Near Perfect GAN Inversion

Qianli Feng
Amazon
Viraj Shah
Amazon
Raghudeep Gadde
Amazon
Pietro Perona
Amazon
Aleix Martinez
Amazon

Abstract

To edit a real photo using Generative Adversarial Networks (GANs), we need a GAN inversion algorithm to identify the latent vector that perfectly reproduces it. Unfortunately, whereas existing inversion algorithms can synthesize images similar to real photos, they cannot generate the identical clones needed in most applications. Here, we derive an algorithm that achieves near perfect reconstructions of photos. Rather than relying on encoder- or optimization-based methods to find an inverse mapping on a fixed generator \( G(\cdot) \), we derive an approach to locally adjust \( G(\cdot) \) to more optimally represent the photos we wish to synthesize. This is done by locally tweaking the learned mapping \( G(\cdot) \) s.t. \( \| x - G(z) \| < \epsilon \), with \( x \) the photo we wish to reproduce, \( z \) the latent vector, \( \| \cdot \| \) an appropriate metric, and \( \epsilon > 0 \) a small scalar. We show that this approach can not only produce synthetic images that are indistinguishable from the real photos we wish to replicate, but that these images are readily editable. We demonstrate the effectiveness of the derived algorithm on a variety of datasets including human faces, animals, and cars, and discuss its importance for diversity and inclusion.

1. Introduction

Generative Adversarial Networks (GANs) have seen dramatic improvement on image photo-realism in recent years [14, 21, 18, 9, 19]. Crucially, GAN-generated images are editable, enabling a number of previously difficult to envision applications. For example, in movies, ads, virtual reality, video games and e-commerce, we can now edit existing photos to improve or showcase image variants that are difficult, costly, or impossible to film. For instance, a face can be shown to be younger or have a different hair color or style.

To edit real photos though, we need a GAN inversion algorithm, Fig. 1. More formally, the generator function \( G(z) = x \) maps a latent vector \( z \in \mathcal{Z} \subseteq \mathbb{R}^q \) into an image \( x \in \mathcal{X} \subseteq \mathbb{R}^p \), with \( q \ll p \). GAN inversion is the problem of finding \( z \) given \( x \), i.e., \( z = G^{-1}(x) \). However, unlike Normalizing Flows and auto-encoders, inverting \( G(\cdot) \) in GANs is generally impossible. To solve this problem, researchers use encoder or optimization approaches to solve

\[
\min \| G(h(x)) - x \| \quad (1)
\]

with the algorithm converging to a solution iff \( \| G(h(x)) - x \| < \epsilon \), and where \( \| \cdot \| \) is a metric in image space \( \mathcal{X} \), \( h : \mathcal{X} \mapsto \mathcal{Z} \), and \( \epsilon > 0 \) is small.

Figure 1: (a) Two real photos. The goal of GAN inversion is to find a latent code that synthesizes an image that is as similar as possible to that in (a). Synthesizing these photos is hard as made evident by the results obtained using state-of-the-art GAN inversion algorithms as shown in (b) and (c). In (d) we show the results of the method proposed in this paper. As we can see, the synthesized images are almost identical clones with only tiny differences not visible to the naked eye. Furthermore, these clones are readily editable using standard GAN editing algorithms as shown in (e-f). Results best seen at a 600% zoom.
Optimization algorithms find the latent vector representation of a given image by optimizing \( h(\cdot) \) as in Eq. (1) from a starting guess [24, 13, 1, 2]. In encoder approaches \( h(\cdot) \) is usually a deep network trained to map from \( \mathcal{X} \) onto \( \mathcal{Z} \) and optimize Eq. (1) over a training set [26, 26, 28, 4, 10]. There are also hybrid approaches [11, 6].

Hence, crucially, optimization and encoder approaches assume that the mapping \( G(\cdot) \) is able to synthesize the desirable photo \( x \). However, given the large variety of all possible photos \( \mathcal{X} \), this assumption will typically not hold, resulting in sub-par reconstruction results, Fig. 1(b, c).

While optimization- and encoder-based approaches optimize or learn \( h(\cdot) \) while fixing \( G(\cdot) \), our key contribution is to show we can fix \( h(\cdot) \) and locally tune \( G(\cdot) \) instead.

Fig. 1(d) shows the reconstruction results of our approach compared to state-of-the-art methods Fig. 1(b, c). GAN editing can then be applied to the reconstructed images, yielding photo-realistic image variants, Fig. 1(e-g).

The derived solution is not only scientifically novel, it is also essential for many real-world applications. Semantic manipulation of real photos is only meaningful if the edited images are seen as realistic variants of the original photos, e.g., the exact same face and background but with a different eye gaze, mouth shape or facial hair as illustrated in Fig. 1(e-g). Additionally, as out-of-sample photos are more likely to be ill-inverted, most current algorithms cannot be used on photos of groups under-represented in the training data, which may include ethnic, racial, gender, age, religious, cultural, job/occupation, etc.

We provide extensive comparative evaluations on several datasets against the state of the art, showing that our reconstructed images keep their photo-realism even after being edited, and demonstrate that our derived algorithm is equally applicable to under-represented groups, increasing diversity and inclusion.

2. Related Work

Two GAN inversion algorithms have been proposed in the literature: “projection via optimization” and “projection via a feedforward network”, i.e., optimization- and an encoder-based method [39].

2.1. Optimization-based inversion

Optimization-based methods require making three choices. First, a loss to measure the similarity between the real photo and the reconstructed image; second, a latent space over which the loss function may be optimized; and, third, the optimization criterion. Early works like [24] explore the usage of the likelihood loss in a Gaussian latent space. Others [13] perform stochastic clipping, limiting the magnitude of the gradient as it optimizes image similarity.

2.2. Encoder-based inversion

For encoder-based or learning-based methods as in [36], we require first, an encoder network; second, a loss to measure the similarity between the real photo and the reconstructed image, and third, the output vectors on the latent space. [26] proposed an auto-encoder architecture incorporating a StyleGAN generator, with an explicit mapping learnt from synthetic image. Others use different mapping or latent space representations [28].

Compared to optimization-based methods, encoder-based algorithms enjoy the benefit of faster inference time. To further improve their results, ReStyle [4] defines an iterative residual-based encoder. [10] proposed a latent space encoder with masked input to study the compositionality in GANs latent space. There are also hybrid methods [11, 6], combining the advantages of encoder- and optimization-based method.

2.3. Limitations of these methods

The pre-trained generator is fixed in both the optimization- and encoder-based methods. Since there is no guarantee that the photos we wish to replicate synthetically can be generated by \( G(\cdot) \), these methods typically yield disappointing results (Fig. 1). Moreover, groups that are under-represented in the training set cannot be reproduced accurately, lowering diversity and decreasing inclusion.

In our work we propose a third way: we allow the generator to be updated locally about the query latent vector. We optimize this update to obtain a faithful reproduction of the photo of interest. This yields better reconstruction accuracy while maintaining editability of the image.

3. Method

3.1. Problem definition

For a given query photo \( x \in \mathcal{X} \), we want to obtain its corresponding latent code \( z \in \mathcal{Z} \) that reconstructs \( x \) as accurately as possible, i.e., \( G(z) = x^* \), with \( \|x^* - x\| < \epsilon \).

Previous encoder- and optimization-based methods focus on optimizing \( h(\cdot) \) while keeping \( G(\cdot) \) frozen. To reconstruct \( x \), these previous methods have to operate under the assumption that \( x^* \) is on the manifold defined by \( G(\cdot) \), i.e., \( \exists z \in \mathcal{Z} \) s.t. \( G(z) = x^* \).

When \( x \) does not lie on or very close to the pre-trained manifold defined by \( G(\cdot) \), the best these methods can do is to retrieve its nearest projection \( \hat{x} \) as illustrated in Fig. 2. This figure shows an example where the query photo \( x \) is not on the manifold defined by \( G(\cdot) \), which is shown as an orange manifold. Thus, GAN inversion methods can at best synthesize the image \( \hat{x} \). \( \hat{x} \) is the image on the manifold defined by \( G(\cdot) \) that is closest to \( x \); here closeness is given by an orthographic projection onto the manifold. Our proposal
Figure 2: The query photo we wish to synthesize is given by $x$. Given a pre-trained GAN model, its generator function $G(\cdot)$ can, at best, synthesize image $\hat{x}$. That is, $\hat{x}$ is the point on the manifold that is closest to $x$. This paper proposes an algorithm to locally tweak $G(\cdot)$ to include an image $x^*$ s.t. $\|x - x^*\| < \epsilon$. We refer to the new tweaked manifold as $G^*(\cdot)$. The original $G(\cdot)$ is shown in orange and the tweaked $G^*(\cdot)$ in blue. The latent space is represented as a gray plane at the bottom.

is an algorithm that locally tweaks this manifold $G(\cdot)$ to include the image $x^*$, an image that is as close as possible to the query photo $x$. This is shown as a blue extension of the manifold in the figure.

3.2. Tweaking the manifold locally

Let us now derive the method to update the manifold defined by the generator locally.

Note we cannot simply modify the manifold $G(\cdot)$ in any random way that happens to include $x$. This is because in addition to including our query image $x$, the manifold should only change locally and in a way that allows us to edit the synthesized version of $x$ as easily as we edit any other synthetic image. In addition, we need to ensure that these edits yield synthetic image variants that look as realistic as the original photo.

To successfully edit the manifold locally, we first need to find $\hat{x}$, i.e., the closest point to $x$ we can find on the manifold. To this end, we can use any of the existing approaches described above. That is, we optimize $h(\cdot)$ by keeping $G(\cdot)$ fix.

Once we have $\hat{x}$, we fix $h(\cdot)$ and let $G(\cdot)$ change locally about $\hat{x}$ to include $x^*$, Fig. 2. Our goal is to make the smallest change possible while maintaining the desirable properties of the pre-trained $G(\cdot)$, e.g., we can edit images in a number of controllable ways.

We do this by combining two loss functions. The first

loss function $L_{\text{local}}$ is tasked to locally tweak the manifold to include $x$ by making the distance from $x$ to $x^*$ as small as possible and keeping the properties of the manifold intact. The second loss function $L_{\text{global}}$ ensures the rest of the manifold does not change.

3.3. Local loss

For the manifold defined by $G(\cdot)$ to generate $x$, there needs to be a latent vector $z$ s.t. $G(z)$ is as similar to $x$ as possible. We can compute this using a reconstruction loss function $L_{\text{recon}}(x_1, x_2)$ that measure the similarity between $x_1$ and $x_2$, with $x_i \in X$.

Because our goal is to synthesize an image that is as visually similar to the query photo as possible, we choose to use the Laplacian pyramid [3, 8] loss function as $L_{\text{recon}}(\cdot, \cdot)$. Note, however, that other similarity losses could be used.

Let $L_{\text{recon}} = \text{LaplacianPyramid}(x, G(z))$ be the reconstruction loss computed using the Laplacian pyramid, calculated by summing over mean L1 differences across all levels of a Laplacian pyramid of $x$ and $G(z)$. We can now find $z$ s.t. $G(z) = \hat{x}$ and then optimize $L_{\text{recon}}$ until $L_{\text{recon}} < \epsilon$.

To encourage that the tweaked manifold is editable and maintains all other desirable properties, we regularize this solution with an adversarial loss, $L_{\text{adv,local}}$. Specifically, we consider, $L_{\text{adv,local}} = \log D(x) + \log(1 - D(G(z)))$, where $D(\cdot)$ is the discriminator.

The combined local loss is thus given by,

$$L_{\text{local}} = L_{\text{recon}} + \lambda L_{\text{adv,local}},$$

(2)

where $\lambda$ is the regularizing term. Here too we find $z$ s.t. $G(z) = \hat{x}$ and then optimize $L_{\text{local}}$.

3.4. Global cohesion

We still need to ensure that the rest of the manifold does not change. This is to make sure that the model keeps any previous training and tweaking we have applied. We do this by computing a global loss function to enforce overall stability of the manifold.

To this end we use the loss of the pre-trained GAN model. For example, when using StyleGAN2, our global loss will be

$$L_{\text{global}} = \mathbb{E}_{x \sim p_x} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z))]),$$

(3)

where $x$, $z$, $p_x$ and $p_z$ are defined exactly as in the pre-trained GAN. If the generator architecture changes, it will be necessary to use the $L_{\text{global}}$ associated to that GAN model.

Putting everything together, the final loss function to optimize is given as

$$\mathcal{L} = \mathbb{1}[p][L_{\text{local}}] + L_{\text{global}},$$

(4)
Algorithm 1 Pseudo-code of Clone, the proposed GAN inversion algorithm.

Input: Query image \( x \), pre-trained model \( G(\cdot) \) and \( D(\cdot) \), the training set \( X_{train} \), hyper-parameters \( \{ p, \lambda, \epsilon \} \), an inversion function \( h(\cdot) \), maximum number of iterations \( T \).

Output: Synthesized image \( x^* \), generator \( G^+(\cdot) \).

1: Let \( z \leftarrow h(x) \).
2: Let \( G^{(0)} \leftarrow G \).
3: Let \( D^{(0)} \leftarrow D \).
4: for \( t = 1 \) to \( T \) do
5: \( u \sim \text{Uniform}[0, 1] \)
6: if \( u < p \) then
7: \( \text{Calculate} \ L_{\text{local}} \text{as in Eq. (2)} \)
8: else
9: \( L_{\text{local}} = 0 \)
10: Compute \( L_{\text{global}} \) as in Eq. (3).
11: \( L = L_{\text{local}} + L_{\text{global}} \).
12: \( G^{(t+1)} \) is updated with \( \partial L / \partial G^{(t)} \)
13: \( D^{(t+1)} \) is updated with \( \partial L / \partial D^{(t)} \).
14: if \( L_{\text{recon}} < \epsilon \) then
15: Stop
16: \( G^+ (\cdot) \leftarrow G^{(t+1)} (\cdot) \).
17: \( x^* \leftarrow G^+(z) \).

where \( \mathbb{1}_p \) is an indicator function activated with probability \( p \), with \( p \) generally kept small to attain good convergence and stability on \( G(\cdot) \). The lower the \( p \), the less frequent \( L_{\text{local}} \) term updates during training. For example, when \( p = 1/8, L_{\text{local}} \) will be updated once for every eight updates on \( L_{\text{global}} \).

The proposed GAN inversion method is summarized in Algorithm 1. We call this algorithm Clone since the synthesized image is a near clone of the real photo.

Once this process is completed, the updated generator \( G^+ \) is used to edit the image \( x^* \). Editing techniques that do not require training (e.g., StyleSpace [35]) can be directly applied on \( G^+ \). Methods that require model training (e.g., WarpedGANSpace [33], Image2StyleGAN [2]) can also be directly applied without re-training on \( G^+ \), since \( G^+ \) preserve most manifold structure as the pre-trained generator.

4. Experiments

In this section, we provide quantitative as well as qualitative comparisons of the proposed GAN inversion algorithms against state-of-the-art techniques.

4.1. Implementation details

We demonstrate the performance of the proposed GAN inversion algorithm on six datasets, Flickr-Faces-HQ Dataset (FFHQ) [20, 22], CelebA-HQ [17], Stanford Cars [23], LSUN-Cars [37], Animal Faces HQ in-the-Wild (AFHQ-Wild) [12], and LSUN-Horses [37]. The resolution of these images go from from 1, 024 × 1, 024 to 256 × 256 pixels.

The generator in StyleGAN2 uses a neural network architecture that receives a style latent code at each of its layers. To achieve a disentangled latent space, the StyleGAN2 architecture employs two decoupled network components referred to as the mapping network and the synthesis network.

A Normal distribution in \( Z \)-space is transformed to \( W \)-space via the mapping network, which is further extended to create \( W^+ \) [2]. In the experiments shown below, we recover the latent codes in \( W^+ \) space. And, in our experiments, we start with pre-trained StyleGAN2 models and apply the proposed GAN inversion algorithms detailed in Section 3.

In our experiments, we use \( p = 1/4, 1/8, 1/32, 1/16 \) for the Faces (FFHQ, CelebA-HQ), Cars (LSUN-Cars, Stanford-Cars), AFHQ and LSUN-Horses experiments, respectively. We use \( \lambda = 10 \) in Eq. (2) for Faces, Cars and AFHQ experiments and \( \lambda = 20 \) for LSUN-Horses experiment. These parameters are selected by a simple heuristic which is described in Supplemental File. When experimenting with additional values for these parameters, we obtained very similar results to those reported below. In all experiments, we set \( \epsilon < .001 \) and \( T = 100K \). Additional details are given in the Supplementary File.

The starting point for \( x^* \) is given by the Restyle encoder algorithm [4]. We have also tested the use of optimization-based algorithms to initialize \( x^* \). This yielded identical results.

4.2. Qualitative results on human faces

We first show our algorithm is able to solve the inversion problem of the target photos with extreme accuracy, even for out-of-sample photos, Fig. 3. The figure provides comparative results against Restyle [4], BDInvert [16], High-Fidelity GAN Inversion (HFGI) [34], and Ensemble (a hybrid optimization+encoder approach) [11].

As seen in Fig. 3, the proposed algorithm is able to synthesize image clones that are basically identical to the real photos. The synthesized images include difficult-to-reconstruct, high-frequency details such as hair and skin texture. (Figure results best seen at a 600% zoom.)

Second, to demonstrate that such accurate reconstructions come at no adverse effect on the ability to edit the synthetic clone, we show a number of image modifications using standard algorithms [35, 29], Fig. 4.

Specifically, we chose two state-of-the-art GAN editing techniques. The first is StyleSpace [35], which discovers dimensions in latent space that control specific image attributes such as eye gaze, hair color, skin tone, and mouth shape. The second is InterfaceGAN [29], which finds linear
traversals that modify a specific semantic attribute.

4.3. Diversity and inclusion

The biases of computer vision and machine learning models are well known and widely reported [27, 32, 7]. These are especially problematic when dealing with human faces due to our attachment of self-worth, identity and cultural values [25]. GANs are not immune from this problem, with pre-trained models carrying the biases of the photos used to define their training sets [15, 5].

The GAN inversion algorithm derived in this paper solves this important problem. Our algorithm is especially good at synthesizing out-of-sample data points. This means, the proposed algorithm successfully synthesizes photos of
Figure 4: The synthetic clones recovered by the proposed algorithm are readily editable using standard, off-the-shelf algorithms. While other GAN inversion methods require training their own traversal functions, our proposed solution allows for the use of standard, pre-computed traversals. Here, we show eight edits using StyleSpace [35] on the clone of the photo shown in Figure 3(a). Note the high photo-realism of these images.

people and cultures under-represented in the training set. In fact, the proposed inversion algorithm is still able to synthesize a near identical images to their query photos.

Fig. 5 shows several example image clones corresponding to out-of-sample photos. Note how the proposed algorithm is able to synthesize hairstyle, facial tattoos, and cultural jewelry/amulets not included in the training set of the pre-trained GAN model we used.

Importantly, and as shown in the figure, these clones are equally editable to those of in-sample groups and cultures.

4.4. Qualitative results on cars and animals

Fig. 6 shows qualitative results for cars on the Stanford Cars [23] dataset using StyleGAN2 models pre-trained on LSUN-Cars dataset [37]. In (a) we show the query photo we wish to synthesized. Results obtained with state-of-the-art methods are in (b-d). The synthesis results given by our algorithm are in (e). And, in (f), we show a couple edits applied to our synthesized images. As with faces, we see that the proposed approach yields results that are indistinguishable from the original photo.

Fig. 7 shows GAN inversion results of animals on the AFHQ-Wild and LSUN horses datasets. As with faces and cars, all StyleGAN2 models were pre-trained on the same training set and the results shown in the figure computed on an independent set of photos. These two datasets only have pre-train models for two of the GAN inversion models, Projection [22] and ReStyle [4]. Thus, Fig. 7(a) shows the original photo, Fig. 7 (b-c) these state-of-the-art results, and Fig. 7(d) the reconstructions given by the Clone algorithm derived in this paper. In (e), we show three example edits using StyleSpace [35] and the unsupervised approach of [30].

4.5. Quantitative results

Previous sections provided a number of qualitative comparative results on faces, cars and animals.

Table 1 shows the corresponding quantitative results. The first results in this table are computed on the CelebA-HQ [17] face dataset using StyleGAN2 models pre-trained on FFHQ [20]. The second results are on the testing set of the AFHQ-Wild [12] animal dataset using StyleGAN2 models pre-trained on the training set of the same database. The third results are on the testing set of the Stanford Cars [23] dataset using StyleGAN2 models pre-trained on the the training set of the same database.

We used two metrics to report our quantitative results in Table 1. The first is the Mean Squared Error (MSE). This is simply the norm-2 distance between the query photo and the image synthesized by each of the five state-of-the-art methods plus the algorithm presented in this paper. Results are the average over all testing images. Obviously, the lower the MSE, the better, with zero indicating the original photos and the synthesized clones are 100% identical. The second metric is the Learned Perceptual Image Patch Similarity (LPIPS) [38]. LPIPS computes the perceptual similarity of the synthesized image to the query photo. The perceptual similarity is calculated using a visual neural network model. As it is most commonly done, we use VGG [31]. As with MSE, the lower the value of LPIPS, the more visually similar the synthesized images are to the query photos.

As we see in the table, our proposed algorithm achieves MSE values that are at least an order of magnitude lower than those obtained by state-of-the-art methods. This is true regardless of the database and type of object we wish to synthesize. LPIPS confirms the visual similarity of our synthesized images to their corresponding query photos. Not all GAN inversion algorithms have a model available for comparison. When that’s the case, we indicate this with ‘–’ entries in the table.

We refer the reader to the Supplementary Files for additional quantitative and qualitative results.
Figure 5: State-of-the-art GAN inversion algorithms cannot accurately synthesize photos of people and cultures that are under-represented in the dataset used to pre-train the GAN. This is because $G(\cdot)$ is fixed and has not been trained to represent those groups and cultures. This leads to issues of diversity and inclusion. The algorithm described in this paper addresses these limitations by allowing local tweaks to $G(\cdot)$. In (a-b), we show results of our GAN inversion algorithm on photos of people with hair styles as well as facial scarification, tattoos, and cultural marks not well represented or not represented at all in the set used to pre-train the GAN. In (c-d), we show edits of the images synthesized by our approach using the algorithm of [35]. (e-g) shows edits using the algorithm of [29].

Figure 6: Inversion results on Stanford Cars [23]. (a) Original photo. (b-d) State-of-the-art results. (e) Our results. (f) A couple edits on our reconstruction using [35] and [30].

5. Assumptions and Limitation

Additionally, the almost perfect GAN inversion results shown above come at an additional small computational cost compared to previous methods. Since previous algorithms optimize $h(\cdot)$, their cost is associated to the number of iterations required to get good convergence. The approach proposed in this paper locally tweaks $G(\cdot)$. Note that this is not the same as re-training the GAN model. This local tweak is successfully completed in just a few iterations, taking typically several seconds to a few minutes. In
Figure 7: Reconstruction results of the real photos shown in column (a). Projection (an optimization method) [22] results are in column (b), ReStyle (an Encoder-based method) [4] results in column (c), and the results of the approach described in this paper in column (d). In (e), we show three image edits using StyleSpace [35] and the unsupervised algorithm of [30]. The first two rows show results on the AFHQ-Wild database [12]. The last two rows show results on the LSUN Horse dataset [37].

Table 1: Quantitative metrics of the proposed algorithm compared to results given by state-of-the-art baseline methods. MSE is the Mean Squared Error between the query photo and its synthesized copy as given by each of the listed algorithms. LPIPS (Learned Perceptual Image Patch Similarity) [38] computes the perceptual similarity between the original photo and the synthesized clones using a VGG network. LPIPS and MSE are averaged across samples. The GAN inversion method (optimization, encoder, hybrid) is specified as: E: encoder-based, O: optimization-based, H: hybrid method. ‘-‘ indicates that the pre-trained models for the corresponding datasets are not publicly available.

The worse cases, where our algorithm’s solution $x^*$ is significantly far from $x$, the Clone algorithm may take several minutes. This may limit the use of the proposed technique in applications that require close to real-time results.
6. Conclusion

GAN inversion is a hard problem, with limitations on which photos can and cannot be synthesized based on GAN architectures, loss functions, and training sets, among others. Here, we have defined a solution to these limitations by allowing the generator function \( G(\cdot) \) to be locally updated to include the image we wish to synthesize. Using extensive experimental results, we have shown that the proposed approach yields near perfect real photo reconstruction that are editable, with quantitative evaluations yielding results an order of magnitude (or more) better than current state-of-the-art algorithms.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4431–4440, 2019.
[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? CVPR, pages 8289–8302, 2020.
[3] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. \textit{RCA engineer}, 29(6):33–41, 1984.
[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In \textit{Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)}, October 2021.
[5] Guha Balakrishnan, Yuanjun Xiong, Wei Xia, and Pietro Perona. Towards causal benchmarking of bias in face analysis algorithms. \textit{arXiv e-prints}, pages arXiv–2007, 2020.
[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William S. Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. ICCV, pages 4501–4510, 2019.
[7] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In \textit{Proceedings of the European conference on computer vision (ECCV)}, pages 456–473, 2018.
[8] Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the latent space of generative networks. In \textit{International Conference on Machine Learning}, pages 600–609. PMLR, 2018.
[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In \textit{International Conference on Learning Representations}, 2018.
[10] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. \textit{ICLR}, 2021.
[11] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In \textit{CVPR}, 2021.
[12] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In \textit{Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition}, pages 8188–8197, 2020.
[13] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. \textit{IEEE Transactions on Neural Networks and Learning Systems}, 30:1967–1974, 2019.
[14] Raghudeep Gadde, Qianli Feng, and Aleix M. Martinez. Detail me more: Improving gan’s photo-realism of complex scenes. In \textit{Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)}, pages 13950–13959, October 2021.
[15] Eun Seo Jo and Timnit Gebru. Lessons from archives: Strategies for collecting sociocultural data in machine learning. In \textit{Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency}, pages 306–316, 2020.
[16] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations. ICCV, 2021.
[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In \textit{International Conference on Learning Representations}, 2018.
[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In \textit{Proc. NeurIPS}, 2021.
[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.
[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In \textit{Proceedings of the IEEE conference on computer vision and pattern recognition}, pages 4401–4410, 2019.
[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In \textit{Proc. CVPR}, 2020.
[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In \textit{Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition}, pages 8110–8119, 2020.
[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. \textit{2013 IEEE International Conference on Computer Vision Workshops}, pages 554–561, 2013.
[24] Zachary Chase Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. \textit{ArXiv}, abs/1702.04782, 2017.
[25] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In \textit{Proceedings of the ieee/cvf conference on computer vision and pattern recognition}, pages 2437–2445, 2020.
[26] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In \textit{Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition}, pages 14104–14113, 2020.
[27] Inioluwa Deborah Raji and Joy Buolamwini. Actionable auditing: Investigating the impact of publicly naming biased
performance results of commercial ai products. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pages 429–435, 2019.

[28] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. CVPR, 2021.

[29] Yujun Shen, Ceyuan Yang, Xiaou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE transactions on pattern analysis and machine intelligence, 2020.

[30] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In CVPR, 2021.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 2015.

[32] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In CVPR 2011, pages 1521–1528. IEEE, 2011.

[33] Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. WarpedGANSpace: Finding non-linear rbf paths in GAN latent space. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 6393–6402, October 2021.

[34] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. arxiv:2109.06590, 2021.

[35] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12863–12872, 2021.

[36] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. arXiv preprint arXiv:2101.05278, 2021.

[37] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arxiv preprint arXiv:1506.03365, 2015.

[38] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 586–595, 2018.

[39] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In ECCV, 2016.
Figure S1: Clones and edits using a StyleGAN2 model pre-trained on FFHQ. Leftmost column: Shown here are the originals and synthetic clones given by our proposed algorithm of a picture of Michelangelo’s statue of David, a drawing (sketch) of a human face, and Da Vinci’s Mona Lisa. Other columns: Edits of our synthetic clones using the methods of [1].
Figure S2: The proposed algorithm can be applied to images with multiple faces. Here, we use a face detector and a facial landmark detector to crop and align all faces. Next, our Clone algorithm is used to synthesize these faces, followed by edits given by off-the-shelf algorithms. Finally, the edited synthetic images are added to the original photo. (a) Original photo. (b) In this image, we have substituted the original faces by their synthetic clones edited to include eye glasses.
Figure S2 (Cont.): (c) Here, we have edited all faces to look angry. (d) And, in this example, we have edited all faces to have an open mouth (showing teeth), giving the impression of a ‘fake’ smile.
Figure S2 (Cont.): (e) In this last example, we have edited the eye gaze direction of the individuals in the photo, giving an impression of being distracted.

Figure S3: Additional qualitative results of our proposed Clone algorithm on images of Celeb-HQ using a StyleGAN2 model trained on FFHQ.
Figure S4: Qualitative results of our proposed Clone algorithm on under-represented sample images downloaded from the internet using a StyleGAN2 model trained on FFHQ.
Figure S5: Qualitative results of our proposed Clone algorithm on under-represented sample images downloaded from the internet using a StyleGAN2 model trained on FFHQ.
Figure S6: Synthetic image reconstructions as given by state-of-the-art GAN inversion methods and our proposed algorithm. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S7: Synthetic image reconstructions as given by state-of-the-art GAN inversion methods and our proposed algorithm. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S8: Synthetic image reconstructions as given by state-of-the-art GAN inversion methods and our proposed algorithm. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S9: Synthetic image reconstructions as given by state-of-the-art GAN inversion methods and our proposed algorithm. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S10: Synthetic image reconstructions as given by state-of-the-art GAN inversion methods and our proposed algorithm. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S11: Additional comparative qualitative image synthesis and edits on images of animal faces. These images are from AFHQ-Wild. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S12: Additional comparative qualitative image synthesis and edits on images of horses. These images are from LSUN-Horses. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
Figure S13: Additional comparative qualitative image synthesis and edits on images of cars. These images are from Stanford-Cars. Below each reconstruction, we show the pixel-to-pixel (2-norm) difference between the original photo and the synthesized image.
S1. Supplementary Documentation Overview

This Supplementary File provides additional details of the proposed algorithm and experimental results. We also provide additional results and illustrate the use of the proposed approach on novel applications. Sec. S2 provides additional details on the training of the models used in the main paper. Sec. S3 describes the heuristics used for hyper-parameter selection. Sec. S4 further details the assumptions and limitation of our algorithm. Sec. S5 discusses ethical and potential societal impact of the work. Figs. S1 to S13 provides additional qualitative results for real photo inversions on human faces, cars, animal faces and horses.

S2. Model Training Details

In the Faces experiment, we use a StyleGAN2 model pre-trained on FFHQ dataset. The quantitative evaluation is performed on a set of 500 CelebA-HQ images selected at random with no pre-processing. In our qualitative evaluation, we show inversions and editings on out-of-sample images selected from the internet. In these images, we first detect the face and estimate the facial landmarks using Dlib facial landmarks estimator. The face is then scaled, translated and rotated to an upfront canonical position. Then, we apply a Gaussian blur around the image boundary to emphasize the face. The resolution of the images used in this experiment is $1024 \times 1024$ pixels.

In the Cars experiment, we use a StyleGAN2 model trained on LSUN-Cars. Our test is based on 500 images from the Stanford Cars database selected at random, with no pre-processing performed. The results provided by the ReStyle encoder are done at a $512 \times 384$ pixel resolution, while the generator is pre-trained to produce images at $512 \times 512$ pixels with top-bottom zero-padding. We follow the same practice which means that the $L_{\text{recon}}$ is performed only on the $512 \times 384$ center crop while both $L_{\text{adv-local}}$ and $L_{\text{global}}$ are applied on the full $512 \times 512$-pixel image (with padding).

In the Animal Faces experiment, we use a StyleGAN2 trained on the AFHQ-Wild training set and test on the full AFHQ-Wild test set (500 images), with no pre-processing. The image resolution in this experiment is $512 \times 512$ pixels.

In the Horses experiment, we use a StyleGAN2 model trained on the LSUN-Horses dataset and randomly sample 500 testing images from the full set. No pre-processing is performed. The image resolution in this experiment is $256 \times 256$ pixels.

S3. Hyper-Parameter Selection

As described in the main paper, we use different sets of hyper-parameter values for different experiments. A value of StyleGAN2 hyper-parameter $\gamma$ is also necessary as in the original StyleGAN2 training. Tab. S1 shows the parameter values used for each experiment shown in the main paper.

| Experiment                  | $p$ | $\lambda$ | $\gamma$ |
|-----------------------------|-----|-----------|----------|
| Faces (CelebA-HQ, FFHQ)     | 1/4 | 10        | 10       |
| Cars (Stanford-Cars, LSUN-Cars) | 1/8 | 10        | 5        |
| AFHQ-wild                   | 1/32 | 20         | 15       |
| LSUN-Horses                 | 1/16 | 10        | 10       |

Table S1: Hyper-parameter values used in the experiments. $\gamma$ (gamma) is a StyleGAN2 specific parameter.

The heuristic we used to select the parameters is:

1. Initialize with $p = 1/8$, $\lambda = 10$ and the value of $\gamma$ equal to that in the pre-training model.
2. When both FID and reconstruction error are high, change GAN model specific hyper-parameters (e.g., gamma in StyleGAN2)
3. When FID is low but reconstruction error is high, increase $\lambda$.
   - if this leads to increase FID, then decrease $p$.
4. When reconstruction error is low but FID is high, decrease $p$.
   - if this leads to increase reconstruction error, then increase $\lambda$.

S4. Assumptions and Limitation

The proposed approach assumes that the pre-trained GAN model is able to generate photo-realistic images. If the GAN model is not able to generate photo-realistic images or has not been properly trained, the algorithm defined in this paper will not be able to find a synthetic clone.

Additionally, the almost perfect GAN inversion results shown above come at an additional small computational cost compared to previous methods. Since previous algorithms optimize $h(\cdot)$, their cost is associated to the number of iterations required to get good convergence. Optimization-based approaches generally require more iterations and, hence, are typically more expensive than encoder-based methods which require only one or a few iterations. In practice, this means optimization-based approaches may take up to a few minutes to solve the GAN inversion problem whereas encoder-based approaches may provide results in a few seconds of less.

The approach proposed in this paper, locally tweaks $G(\cdot)$. It needs to be noted that this is not the same as retraining the GAN model. This local tweak is successfully
completed in just a few iterations, taking typically several seconds and up to a few minutes. In the worse cases, where our algorithm’s solution \( \mathbf{x}^* \) is significantly far from \( \mathbf{x} \), the Clone algorithm may take several minutes. This may limit the use of the proposed technique in applications that require close to real-time results.

**S5. Potential Societal Impact**

This paper poses a similar societal impact as other GAN inversion techniques or realistic image editing methods.

On the one hand, the near perfect inversion and editing can be used for editing memorable photos, helping people re-experience the moments they treasured. The method can also provide a chance to correct unsatisfactory poses and expressions after a photo has been captured, especially when the photo cannot be easily re-taken. We believe these new editing opportunities provided by our algorithm can be used to save time and making valuable memory more vivid and lively. There are also a number of potential applications in film and e-commerce that were not possible with previous GAN inversion algorithms due the lack of almost-perfect reconstructions. Moreover, as described in the main paper, our method can reconstruct photos even if the people in the photo are under-represented in the training set.

On the other hand, the method can be used to manipulate photos to spread misinformation by changing eye gaze, facial expressions and other semantic attributes. The negative aspect of photo editing is not a new concern, as realistic editing can be already achieved using “Photoshoping” and other computer graphics, computer vision and machine learning techniques. Thus, we think it is important to develop algorithms that can tell these edited images apart. AI regulations may also be warranted.

**S6. Image Attribution**

This paper uses and edits the images with Creative Commons license described in Tab. S2.

**References**

[1] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021.
| URL                                                                 | Authors         | License       |
|--------------------------------------------------------------------|-----------------|--------------|
| https://commons.wikimedia.org/wiki/File:M%C3%A9lanie_de_Jesus_dos_Santos_-_Glasgow_2018_-_01_cropped.jpg | TwoWings        | CC-BY-SA-4.0  |
| https://commons.wikimedia.org/wiki/File:Alice_in_Chains__2019158181605_2019-06-07_Rock_am_Ring__1163__B7018463.jpg | Sven Mandel     | CC-BY-SA-4.0  |
| https://commons.wikimedia.org/wiki/File:Bahjat_Artist_.jpg         | Teambahjat      | CC-BY-SA-4.0  |
| https://commons.wikimedia.org/wiki/File:Young_Woman_28Adriana29_at_Tourist_Office_28Cachoeira_29_Bahia_28Brazil_.JPG | Adam Jones²     | CC-BY-SA-3.0  |
| https://commons.wikimedia.org/wiki/File:Maori_Man_28Imagicity_103429.jpg | Graham Crumb³   | CC-BY-SA-3.0  |
| https://commons.wikimedia.org/wiki/File:Alethea_Arnaquq-Baril_in_Toronto.jpg | Aarnaquq       | CC-BY-SA-4.0  |
| https://live.staticflickr.com/65535/49535/4953727621_0e254349bb_o_d.jpg | Rod Waddington  | CC-BY-SA-2.0  |
| https://live.staticflickr.com/65535/49535/49542075471_9fde3d42a5_o_d.jpg | Rod Waddington  | CC-BY-SA-2.0  |
| https://www.maxpixel.net/Warrior-Man-Strong-2152851 | Max Pixel       | CC0 Public Domain |

Table S2: Image license information