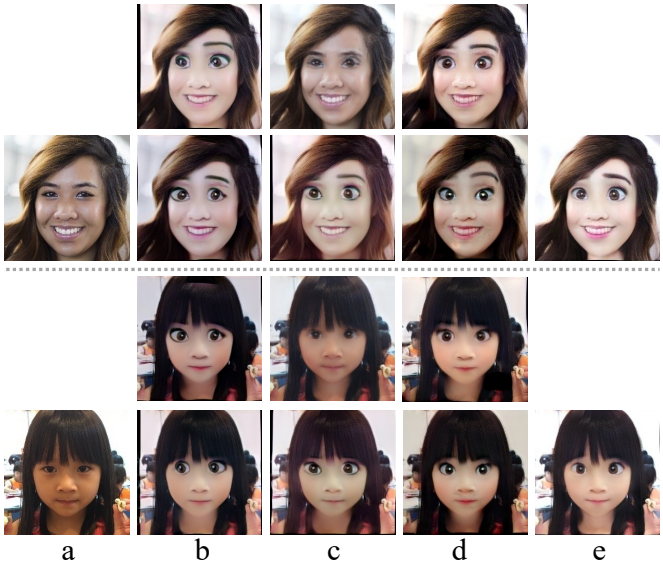

Fig. 1: Ablation study of hyper-parameters. Column a shows portrait images, column b shows results of tuning λ_1, column c shows results of tuning λ_2, column d shows results of tuning λ_3, and column shows results with default settings. Upper row shows results of reducing the hyper-parameter and lower row shows results of decreasing the hyper-parameter.

I. Ablation Study for Hyper-parameters

We conducted an ablation study to explore the optimal hyper-parameters. To balance the influence of each loss term, we set the default value of λ_1, λ_2 and λ_3 to be 2, 10, and 10 respectively, based on our analysis of the effect and numerical scale of each loss term. To explore the influence of λ_1, λ_2 and λ_3, we designed a grid search on all three hyper-parameters and train 7 different models, with one parameter changed and the rest two fixed. The experiments are conducted on animation dataset with the large generator model, and the performance is evaluated by Art-FID. We show the quantitative results in Table I and qualitative results in Fig 1, where we can see the model with default hyper-parameters achieves the best performance, while increasing or decreasing the weight of each loss term will result in performance drop. This proves the chosen hyper-parameters are optimal for our model design.

Corresponding author: Xinrui Wang.
E-mail: secret_wang@weblab.t.u-tokyo.ac.jp.
Xinrui Wang, Yusuke Iwasawa, Yutaka Matsuo and Jiaxian Guo are with The University of Tokyo, Tokyo, Japan.
Zhuoru Li is with Project HAT, Shenzhen, Guangdong, China.
Xuanyu Yin is with Meituan.Inc, Beijing, China.
Xiao Zhou is with Hefei Normal University, Hefei, Anhui, China.

TABLE I: Ablation study of hyper-parameters evaluated by Art-FID

	$\times 0.1$	Default	$\times 10$
λ_1	91.27	78.36	88.61
λ_2	105.38	78.36	117.96
λ_3	89.64	78.36	86.70

II. Network Architecture

Here we denote the generator network as encoder and decoder to simplify the description. Let $K7S2C64$ denotes a convolution layer with 7×7 kernel size, 64 filters and stride 2, $K7U2C64$ denotes a convolution layer with 7×7 kernel size, 64 filters and a bilinear interpolation for 2 times upsample, and $9 \times R256$ represents 9 stacked residual block with each consisting of two convolution layer with 3 ×3 kernel size and 256 filters. then the architecture of the proposed generator and discriminator are illustrated as below:

Large Model:

- Encoder: $K7S2C32$, $K3S1C32$, $K3S2C64$, $K3S1C64$, $K3S2C128$, $K3S1C128$
- Decoder: $6 \times R128$, $K3S1C128$, $K3U2C64$, $K3S1C64$, $K3U2C32$, $K3S1C32$, $K7S2C3$
- Discriminator: $K4S2C32$, $K4S2C64$, $K4S2C128$, $K4S1C128$, $K1S1C1$

Small Model:

- Encoder: $K7S2C16$, $K3S1C16$, $K3S2C32$, $K3S1C32$, $K3S2C64$, $K3S1C64$
- Decoder: $4 \times R64$, $K3S1C64$, $K3U2C32$, $K3S1C32$, $K3S2C16$, $K3S1C16$, $K7S2C3$
- Discriminator: $K4S2C16$, $K4S2C32$, $K4S2C64$, $K4S1C64$, $K1S1C1$

All the convolution layers in the generator except the last one are followed by an Instance Normalization layer [1] and a ReLU activation layer [2], and all the convolution layers in the discriminator except the last one are followed by a Leaky ReLU activation layer. The last layer of generator and discriminator directly return the result of convolution operation.

III. Images Shown in User Study

In the supplementary material, we present the user interface of the user study and the images shown to people surveyed in the user study. For each style, we prepare 16 groups of images and randomly show 8 of them to the participant. In Fig 2, We show the user interface for the user study. In Fig 3 to Fig 10, from left to right, we show portrait image and the results

of CycleGAN [3], AgileGAN [4], UGATIT [5], DRIT++ [6], SCGAN [7], CocosNet V2 [8], DCTNet [9], StableDiffusion [10] with ControlNet [11] and LoRA [12], and our proposed model respectively.

IV. Results of Stylized Video Frames

We also show addition results of stylized video frame sequences, and compare our proposed method with previous methods. Considering the results of user study, we only compare with 4 previous methods that achieved relatively higher preference by users investigated: CycleGAN, SCGAN, DCTNet and StableDiffusion with ControlNet and LoRA. In Fig 11 to Fig 14, from top to bottom, we show the original portrait photo and the results of CycleGAN, DCTNet, SCGAN, StableDiffusion with LoRA, and our proposed method respectively.

References

[1] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.
[2] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Icml, 2010.
[3] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of IEEE International Conference on Computer Vision, 2017.
[4] G. Song, L. Luo, J. Liu, W.-C. Ma, C. Lai, C. Zheng, T.-J. Cham, Agilegan: stylizing portraits by inversion-consistent transfer learning, ACM Transactions on Graphics (TOG) 40 (4) (2021) 1–13.
[5] J. Kim, M. Kim, H. Kang, K. Lee, U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation, arXiv preprint arXiv:1907.10830 (2019).
[6] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, M.-H. Yang, Drit++: Diverse image-to-image translation via disentangled representations, International Journal of Computer Vision 128 (10) (2020) 2402–2417.
[7] H. Deng, C. Han, H. Cai, G. Han, S. He, Spatially-invariant style-codes controlled makeup transfer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6549–6557.
[8] X. Zhou, B. Zhang, T. Zhang, P. Zhang, J. Bao, D. Chen, Z. Zhang, F. Wen, Cocosnet v2: Full-resolution correspondence learning for image translation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11465–11475.
[9] Y. Men, Y. Yao, M. Cui, Z. Lian, X. Xie, Dct-net: domain-calibrated translation for portrait stylization, ACM Transactions on Graphics (TOG) 41 (4) (2022) 1–9.
[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
[11] L. Zhang, M. Agrawala, Adding conditional control to text-to-image diffusion models, arXiv preprint arXiv:2302.05543 (2023).
[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

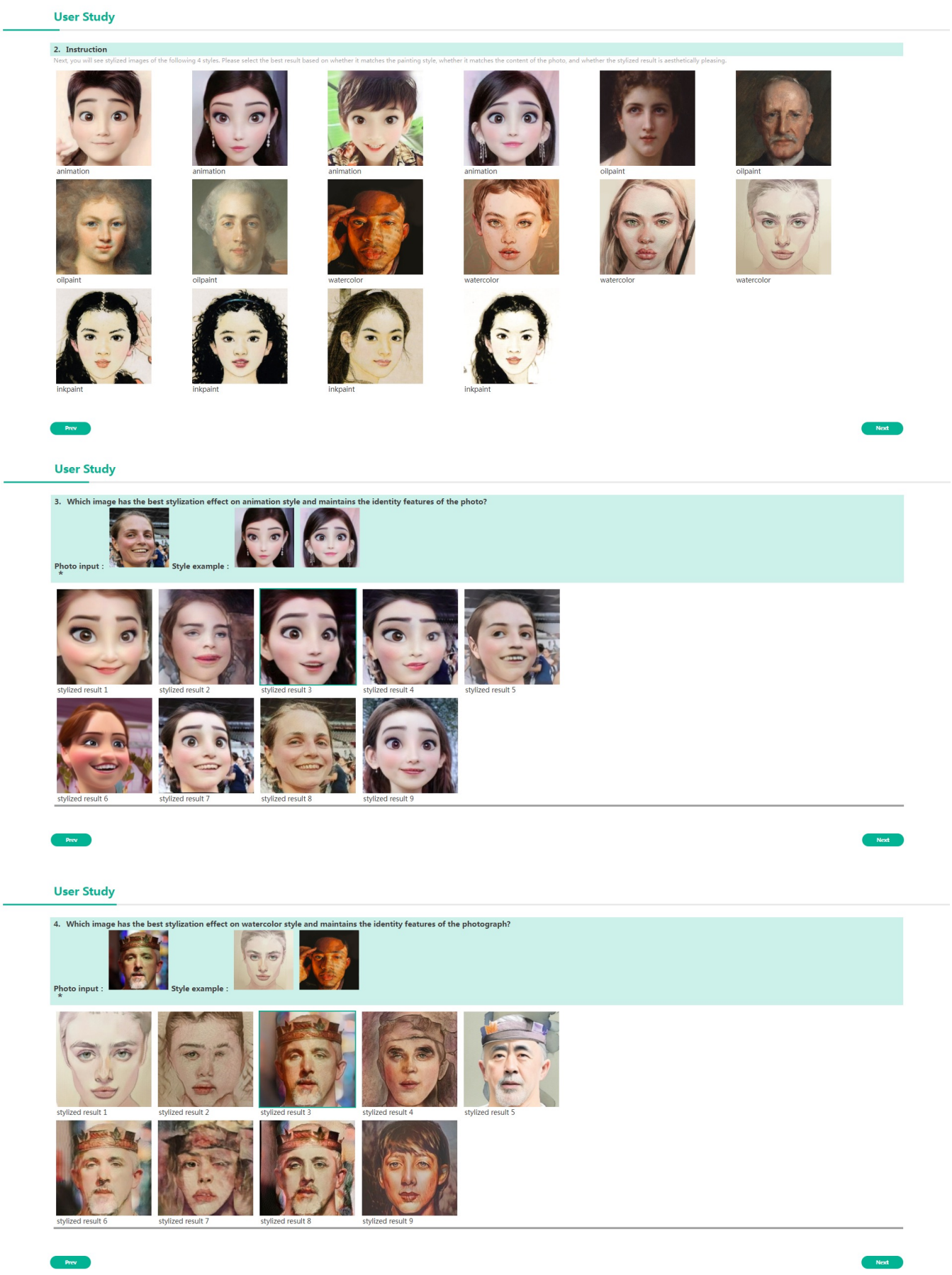

Fig. 2: User Interface of the user study.

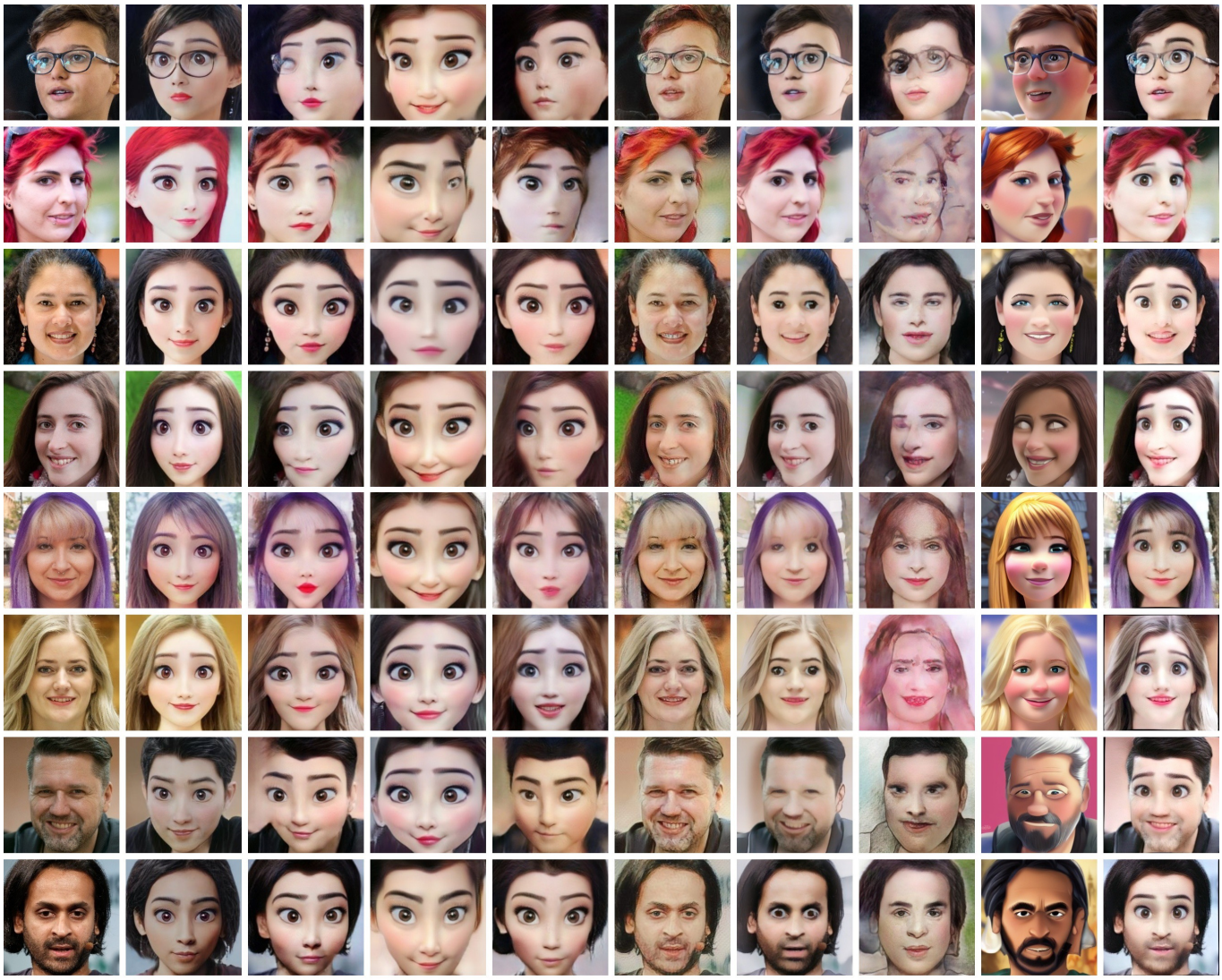

Fig. 3: Results of animation style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

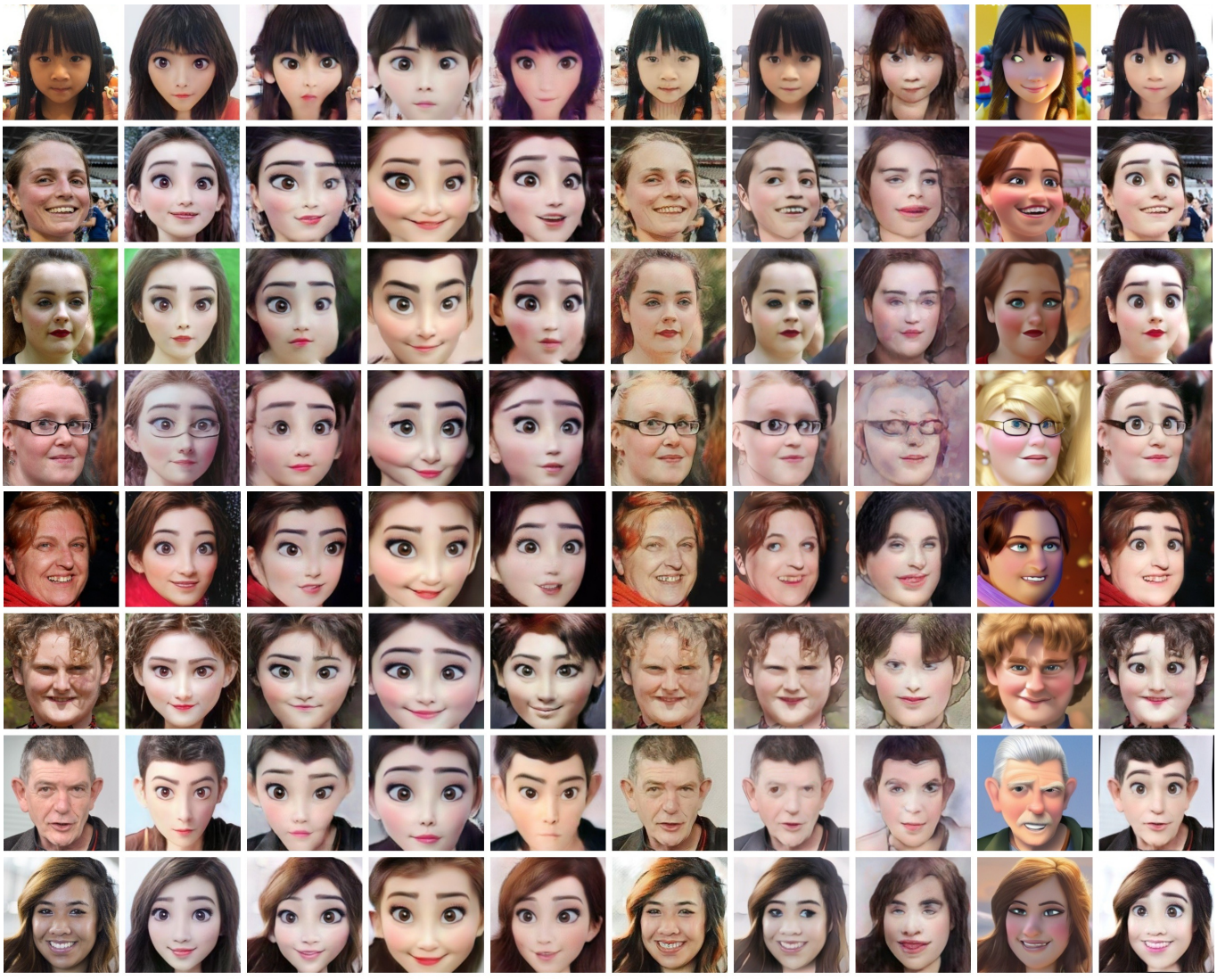

Fig. 4: Results of animation style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

Fig. 5: Results of ink paint style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

Fig. 6: Results of ink paint style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

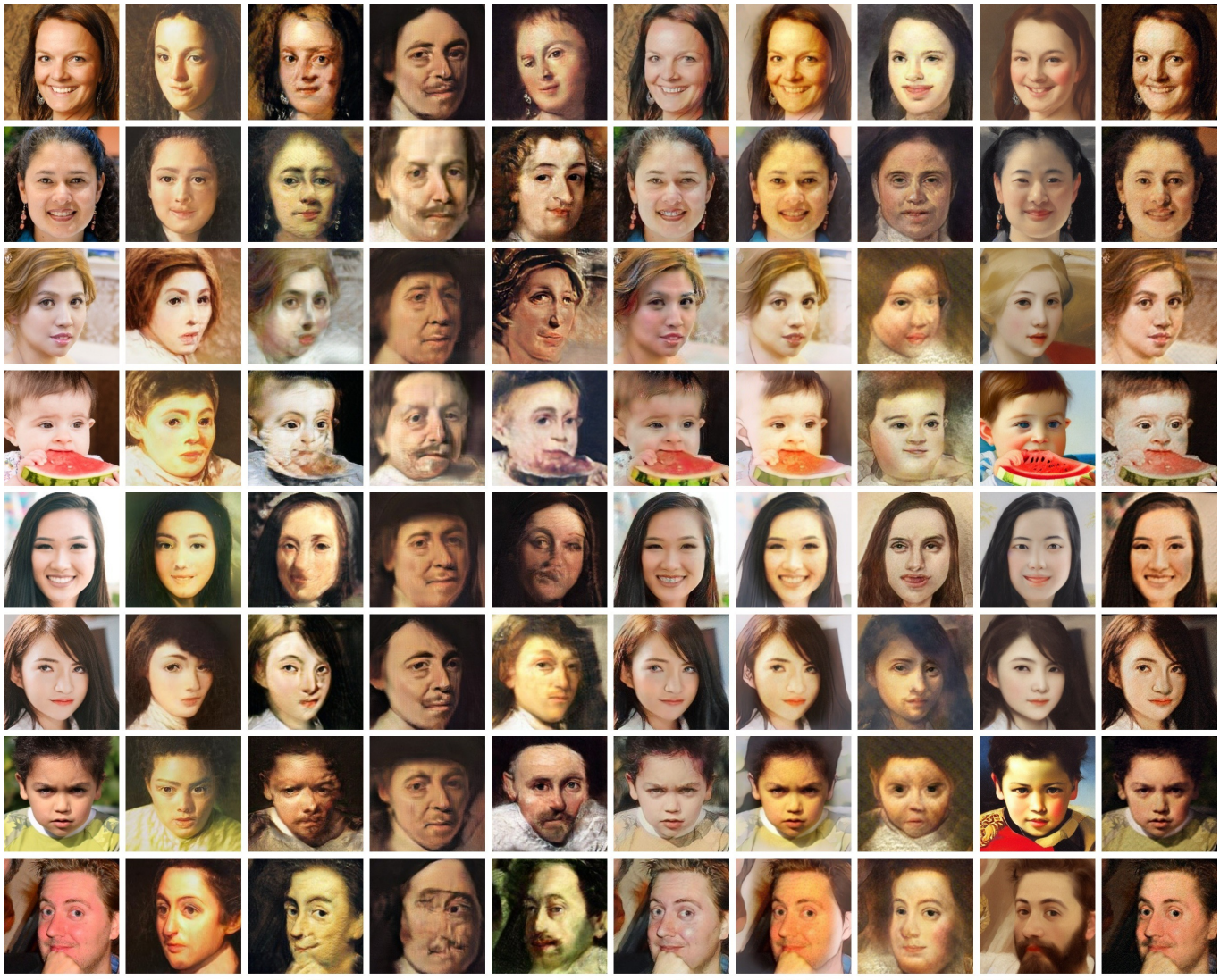

Fig. 7: Results of oil paint style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

Fig. 8: Results of oil paint style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

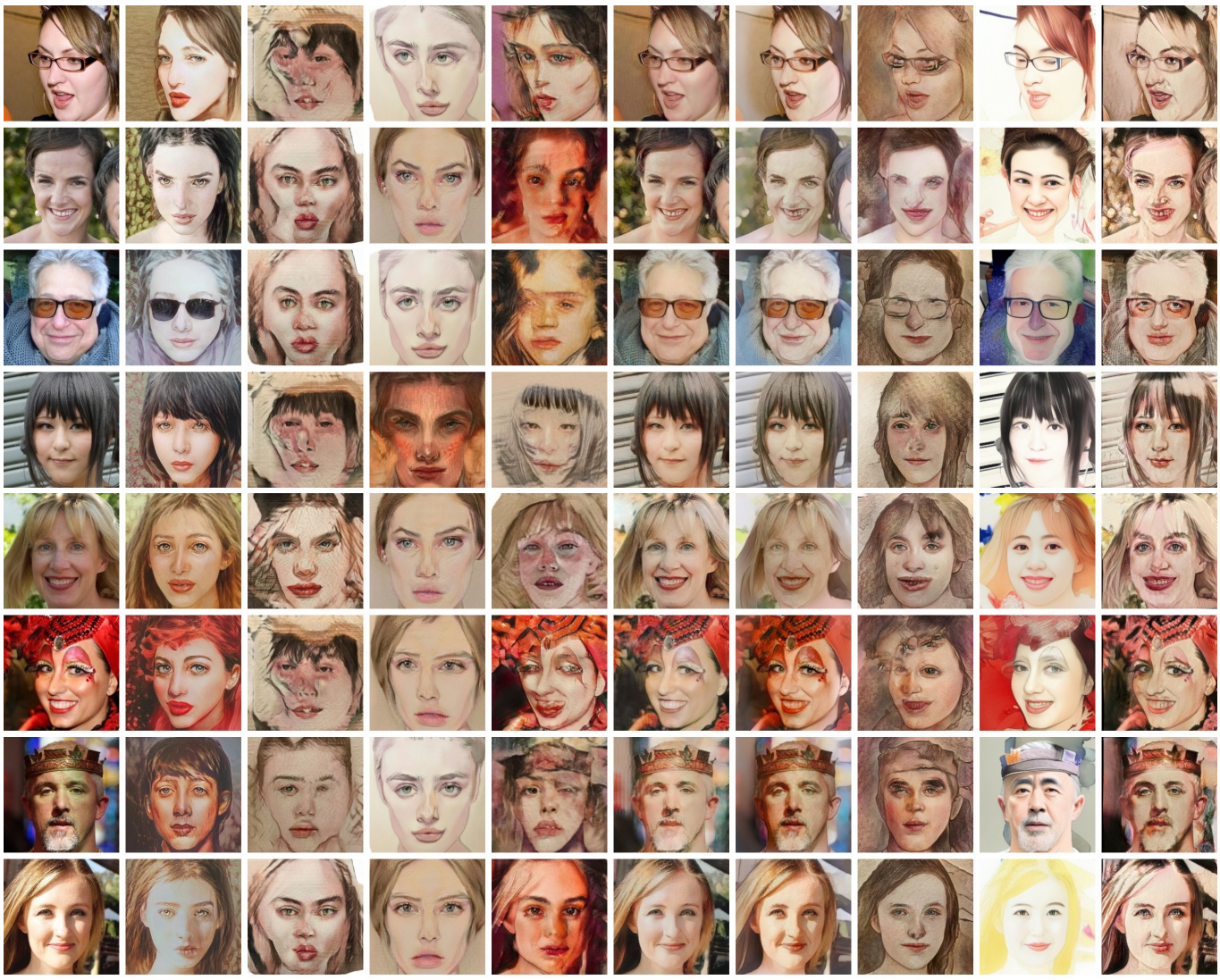

Fig. 9: Results of water color style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

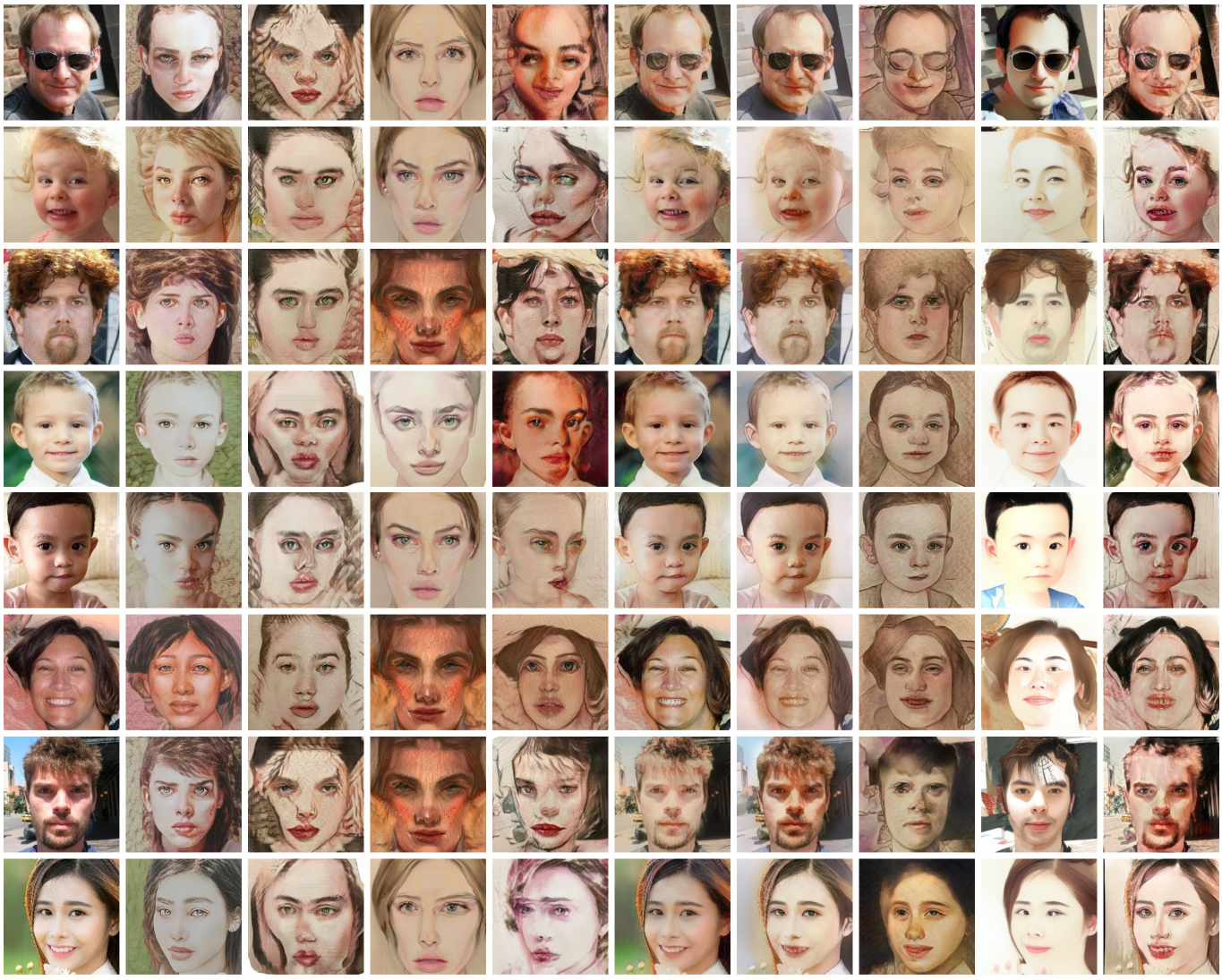

Fig. 10: Results of water color style shown in the User Study. From left to right we show the portrait image and the results of CycleGAN, AgileGAN, UGATIT, DRIT++, SCGAN, CocosNet V2, DCTNet, LDM with LoRA, and our proposed model.

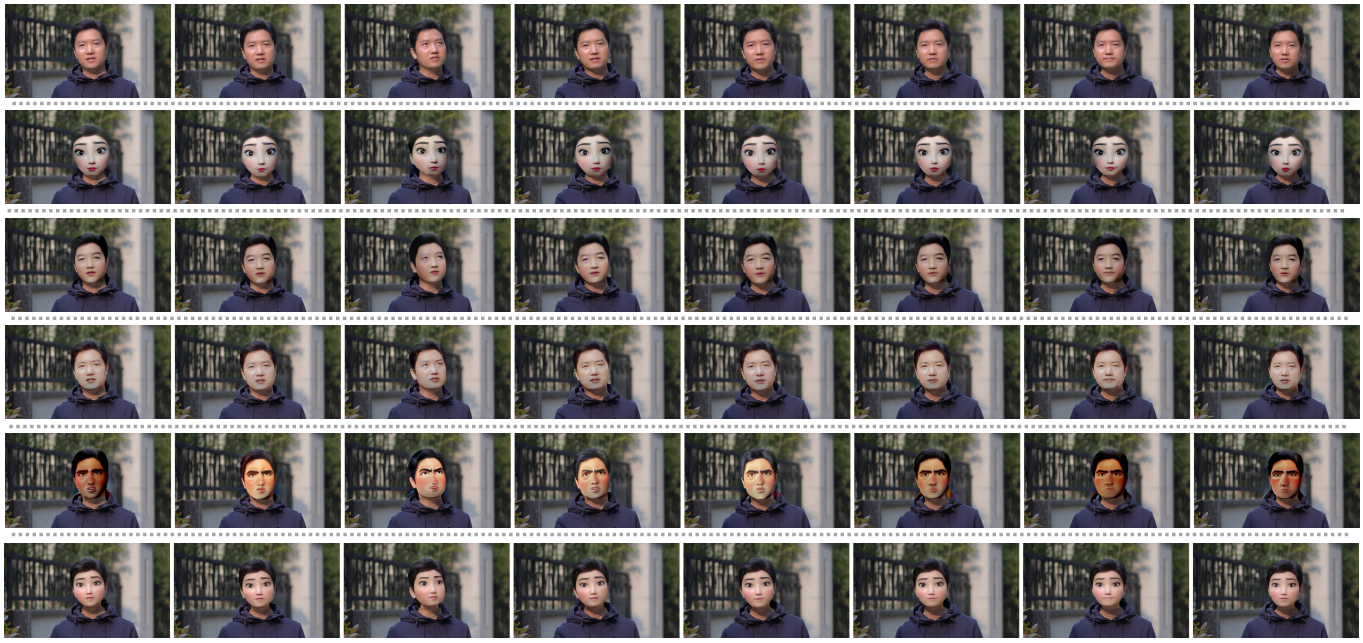

Fig. 11: Comparison of stylized video frame sequences between previous methods and our method. From top to bottom we show the input frames and the results of CycleGAN, DCTNet, SCGAN, LDM with LoRA, and our proposed model.

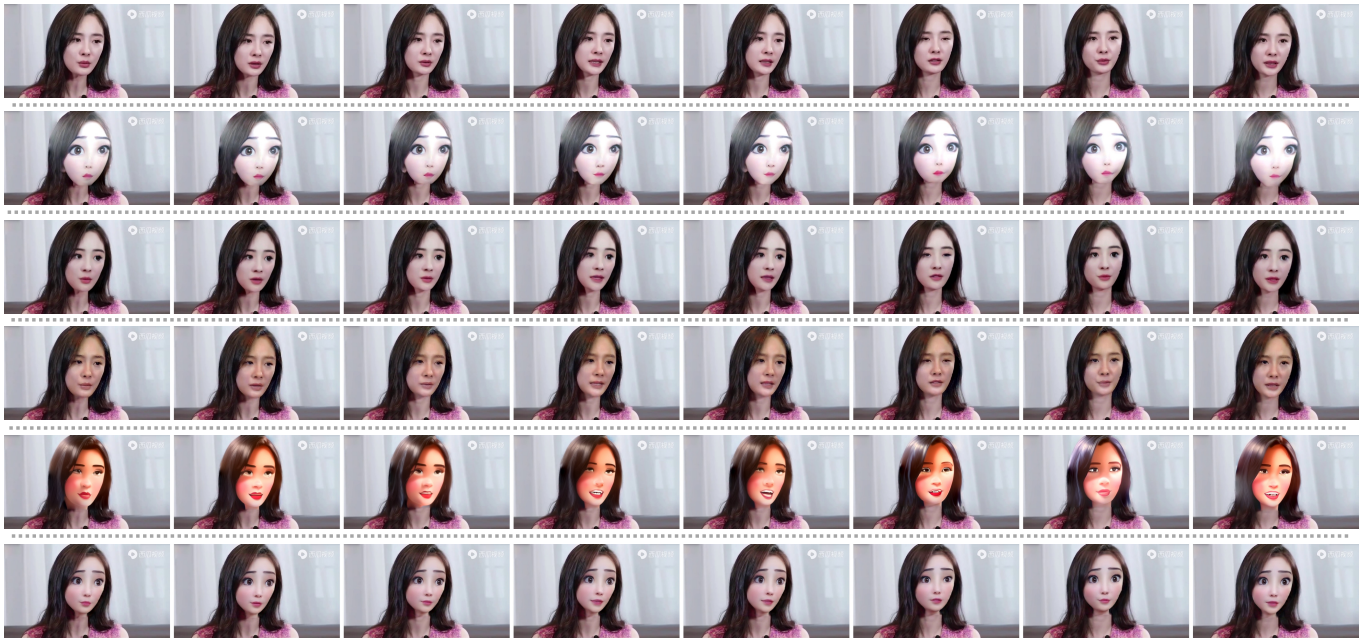

Fig. 12: Comparison of stylized video frame sequences between previous methods and our method. From top to bottom we show the input frames and the results of CycleGAN, DCTNet, SCGAN, LDM with LoRA, and our proposed model.

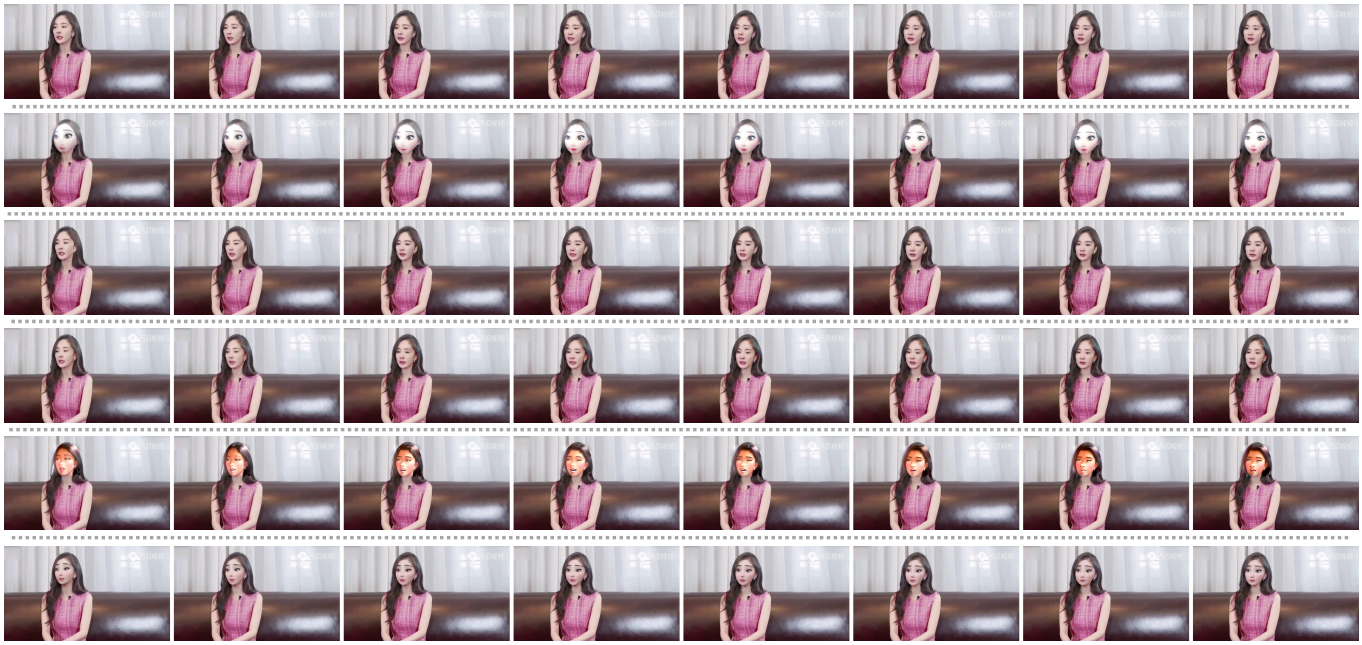

Fig. 13: Comparison of stylized video frame sequences between previous methods and our method. From top to bottom we show the input frames and the results of CycleGAN, DCTNet, SCGAN, LDM with LoRA, and our proposed model.

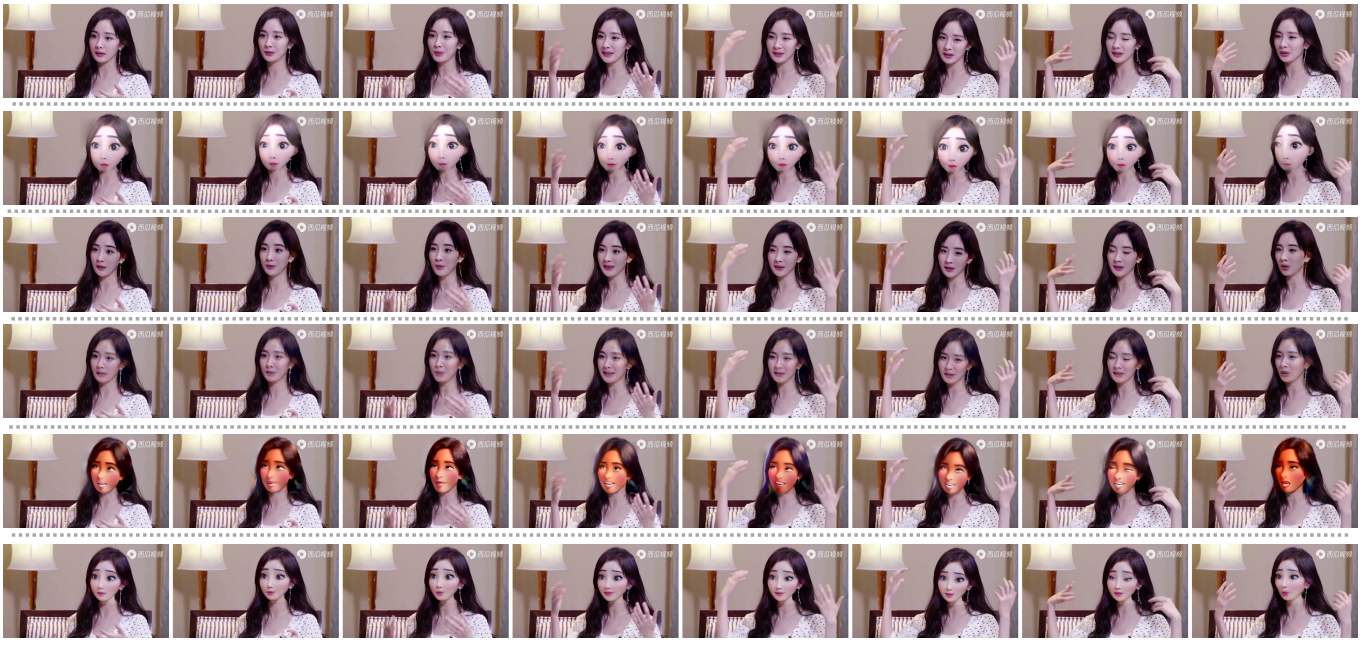

Fig. 14: Comparison of stylized video frame sequences between previous methods and our method. From top to bottom we show the input frames and the results of CycleGAN, DCTNet, SCGAN, LDM with LoRA, and our proposed model.