Supplementary Materials:
Towards Vivid and Diverse Image Colorization with Generative Color Prior

Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, Ying Shan
Applied Research Center (ARC), Tencent PCG
{yanzewu,xintaowang,ianyli,honlanzhang,emmaxunzhao,yingsshan}@tencent.com

Abstract

This supplementary material includes:

- GAN inversion examples of our method in Sec. 1.
- Ablation study on GAN inversion losses in Sec. 2.
- Comparisons with SOTA GAN inversion method DGP [2] in Sec. 3.
- More diverse colorization results in Sec. 4
- User study results in Sec. 5.

1. GAN Inversion Results

We show the GAN inversion results in Fig. 1. It is observed that the GAN inversion could generate colorful images that share the similar semantic contents to the gray inputs.

Figure 1: Visual results of BigGAN inversion. It is observed that the GAN inversion could generate colorful images that share the similar semantic contents to the gray inputs. **Left Column**: input gray-scale images. **Middle Column**: the GAN inversion results. **Right Column**: ground-truth colorful images.

2. Ablation Study of BigGAN Inversion

Despite great success of GAN inversion in StyleGAN[7], optimizing an encoder-based BigGAN inversion model is still a challenging task [1, 2]. Simply optimizing the latent code z leads to large L_2 norm of latent code z and low-quality inverted images. We conjecture that this is because the self-attention architecture in BigGAN brings optimization challenges. To tackle this issue, we directly add an extra L_2 norm penalty to the latent code and empirically found that it could alleviate this issue and generate plausible inversion results. Besides, We also employ the discriminator feature loss $\mathcal{L}_{inv\_ftr}$ instead of the perceptual loss trained on classification task, since the feature space of pretrained discriminator is more coherent with that of inversion images. Recently important progress has been achieved in StyleGAN inversion task, some useful tricks are introduced to generate high-quality inversion images. However, not all of those useful tricks are suitable for BigGAN, *e.g.*, we found in practice that the domain-guided encoder introduced in [7] will not bring benefits in BigGAN. Hence, the inversion of the encoder-based BigGAN still remains a challenge.

In addition to the discriminator feature loss and L_2 norm scheme, we also make some attempts to improve BigGAN inversion but their results are no better than the aforementioned method. Those attempts include adopting more discriminator layers to calculate the feature loss and introducing pixel loss. The comparisons of these attempts can be found in Fig. 2. It is observed that neither adding more discriminator layers nor introducing the pixel loss will generate better inversion images, but result in blurry images with fewer details.

Input | GT | L_{dis7} | L_{dis3} | $L_{dis3} + L_{pix}$

Figure 2: Ablation study of loss functions in BigGAN inversion. $\mathcal{L}_{dis7}$ denotes using more discriminator layers (*i.e.*, the last 7 layers) to calculate the feature loss. $\mathcal{L}_{dis3}$ denotes using the last 3 discriminator layers to calculate the feature loss. $\mathcal{L}_{dis3} + \mathcal{L}_{pix}$ denotes using the last 3 discriminator layers and pixel loss. It is observed that neither adding more discriminator layers nor introducing the pixel loss will generate better inversion images, but result in blurry images with fewer details. Therefore, we choose $\mathcal{L}_{dis3}$ as our default setting.

3. Comparison with GAN Inversion

As GAN inversions also incorporate pretrained GANs as a prior to guide the colorization process, we further compare the results of our method against results of a GAN inversion method - DGP [2]. Selected representative cases are presented in Figure 3 for a qualitative comparison. As shown in the figure, we can observe that color boundaries are not clearly separated in the results of DGP. The blurred color boundaries produced by DGP are inevitable considering the information loss during GAN inversion process. The low-dimensional latent codes generated in DGP fail to encode the spatial information in the image, thereby DGP is neither capable of retaining the local details nor preserving the shape of objects. On the other hand, our method, which employs a SPADE modulation to incorporate the spatial features, has obtained better results in terms of texture faithfulness. Apart from improving colorization quality, our method also reduces the computational cost by adopting straightforward feed-forward inference to avoid iterative optimization training process used in DGP.

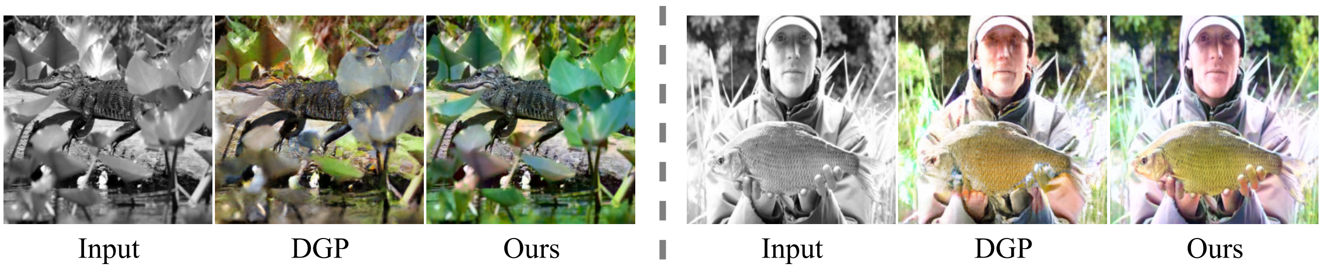

Figure 3: Comparisons with DGP.

4. More Diverse Colorization Results

In this section, we provide comparisons of diverse colorization with stochastic sampling based method PIC [3]. PIC adopts an autoregressive network PixelCNN to model the full joint distribution of pixel color values, by sampling from this distribution, diverse colorizations can be obtained. As shown in Fig. 4 and Fig. 5, PIC usually generates **spatially incoherent** and **unrealistic** colorization results. On the contrary, our method does not suffer from these issues. Besides, our method could attain smooth and controllable colorizations.

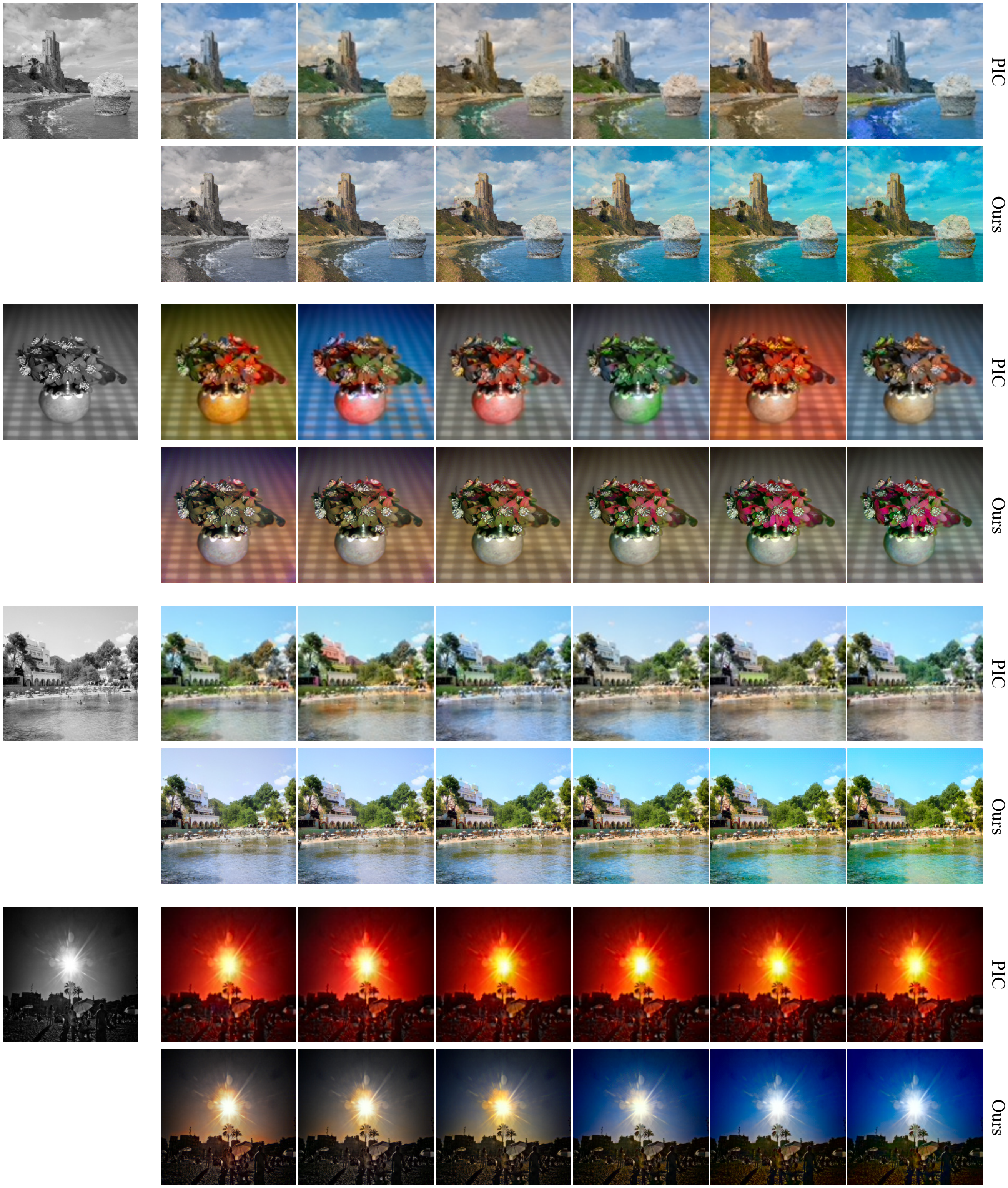

Figure 4: Comparisons of diverse colorization with PIC (**First row**: PIC; **Second row**: Ours). Zoom in to see the spatially incoherent and unrealistic colorizations from PIC. Our method could generate spatially coherent and more natural colorization results with smooth control.

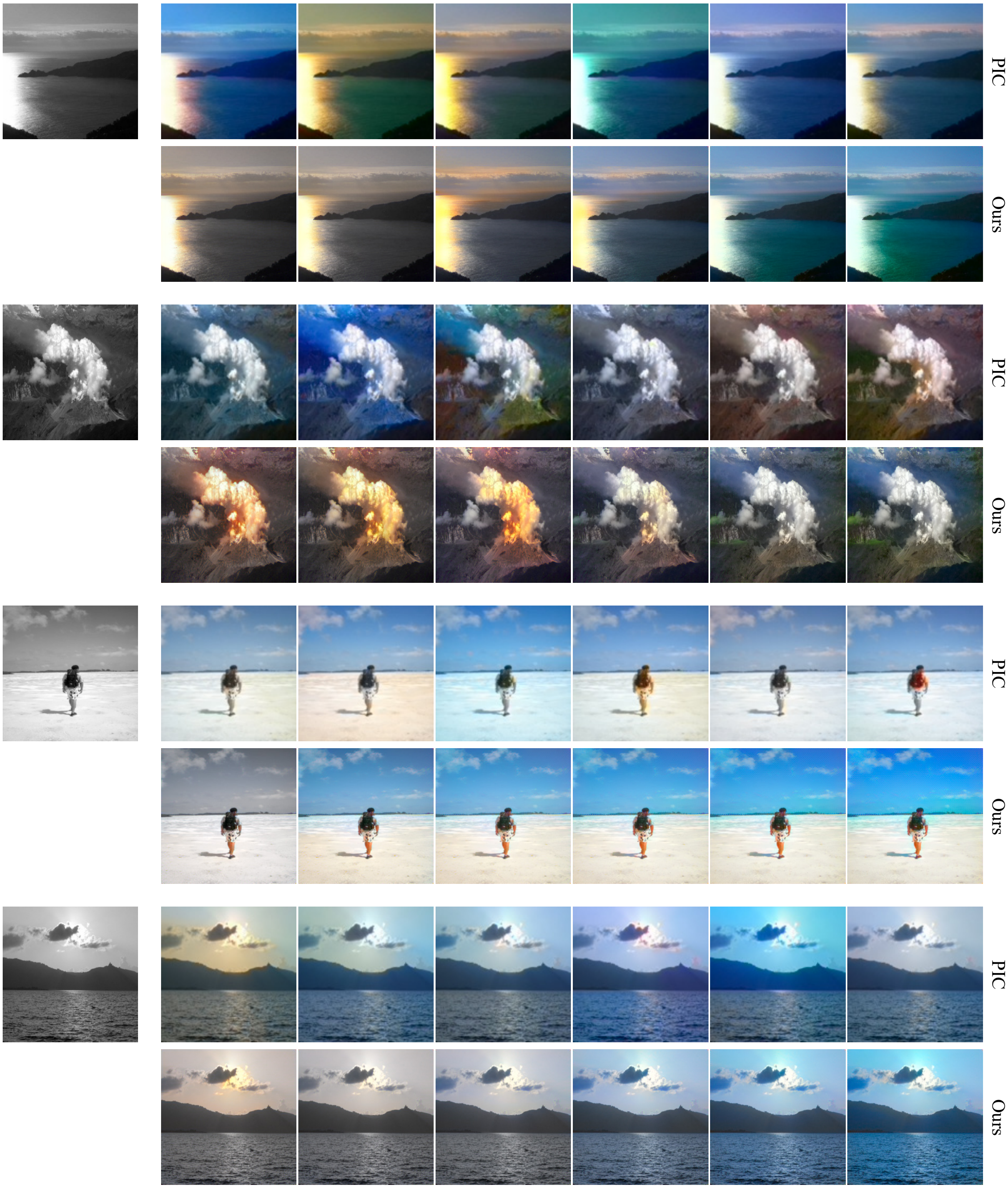

Figure 5: Comparisons of diverse colorization with PIC (**First row**: PIC; **Second row**: Ours). Zoom in to see the spatially incoherent and unrealistic colorizations from PIC. Our method could generate spatially coherent and more natural colorization results with smooth control.

5. User study results

We show some cases from our user study in Fig. 6 - 29. Our method got a significant preference by users than other colorization methods (CIC [6], ChromaGAN [5], DeOldify and InstColorization [4]), showing distinct advantage on producing natural and vivid results.

Figure 6: User preferences for different methods.
From column 2-6: 10%(CIC) : 3%(ChromaGAN) : 13%(DeOldify) : 3%(InstColor) : 70%(Ours).

Figure 7: User preferences for different methods.
From column 2-6: 10%(CIC) : 7%(ChromaGAN) : 23%(DeOldify) : 10%(InstColor) : 50%(Ours).

Figure 8: User preferences for different methods.
From column 2-6: 3%(CIC) : 3%(ChromaGAN) : 30%(DeOldify) : 3%(InstColor) : 60%(Ours).

Figure 9: User preferences for different methods.
From column 2-6: 7%(CIC) : 10%(ChromaGAN) : 20%(DeOldify) : 10%(InstColor) : 53%(Ours).

Figure 10: User preferences for different methods.
From column 2-6: 0%(CIC) : 7%(ChromaGAN) : 0%(DeOldify) : 23%(InstColor) : 70%(Ours).

Figure 11: User preferences for different methods.
From column 2-6: 7%(CIC) : 13%(ChromaGAN) : 17%(DeOldify) : 23%(InstColor) : 40%(Ours).

Figure 12: User preferences for different methods.
From column 2-6: 0%(CIC) : 13%(ChromaGAN) : 3%(DeOldify) : 33%(InstColor) : 50%(Ours).

Figure 13: User preferences for different methods.
From column 2-6: 17%(CIC) : 10%(ChromaGAN) : 27%(DeOldify) : 0%(InstColor) : 47%(Ours).

Figure 14: User preferences for different methods.
From column 2-6: 7%(CIC) : 7%(ChromaGAN) : 0%(DeOldify) : 27%(InstColor) : 60%(Ours).

Figure 15: User preferences for different methods.
From column 2-6: 3%(CIC) : 7%(ChromaGAN) : 13%(DeOldify) : 3%(InstColor) : 73%(Ours).

Figure 16: User preferences for different methods.
From column 2-6: 13%(CIC) : 7%(ChromaGAN) : 0%(DeOldify) : 0%(InstColor) : 80%(Ours).

Figure 17: User preferences for different methods.
From column 2-6: 7%(CIC) : 13%(ChromaGAN) : 17%(DeOldify) : 10%(InstColor) : 53%(Ours).

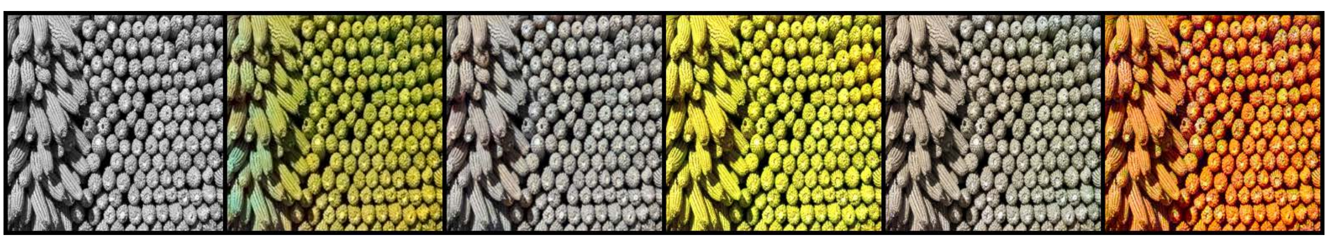

Figure 18: User preferences for different methods.
From column 2-6: 7%(CIC) : 0%(ChromaGAN) : 20%(DeOldify) : 0%(InstColor) : 73%(Ours).

Figure 19: User preferences for different methods.
From column 2-6: 3%(CIC) : 0%(ChromaGAN) : 23%(DeOldify) : 13%(InstColor) : 60%(Ours).

Figure 20: User preferences for different methods.
From column 2-6: 0%(CIC) : 13%(ChromaGAN) : 13%(DeOldify) : 0%(InstColor) : 73%(Ours).

Figure 21: User preferences for different methods.
From column 2-6: 13%(CIC) : 0%(ChromaGAN) : 10%(DeOldify) : 13%(InstColor) : 63%(Ours).

Figure 22: User preferences for different methods.
From column 2-6: 0%(CIC) : 30%(ChromaGAN) : 0%(DeOldify) : 0%(InstColor) : 70%(Ours).

Figure 23: User preferences for different methods.
From column 2-6: 0%(CIC) : 10%(ChromaGAN) : 7%(DeOldify) : 0%(InstColor) : 47%(Ours).

Figure 24: User preferences for different methods.
From column 2-6: 3%(CIC) : 43%(ChromaGAN) : 0%(DeOldify) : 7%(InstColor) : 83%(Ours).

Figure 25: User preferences for different methods.
From column 2-6: 7%(CIC) : 0%(ChromaGAN) : 3%(DeOldify) : 10%(InstColor) : 80%(Ours).

Figure 26: User preferences for different methods.
From column 2-6: 10%(CIC) : 3%(ChromaGAN) : 0%(DeOldify) : 0%(InstColor) : 87%(Ours).

Figure 27: User preferences for different methods.
From column 2-6: 10%(CIC) : 23%(ChromaGAN) : 3%(DeOldify) : 0%(InstColor) : 63%(Ours).

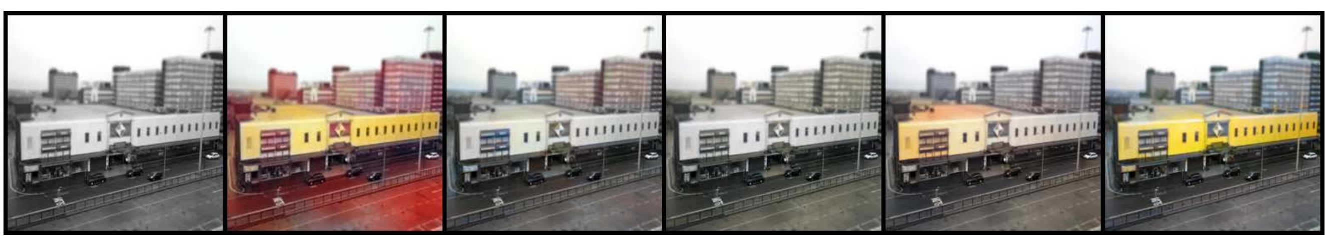

Figure 28: User preferences for different methods.
From column 2-6: 17%(CIC) : 7%(ChromaGAN) : 3%(DeOldify) : 3%(InstColor) : 70%(Ours).

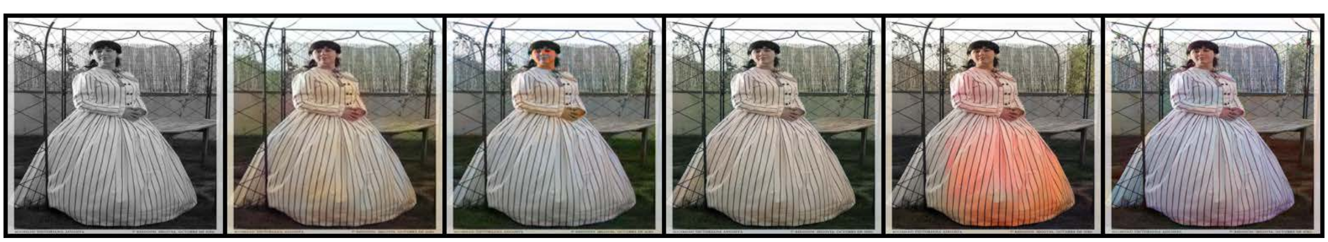

Figure 29: User preferences for different methods.
From column 2-6: 17%(CIC) : 13%(ChromaGAN) : 13%(DeOldify) : 3%(InstColor) : 53%(Ours).

References

[1] Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *CVPR*, 2020. 3

[2] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV*, 2020. 1, 3, 4

[3] Amelie Royer, Alexander Kolesnikov, and Christoph Lampert. Probabilistic image colorization. In *BMVC*, 2017. 4

[4] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *CVPR*, 2020. 7

[5] Patricia Vitoria, Lara Raad, and Coloma Ballester. Chromagan: Adversarial picture colorization with semantic class distribution. In *WACV*, 2020. 7

[6] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016. 7

[7] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020. 3