Improved StyleGAN Embedding: Where are the Good Latents?

Peihao Zhu[1] Rameen Abdal[1] Yipeng Qin[2] John Femiani[3] Peter Wonka[1]

[1]KAUST [2]Cardiff University [2]Miami University

{peihao.zhu, rameen.abdal}@kaust.edu.sa qiny16@cardiff.ac.uk

femianjc@miamioh.edu pwonka@gmail.com

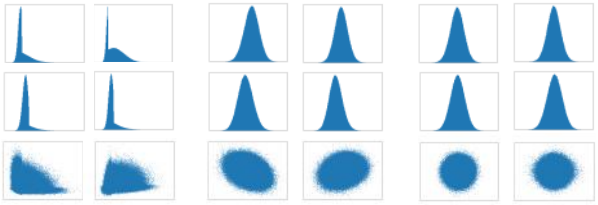
Abstract

StyleGAN is able to produce photorealistic images that are almost indistinguishable from real ones. The reverse problem of finding an embedding for a given image poses a challenge. Embeddings that reconstruct an image well are not always robust to editing operations. In this paper, we address the problem of finding an embedding that both reconstructs images and also supports image editing tasks. First, we introduce a new normalized space to analyze the diversity and the quality of the reconstructed latent codes. This space can help answer the question of where good latent codes are located in latent space. Second, we propose an improved embedding algorithm using a novel regularization method based on our analysis. Finally, we analyze the quality of different embedding algorithms. We compare our results with the current state-of-the-art methods and achieve a better trade-off between reconstruction quality and editing quality.

1. Introduction

GAN inversion (embedding) refers to the task of computing a latent code for a given input image. The goal of the embedding is typically to perform some subsequent image processing tasks such as image interpolation or semantic image editing. Due to the high visual quality of generated images and comparatively lightweight architecture, most recent papers build on StyleGAN [15] and StyleGAN2 [16, 14] to develop GAN inversion algorithms. The resulting embedding methods represent a tradeoff of two main concerns: the reconstruction quality and the editing quality. The reconstruction quality considers how similar an embedded image is to the input image and how realistic the embedded image is. The editing quality describes the visual quality of images after performing editing operations in latent space, e.g., editing the pose, lighting, or age of in face image [11].

In this paper, we set out to provide some analysis of existing embedding methods. One important consideration is the choice of the embedding space, e.g., Z space, W space, or W^+ space. We propose a suitable space, called P_N space, to perform the analysis. In P_N space, the distribution of StyleGAN latent codes has a surprisingly simple structure. This enables an explanation of where good latent codes can be found. A major insight of the analysis is that the quality of a latent code is closely related to the distance from the center of P_N space. The L_2 norm in P_N space is a Mahalanobis distance of latent codes, so that L_2 regularization in this space will bias embeddings towards more probable solutions. We find that reconstruction quality favors latent codes that are far from the origin in P_N space, but editing quality favors latent codes close to the origin. This requires a trade-off between reconstruction quality and editing quality. Investigating the individual dimensions of latent codes in P_N space gives a strong understanding of the trade-offs made by current methods. For example, the step size and the learning rate in Image2StyleGAN [1] controls the trade-off between reconstruction quality and editing quality. By contrast, current embedding networks, e.g. [17, 16, 22], create embeddings close to the origin in P_N space and favor editing quality over reconstruction quality. We propose



(a) Marginal and pairwise distributions in $\boldsymbol{W}$ space (b) Marginal and pairwise distributions in $\boldsymbol{P}$ space (c) Marginal and pairwise distributions in $\boldsymbol{P_N}$ space

Figure 1: Marginal and pair-wise distributions of different spaces in StyleGAN. First and second rows: the marginal distribution of a randomly picked latent variable. Third row: the pairwise joint distribution of two randomly picked latent variables.

an improved embedding algorithm by introducing a regularizer that encourages embeddings to stay closer to the origin of the P_N space. The advantage of our regularizer is that it explicitly controls the distance to the origin and does not have undesirable side effects. Finally, we present the first comprehensive evaluation comparing different state-of-the-art embedding algorithms. The evaluation does not just consider reconstruction quality, but also the effect of a variety of edits (pose, lighting, and age) and the impact of embedding algorithms on conditional embedding tasks (super-resolution, image colorization, and inpainting). Our main contributions are:

- the introduction of P_N space for analyzing and regularizing StyleGAN embeddings.
- a new regularizer for StyleGAN embedding that provides the best trade-off between reconstruction and editing quality.
- a comprehensive evaluation to many state-of-the-art embedding algorithms. Our evaluation not only emphasizes reconstruction but also the impact on downstream editing tasks.

2. Related Work

The seminal paper by Goodfellow et al. [8] started an avalanche of GAN papers that together lead to impressive improvements over the last years [20, 4, 13, 19, 15, 16, 14]. Similar to most other work in GAN image embedding, we build on StyleGAN [16, 15, 14] due to its exceptional visual quality and comparatively lightweight architecture.

Since the recent inception of Image2StyleGAN (I2S) [1] which first proposed a feasible method to embed a given image into the W^+ latent space of the StyleGAN generator, there have been many works trying to improve upon the initial idea.

Most of the existing methods have the same goal: finding a balance between the reconstruction quality and the editing quality. For example, I2S++ [2] improved the reconstruction quality of I2S by incorporating a noise optimization step to restore the high-frequency details in the input image. Similarly, StyleGAN2 [16] uses an additive ramped-down noise to help the latent space exploration. These two methods lead to different trade-offs: I2S++ embeds images into the W^+ space that sacrifices a bit of editing quality to achieve better reconstruction quality; while on the contrary, the StyleGAN2 embeds images in the W space that enables better editing but at the cost of worse reconstruction. Following a different approach, PIE [25] first embeds an image into the W space for better editing quality and then improves the reconstruction quality by optimizing the latent obtained in the W^+ space. Such a two-stage encoding process is also employed by several concurrent works based on encoder networks [6, 30, 10, 5], which to our knowledge first appears in iGAN [31]. In their methods, an initial latent code is first obtained by passing the input image through a pre-trained StyleGAN encoder. Then, the initial latent code is further optimized to improve its reconstruction quality. Using only the encoder network by itself leads to poor reconstruction quality.

Unlike previous methods, PULSE [17] formulated embedding as traversing the manifold consisting of good latents. However, they regularized the latents to be on the surface of a hypersphere that only contains a subset of good latents. As a results, their method usually leads to poorer reconstruction quality. Inspired by PULSE, in this work, we provide a thorough analysis of where the good latents are and how to evaluate an embedding. Based on our analysis, we propose an improved embedding algorithm that outperforms all existing methods.

3. Which Space to Use: a Statistical Analysis of Latent Distributions

The goal of this section is to identify a space where the distribution of latent codes has a simple structure. A suitable space will make it easier to reason about good and bad latent codes for the StyleGAN generator [15, 16]. However, the interpretability challenge of deep neural networks makes it difficult to determine which latent space is better. Following the philosophy of *Occam's razor* [7], we conjecture that a good latent space is one in which the latent distribution can be characterized in the simplest possible way. To find such a latent space, we conduct a statistical analysis on the latent distributions in different latent spaces as follows (See Fig. 2).

***W* Space:** The W space proposed in StyleGAN [15] is the most straightforward choice of latent space. Therefore, we start our investigation by sampling 1 million latents in W space and visually checking the statistics of their marginal distributions. As Fig. 1 (a) shows, the marginal distributions are heavily skewed and do not follow any obvious patterns. Thus, the W space latent distribution is difficult to characterize and not suitable for the classification of good and bad latents.

***P* Space:** To get rid of the skewness of marginal distributions, we transformed the W space to the P space by inverting the last Leaky ReLU layer in the StyleGAN mapping network. The name is derived from PULSE [17] even though it is not described in the paper. They use it in their code to perform a projection operation that was supposed to be performed in Z space. Since the last Leaky ReLU uses a slope of 0.2 we invert it using

$$\mathbf{x} = \mathrm{LeakyReLU}_{5.0}(\mathbf{w}) \tag{1}$$

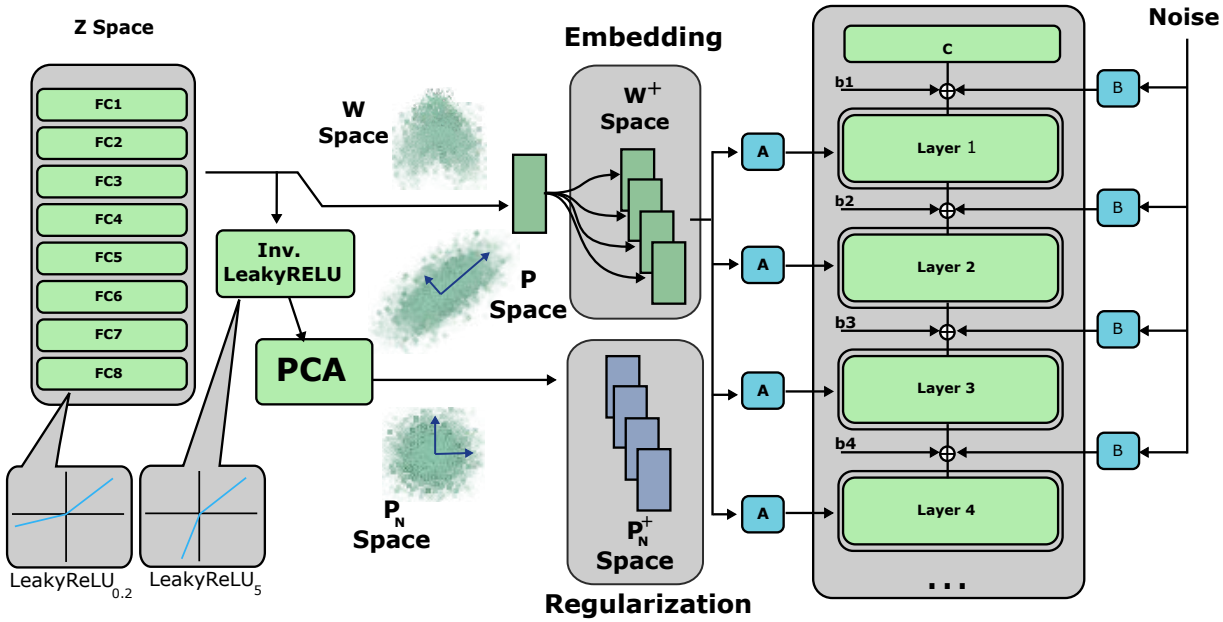


Figure 2: We show six different spaces into which projection of a given image is possible. These are: Z space, W space, W^+ space, P Space, P_N space and P_N^+ space. Notice that translation from W to P or P to W space goes through an invertible function (Leaky ReLU).

where $\mathbf{w}$ and $\mathbf{x}$ are latents in W and P space respectively. Similar to those in W space, we plot the marginal distributions of latents in P space and observed that they nicely follow a simple Gaussian-like distribution. As a side note, it was fascinating for us that the GAN learns a latent space following a multi-dimensional Gaussian distribution while the VAE fails even though explicitly optimizing for it. Although this is a great start, it is unclear how much the latents depend on each other. Therefore, we plot the pairwise joint distribution of latents (See Fig. 1b) and observed that the dimensions are indeed dependent. We make the the simplest assumption that the joint distribution of latents is approximately a multi-variate Gaussian distribution. This motivates us to introduce the P_N Space in which the latent distribution is easier to characterize.

P_N Space: Our P_N space aims to i) eliminate the dependency among latent variables; ii) remove redundancy and only capture the major information of the latent distribution. To fulfill these two aims, we define an affine transformation from P to our P_N space as:

$$\hat{\mathbf{v}} = \mathbf{\Lambda}^{-1} \cdot \mathbf{C}^T(\mathbf{x} - \boldsymbol{\mu}) \tag{2}$$

where $\mathbf{\Lambda}^{-1}$ is a scaling matrix, $\mathbf{C}$ is an orthogonal matrix and $\boldsymbol{\mu}$ is a mean vector. They are estimated by applying Principal Component Analysis (PCA) to $\mathbf{X} \in \mathbb{R}^{10^6 \times 512}$ consisting of 1 million latent samples in P space. The latents are first sampled in Z space, then passed through the mapping network to W space and transformed to P space following Eq. 1.

$$\mathbf{C}, \mathbf{\Lambda}, \boldsymbol{\mu} = \mathrm{PCA}(\mathbf{X}) \tag{3}$$

The rows of $\mathbf{C}$ are the principal component axes; $\boldsymbol{\mu}$ is the mean vector of $\mathbf{X}$; and $\mathbf{\Lambda}$ is a matrix containing the corresponding eigenvalues of $\mathbf{C}$. Intuitively, the above transformation normalizes the distribution of each latent variable to be of zero mean and unit variance. As a result, the latent distribution in our P_N space will look like a ball that is isotropic in all directions (Fig. 1c).

The $L2$ norm $\|\hat{\mathbf{v}}\|$ is related to the Mahalanobis distance $d_M(\cdot)$ of points $\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = \mathbf{C}\mathbf{\Lambda}^2\mathbf{C}^T$

$$d_m^2(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{4}$$

$$= (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}\mathbf{\Lambda}^{-2}\mathbf{C}^T(\mathbf{x} - \boldsymbol{\mu}) \tag{5}$$

$$= \hat{\mathbf{v}}^T\hat{\mathbf{v}} \tag{6}$$

This space has useful properties for regularizing our embedding algorithm because it biases the solution towards the mode $\boldsymbol{\mu}$ of embeddings and it is more sensitive to directions of low variance, resulting in more regularization to

deviations from the mode that are unlikely in the embedding space and would be considered artifacts, and less regularization for deviations where one expects variance in the images. For reconstruction tasks, the realistic/recognizable components original image is unlikely to be penalized.

Without regularization, editing tasks are effected because real images contain variations that are not modeled by the GAN so that their embedding can end up in low-probability region of the embedding space even though the image seems plausible. An edit operation shifts the embedding to a new location, so if the original embedding was in a high probability part of the embedding space the perturbed embedding is also likely to be nearby in a high probability region. However in the low-probability portion of the space, a perturbed image will also have a low-probability and since it is not a 'real world' image it may be perceived as implausible.

To verify the validity of our P_N space, we sample from a standard Normal distribution in P_N space and compare the FID scores we get with those of StyleGAN2 [16]. P_N space sampling yields an FID of 3.28 compared to an FID of 2.81. Such as small difference supports the validity of our P_N space. By contrast, fitting a normal distribution to W space and sampling from it yields an FID of 76.63. Although good for characterization of latent distributions, P_N space is too restrictive for image embedding. To this end, we extend it to P_N^+ space following I2S [1].

P_N^+ Space: Similar to the extension from W space to W^+ space in which 18 different 512-dimensional W latents ($\mathbf{w}_i$) are concatenated [1],

$$\mathbf{w}^+ = \{\mathbf{w}_i\}_{i=1}^{18} \tag{7}$$

we extend P_N space to P_N^+ space as

$$\mathbf{v} = \{\Lambda^{-1}\mathbf{C}^T(\mathbf{x}_i - \boldsymbol{\mu})\}_{i=1}^{18} \tag{8}$$

where $\mathbf{x}_i = \mathrm{LeakyReLU}_{5.0}(\mathbf{w}_i)$. Each of the latents is used to demodulate the corresponding StyleGAN feature maps at different layers. We propose to use P_N^+ space to analyze and regularize StyleGAN inversion algorithms.

4. Improved I2S

4.1. Distance Regularizer

Based on our results, it seems most promising to devise a regularizer that penalizes an embedding if it goes too far from the center. To this effect, there seem to be three simple strategies to evaluate for regularizing a latent code in P_N^+ space. Let I be the input image, $L = L_{LPIPS} + L_2$ be a loss function consisting of LPIPS and pixel-wise L2 loss terms, $\mathbf{w}^+$ and $\mathbf{v}$ be the latents in W^+ and P_N^+ space respectively, λ, ϵ, and k are hyperparameters, we have:

1. Regularizing the $L2$ distance from the center, possibly using separate weights along each dimension.

$$\mathbf{w}^{+^*} = \arg\min_{\mathbf{w}^+} L\left(I, G\left(\mathbf{w}^+\right)\right) + \lambda\|\mathbf{v}\|^2 \tag{9}$$

2. Clipping values that are over a given threshold, possibly using a separate threshold per dimension.

$$\mathbf{w}^{+^*} = \arg\min_{\mathbf{w}^+} L\left(I, G\left(\mathbf{w}^+\right)\right) \text{ s.t. } \|\mathbf{v}\|_\infty \leq \epsilon \tag{10}$$

3. Projecting to a sphere / ellipsoid in a suitable space.

$$\mathbf{w}^{+^*} = \arg\min_{\mathbf{w}^+} L\left(I, G\left(\mathbf{w}^+\right)\right) \text{ s.t.} \|\mathbf{v}\| \leq k\epsilon \tag{11}$$

While all three versions are viable choices to achieve a trade-off between editing quality and reconstruction quality, we only focus on the P_N^+ $L2$ regularizer (9) in this paper for reasons discussed in the supplementary.

Remark Our $L2$ distance regularizer can be interpreted as incorporating a Gaussian prior in P_N space. Thereby, GAN embedding is extended from maximum likelihood estimation (MLE) to maximum a-posteriori (MAP) estimation. Since the negative logarithm of a Gaussian prior is a squared $L2$ norm (*i.e.* our $L2$ distance regularizer), incorporating it into the embedding algorithm extends it to MAP. In this interpretation, I2S [1] is the MLE version of the algorithm, corresponding to $\lambda = 0$.

4.2. Improved Perceptual Regularizer

One important aspect of I2S is to use a perceptual regularizer based on VGG-perceptual loss [24]. However, we find that using LPIPS [28] results in a noticeable improvement. Following PULSE [17] we also propose to use bicubic downsampling on the generated images and Lanczos downsampling [27] on the reference image. We did not use Lanczos for both as it is not differentiable. Nevertheless, we observed that the current setup works better than using bicubic sampling for both cases.

4.3. Implementation Details

For our method, we used a learning rate of 0.01 with 1300 steps for all images. We used the ADAM optimizer with standard settings to obtain the results.

5. How to Evaluate an Embedding?

The purpose of this section is to provide an extensive evaluation and comparison of different embedding algorithms. We propose a sequence of tasks for that purpose.

In general, a good embedding should have the following three properties: i) it should faithfully reconstruct the input image ii) it should be a realistic face image and iii) it should

Figure 3: Reconstruction quality of different embedding methods using the StyleGAN2 generator.

enable realistic editing. We consider i) and ii) to be part of the reconstruction quality and iii) as editing quality. We make a subtle distinction between i) and ii) because some embedding methods have good reconstruction quality when considering traditional similarity metrics, but still have artifacts, e.g. unrealistic eyes and teeth (see supplementary).

We extensively tested many methods that have code available and selected the methods that performed best for a more detailed comparison. We compare our proposed II2S method (Section 4) with the following state-of-the-art methods:

- I2S [1] is the baseline method we would like to improve upon which embeds images into the W^+ space.
- PULSE [17] observed the "soap bubble" effect of high dimensional Gaussian distributions and proposed embedding onto the surface of a hypersphere in Z space.
- StyleGAN2 [16] embeds images into the W space with the help of noise regularization.
- StyleGAN2Encoder [22] embeds images into the W^+ space using logcosh image loss, MS-SSIM loss, L1 penalty on latents and several other losses. In addition, they also use early stopping to prevent overfitting.
- pSp [21] proposes a novel encoder architecture to map the given image to the W^+ space. The method is concurrently developed and only available on arXiv. However, we include it to make the comparison stronger.

Note that we updated I2S and PULSE to use StyleGAN2 and adjusted the hyper-parameter settings for I2S to limit overfitting. We do not compare to the IDInvert method [30] and Multi-code embedding [9] because they only work on the original StyleGAN architecture [15] at a low resolution of 256×256. We also do not use noise embedding as proposed by I2S++ [2] because it is too hard to control with respect to editing quality.

For the evaluation, we collected a dataset of 120 images from the website Unsplash. The remaining subsections describe the tasks and metrics used to evaluate the system.

5.1. Reconstruction Quality

In this work, we measure the reconstruction quality of an embedding both i) absolutely using RMSE, PSNR and ii) perceptually using SSIM, VGG perceptual similarity, LPIPS perceptual similarity and the FID [12] score between the input and embedded image. The results are shown in Table 1. Our method II2S with $\lambda = 0.001$ and I2S are either ranked first or second across all metrics, except SSIM. However, we note that the metrics are less important than perceptual user evaluations. Table 2 presents a user study for a stronger regularized setting with $\lambda = 0.005$. This is our recommended setting for the best compromise between editing quality and reconstruction quality. Here, II2S reconstruction results were accepted by users more than any competing method except I2S. In Fig. 3, we show the reconstruction quality. Notice that other methods can have visual artifacts or differ substantially from the original for some examples when viewing the embedding results at higher resolution.

5.2. Editing Quality

Unlike reconstruction, the editing quality of an embedding has not been studied, because competitive editing frameworks just became available very recently [23, 11, 26, 3]. Note that these editing frameworks are orthogonal to

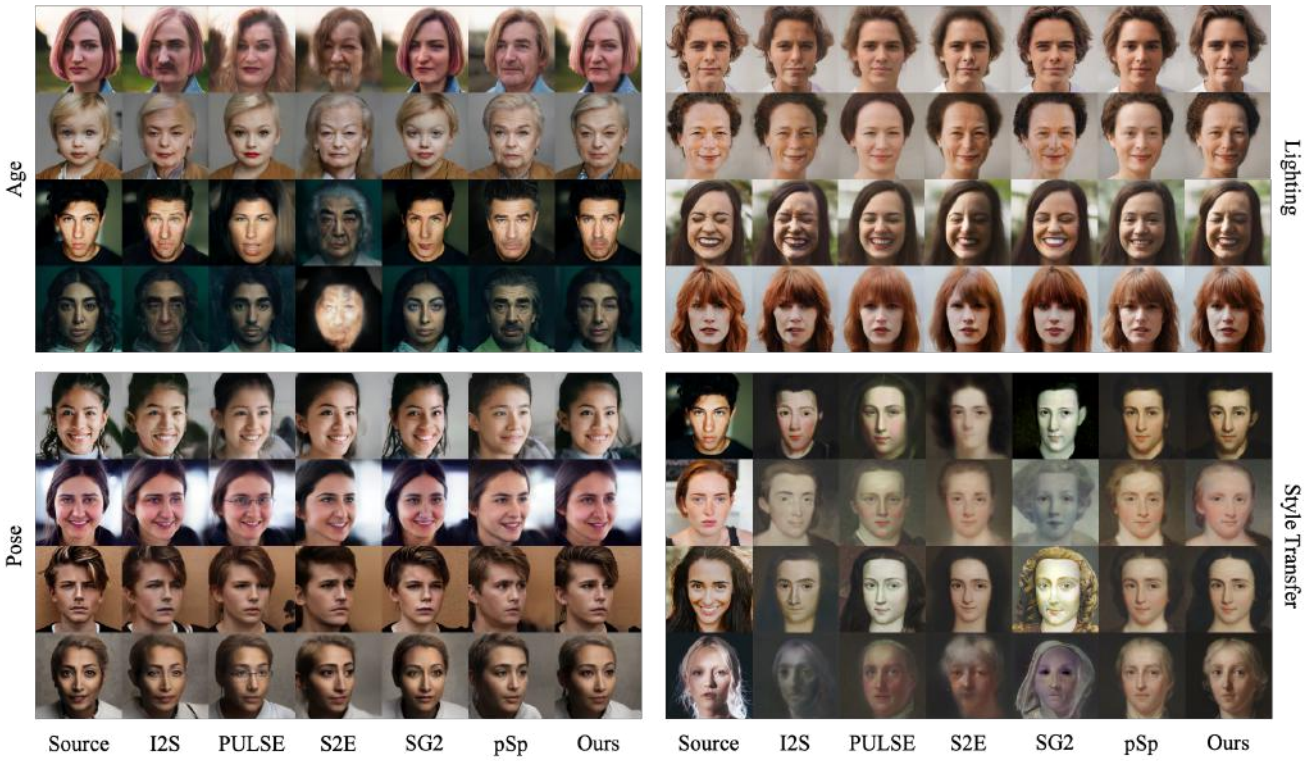


Figure 4: Four selected examples each for editing age, pose, lighting, and style transfer.

Method	SSIM	RMSE	PSNR	VGG	LPIPS	FID
I2S [1]	0.83	**0.07**	**23.45**	**0.63**	0.21	**36.60**
PUL [17]	0.81	0.09	21.63	1.05	0.37	64.63
S2E [22]	**0.84**	0.08	22.50	0.93	0.32	70.70
SG2 [16]	0.79	0.12	19.47	0.80	0.21	46.49
pSp [21]	0.81	0.09	21.35	0.91	0.30	54.80
$\lambda = 0.01$	0.81	0.09	20.64	0.90	0.27	53.89
$\lambda = 0.005$	0.82	0.08	21.45	0.85	0.25	49.91
$\lambda = 0.001$	0.83	**0.07**	23.00	0.76	**0.20**	43.99

Table 1: Quantitative comparison of reconstruction quality using similarity metrics SSIM, RMSE, PSNR, VGG [24], LPIPS [28], and FID [12]. Two best results are underlined; best result is bold.

our work as they do not propose new embedding algorithms. We choose StyleFlow [3] as our main evaluation method for editing as it generally leads to the highest quality edits.

We propose to test the editing quality using the following editing operations: pose, age, and lighting. For the pose editing task, we set the target yaw of each image to 20°. To edit age, we set the target age to 50 years old. In lighting change experiments, we set the target light direction to point towards viewer's right. In Fig. 4, we visually compare editing results of different methods. Notice how our method keeps the identity of the original image and preserves the realism after the edits.

The user study results presented in Table 2 show that editing operations on latent codes derived by II2S maintain image quality across different editing tasks. Our editing results are preferred to the edits of other methods for all tasks. Somewhat surprisingly SG2 is the second best method for editing. Based on our visual analysis of the results we observe the following problem. Some embedding algorithms compute latent codes that cannot be edited well. After an edit, the new latent code may produce a new image that is very similar and did fully accomplish the editing task. For example, when performing an edit to change the age, PULSE produces results that look noticeably younger than the target age. We further study this phenomenon using the state-of-the-art classifier Microsoft Face API [18] in supplementary materials.

5.3. Conditional Embedding Quality

We consider four conditional embedding applications: image colorization, inpainting, super resolution, and style transfer. Let I be an input "condition" image, and G be a StyleGAN generator. Conditional embedding aims to locate an optimal latent code $\mathbf{w}^{+*}$ in W^+ space so that the embedded image $G(\mathbf{w}^{+*})$ i) faithfully captures the "condition" of I and ii) is a realistic face image. Accordingly, we define

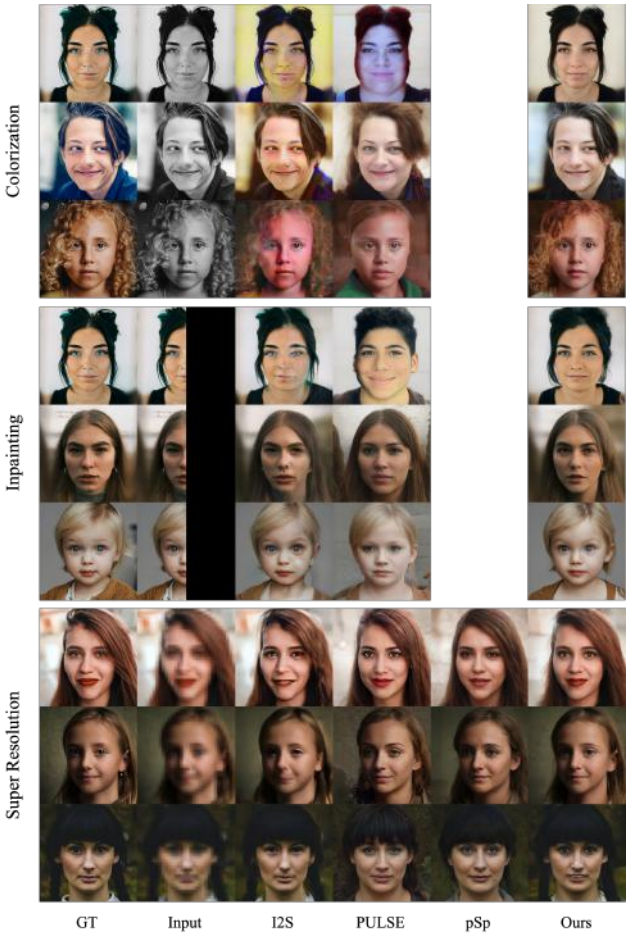


Figure 5: Examples of conditional embedding results. Three example of each task are shown from top to bottom: Image colorization, inpainting, and super resolution.

conditional embedding as:

$$\mathbf{w}^{+*} = \arg\min_{\mathbf{w}^+} L\left(I, f\left(G\left(\mathbf{w}^+\right)\right)\right) + \lambda \cdot R(\mathbf{w}^+) \quad (12)$$

where f is a "condition" function that modifies an image to satisfy a pre-defined condition (*e.g.* grayscale), L is a loss function measuring the similarity between two images (*e.g.* pixel-wise $L2$ loss, perceptual loss), λ is a hyperparameter, and R is the regularizer. Note that the only difference between ordinary and conditional embedding is the incorporation of f.

For image colorization, the input image I is a grayscale image and the condition function f coverts a color image to grayscale. For inpainting, the input image I is an incomplete image and the condition function f is a *mask* function that erases the pixels in a given region. In our tests the missing region is half the image. For super resolution, the input image I is a 32×32 low-resolution image and the condition function f is a *downsampling* function.

The style transfer task is accomplished using a different approach. Let G and G' be two StyleGAN generaters, G is trained on the FFHQ dataset and G' is a variant of G fine-tuned on the MetFace [14] dataset. Our style transfer is implemented by embedding the input image into G and then evaluating the resulting latent code using G'.

The corresponding user study results for all four applications are presented in Table 2. Examples of visual results for the first three tasks are shown in Fig. 5 and results for style transfer are shown in Fig. 4. From the user study we can note that users ranked our image colorization results to be the most successful. Surprisingly, they are ranked even more realistic than the ground truth images. We attribute this to the fact that several images are stylized photographs. For the superresolution results we asked users which image was closest to the ground truth. Our result is clearly preferred to all published competitors I2S and PULSE. While our results are also better than pSp, they are not statistically significant. For inpainting we also significantly outperform all competitors. This is the only result where our method clearly looses to ground truth. We ascribe this to the fact that we do inpainting for a very large missing region (half the image). This is a very challenging task. For pSp we could not complete the colorization and inpainting comparisons because pSp needs to pre-train a separate encoder for each task. Corresponding encoders were not provided by the authors. Similarly, we did not compare to SG2 and S2E on the first three tasks. This would have required us to reimplement conditional embedding in Tensorflow.

5.4. User Studies

We carried out a series of user studies using Amazon Mechanical Turk (MTurk). For each task, images were generated using II2S and a competing method. Each pair of images was presented to workers on Amazon Mechanical turk twice, once in the order (II2S, Source, Other), and once in the reverse order so that preferences for the image based on its position in the survey would not be an issue. Users were given a prompt to select the image that best accomplishes the task with the fewest artifacts. Table 2 shows the percentage of responses that selected a competing method, and a number below 50% means that the II2S solution was selected more frequently. Each survey received 120 responses, so any number below 40% is statistically significant at 95% confidence. II2S outperforms competing methods on most tasks, with only a few exceptions. Reconstruction results are better with I2S, however I2S is effectively the same approach without regularization and thus one would expect it to do better at reconstruction and poorer for editing tasks. For conditional GAN tasks II2S is outperformed by Ground Truth, which is expected. To summarize, we argue that our evaluation demonstrates that our proposed regularizer has a significant impact on downstream applica-

tions.

	GT	I2S	SG2	S2E	PUL	pSp
Reconstruction	–	**60.8**	41.7	28.3	16.7	45.0
Edit:Age	–	20.0	42.5	2.5	35.8	26.7
Edit:Pose	–	30.8	45.0	20.8	30.8	34.2
Edit:Light	–	20.8	35.8	11.7	23.3	31.7
Style Trans.	–	22.5	15.8	23.3	26.7	44.2
Colorization	35.6	5.0	–	–	20.6	–
Inpainting	**71.7**	34.2	–	–	24.2	–
Super Res.	**56.7**	5.83	–	–	34.2	47.5

Table 2: A user study comparing II2S to the Ground Truth (GT), I2S, StyleGan2 (SG2), StyleGAN2 Encoder (S2E), Pulse (PUL) and pSp. Cases that outperform II2S are in bold.

5.5. Ablation Study

The most important parameter in our method is the strength of the regularizer. Based on many experiments we manually selected three settings of λ to compare to: 0.01, 0.005, and 0.001. These settings were selected such that a visual difference is noticeable between the embeddings. In Fig. 3 we show the visual difference between $\lambda = 0.005$ and $\lambda = 0.001$. In Tab. 1 we show how λ affects reconstruction metrics. A user study to evaluate different choices of λ is shown in Table 3, and we picked $\lambda = 0.005$ for all tests in the paper as a trade-off between reconstruction and editing quality.

	.01	.005	0.001
Reconstruction	28.8	49.6	**71.7**
Edit:Age	56.7	**60.0**	33.3
Edit:Pose	47.9	48.3	**53.8**
Inpainting	48.8	**53.8**	47.5
SuperResolution	**61.3**	53.3	35.4
Average	53.6	**53.9**	42.5

Table 3: User Study of λ. Each value is the percentage of times an image was selected as the best when compared to an image form *either* of the other two columns; higher numbers are better.

5.6. Histogram of the Embeddings

We visually analyze the histograms in the P_N^+ space to obtain additional insights about the different embedding algorithms. To compute the histograms we embed the 120 images of our dataset using the different algorithms. Then we take the 20^{th} dimension of each latent code in P_N^+. For all methods except SG2 there are 18 such latent codes per embedded image. For SG2 we repeat the same value 18 times. In Fig. 6 we show the histograms. For the first histogram, denoted by I2S Overfitting, we run I2S with a high learning rate. As we know that this setting leads to overfitting, we can observe that overfitting correlates with a histogram that is wider, has higher variance, and more outliers. The histograms of the encoders, pSp and S2E, have the lowest variance indicating that the embedded latent codes are closer to the origin of P_N^+. It is our conjecture that narrow histograms correlate with underfitting. The histogram for SG2 looks very strange. This might be due to the fact that SG2 has fewer unique samples. Still, the variance is significantly higher than we would have expected. Our method with the truncation regularizer, denoted as ours (clip) also has a unique, but expected histogram. Even though we cannot really observe the impact on downstream editing tasks, it is somehow worrying to have so many values clipped to exactly the same number.

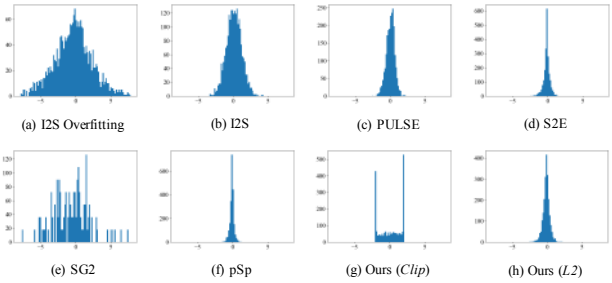


Figure 6: Histograms showing the distribution of latents in the twentieth dimension of the P_N^+ Space.

6. Limitations, Future Work, and Conclusion

Our work has some limitations that we leave to future work. One limitation is that we did not investigate correlations between the (18) different components of an extended latent space. From sampling new latent codes in Z^+ space, we know that sampling the 18 components independently leads to poor results. However, it is unclear if their similarity should be enforced by an additional regularizer, as we could not really observe a consistent improvement when experimenting with different regularizers proposed in the literature (e.g. hierarchical optimization or an $L2$ similarity term). In the future, we would like to revisit this topic.

We introduced the P_N space as a tool to facilitate improved StyleGAN embedding and to analyze different embedding algorithms. The switch to P_N space can significantly help follow up work. We proposed a new regularizer for StyleGAN embedding that provides the best trade-off between reconstruction quality and editing quality. Finally, we conducted an extensive evaluation highlighting the strength and weaknesses of previous work. The evaluation demonstrates that our results clearly outperform all compet-

ing methods for downstream applications.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019. 1, 2, 4, 5, 6, 11, 12

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 2, 5

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv e-prints*, pages arXiv–2008, 2020. 5, 6, 11

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2

[5] David Bau, Hendrik Strobelt, William Peebles, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, et al. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020. 2

[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4502–4511, 2019. 2

[7] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987. 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[9] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3012–3021, 2020. 5

[10] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *arXiv preprint arXiv:2007.01758*, 2020. 2

[11] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 1, 5, 11, 14

[12] 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 *Advances in neural information processing systems*, pages 6626–6637, 2017. 5, 6

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[14] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 1, 2, 7

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1, 2, 5

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2, 4, 5, 6, 11

[17] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2445, 2020. 1, 2, 4, 5, 6, 11

[18] Microsoft. Azure face, 2020. 6, 11

[19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2

[20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[21] 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. *arXiv preprint arXiv:2008.00951*, 2020. 5, 6, 11

[22] robertluxemburg. Github repository: stylegan2encoder. https://github.com/robertluxemburg/stylegan2encoder, 2020. 1, 5, 6, 11

[23] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *arXiv preprint arXiv:2005.09635*, 2020. 5

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4, 6

[25] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, Christian Theobalt, et al. Pie: Portrait image embedding for semantic control. *arXiv preprint arXiv:2009.09485*, 2020. 2, 12

[26] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020. 5

[27] Ken Turkowski. Turkowski filters for common resampling tasks 10 april 1990 filters for common resampling tasks. 4

[28] 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. 4, 6

[29] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W. Jacobs. Deep single portrait image relighting. In *International Conference on Computer Vision (ICCV)*, 2019. 11

[30] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2, 5, 12

[31] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016. 2

A. Metrics

A.1. Comparison of Editing Quality

We choose StyleFlow [3] as our main evaluation method for editing as it generally leads to the highest quality edits. In addition, we compare against another editing approach, GANSpace [11] in this supplemental material.

We propose to test the editing quality using the following editing operations: pose, age, lighting, gender and eyeglasses. For the pose editing task, we set the target yaw of each image to 20°. To edit age, we set the target age to 50 years old. In lighting change experiments, we set the target light direction to point towards viewer's right. For the gender transfer task, we swap the gender of the input image. In eyeglasses experiments, we add eyeglasses to each image. Note that the only conditional edit is the gender swap. Here the goal of the edit depends on the input image. For other edits, there is a fixed goal, such as adding eyeglasses. If the person had eyeglasses before, the image is not supposed to change. Similarly, for the pose, there is a fixed target pose. Some images are expected to change more than others under these edits. Images that are already in the target pose are expected not to change.

We already show examples of pose, age, and lighting in the main paper. In Fig. 7, we visually compare the remaining editing results (gender, eyeglasses and lighting+pose) of different methods. Notice how our method keeps the identity of the original image and preserves the realism after the edits. Quantitative results for various edits are presented in Table 4. To create the table, attributes were measured using Microsofts face API [18]. Lighting is measured by a different network [29]. Overall, our method has the best results. pSp is very slightly better than ours in two edits, but clearly worse in two others.

Method	Age	Pose	Lighting	Gender	Eyeglasses
	50	20	R→L	Swap	Add
I2S [1]	53.4	18.1	0.90	0.85	0.71
PUL [17]	33.8	**18.3**	0.88	0.71	0.76
S2E [22]	60.1	25.7	0.91	0.93	0.83
SG2 [16]	29.4	13.2	0.85	0.33	0.38
pSp [21]	55.7	27.5	**0.95**	**1.00**	**0.98**
Ours	**50.2**	23.5	0.94	0.99	**0.98**

Table 4: Quantitative comparison of editing quality.

In addition to using StyleFlow [3] as the evaluation method of editing quality, for the pose change experiments, we use the right-left pose edit from GANSpace [11]. Specifically, for a latent code in the GANSpace coordinate system, we can set the coordinate corresponding to pose to five different values: -2σ, -σ, 0, σ, and +2σ, where σ is the eigenvalue for the direction. Fig. 8 shows the qualitative results of pose change edits. Quantitative results are presented in Table 5. We can observe that the strength of an edit depends on the initial latent code and the embedding method. This poses a challenge for the evaluation of different embedding methods using GANSpace.

Method	-2σ	-1σ	0	1σ	2σ
I2S	18.42	8.29	-0.81	-9.27	-18.49
PUL	20.12	10.16	-0.31	-9.66	-17.99
S2E	25.75	11.66	-1.34	-13.52	-26.35
SG2	12.49	4.84	-2.04	-8.36	-14.75
pSp	**28.86**	15.85	-0.77	-15.62	**-26.81**
Ours	26.23	13.16	-0.01	-12.07	-23.42

Table 5: Pose metric for testing the editing quality using GANspace [11]. We mark the highest ranges. These numbers by themselves are not conclusive with regards to quality, but we can observe that our method and pSp react stronger to this GANSpace edit than other methods. We can also observe that the dependence of the strength of an edit on the initial embedding is quite strong.

A.2. Linear Interpolation

Linear interpolation of latent codes can generate a morph between two embedded images. It is also a powerful tool to determine whether an embedding is semantically meaningful. The nature of the latent space ensures that intermediate images are of high quality as long as the two images being interpolated are of high quality. In other words, we can tell if an embedding is good if the interpolated images between it and a randomly sampled image are of high quality. To this end, we test the interpolation results in both the W^+ and P_N^+ space. Specifically, we compute the FIDs between the 120 images of our dataset and 2,500 interpolated images obtained by i) sampling 500 image pairs from the dataset and ii) interpolating 5 images for each pair at proportions 0.1, 0.25, 0.5, 0.75 and 0.9. Table 6 shows the results on various methods. We note that when I2S embedding is projected into the P_N^+ space, the interpolation FID becomes worse, again indicating that the distribution is far from the original StyleGAN latent distribution. Qualitative results are shown in Fig. 9 and Fig. 10. While we have the best quantitative results together with I2S one has to consider that FID is only calculated on a smaller set of images (2500 instead of 50000). While several other authors also report FID on smaller sets of images for editing and manipulation tasks, the metric is not entirely conclusive. For the qualitative results, it is important to mention that our method has the best reconstruction quality.

A.3. Style Mixing

As two sides of the same coin, the results of style mixing and structural edits (*e.g.* pose editing) both reflect how

Method	Style Mixing	Interp. in W^+	Interp. in P_N^+
I2S	57.36	**43.03**	55.25
PUL	65.05	63.73	63.39
S2E	66.22	67.84	69.91
SG2	58.48	49.40	54.19
pSp	64.73	66.69	66.00
Ours	**55.69**	48.61	**49.15**

Table 6: FID of style mixing and interpolation in different spaces.

good a latent disentangles the style and contents at different layers of the StyleGAN generator. Thus, it is also worthwhile to check whether an embedding enables high quality style mixing. We take the variables from the first 7 layers from one image and from the remaining 11 layers from another image and test the FID of the results with the original images.

Table 6 shows the result of style mixing on various methods. Qualitative results are shown in Fig. 11. Again, even though our method has the best results, FID is calculated for a smaller set of 500 images. Also for the qualitative results, it's important to consider that our method has the best reconstruction quality.

B. Ablation Study

We conducted two additional ablation studies for the supplementary materials. The first ablation study compares the results when embedding in Z^+ space rather than W^+ space. While other authors advocate for embedding in Z^+ space, we obtain clearly worse results in our implementation. Specifically, we observe artifacts for several of the embedded images. A quantitative evaluation in shown in Table 7.

Another ablation study investigates the result of the different downsampling operation and the effect of choosing LPIPS as regularizer over VGG. The different downsampling operation has a very minor effect on the results, but exchanging VGG for LPIPS leads to significant and important differences. We show quantitative results in Table 8 and qualitative results in Fig. 12.

Space	SSIM	RMSE	PSNR	VGG	LPIPS	FID
Z^+	0.79	0.10	19.99	0.99	0.31	63.57
W^+	**0.83**	**0.07**	**23.00**	**0.76**	**0.20**	**43.99**

Table 7: Ablation study of optimization in different spaces. The hyperparameter λ of the regularizer is set to 0.001.

Method	SSIM	RMSE	PSNR	VGG	LPIPS
Bilinear down.	0.82	0.08	22.95	0.77	0.21
VGG percep.	0.82	0.08	22.45	0.70	0.24
Ours	**0.83**	**0.07**	**23.00**	**0.76**	**0.20**

Table 8: Ablation study of perceptual loss and down sampling method. The hyperparameter λ of the regularizer is set to 0.001.

C. Looking at the "Eyes" and "Mouth"

Several existing evaluations in previous work are flawed, because they overemphasize reconstruction quality in the hyperparameter settings. For example, we could observe that Zhu et al. [30] and Tewari et al. [25] use suboptimal I2S [1] hyperparameter settings for some of the comparisons. One simple test is to check the visual realism of the eyes and teeth during the embedding to observe the transition from reasonable editing quality to overfitting and poor editing quality. In Fig. 15 we show a progression of the embedding process with closeup insets of the eyes of the original I2S algorithm. This simple test is of high practical importance. Our proposed regularizer is very important, because it is too difficult and impractical to set the number of iterations for each image separately for I2S. While it is too time consuming to do this for our complete dataset, we believe that I2S results would improve using per image finetuning. See Fig. 14 for a visualization how the regularizer can prevent overfitting. This makes a big difference for eyes and teeth.

D. Visual Explanation of the Pulse Regularizer Problem

To better illustrate the problem of the Pulse regularizer, we perform an experiments of repeatedly embedding the same image. For most methods, embedding the same image multiple times leads to almost the same result. However, the Pulse regularizer uses a projection onto a subset of the latent space and a random initialization. This Pulse subset has high quality, but overall the restriction is unnecessarily restrictive. Also, the initial starting point and the location of the initial projection has a big influence on the final result. We believe Pulse is the only method embedding method that gives widely different results for the same input. Also, if the random initialization of Pulse is unlucky, the result of the embedding will be very different from the input image. See Fig. 13 for an example.

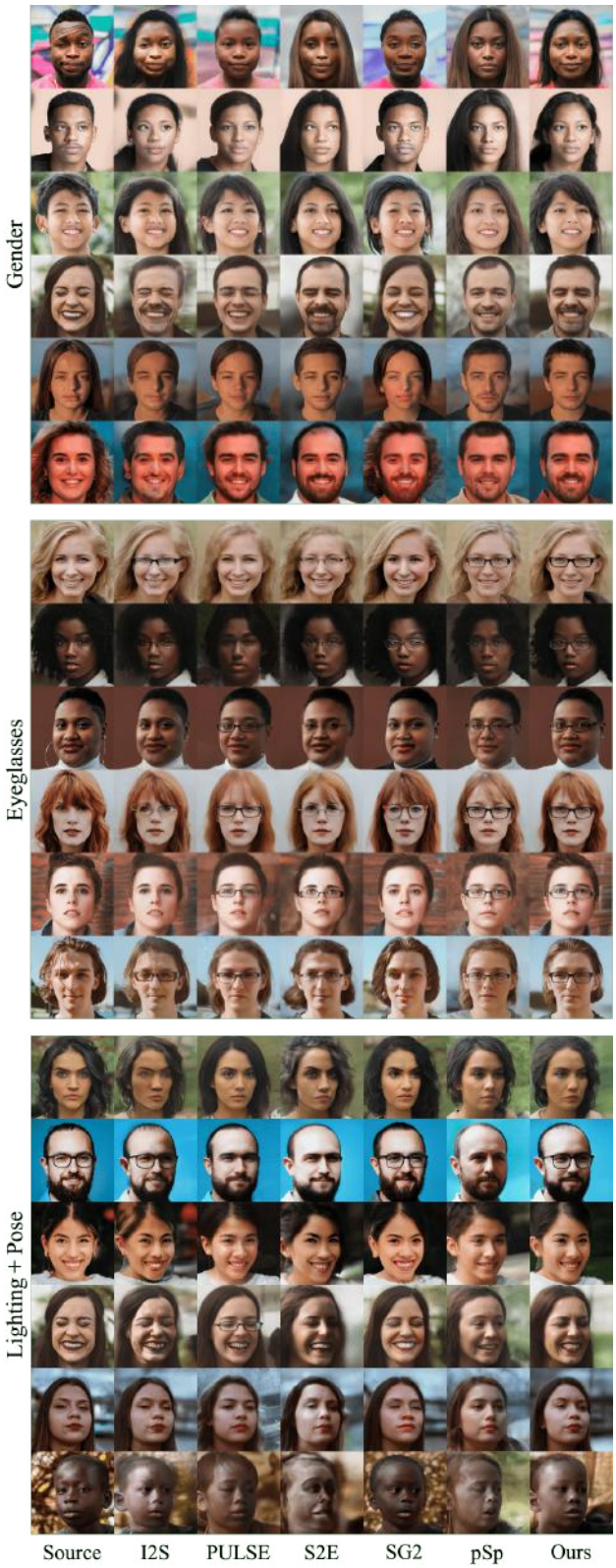


Figure 7: Examples of editing gender, eyeglasses and lighting + pose.

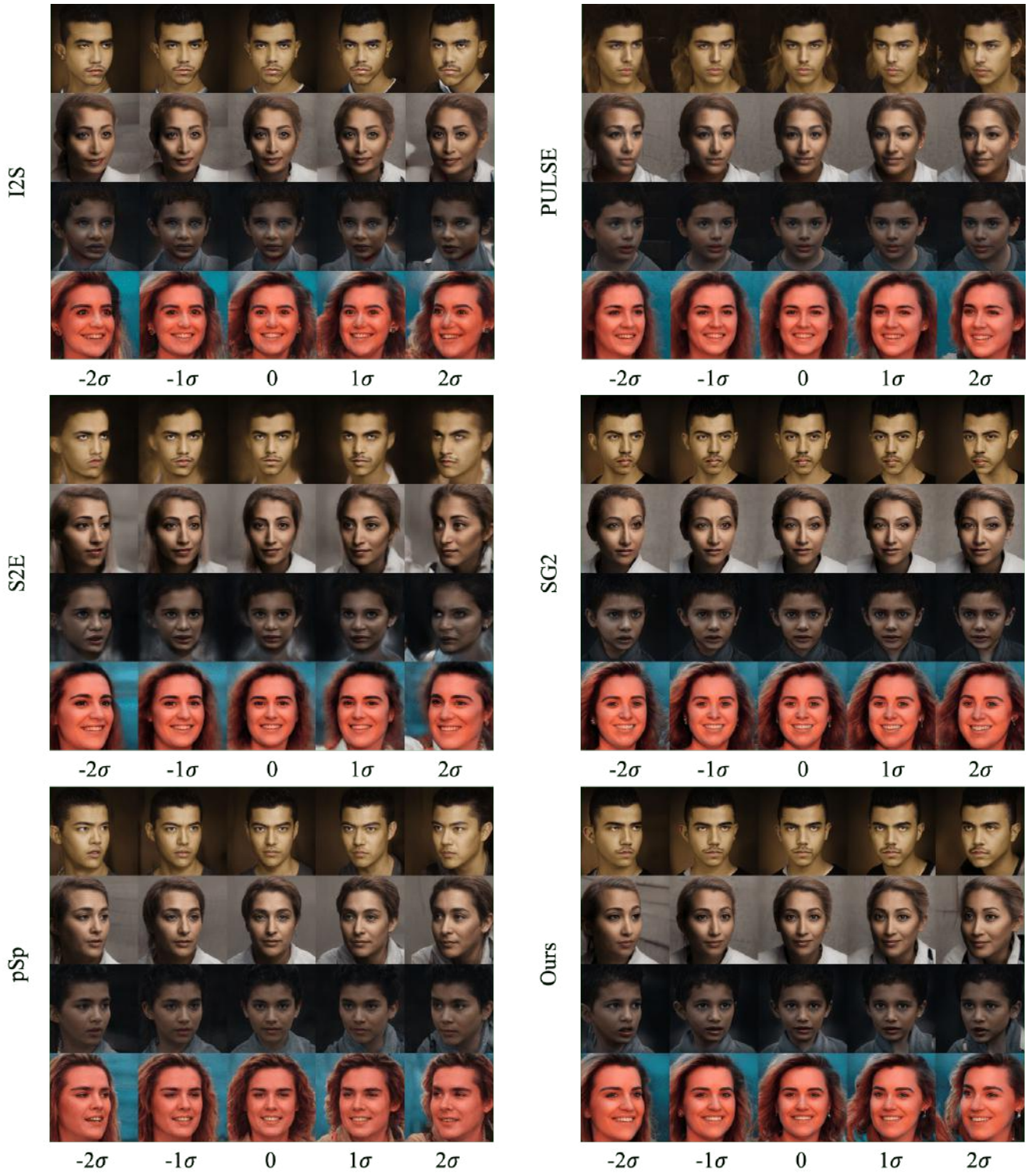


Figure 8: Examples of pose editing using GANspace [11].

Figure 9: Examples of Interpolation in W^+ space.

Figure 10: Examples of Interpolation in P_N^+ space.

Figure 11: Examples of style mixing.

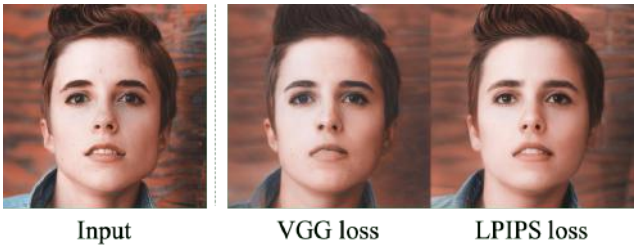


Figure 12: LPIPS perceptual loss vs. VGG perceptual loss.

Figure 13: Comparing the stability of the algorithms using repeated embeddings. Each run of PULSE will provide different results, while our results are relatively stable. This is a visual demonstration of the problem that the PULSE regularizer creates.

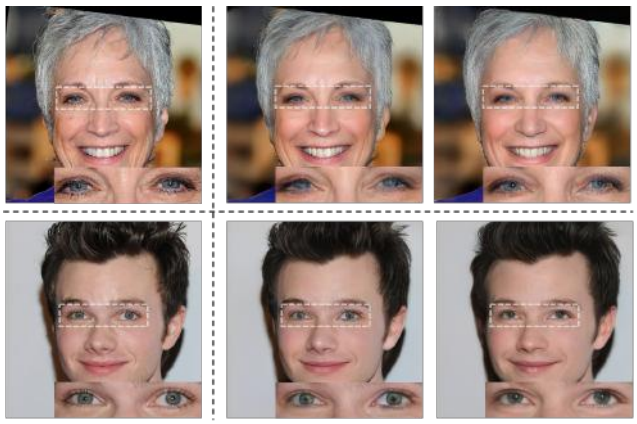


Figure 14: Eye correction obtained by applying the regularizer in the P_N^+ space on W^+ embedding. Left: original image, middle: W^+ embedding, right: W^+ embedding with P_N^+ regularizer.

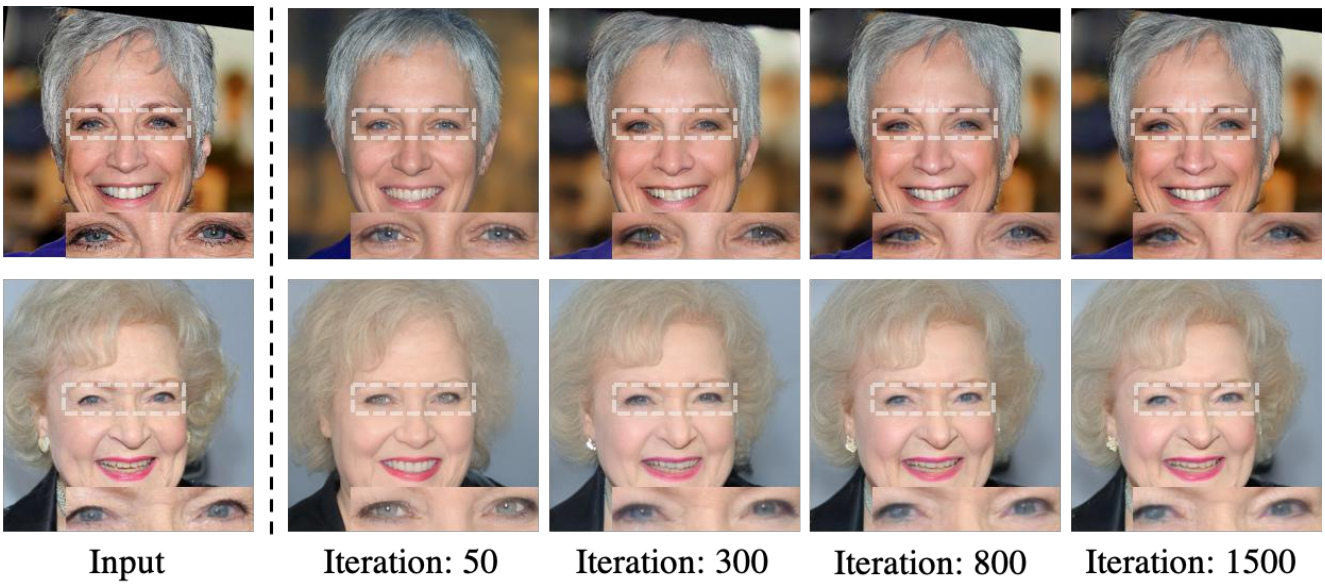


Figure 15: Progression of embedding using I2S. **Notice that the eyes eventually lose their realism with 800 iterations.** The original image is shown to the left.