**Image Processing Using Multi-Code GAN Prior**

Amil Anilkumar

**Abstract**

While Generative Adversarial Networks (GANs) have achieved remarkable success in image generation, their application to real-world image processing tasks remains challenging. Traditional methods typically invert a target image into the latent space using either back-propagation or by training an additional encoder. However, these approaches often lead to poor reconstructions. In this paper, we introduce a novel approach called mGANprior, which leverages pre-trained GANs as a powerful prior for a wide range of image processing tasks. Specifically, our method utilizes multiple latent codes to generate several feature maps from intermediate layers of the GAN’s generator, which are then combined with adaptive channel weighting to accurately reconstruct the input image. This over-parameterization of the latent space significantly enhances image reconstruction quality, outperforming existing methods. The high-fidelity reconstructions produced by this approach make it possible to use pre-trained GANs as priors for various real-world applications, including image colorization, super-resolution, inpainting, and semantic manipulation.

1. **Introduction**

Recently, Generative Adversarial Networks (GANs) have made significant strides in image generation, enhancing the quality of synthesized images and improving the stability of the training process. This ability to generate high-quality images has opened up many possibilities for GANs in various image processing tasks, such as semantic face editing, super-resolution, and image-to-image translation. However, most GAN-based methods require task-specific modifications in network design or loss functions, which limits their versatility and ability to generalize across different applications. In contrast, large-scale GAN models, such as StyleGAN and BigGAN, are capable of generating photo-realistic images after being trained on massive datasets. These models have been found to capture different levels of semantic information within the data they process. While reusing these pre-trained models as priors for real-world image processing tasks with minimal additional effort could greatly expand their range of applications, this approach has not been fully explored.

The main challenge in using GANs for real image processing is that these models are originally designed to generate images from random noise, meaning they are not capable of handling real images for post-processing tasks. A common solution is to invert a real image into a latent code that can be used by the generator to recreate the image. This latent code can then be utilized for further processing. There are two primary approaches for reversing the generation process. The first involves directly optimizing the latent code by minimizing the reconstruction error through back-propagation. The second approach trains an additional encoder to learn the mapping between the image space and the latent space. However, both methods often produce poor reconstructions, particularly when dealing with high-resolution images. As a result, the reconstructed image quality is insufficient for effective use in image processing tasks.

It is fundamentally impossible to recover every detail of an arbitrary real image using a single latent code, as that would imply having an ideal image compression method. The latent code’s expressiveness is constrained by its finite dimensionality. To more accurately reconstruct a target image, we propose the use of multiple latent codes and combining their corresponding feature maps at an intermediate layer of the generator. By leveraging multiple latent codes, the generator can recover the target image by utilizing all the compositional knowledge embedded in the deep generative representation. Our experiments demonstrate that this approach significantly enhances the quality of image reconstruction. More importantly, this improved reconstruction capability enables various real image processing tasks by using pre-trained GAN models as a prior, without the need for retraining or model modifications.

mGANprior (multi-code GAN prior), an advanced GAN inversion technique that utilizes multiple latent codes and adaptive channel importance to effectively reconstruct real images, surpassing existing methods. The mGANprior method is applied to a variety of real-world tasks, including image colorization, super-resolution, image inpainting, and semantic manipulation, showcasing its versatility in real image processing. Additionally, we conduct an analysis of the internal representations at different layers of a GAN generator by composing the features extracted from the inverted latent codes at each respective layer.

1. **Related Work**

**GAN Inversion: GAN Inversion** refers to the task of mapping a given image back to its latent code using a pre-trained GAN model. This process is a crucial step in applying GANs to real-world tasks, and it has recently garnered significant attention. Existing methods for inverting a fixed GAN generator typically fall into two categories: optimization-based approaches, which iteratively refine the latent code using gradient descent, and encoder-based methods, where an additional encoder is learned to project the image back to the latent space. For instance, Bau et al. proposed using an encoder to better initialize the optimization process. Some models even integrate invertibility during the training phase. However, all of these methods rely on using a single latent code to reconstruct the input image, and the results often fall short, especially when the test image deviates significantly from the training data. This limitation arises because the input image may not lie within the generator's synthesis space, making a perfect inversion with just one latent code impossible. In contrast, we propose using multiple latent codes, which significantly enhances the inversion quality, whether the target image is within or outside the model's training domain.

**Image Processing with GANs**: GANs have become highly effective for real image processing tasks, thanks to their ability to generate photorealistic images. These applications include tasks such as image denoising, image inpainting, super-resolution, image colorization, style mixing, and semantic image manipulation. However, many existing GAN-based models are designed with specialized architectures or loss functions tailored to a specific task. These models are typically trained using paired data, where one image is input and another serves as the supervision. In contrast, our approach leverages the knowledge embedded in a pre-trained GAN model, allowing a single GAN to be used as a prior for a wide range of image processing tasks without the need for retraining or modifications. Notably, our method achieves comparable or even superior results compared to existing GAN-based models that are trained specifically for individual tasks.

**Deep Model Prior**: The impressive performance of deep convolutional models is largely due to their ability to capture statistical patterns from large datasets, which serve as a prior for image generation and reconstruction. This prior knowledge has been applied inversely for various image processing tasks. For instance, Upchurch et al. inverted a discriminative model starting from deep convolutional features to achieve semantic image transformation. Ulyanov et al. demonstrated that a U-Net structure can effectively reconstruct a target image by capturing low-level image statistics even before any learning occurs. Athar et al. developed a universal image prior applicable to multiple image restoration tasks. While some studies have explored the theoretical aspects of the prior provided by deep generative models, the results when applying GAN priors to real image processing tasks have been limited. A recent approach applied generative image priors to semantic photo manipulation, but it was only able to edit specific regions of the input image and could not handle tasks like colorization or super-resolution. This limitation arises because it inverts the GAN model to an intermediate feature space, rather than fully reversing the generative process back to the original latent space. In contrast, our method inverts the entire generative process, from the image space to the initial latent space, allowing for a broader range of flexible image processing tasks.

1. **Multi-Code GAN Prior**

A well-trained GAN generator G(⋅)can produce high-quality images by sampling latent codes from the latent space Z. Given a target image x, the task of GAN inversion is to reverse the generation process and find the appropriate latent code that can reconstruct x. This can be formulated as:

A math equation with black text

Description automatically generated with medium confidence

However, due to the highly non-convex nature of this optimization problem, previous methods often struggle to accurately reconstruct an arbitrary image by optimizing a single latent code. To address this, we propose utilizing multiple latent codes and combining their corresponding intermediate feature maps with adaptive channel importance to improve the reconstruction process.

**GAN Inversion with Multiple Latent Codes:**

A single latent code may not have sufficient expressiveness to capture all the details of a given image. So, what if we use NNN latent codes ​, where each code helps to reconstruct different sub-regions of the target image? Below, we describe how to effectively leverage multiple latent codes for GAN inversion.

*Feature Composition*

To enhance the process of GAN inversion using multiple latent codes, we introduce the concept of **intermediate feature maps**. The generator G(⋅) is divided into two sub-networks: G1(‘)(⋅)n and G2(‘)(⋅). The index ‘`‘ refers to the specific intermediate layer of the generator where the feature composition occurs. This division allows us to extract the corresponding spatial feature Fn(‘)=G1(‘)(zn) for each latent code zn ​, which can then be used for further processing. A major challenge when using multiple latent codes is determining how to integrate them during the generation process. A simple approach might be to combine the images generated by each znz\_nzn​ in the image space XXX. However, since XXX is not a naturally linear space, linearly combining the generated images may not result in a meaningful image, let alone a faithful reconstruction of the original input.

A recent study highlighted that inverting a generative model from the image space to an intermediate feature space is significantly easier than inverting directly to the latent space. Based on this insight, we propose to combine the latent codes by composing their corresponding feature maps at an intermediate layer within the generator. This approach allows for a more flexible and accurate integration of multiple latent codes in the image generation process.

*Adaptive Channel Importance*

We aim to make each latent code znz\_nzn​ focus on reconstructing specific regions of the target image. To achieve this, we leverage the observation by Bau et al. [4] that different channels in the generator are responsible for generating distinct visual concepts (e.g., objects, textures). Based on this, we introduce **adaptive channel importance** αn\alpha\_nαn​ for each latent code znz\_nzn​. Here, where C is the number of channels in the intermediate layer ‘`‘ of the generator. Thus, the final reconstructed image xinv​ is generated by:

Thus, the final reconstructed image xinv​ is generated by:

A math equations and formulas

Description automatically generated

Here:

* i and j represent the spatial locations,
* c refers to the channel index.

***Optimization Objective***

With the introduction of feature composition and adaptive channel importance to integrate multiple latent codes, the optimization now involves 2N sets of parameters. As a result, we reformulate the objective function as follows:

A black text on a white background

Description automatically generated

To enhance the image reconstruction quality, the objective function is defined to incorporate both low-level pixel-wise errors and high-level perceptual features. Specifically, we use the pixel-wise reconstruction error and the L1 distance between perceptual features extracted from both images. The objective function becomes:



Here, ϕ(⋅) represents the perceptual feature extractor. To solve this, we use gradient descent to optimize the latent codes and the corresponding channel importance values.

1. **Knowledge Representation in GANs**

As mentioned previously, one of the key limitations when using a single latent code is its restricted expressiveness, especially when there is a domain gap between the training data and the test image. To address this, we investigate whether utilizing multiple latent codes can alleviate this issue. Specifically, we test GAN models trained for synthesizing faces, churches, conference rooms, and bedrooms, and attempt to invert a bedroom image using these models.

when only a single latent code is used, the reconstructed image often remains within the domain of the original training data (for instance, the image reconstructed using the PGGAN CelebA-HQ model appears as a face rather than a bedroom). In contrast, our approach, which uses multiple latent codes, allows us to effectively reconstruct a bedroom image regardless of the original domain the GAN model was trained on.

We also analyze the knowledge represented across different layers of a well-trained GAN model. By performing feature composition at various layers, we observe that the reconstruction quality improves as we move to higher layers. This is because higher layers focus on recovering more detailed content, while lower layers tend to capture more abstract semantics. For instance, we found that the 4th layer is sufficient to invert a bedroom image when using the bedroom-trained GAN, while other models (trained for faces, churches, or conference rooms) require the 8th layer for satisfactory reconstruction. This indicates that higher-level content knowledge in the bottom layers is not as useful when the test image significantly differs from the training domain.

To further analyze the effectiveness of our approach, we apply it to image colorization and image inpainting tasks, The colorization task yields the best results when using the 8th layer, while the inpainting task performs best with the 4th layer. This distinction arises because colorization primarily involves low-level rendering tasks, while inpainting requires filling in missing content with semantically meaningful objects. This supports our earlier findings that low-level knowledge from GAN priors is more useful at higher layers, while high-level content knowledge is captured in the lower layers.

1. **Processed Images**

*MultiCode Inversion*

**A bedroom with a red bed and a red dresser

Description automatically generated** 

*Colorization*

*A bed with a striped bedding

Description automatically generated A bed with a wood headboard

Description automatically generated*

*Face Semantic editing*

**A person with dark hair and a white shirt

Description automatically generated A blurry image of a person

Description automatically generated**

*Inpainting*

A black and white image of a black background

Description automatically generatedA blurry image of a building

Description automatically generated

1. **Conclusion**

We introduce **mGANprior**, a method that utilizes multiple latent codes to reconstruct real images using a pre-trained GAN model. This approach leverages the power of GAN models as a strong prior for various image processing tasks

1. **References**
2. Image Processing Using Multi-Code GAN Prior Jinjin Gu, Yujun Shen , Bolei Zhou
3. Martin Arjovsky, Soumith Chintala, and Leon Bottou. ´ Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.
4. ShahRukh Athar, Evgeny Burnaev, and Victor Lempitsky. Latent convolutional models. In ICLR, 2019.
5. David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. In SIGGRAPH, 2019.
6. David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In ICLR, 2019.
7. David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Inverting layers of a large generator. In ICLR Workshop, 2019