MAGECally invert images for realistic editing

Asya Grechka[12]
asya.grechka@lip6.fr
Jean-Francois Goudou[1]
jean-francois.g@meero.com
Matthieu Cord[2]
matthieu.cord@lip6.fr

[1] Meero
Paris, France

[2] Sorbonne Université
Paris, France

Abstract

Generative Adversarial Networks (GANs) are now able to generate astonishingly realistic high-resolution images. Recent work has shown the emergence of semantically-meaningful manipulations simply by editing the corresponding latent vector. However, a real image must first be inverted into its GAN latent code before editing. Previous work usually achieves accurate reconstruction, but poor-quality latent vectors: applying known editing methods onto these latent codes results in artifacts and erroneous edits.

We aim to bridge the gap between reconstruction and editability. We propose a novel instance-optimization based inversion method, which specifically aims to maximize the semantic information of the latent vector, all while producing an accurate reconstruction. We introduce the i**MAG**e-lat**E**nt **C**onsistency loss ("MAGEC"), which allows supervision in the latent space, encouraging editability of the resulting latent vector. We provide extensive qualitative and quantitative evaluation to validate our method, using the recent state-of-the-art *StyleGAN* and show that our method outperforms baseline inversion methods, opening the door to new realms of real-image editing.

1 Introduction

Editing real images in a fast and realistic way is highly lucrative for obvious reasons. Users could use such a tool to visualize their personal curiosities before committing more time or energy to something more permanent - How would a certain hairstyle look on me? How would my house look with brick walls? Would my car look nice with different tires? Moreover, a professional photographer spends around 1.5 hours of editing for a portrait image, with the most costly operations being those requiring complicated semantic changes [1]. *Automatic semantic editing* is thus an application which could interest and benefit many.

Generative Adversarial Networks (GANs) [12] have made such ideas possible in recent years. Recently, the celebrated StyleGAN [19, 20] has allowed unconditional generation of unparalleled quality, resolution (1024×1024) and realism. What's more, this GAN was the first of its kind to show such powerful *disentanglement qualities* of its latent space. For example, mixing two latent codes corresponding to two images results in a coherent output image containing properties from both former images [19]. This led to a surge of research into studying and manipulating this latent space, in which high-level semantic concepts of

Figure 1: From left to right: original image, projection in latent space using MAGEC loss, various edits (GANSpace, InterfaceGAN and StyleFlow respectively). Edits are of high-quality, and conform to expectations of the various editing methods. Best viewed zoomed.

generated images can be edited [4, 9, 16, 30, 31, 34, 41]. These editing methods produce realistic and high-quality global or local changes in the output image.

The extension to real images is natural but not trivial. A latent code first needs to be found such that, when inserted into the *StyleGAN* network, outputs the original image. Although inversions are able to achieve high reconstruction quality [7, 8], problems arise when applying known editing methods onto the inverted images: the editing either does not work at all, or they produce output images of poor quality, presenting artifacts and blur [37, 39].

In this paper, we propose a learning framework to improve editability of the projected latent vector of a real image while maintaining good reconstruction. We reframe the strategy of the standard optimization framework for GAN inversion. More precisely, we add a term of editability and image-latent-consistency into the global loss function by using a simple yet effective procedure based on recent latent-manipulation methods. This allows us to explicitly incorporate known editing methods into the GAN inversion optimization scheme to ensure better editability. We validate our method with extensive qualitative and quantitative evaluation. Notably, we introduce a novel "edit-consistency score" which specifically evaluates the quality of a projection method in terms of editability. We show that our method outperforms existing baseline methods, and opens the door to new possibilities of editable high-quality image inversions.

2 Related Work

Generative Adversarial Networks Generative Adversarial Networks (GANs) [12] have sparked massive interest since their introduction in 2014. In recent years, breakthrough research in architecture design, loss functions, and training dynamics have allowed unconditional GANs to synthesize images of unprecedented quality for resolutions up to 1024×1024 [7, 18, 19, 20]. BigGAN [6] has likewise made a pushed state of the art by allowing high-quality class-conditioned generation on ImageNet. Image-to-image translation models like [17, 27, 33] typically all use GANs in some form. In this paper, we will work with the current state-of-the-art generator network StyleGAN2 [19, 20], whose novel architecture notably in-

cludes a separate mapping network which transforms the normal distribution $\mathcal{Z}$ space into the more disentangled $\mathcal{W}$ space, which not only permits higher-quality image-generation, but also a much smoother, more interpretable and semantically-rich latent space.

Latent Space Manipulation With the introduction of StyleGAN and its newly disentangled latent space, a surge of research has come forward to provide various ways of interpreting and manipulating this latent space $\mathcal{W}$. In the original StyleGAN paper, the authors showed that simply performing linear interpolation between two latent vectors gives an "interpolated" image between the two former vectors. Another straight-forward approach is to find linear directions in the latent space for binary attributes learned in a fully-supervised manner [11, 30, 41]. By using an auxiliary classifier in the image space, we can find the desired linear boundaries in the latent space, and then simply "walk" in the direction of a given attribute to change the output image. StyleFlow [4] likewise uses an auxiliary classifier for ground-truth labels, but instead uses a flow network to learn non-linear directions in the latent space, which allows for better preservation of unedited attributes as well as sequential edits. GANSpace [16] doesn't use an auxiliary classifier, but instead performs PCA analysis on the latent space. StyleRig [31] and [37] use a pre-trained StyleGAN network along with a 3D differentiable renderer in order to perform 3D edits on generated images via latent space manipulations. Recently, [26] explicitly models the generator network to disentangle 3D properties, allowing its latent code to perform powerful changes in the 3D space.

Latent Space Embedding The embedding of a real image into the latent space of a deep network has a long history of research and is generally dealt a few ways. Instead of using a pre-trained network, networks such as Variational Auto-Encoders [22] (VAEs) contain native encoders directly in their architecture, and are trained jointly with the decoder. Although variants of these are used for image editing [5, 28], these networks still don't have the powerful semantic information like StyleGAN does.

Another possibility is to train a new encoder from scratch for a previously-trained GAN. There has been much work in recent years applying such encoders to StyleGAN, requiring a very specific architecture that mimics the StyleGAN's hierarchical structure [29, 32, 35]. Encoder-based inversion methods have the advantage of giving instantaneous results, but often inadequate reconstructions. Only recently has there been any interest in encoding an image specifically for the aim of editing the final image. The task requires that the resulting latent vector be as "in-domain" as possible. [39] addressed this by using a novel encoder, trained in an adversarial manner, to encourage predicted latent vectors to follow the original latent distribution. Recently, [32] proposed its own "encoder for editing" which builds on the StyleGAN2-specific pSp [29] encoder, but is also trained in an adversarial manner along with StyleGAN2-specific loss functions. Although the resulting latents are indeed more editable, reconstruction is typically compromised.

Finally, the third possibility is to use an instance-based optimization method, still the most classic method for GAN inversion [2, 3, 13, 19, 40]. For one given image, we aim to find the corresponding latent code by optimizing the latent code directly. Although costly in nature, [2, 3] have shown that virtually any image can be correctly reconstructed in the extended $\mathcal{W}+$ space of StyleGAN. Nevertheless, these optimized latent codes respond poorly to known editing methods, showing that the obtained latent code is out-of-domain [39]. This is because the instance-optimization problem is poorly constrained, and overfits to the input image at the expense of an out-of-domain latent code.

This observation is the basis of our paper, which aims to constrain the problem specifically to ensure image editing, and thus, to ensure a more "in-domain" latent vector. As far as we know, our work is the first to address this question through instance-based optimization.

3 Methodology

GAN-Inversion Framework Given a pre-trained GAN network (generator G, discriminator D) and a real image $x_{real} \in \mathbb{R}^{C\times H\times W}$ such that: $G : z \in \mathbb{R}^d \longrightarrow x_{gen} \in \mathbb{R}^{C\times H\times W}$. The GAN-Inversion goal is to find a latent code $z_{inv} \in \mathbb{R}^d$ such that: $G(z_{inv}) = x_{rec} \approx x_{real}$. For any real x_{real} image, the instance-based optimization problem for GAN inversion is:

1. Initialize $z = z_0$. This can be a random latent vector [2], an "average" latent vector z_{avg} [2, 3, 19], or the output of a pre-trained encoder on x_{real} [39, 40].

2. Minimize over z a loss $\mathcal{L}$ between the synthesized image $G(z)$ and the original one x_{real}. The basic scheme works with a $\mathcal{L} = \mathcal{L}_2$ loss for the reconstruction, recently improved with some sort of perceptual loss $\mathcal{L}_{percept}$. The current general framework minimizes the following loss:

$$z_{inv} = \min_z \; \mathcal{L}_2(G(z), x_{real}) + \lambda \mathcal{L}_{percept}(G(z), x_{real}). \tag{1}$$

Image2StyleGAN++ [3], the current reference for inversion based on instance-based optimization, uses two perceptual losses of an ImageNet pre-trained VGG-16 network. Recently, it has been observed that using the LPIPS [36] perceptual loss gives more robust results [13, 29], as well as adding an "ID" loss which aims at preserving identity [29, 32].

This classic GAN-inversion framework fares poorly when known editing methods are applied to the inverted latents (see Fig. 3) largely due to the latent vector overfitting to the image space. Our goal is to preserve accurate reconstruction, but improve editability of the latent vector. We solve this using a 2-part strategy. First, like recent latent-manipulation methods, we learn the structure of the latent space with respect to image features. Second, and unlike previous work, we explicitly use this learnt structure to constrain our optimization process. The method is summarized in Fig. 2: the novel projection strategy continues to optimize the image-space loss of equation 1 (shown in pink), but now also explicitly optimizes at the latent-level (shown in blue). With this new modeling, we can explicitly add editability directly to the loss term. As we can see with in Fig. 1, this allows diverse, high-quality edits of our projected latent vectors all while maintaining high reconstruction fidelity.

3.1 Latent-Space Supervision

The lack of editability of previous optimization-based methods reveal that the latent vector overfits to the image space. We thus aim to supervise the latent vector directly through the latent space. Inspired by work on latent-space manipulation [2, 16, 30, 31, 41], we also *link* the latent space with the image space, but here with the explicit goal to supervise our loss.

Consider a pre-trained deep network F which inputs an image and outputs some kind of image descriptor d. This could be attributes, keypoints, segmentation map, etc.

As we know from previous work [35], the latent space of recent style-based generative models has high discriminative capacity. Our method aims to link the image descriptors to the latent vector with the simplest network possible so as to avoid any form of "re-learning" some part of the GAN network (and thus keep supervision only at the latent-level). We generate N image-latent pairs and train a simple linear model *LinkNet* which predicts image descriptors d from the latent vectors z.

Concretely, we train *LinkNet* using the same exact loss as the deep image descriptor network F was trained with, using the predictors of F as the ground-truth labels. This simple *LinkNet* will be sufficient for supervising our latent vector directly in the latent space.

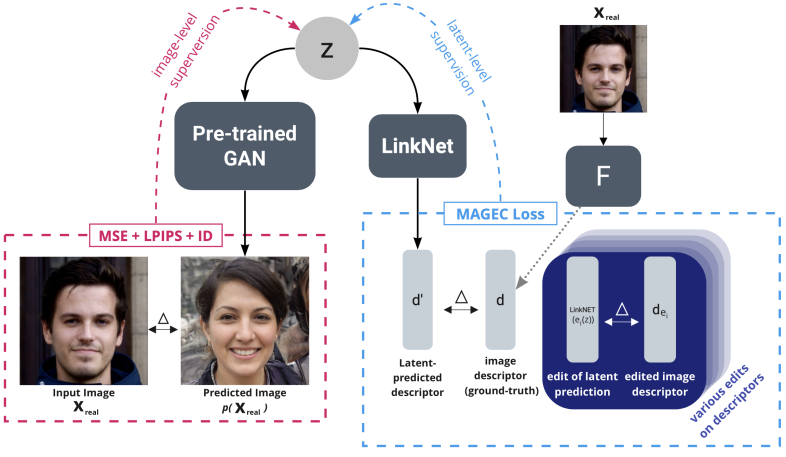


Figure 2: Our proposed optimization framework. We initialize our z vector with z_0 (the "average" latent vector). Then, we generate the associated image with the pre-trained generator to obtain the image-level loss. We predict the latent-level descriptor d' with our pre-trained *LinkNet*. We add a consistency loss of these features with the "ground-truth" features d evaluated from our feature-extractor network F. We add another consistency loss over the edited descriptors (using a differentiable editor e) and the "ground-truth" edits which we obtain by modifying d. This MAGEC loss gives us latent-level optimization which promotes editability and an in-domain output vector.

3.2 MAGEC Loss

Now that we have linked our image and latent space with the simple *LinkNet*, we can define our i**MAG**e-lat**E**nt **C**onsistency "MAGEC" loss in the following way, built from Eq.1.

First, we add an image-latent consistency loss over the input image and the latent to optimize. We obtain the "ground-truth" image descriptor d using F, then we use our *LinkNet* to predict the image descriptor d' from latent vector z. If the latent vector correctly represents the image, it should be able to predict the associated image predictors.

Second, we add an image-latent edit consistency loss. Let e be a differentiable known editor with i editing operations (for example, an editing operation can be *add glasses*, *remove wrinkles*, *to woman*, etc.). We perform i edits of the latent vector $e_i(z)$ and edit the "ground-truth" image descriptor d accordingly to obtain d_{e_i}. Our edited latent vector should be able to predict d_{e_i} (with *LinkNet*). See Fig. 2 for detailed illustration.

The MAGEC loss is given as follows:

$$\mathcal{L}_{MAGEC}(z, x_{real}) = \underbrace{\mathcal{L}_F(LinkNet(z), d)}_{\text{image-latent consistency}} + \underbrace{\frac{1}{|edits|} \sum_{i \in edits} \mathcal{L}_F(LinkNet(e_i(z)), d_{e_i})}_{\text{image-latent edit consistency}} \quad (2)$$

where $d = F(x_{real})$ and d_{e_i} is the modified d according to edit i.

Our final loss is

$$\begin{aligned}\mathcal{L}(z,x_{real}) = \lambda_{MSE}\mathcal{L}_2(G(z),x_{real}) + \lambda_{LPIPS}\mathcal{L}_{LPIPS}(G(z),x_{real}) \\ + \lambda_{ID}\mathcal{L}_{ID}(G(z),x_{real}) + \lambda_{MAGEC}\mathcal{L}_{MAGEC}(z,x_{real})\end{aligned} \tag{3}$$

where $\mathcal{L}_{ID}$ is the *ID loss* introduced in [29, 32] based on a pre-trained ArcFace [10] network (or ResNet-50 [14] network trained with MOCOv2 [8] for non-face images).

As we can see in Fig. 2, our framework allows dual supervision, simultaneously in the image and latent spaces. We are able to provide latent editing directly in the loss term, which not only allows to generate editable latents, but also helps to visualize which images are inherently out-of-domain for the GAN in question. Moreover, this framework can be incorporated into any pre-trained GAN, and allows potential further constraint by combining several differentiable editing methods together.

4 Experiments

Configurations We use StyleGAN2, pre-trained on FFHQ for 1024×1024 output resolution as our pre-trained GAN. We use the mapped latent space $\mathcal{W}$ and similarly to [2, 3, 29, 32, 39], we extend this space to $\mathcal{W}+$ by allowing each of the 512-dimensional *style vectors* to be independent of each other.

For the feature extractor F, we train an attribute-classifier on CelebA [25] to predict 40 binary attributes. Finally, we use InterfaceGAN [30] as our editor e to supervise our MAGEC loss, which performs add/remove operations on all 40 attributes.

Training Protocol We initialize z with z_{avg}, by sampling $N = 50000$ latent vectors with StyleGAN2 and taking the average vector. We use the Adam[21] optimizer with the default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e^{-08}$). We perform our training in two parts. First, we use $\lambda_{MAGEC} = 3e^{-4}$, $\lambda_{LPIPS} = 5e^{-1}$, $\lambda_{MSE} = 1e^{-3}$ and $\lambda_{ID} = 3e^{-4}$ and perform 100 optimization steps. Then, we set $\lambda_{MSE} = \lambda_{LPIPS} = 5e^{-1}$ for another 100 optimization steps, leaving the other loss coefficients unchanged. The learning rate begins at 0.07 and is exponentially decayed with a decay factor of 0.8 every 25 epochs. [1]

Datasets and Editors We perform thorough evaluation of our method using 1000 random samples from the CelebAHQ [18] dataset. We also evaluate our method on random images from the Stanford Cars dataset [23], which we provide in the supplementary material. We evaluate our projection on four well-known editing methods: InterfaceGAN [30], GANSpace [16], StyleFlow [4] and random interpolation between latent vectors [19].

Baselines We compare our method with Image2StyleGAN++ [3], the current state-of-the-art optimization-based method for GAN-Inversion. This method uses the classic loss from Eq. 1 (using the LPIPS[36] loss as the perceptual loss) and first optimizes the latent vector for 1000 iterations, then the noise vector for 1000 iterations. This loss can be seen as an ablation of the MAGEC and ID losses from Eq. 3, but minimized over more iterations. We also perform an ablation study using our training protocol to optimize Eq. 3 but without the MAGEC loss. All methods initialize z with z_{avg}.

[1] Our two-steps procedure is motivated by the fact that instance-based optimization often fails at adding semantic information in the latent vector with small optimization steps. When placing a higher weight on λ_{MAGEC}, we direct our latent vector z to a strong point which captures this semantic information. Once the latent vector is within the correct "area" of the latent space, we can give more relative weight to the loss coefficients related to image reconstruction.

Figure 3: Comparison of our method vs Im2StyleGAN++. Left: original images. Top rows: Im2StyleGAN++ projection. Bottom rows: Our projection with MAGEC. Edits are made using GANSpace. While Im2StyleGAN++'s projection is accurate, edits present strong artifacts or absurdities. Our reconstructions are also accurate, but react correctly to editing operations, suggesting that it follows the native distribution of StyleGAN more closely.

	MSE ↓	LPIPS ↓	Nb opt. steps ↓	Time (s) ↓
Full Method	0.0040	0.053	200	34.6
w/o MAGEC	0.0094	0.062	200	29.7
w/o MAGEC + extra opt. steps	0.0078	0.050	300	44.5
Im2StyleGAN++	0.0012	0.018	2000	242.2

Table 1: Reconstruction evaluation of projection, showing averages per image. We should note that the time of our method directly depends on the number of attributes we add to our MAGEC loss (Eq. 2). Here, adding editing constraints for 40 attributes leads to a cost of about 5 seconds per image.

4.1 Qualitative Results

Fig. 1 shows our method and various edits applied on notorious figures. As we can see in Fig. 3, our method visually significantly outperforms Im2StyleGAN [3] in terms of editing quality. While [3] produces artifacts and blur during edits, our method produces sharp, realistic edits. We further provide a host of edits in the supplementary material to further convince the reader of our method.

4.2 Quantitative Evaluation

Reconstruction Tab. 1 shows the various metrics for reconstruction. Notice that adding our MAGEC loss in Eq. 3 leads a better reconstruction. Even when allowing 50% more iterations, our method still performs on par in terms of reconstruction. The MAGEC loss could thus be seen as a prior which speeds up optimization. As expected, [3] performs excellent reconstruction, given the time (over 4 minutes) and the under-constrained loss function.

Editability We perform random edits on the 1000 images to obtain 20000 edited images per projection method. The semantic editability of the inverted latent vector is of utmost importance when performing GAN Inversion, but there is no standard metric for measuring this.

	InterfaceGAN		StyleFlow		GANSpace	Interpolations
	realism	*t*	*realism*	*t*	*realism*	*realism*
Im2StyleGAN++	0.973	0.096	0.929	**0.211**	0.960	1.00
w/o MAGEC loss	**0.994**	0.097	**0.976**	0.148	**0.985**	1.03
Full Method	**0.998**	**0.122**	**0.982**	**0.202**	**0.984**	**1.04**

Table 2: *Realism scores* and "improved target" scores of random image edits. For reliable interpretation, we perform a paired Student's t-test between our method's metrics and the competing method. The bold values are in line with this significance (p-value < 0.05). Our method consistently produces realistic and coherent images for the task at hand.

We first evaluate using common metrics before introducing our novel "editability score".

We aim to evaluate the realism and "coherence" of a given edit. The FID score [15] is not adapted to measure the quality of the edits, firstly because the original sample size (1000 original images) is well-below the recommended 50,000 needed for an accurate FID score, and secondly because our edits inherently lack diversity (edits of the same image resemble each other). Instead, we use the *realism score* [24], which evaluates an image instead of a distribution. This is a nearest-neighbor based method (higher is better) in which the *realism threshold* is set to 1 (a score above 1 is a realistic image).

To measure the "coherence" of an edit, we calculate a simple "improved-target" score t to measure the net difference of the predicted attribute probability before and after the attribute-targeted edit by using a pre-trained attribute classifier. A higher value means that the attribute prediction increased after applying the editing method, meaning it reacted accordingly. Note that $t \in [0,1]$. Remark that this metric is only applicable to the editors which make attribute-specific edits (InterfaceGAN [30] and StyleFlow [4]).

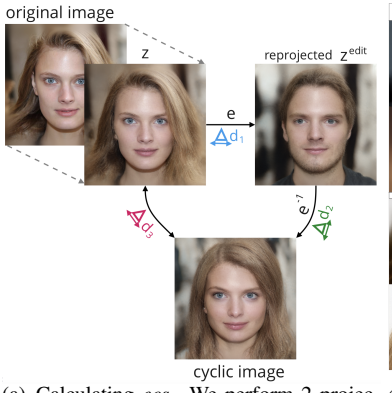
For reliable interpretation of these metrics, we performed a paired Student's t-test on our method and a competing method, as the edit operations were the same for all projected images. Our results are summarized in Tab. 2. As we can see, our method performs well on these metrics, always among the best in terms of realism or "coherence". However, these results are not entirely conclusive, and we investigate a better metric in order to evaluate the editability of a given inversion.

Edit-Consistency Score We introduce a new metric which aims at measuring the projection in terms of editability (see Fig. 4(a)). We first use our projection method p to obtain vector z from x_{real}. Then, we use a known editing method e to edit the vector with respect to a certain attribute, giving us x_{edit}. Then, we re-project it into the latent space to obtain z'_{edit}. Finally, we apply the inverse editing method onto z'_{edit} to obtain a cyclic image x_{cyc}, which should ideally match the initial projection. We define the edit-consistency loss as follows:

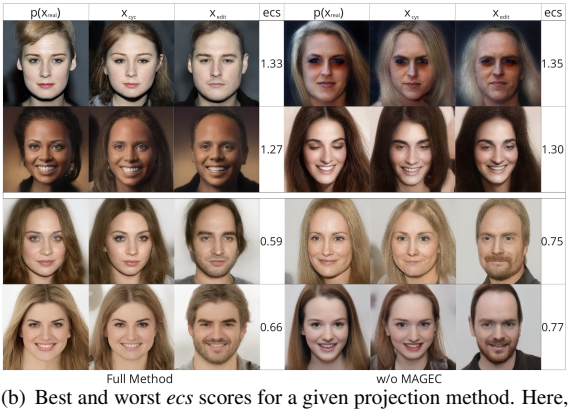
$$ecs(p, x_{real}) = \frac{2 \times \mathcal{L}_{LPIPS}(p(x_{real}), x_{cyc})}{\mathcal{L}_{LPIPS}(p(x_{real}), x_{edit}) + \mathcal{L}_{LPIPS}(x_{edit}, x_{cyc})} \tag{4}$$

The intuition is that the cyclic image should resemble the projected image, but images should also react accordingly to editing methods. Remark that a "perfect" *ecs* score is 0. An *ecs* of 1 can be seen as a "quality" threshold, since $ecs > 1$ means that the distance $\mathcal{L}_{LPIPS}(p(x_{real}), x_{cyc})$ is larger than one of the two edit operations. See Fig. 4(b) for examples of varying *ecs* scores. We can now define the edit-consistency score of a projection :

$$ECS(p) = \frac{1}{|X|} \sum_{x \in X} ecs(p, x). \tag{5}$$



(a) Calculating *ecs*. We perform 2 projections per *ecs*: 1 from the original image, and one from the edited latent. A better (lower) *ecs* means that d_3 is smaller than d_1 and d_2



(b) Best and worst *ecs* scores for a given projection method. Here, the editing method is *to male* with GANSpace. Notice how the worst scores correspond to poorer editability, for example, the woman in the second row on the right did not transform into a man.

Figure 4: Edit Consistency Score - Illustration and Intuition

Tab. 3 compares ECS results between our method and the two baselines. Importantly, notice how our method gives better scores for an editing method not utilized to supervise the loss (GANSpace), suggesting that the latent vector doesn't overfit to one editing method, but is encouraged to become "in-domain".

	InterfaceGAN		GANSpace	
	Male ↓	*Smile* ↓	*Male* ↓	*Smile* ↓
Im2StyleGAN++	0.97	1.00	1.07	0.98
w/o MAGEC	1.01	0.95	1.06	0.90
Full Method	**0.84**	**0.87**	**0.95**	**0.79**

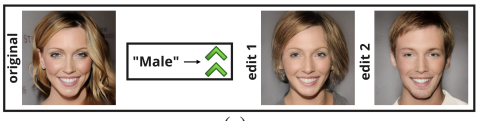
Table 3: *ECS* evaluation. Our MAGEC loss significantly improves *ECS* for all edits, notably ones not used to supervise our loss (GANSpace). Scores are evaluated on images not containing the target attribute.

Human Evaluation While automatic quantitative methods allow quick comparisons to baseline methods, many have observed [53] that human judgment is still the most reliable metric for evaluating image quality. We thus performed a user study in which experts of photography were asked to judge photo edits between each other, each one corresponding to a different projection method. See Fig. 5 for details. Tab. 5(b) shows the results, showing that our method was strongly preferred over the baseline methods.

5 Limitations of our Method

Our method aims at producing an editable latent vector, which often means a vector within the domain of the pre-trained GAN. Using real images that are clearly out of domain produces low-quality results, both in terms of reconstruction and editability. This is because the trade-off between these two objectives is too strong, and our method struggles to find a suitable compromise. See the top row of Fig. 6 for an example.

Our method also fails when faced with rare or challenging semantics. The MAGEC loss is not sufficient to push the z in the ideal direction, and some semantic concepts become lost. As we see in the bottom row of Fig. 6, the keffiyeh becomes semantically represented as hair.



(a)

Displayed Methods	% Preferred Ours
Ours vs. w/o MAGEC	72%
Ours vs. Im2StyleGAN++	78%

(b)

Figure 5: User Study: 30 photography experts judged edit operations resulting from 3 projection methods: (1): Our full method, (2): Ablation of MAGEC loss from our Eq. 3, and (3): Im2StyleGAN++[3]. Fig. 5(a) shows the interface in which the expert had to choose their preferred edit according to quality and coherence to the edit operation. The user judged for five minutes, and an average of 30 edit pairs were judged per user. Each edit operation consisted in changing one of 10 possible facial attributes to a random new value, using either [16] or [4]. For fairness, [30] was not included in the edits, as this method was used to supervise our loss. The results (Fig. 5(b)) show the strong preference for our method.

Figure 6: Problematic images for our method. In the first row, an atypical face produces a low-quality reconstruction since the image space loss and the latent space loss oppose each other: the input image is too much out-of-domain. In the second row, a challenging semantic concept (the keffiyeh) struggles to be represented with our method. The projection represents the keffiyeh as hair, evidenced by the subsequent edits (*straight hair*, *old age*).

6 Conclusion

We propose a novel GAN-inversion optimization strategy which allows supervision on two levels: the image space and the latent space. In this way, we are able to integrate editability directly in the loss term, resulting in more editable latent vectors when applying editing methods. In particular, editing methods not used to supervise the loss perform better than baseline methods, suggesting that performing optimization in this way discovers a more "in-domain" latent vector. We evaluate qualitatively and quantitatively, and notably introduce a novel *edit consistence score* which specifically evaluates the performance of a projection method in terms of editability. Our method takes a step forward in performing realistic and high-quality edits on real images.

References

[1] How Long does a Professional Photographer Take to Edit Photos? https://www.belindajiao.com/blog/how-long-professional-photographer-edit-photos, 2020.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019.

[3] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020.

[4] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 2021.

[5] Ivan Anokhin, Pavel Solovev, Denis Korzhenkov, Alexey Kharlamov, Taras Khakhulin, Aleksei Silvestrov, Sergey Nikolenko, Victor Lempitsky, and Gleb Sterkin. High-resolution daytime translation without domain labels. In *CVPR*, 2020.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[7] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. In *ICLR*, 2019.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *CVPR*, June 2020.

[10] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[11] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. *arXiv preprint arXiv:1906.10112*, 2019.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*. 2014.

[13] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *CoRR*, abs/2007.01758, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[16] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[22] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013.

[24] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[26] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021.

[27] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

[28] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *NeurIPS*, 2020.

[29] 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. In *CVPR*, 2021.

[30] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

[31] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Perez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, 2020.

[32] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.

[33] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

[34] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021.

[35] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021.

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[37] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *ICLR*, 2021.

[38] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. Hype: A benchmark for human eye perceptual evaluation of generative models. In *NeurIPS*, 2019.

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

[40] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.

[41] Peiye Zhuang, Oluwasanmi O Koyejo, and Alex Schwing. Enjoy your editing: Controllable gans for image editing via latent space navigation. In *ICLR*, 2021.