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

Colorization has attracted increasing interest in recent years. Classic reference-based methods usually rely on external color images for plausible results. A large image database or online search engine is inevitably required for retrieving such exemplars. Recent deep-learning-based methods could automatically colorize images at a low cost. However, unsatisfactory artifacts and incoherent colors are always accompanied. In this work, we aim at recovering vivid colors by leveraging the rich and diverse color priors encapsulated in a pretrained Generative Adversarial Networks (GAN). Specifically, we first “retrieve” matched features (similar to exemplars) via a GAN encoder and then incorporate these features into the colorization process with feature modulations. Thanks to the powerful generative color prior and delicate designs, our method could produce vivid colors with a single forward pass. Moreover, it is highly convenient to obtain diverse results by modifying GAN latent codes. Our method also inherits the merit of interpretable controls of GANs and could attain controllable and smooth transitions by walking through GAN latent space. Extensive experiments and user studies demonstrate that our method achieves superior performance than previous works.

1. Introduction

Colorization, the task of restoring colors from black-and-white photos, has wide applications in various fields, such as photography technologies, advertising or film in-

Figure 1: By leveraging the generative color prior and the delicate designs, our method is capable of generating vivid and diverse colorization results without relying on external exemplars. Besides, our method could attain controllable and smooth transitions by walking through the GAN latent space.
As colorization requires estimating missing color channels from only one grayscale value, it is inherently an ill-posed problem. Moreover, the plausible solutions of colorization are not unique (e.g., cars in black, blue, or red are all feasible results). Due to the uncertainty and diversity nature of colorization, it remains a challenging task.

Classic reference-based methods (e.g. [17, 35, 53]) require additional example color images as guidance. These methods attempt to match the relevant contents between exemplars and input gray images, and then transfer the color statistics from the reference to the gray one. The quality of generated colors (e.g., realness and vividness) is strongly dependent on the reference images. However, retrieving desirable reference images requires significant user efforts. A recommendation system for this can be a solution, but it is still challenging to design such a retrieval process. A large-scale color image database or online search engine is inevitably required in the system.

Recently, convolutional neural network (CNN) based colorization methods [7, 11, 23] have been proposed to facilitate the colorization task in an automatic fashion. These methods learn to discover the semantics [31, 58] and then directly predict the colorization results. Though remarkable achievements in visual qualities are achieved, unsatisfactory artifacts and incoherent colors are always accompanied.

In this work, we attempt to exploit the merits of both reference-based and CNN-based methods, i.e., achieving realistic and vivid colorization results on par with reference-based methods, while keeping them automatic at a low cost. Inspired by recent success in Generative Adversarial Network (GAN) [8, 27], our core idea is to leverage the most relevant image in the learned GAN distribution as the exemplar image. The rich and diverse color information encapsulated in pretrained GAN models, i.e., generative color prior, allows us to circumvent the explicit example retrieval step and integrate it into the colorization pipeline as a unified framework. Specifically, we first ‘retrieve’ matched features (similar to exemplars) via a GAN encoder and then incorporate these features into the colorization process with feature modulations. In addition, our method is capable of achieving diverse colorization from different samples in the GAN distribution or by modifying GAN latent codes. Thanks to the interpretable controls of GANs, our method could also attain controllable and smooth transitions by walking through GAN latent space.

We summarize the contributions as follows. (1) We develop a unified framework to leverage rich and diverse generative color prior for automatic colorization. (2) Our method allows us to obtain diverse outputs. Controllable and smooth transitions can be achieved by manipulating the code in GAN latent space. (3) Experiments show that our method is capable of generating more vivid and diverse colorization results than previous works.

### 2. Related Work

**User assisted colorization.** Early colorization methods are mostly interactive and require users to draw color strokes on the gray image to guide the colorization. The propagation from local hints to all pixels is usually formulated in a constrained optimization manner [43] that tries to assign two pixels with the same color if they are adjacent and similar under similarity measures. Attempts in this direction focus on finding efficient propagation [55] with different similarity metrics by hand-crafting [43] or learning [12], building long-range relations [52], or using edge detection to reduce bleeding [19]. While these methods can generate appealing results with careful interactions, it demands intensive user efforts. The work of [59] alleviates this issue by using sparse color points, and introduces a neural network to colorize from these hints. Some methods also propose to use global hints like color palettes [2, 5] instead of dense color points as constraints.

**Reference-based methods** try to transfer the color statistics from the reference to the gray image using correspondences between the two based on low-level similarity measures [34], semantic features [6], or super-pixels [8, 15]. Recent works [17, 35, 53] adopt deep neural network to improve the spatial correspondence and colorization results. Though these methods obtain remarkable results when suitable references are available, the procedure of finding references is time-consuming and challenging for automatic retrieval system [8].

**Automatic colorization** methods are generally learning based [7, 11, 23, 30]. Pre-trained networks for classification are used for better semantic representation [31]. Two branch dual-task structures are also proposed in [22, 48, 60] that jointly learn the pixel embedding and local (semantic maps) or global (class labels) information. The recent work [47] investigates the instance-level features to model the appearance variations of objects. To improve the diversity of generated colors, Zhang et al. [58] uses a class-rebalancing scheme, while some methods propose to predict per-pixel color distributions [10, 39, 44] instead of a single color. Yet, their results typically suffer from visual artifacts like unnatural and incoherent colors.

**Generative priors** of pretrained GANs [3, 26, 27, 28] is previously exploited by GAN inversion [11, 13, 38, 41, 62, 63], which aims to find the closest latent codes given an input image. In colorization, they first ‘invert’ the grayscale image back to a latent code of the pretrained GAN, and then conduct iterative optimization to reconstruct images. Among them, mGANprior [13] attempts to optimize multiple codes to improve the reconstruction quality. In order to reduce the gap between training and testing images, DGP [41] further jointly finetunes the generator and the latent codes. However, these results struggle to faithfully retain the local details, as the low-dimension latent codes
3. Method

3.1. Overview

With a grayscale image $x^l \in \mathbb{R}^{1 \times H \times W}$ as input, the objective of colorization is to predict the two missing color channels $a$ and $b$ in the original color image $x^{lab} \in \mathbb{R}^{3 \times H \times W}$, where $l$ and $a$, $b$ represent the luminance and chrominance in CIELAB color space, respectively. In this work, we aim to leverage rich and diverse color priors encapsulated in a pretrained generative network to guide the colorization. The overview of our framework is shown in Figure 2. It mainly consists of a pretrained GAN $\mathcal{G}$, a GAN Encoder $\mathcal{E}$, and a Colorization Network $\mathcal{C}$.

Given the grayscale image $x^l$, GAN Encoder $\mathcal{E}$ is responsible for mapping $x^l$ into a latent code $z$, which can serve as an input of $\mathcal{G}$. The pretrained GAN $\mathcal{G}$ then generates the most relevant color image (denoted as inversion image $x^{rgb}_{inv}$) to the grayscale input. In addition to using the inversion image directly, the prior features $F_{\text{GAN}}$ from intermediate layers are more informative and are leveraged to guide the colorization. However, these ‘retrieved’ features usually do not spatially align with the grayscale input. To solve this, we first perform an alignment step to warp $F_{\text{GAN}}$ based on the correspondence between $x^l$ and $x^{rgb}_{inv}$. After that, we use the aligned GAN features to modulate the multi-scale spatial features in Colorization Network $\mathcal{C}$ with effective SPADE (spatially-adaptive denormalization) layers [42], producing the final colorized image $\hat{x}^{rgb}$.

3.2. Generative Color Prior and GAN Encoder

Recently, GAN has achieved tremendous advances in unsupervised image generation. Once trained on an adequate number of images, GAN can map arbitrary random noises into realistic and high-resolution images with impressive details and rich colors, which motivates us to employ pretrained GAN models to assist the colorization task. As illustrated above, we need to find a latent code $z$ so that $\mathcal{G}$ can take it as the input and produce the GAN features $F_{\text{GAN}}$ from its intermediate layers which are most relevant to $x^l$. To achieve this goal, we constrain the generated image from code $z$ to have similar content with $x^l$ but contains rich chrominance information. The process of finding a certain $z$ corresponding to a target image is termed as GAN inversion. A typical inversion way is to randomly initialize $z$ and optimize it to reduce the difference between the generated image and the target one. However, such a method requires expensive iterative optimization at inference and thus is not applicable for real-world scenarios. Instead, we choose to train a GAN encoder $\mathcal{E}$ to map arbitrary grayscale images
into latent codes, i.e.,
\[
z = \mathcal{E}(x') = x_{inv}' \circ F_{GAN} = \mathcal{G}(z),
\]
where \(\mathcal{G}(z)\) simultaneously outputs the inversion image \(x_{inv}'\) and the intermediate GAN features \(F_{GAN}\).

**Discussion.** It is well known that the optimization-based inversion methods could generate more accurate inversion images than encoder-based inversion methods \cite{62, 63}. However, in our colorization scenarios, we do not need such precise and high-quality inversion images. A coarse inversion result already contains sufficient color information and the colorization network (illustrated in the next section) will further refine the corresponding inversion features. In addition, we use the GAN features as extra information to guide the colorization, which could provide complementary information to the inversion image.

### 3.3. Colorization Network

The colorization network \(\mathcal{C}\) consists of two downsampling blocks, six residual blocks and two upsampling blocks, which proves to be an effective architecture used in many image-to-image translation tasks \cite{50}. It takes the grayscale image \(x'\) as input and predicts two missing color channels \(\hat{z}^{ab}\). A color image \(\hat{z}^{lab}\) can then be obtained by concatenating the luminance channel \(x'\) and the chrominance channels \(\hat{z}^{ab}\). As the two color spaces RGB and LAB can be converted to each other, \(\hat{z}^{rgb}\) can also be obtained.

In order to use the prior color features \(F_{GAN}\) to guide the colorization, and to better preserve the color information, we use the GAN features as extra information to guide the colorization, which could provide complementary information to the inversion image.

### 3.4. Objectives

**GAN inversion losses.** As we have the ground truth colorful images \(x_{rgb}^{inv}\) during training, we can train GAN encoder \(\mathcal{E}\) by minimizing the difference between \(x_{inv}'\) and \(x_{rgb}^{inv}\). Instead of directly constraining the pixel values of \(x_{inv}'\) and \(x_{rgb}^{inv}\), we choose to minimize the discrepancy between their features extracted by the pre-trained discriminator \(\mathcal{D}^g\) of \(\mathcal{G}\), as this will make the GAN inversion easier to optimize (especially for BigGAN) \cite{9, 41}. Formally, the discriminator feature loss \(\mathcal{L}_{inv,ftr}\) is defined as:
\[
\mathcal{L}_{inv,ftr} = \sum_{l} \left\| \mathcal{D}^g_l(x_{inv}') - \mathcal{D}^g_l(x_{rgb}^{inv}) \right\|_1,
\]
where \(\mathcal{D}^g_l\) represents the feature map extracted at \(l\)-th layer from \(\mathcal{D}^g\).

To keep \(z\) within a reasonable range, we also add a \(L_2\) norm \(\mathcal{L}_{inv,reg}\), otherwise, the GAN inversion with grayscale input will lead to unstable results:
\[
\mathcal{L}_{inv,reg} = \frac{1}{2} \| \hat{z} \|_2.
\]

**Adversarial loss.** We adopt the adversarial loss to help the colorization images looking more realistic. To stabilize the adversarial training, we employ the loss functions in LS-GAN \cite{56}:
\[
\begin{align*}
\mathcal{L}_{adv}^D &= \mathbb{E}[(\mathcal{D}^c(\hat{z}^{lab}) - 1)^2] + \mathbb{E}[(\mathcal{D}^c(\hat{z}^{lab}) - 1)^2], \\
\mathcal{L}_{adv}^G &= \mathbb{E}[(\mathcal{D}^c(\hat{z}^{lab} - 1))],
\end{align*}
\]
where \(\mathcal{D}^c\) is the discriminator to discriminate the colorization images \(\hat{z}^{lab}\) generated from \(C\) and color images \(x_{rgb}^{inv}\) from real world. \(C\) and \(\mathcal{D}^c\) are trained alternatively with \(\mathcal{L}_{adv}^G\) and \(\mathcal{L}_{adv}^D\), respectively.

**Perceptual loss.** To make the colorization image perceptual plausible, we use the perceptual loss introduced in \cite{25}:
\[
\mathcal{L}_{perc} = \| \phi_l(\hat{z}^{rgb}) - \phi_l(x_{rgb}^{inv}) \|_2,
\]
where \(\phi_l\) represents the feature map extracted at \(l\)-th layer from a pretrained VGG19 network. Here, we set \(l = relu5_2\).

**Domain alignment loss.** To ensure that the grayscale image and inversion image are mapped to a shared feature.
space in correlation calculation, we adopt a domain alignment loss [57]:

$$L_{dom} = \| \mathcal{F}_{L \rightarrow S}(x^l) - \mathcal{F}_{RGB \rightarrow S}(x^{rgb}) \|_1.$$  \hspace{1cm} (6)

**Contextual Loss.** The contextual loss [37] can measure the similarity between two non-aligned images. This loss term was originally designed for image translation tasks with non-aligned data, and then extended to exemplar-based colorization [56] to encourage colors in the output image to be close to those in the reference image. In this paper, we also adopt this loss to encourage the colorization image $\hat{x}^{rgb}$ to be relevant to the inversion image $x^{inv}_{inv}$. Formally, the contextual loss $L_{ctx}$ is defined as:

$$L_{ctx} = \sum_{l} \omega_l \left( - \log(\text{C}X(\phi_l(\hat{x}^{rgb})), \phi_l(x^{inv}_{inv})) \right),$$  \hspace{1cm} (7)

where CX denotes the similarity metric between two features and we refer readers to [37] for more details. We use the layer $l = \text{relu}\{3, 2, 4, 2, 5, 2\}$ from the pretrained VGG19 network with weight $\omega_l = \{2, 4, 8\}$ to calculate $L_{ctx}$. We set larger loss weights for deeper features to guide the network to focus more on the semantic similarity instead of pixel-wise color closeness.

**Full objective.** The full objective to train the GAN encoder $E$ is formulated as

$$L_E = \lambda_{inv, ftr} L_{inv, ftr} + \lambda_{inv, reg} L_{inv, reg},$$  \hspace{1cm} (8)

The full objective to train the colorization network $C$ is formulated as

$$L_C = \lambda_{dom} L_{dom} + \lambda_{perc} L_{perc} + \lambda_{ctx} L_{ctx} + \lambda_{adv} L_{adv}.$$  \hspace{1cm} (9)

The full objective to train the discriminator $D_C$ for colorization is formulated as

$$L_{D_C} = \lambda_{adv} L_{adv}^D.$$  \hspace{1cm} (10)

In the above loss objectives, $\lambda_{inv, ftr}$, $\lambda_{inv, reg}$, $\lambda_{dom}$, $\lambda_{perc}$, $\lambda_{ctx}$, and $\lambda_{adv}$ are hyper-parameters that control the relative importance of each loss term.

### 4. Experiments

#### 4.1. Implementation Details

We conduct our experiments on ImageNet [45], a multi-class dataset, with a pretrained BigGAN [3] as the generative color prior. We train our method with the official training set. All the images are resized to 256 x 256.

Our training process consists of two stages. In the first stage, we freeze the pretrained BigGAN generator and train a BigGAN discriminator with $L_E$. We utilize the last three layers of BigGAN discriminator to calculate the feature loss, where $\lambda_{inv, ftr}$ is set to 1.0 and $\lambda_{inv, reg}$ is set to 0.0125. We train the model for totally 10 epochs with an initial learning rate of 1e−5.

In the second stage, the BigGAN encoder is frozen and the rest of networks are trained end-to-end. We set the $\lambda_{perc}$ to 1e−3, $\lambda_{adv}$ to 1.0, $\lambda_{dom}$ to 10.0 and $\lambda_{ctx}$ to 0.1. We train the model for totally 10 epochs with an initial learning rate of 1e−4. In both two stages, we set the batch size to 8 per GPU and decay the learning rate linearly to 0. All the models are trained with the Adam [29] optimizer.

The BigGAN encoder has a similar structure to the BigGAN discriminator, while the class embedding is injected into the BigGAN encoder following the BigGAN generator. To reduce the learnable parameters of $\mathcal{F}_{L \rightarrow S}$ and $\mathcal{F}_{RGB \rightarrow S}$, their backbones are shared. The discriminator for colorization $D^c$ is a three-scale PatchGAN network [23, 30] with spectral normalization [40].

### 4.2. Comparisons with Previous Methods

To evaluate the performance of our method, we compare our results to other state-of-the-art automatic colorization methods including CIC [58], ChromaGAN [48], DeOldify[1] and InstColorization [47]. We re-implement CIC and ChromaGAN in PyTorch based on the original paper.

**Quantitative comparison.** We test all the methods on the ImageNet validation set of 50,000 images. The quantitative results on five metrics are reported in Table 1. Fréchet Inception Score (FID) [18] measures the distribution similarity between the colorization results and the ground truth color images. Our method achieves the lowest FID, indicating that our method could generate colorization results with better image quality and fidelity. Colorfulness Score [16] reflects the vividness of generated images. Our method obtains a better colorfulness score (2nd column) and closest colorfulness score to the ground-truth images (3rd column). The highest colorfulness score of CIC is an outlier, also observed in [56]. This is probably because CIC encourages rare colors in the loss function, and the rare colors are misjudged as vivid colors by this metric. We also provide PSNR and SSIM for reference. However, it is well-known that such pixel-wise measurements may not well reflect the actual performance [4, 17, 39, 47, 48, 60], as plausible colorization results probably diverge a lot from the original color image.

**Qualitative Comparison.** As shown Figure 3 compared to CIC, ChromaGAN and Deoldify, our results tend to obtain more natural and vivid colors. For instance, in contrast to our vivid green, the broccoli color (Column 5) of the other methods looks unnatural and yellowish. In addition, our method generates better results with regards to consistent hues. In column 1, we can observe inconsistent chromaticities in the results of Deoldify and InstColorization, where

---

[1] https://github.com/jantic/DeOldify
Table 1: Quantitative comparison. $\Delta_{\text{Colorful}}$ denotes the absolute colorfulness score difference between the colorization images and the ground truth color images.

<table>
<thead>
<tr>
<th></th>
<th>FID↓</th>
<th>Colorful↑</th>
<th>$\Delta_{\text{Colorful}}$↓</th>
<th>PSNR↑</th>
<th>SSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIC</td>
<td>19.71</td>
<td>43.92</td>
<td>5.57</td>
<td>20.86</td>
<td>0.86</td>
</tr>
<tr>
<td>ChromaGAN</td>
<td>5.16</td>
<td>27.49</td>
<td>10.86</td>
<td>23.12</td>
<td>0.87</td>
</tr>
<tr>
<td>DeOldify</td>
<td>3.87</td>
<td>22.83</td>
<td>15.52</td>
<td>22.97</td>
<td>0.91</td>
</tr>
<tr>
<td>InstColorization</td>
<td>7.36</td>
<td>27.05</td>
<td>11.30</td>
<td>22.91</td>
<td>0.91</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>3.62</strong></td>
<td><strong>35.13</strong></td>
<td><strong>3.22</strong></td>
<td><strong>21.81</strong></td>
<td><strong>0.88</strong></td>
</tr>
</tbody>
</table>

Figure 3: Visual comparisons with previous automatic colorization methods. Our method is able to generate more natural and vivid colors. Though ChromaGAN has consistent tones, it fails to capture the details by only mapping overall blue tones to the image, leading to the blue artifacts on the lady’s face. Instead, our method successfully maintains the consistent tone and captures the details as shown in Column 1. Furthermore, our method can yield more diverse and lively colors for each small instance as shown in column 7, while InstColorization fails to detect such small objects and inherently degrades to use its global image colorization network, whose results are dim and less vivid.
Figure 4: Boxplots of user preferences for different methods. Green dash lines represent the means. Our method got a significantly higher preference rate by users than other colorization methods.

**User study.** In order to better evaluate the subjective quality (i.e., vividness and diverseness of colors), we conduct a user study to compare our method with the other state-of-art automatic colorization methods, i.e., CIC [58], ChromaGAN [48], DeOldify and InstColorization [47]. We recruited 30 participants with normal or corrected-to-normal vision and without color blindness. We randomly select 70 images from various categories of image contents (e.g., animals, plants, human beings, landscape, food, etc.). For each image, the grayscaled image is displayed at the leftmost while five colorization results are displayed in a random manner to avoid potential bias. The participants are required to select the best-colorized image according to the colorization quality in terms of vividness and diverseness. The boxplots are displayed in Figure 4. Our method obtained a significant preference (46.3%) by users than other colorization methods (CIC 12.4%, ChromaGAN 11.7%, DeOldify 16.6%, InstColor 13.0%), demonstrating distinct advantage on producing natural and vivid results.

4.3. Ablation Studies

**Generative color prior.** Generative color prior plays an important role in providing exemplar features for vivid and various colors. When we remove the pretrained GANs, our method degrades to a common automatic colorization method without guidance. As shown in Figure 5, the variant experiment without GAN prior generates dim and bleak colors, while our method could produce bright and joyful images, under the guidance of colorful exemplars in GAN distributions. We could also observe a large drop in FID and Colorfulness score without generative color prior (Table 2).

**Feature guidance vs. image guidance.** Our method adopts multi-resolution intermediate GAN features to guide the colorization. The design could fully facilitate the color and detail information from GAN features. Compared to image guidance, our results are capable of recovering more vivid colors faithful to their underlying classes (e.g., the trees on the beach, the grass on the mountains), as multi-resolution features contain more semantic and color information (Figure 5 and Table 2).

**Spatial alignment.** The prior features from pretrained GANs probably misalign with the input images. To address this problem, we employ spatial alignment to align the GAN features, leading to fewer artifacts and coherent results for colorization. As shown in Figure 5, the tree and square colors of our results are not affected by other objects.

4.4. Controllable Diverse Colorization

In order to achieve diverse colorization, many attempts have been made. For automatic colorization methods, diversity is usually achieved by stochastic sampling [10][14][44]. However, the results are barely satisfactory and difficult to
Table 2: Quantitative comparisons for ablation studies. \(\Delta\)Colorful denotes the absolute colorfulness score difference between the colorization images and the ground truth color images.

<table>
<thead>
<tr>
<th>Variants</th>
<th>FID↓</th>
<th>Colorful↑</th>
<th>(\Delta)Colorful↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Model</td>
<td>3.62</td>
<td>35.13</td>
<td>3.22</td>
</tr>
<tr>
<td>w/o Generative Color Prior</td>
<td>8.40</td>
<td>31.21</td>
<td>7.14</td>
</tr>
<tr>
<td>Image Guidance</td>
<td>4.01</td>
<td>26.12</td>
<td>12.23</td>
</tr>
<tr>
<td>w/o Spatial Alignment</td>
<td>4.59</td>
<td>31.94</td>
<td>6.41</td>
</tr>
</tbody>
</table>

Figure 6: Our method could adjust the latent codes to obtain various inversion results, thus easily achieving diverse colorization results for the parrot.

control ( Detailed comparisons are provided in the supplementary material). For exemplar-based colorization methods [17, 53], it is time-consuming and challenging to find large amounts of reference images with various styles. Different from previous methods, our method could achieve diverse and controllable colorization in a much easier and novel manner.

On the one hand, we can adjust the latent codes with disturbance to obtain various exemplars and their corresponding features. As shown in Figure 6 various inversion results are attained by adjusting the latent codes, and correspondingly, diverse colorization results of parrots are obtained. On the other hand, our method inherits the merits of interpretable controls of GANs [20, 24, 46, 49] and could attain controllable and smooth transitions by walking through GAN latent space. Specifically, we employ an unsupervised method [49] to find color-relevant directions, such as lighting, saturation, etc. As depicted in Figure 7 smooth transitions on background colors, object colors, saturation could be achieved.

4.5. Limitations

Though our method could produce appealing results in most cases, it still has limitations. When the input image is not in the GAN distribution or GAN inversion fails, our method degrades to common automatic colorization methods and may result in unnatural and incoherent colors. As shown in Figure 8 the human on the beach and the cactus are missing from the GAN inversion results. Thus, our method cannot find corresponding color guidance, resulting in unnatural colorization for these objects. This could be mitigated by improving the GAN inversion and common automatic colorization approaches in future works.

5. Conclusion

In this work, we have developed a framework to produce vivid and diverse colorization results by leveraging generative color priors encapsulated in pretrained GANs. Specifically, we ‘retrieve’ multi-resolution GAN features conditioned on the input grayscale image with a GAN encoder, and incorporate such features into the colorization process with feature modulations. One could easily realize diverse colorization by simply adjusting the latent codes or conditions in GANs. Moreover, our method could also attain controllable and smooth transitions by walking through the GAN latent space.
References


