

Improved StyleGAN Embedding: Where are the Good Latents?

PEIHAO ZHU, KAUST, Saudi Arabia
 RAMEEN ABDAL, KAUST, Saudi Arabia
 YIPENG QIN, Cardiff University, UK
 JOHN FEMIANI, Miami University, USA
 PETER WONKA, KAUST, Saudi Arabia

StyleGAN is able to produce photorealistic images that are almost indistinguishable from real photos. 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.

CCS Concepts: • Computing methodologies → Image processing; Reconstruction; Neural networks.

Additional Key Words and Phrases: Image Editing, GAN embedding, StyleGAN

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 [Karras et al. 2019] and StyleGAN2 [Karras et al. 2020a,b] to develop GAN inversion algorithms. The resulting embedding methods represent a trade-off 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. The editing quality describes the visual quality and realism of images after performing editing operations in latent space, e.g., editing the pose, lighting, or age of a face images [Härkönen et al. 2020].

In this paper, we set out to provide some analysis of existing embedding methods and use that analysis as the basis for an improved image-to-StyleGAN embedding method. One important consideration is the choice of the embedding space, e.g., Z space, W space, or $W+$ space. These spaces each impose trade-offs – the Z space, used when training StyleGAN or StyleGAN2, restricts latent codes to a hypersphere. The W space does not have a sphere constraint and uses a mapping network to disentangle latent codes. The W space is therefore less entangled than the Z space. However, both spaces use

Authors' addresses: Peihao Zhu, KAUST, Saudi Arabia, peihao.zhu@kaust.edu.sa; Rameen Abdal, KAUST, Saudi Arabia, rameen.abdal@kaust.edu.sa; Yipeng Qin, Cardiff University, UK, qiny16@cardiff.ac.uk; John Femiani, Miami University, 510 E. High St, Oxford, OH, 45056, USA, femianjc@miamioh.edu; Peter Wonka, KAUST, Saudi Arabia, pwonka@gmail.com.

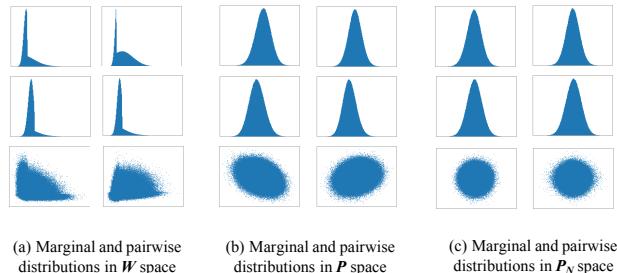


Fig. 1. Marginal and pair-wise distributions of different spaces in StyleGAN. First and second rows: the marginal distribution of a randomly picked latent variable. Third row: the pairwise joint distribution of two randomly picked latent variables.

512 dimensional latent codes. This does not provide enough expressiveness to represent all real-world faces well. In [Abdal et al. 2020a] it was shown that $W+$ codes, in which the degrees of freedom are 18 times larger than W space, are capable of reconstructing images well. However, these additional degrees of freedom make GAN inversion an ill-conditioned problem. Thus, non regularized inversion of $W+$ codes can find embeddings in low-density portions of the latent space, so that even if a particular image is reconstructed well, the reconstruction is in a region that will not normally be reached by the mapping network. Empirically, this leads to undesirable images after GAN-based image manipulations such as interpolation or semantic editing.

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, illustrated in Fig. 1, which we approximate by a multivariate normal distribution. 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 densely sampled regions of the GAN latent space. 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 in P_N space. 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.

We propose an improved embedding algorithm by introducing a regularizer that encourages embeddings to stay closer to the origin

of the P_N space, which is where the W latent-codes of training data have the highest density. We further extend this to the much higher capacity $W+$ space, introducing a corresponding P_N^+ regularizer. The advantage of our regularizer is that it explicitly controls the distance to the origin and does not have undesirable side effects. In particular, a regularizer will bias solutions towards more dense and editable portions of latent space, but it will not prevent optimization from finding solutions that reduce the reconstruction error. Regularizing will slightly reduce reconstruction quality, but much less-so than a constraint. Further, a regularization hyperparameter allows control over the trade-off between reconstruction quality and editability. 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 and 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 of 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. [Goodfellow et al. 2014] started an avalanche of GAN papers that together lead to impressive improvements over the last years [Arjovsky et al. 2017; Karras et al. 2017, 2020a, 2019, 2020b; Miyato et al. 2018; Radford et al. 2015]. Similar to most other work in GAN image embedding, we build on StyleGAN [Karras et al. 2020a, 2019, 2020b] due to its exceptional visual quality and comparatively lightweight architecture.

Since the recent inception of Image2StyleGAN (I2S) [Abdal et al. 2019] 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++ [Abdal et al. 2020a] improved the reconstruction quality of I2S by incorporating a noise optimization step to restore the high-frequency details in the input image. Similarly, StyleGAN2 [Karras et al. 2020b] 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 [Tewari et al. 2020a] 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 [Bau et al. 2020, 2019; Guan et al. 2020; Zhu et al. 2020], which to our knowledge first appears in iGAN [Zhu et al.

2016]. 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 [Menon et al. 2020] formulated embedding as traversing the manifold consisting of good latents. However, they constrained the latents to be on the surface of a hypersphere that only contains a subset of good latents. As a result, 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 [Karras et al. 2019, 2020b]. To find such a latent space, we conduct a statistical analysis on the latent distributions in different latent spaces illustrated in Fig. 2, including W , P_N , $W+$, and P_N^+ spaces.

W Space: The W space proposed in StyleGAN [Karras et al. 2019] is the most straightforward choice of latent space. Therefore, we start our investigation by analyzing the distribution of 1 million latent codes in W space and visually checking the statistics of their marginal distributions. We sample this space by first generating random samples in Z space, and then forward-propagating through the mapping layers of StyleGAN2. We use the same input distribution as StyleGAN, in which the latent-codes in Z space are projected onto a hypersphere before propagating them through the mapping network. As Fig. 1 (a) shows, the marginal distributions are heavily skewed.

P Space: To get rid of the skewness of marginal distributions, we transform the W space to the P space by inverting the last Leaky ReLU layer in the StyleGAN mapping network. Since the last Leaky ReLU uses a slope of $\frac{1}{5}$ we invert it using

$$x = \text{LeakyReLU}_{5.0}(w) \quad (1)$$

where w and x are latent codes in W and P space respectively. Similar to those in W space, we plot the marginal distributions of latent codes in P space and observed that they appear to follow a simple Gaussian-like distribution. We plot the pairwise joint distribution of latent codes (See Fig. 1b) and observe that the dimensions are not statistically independent. Based on the marginal plots, we suspected that the distribution may be a Multivariate Normal Distribution, so we performed a Henze-Zirkler [Henze and Zirkler 1990] test for multivariate normality in 512 dimensions, using 1,000 samples, the resulting test indicates that the data is not normal, with a significance of $\alpha = 5\%$. We also used a Mardia test [Mardia 1970, 1974] of multivariate normality, which is based on a measurement of multivariate skewness, and kurtosis. The Mardia test had similar results; the distribution is not normal, with a significance of 5%.

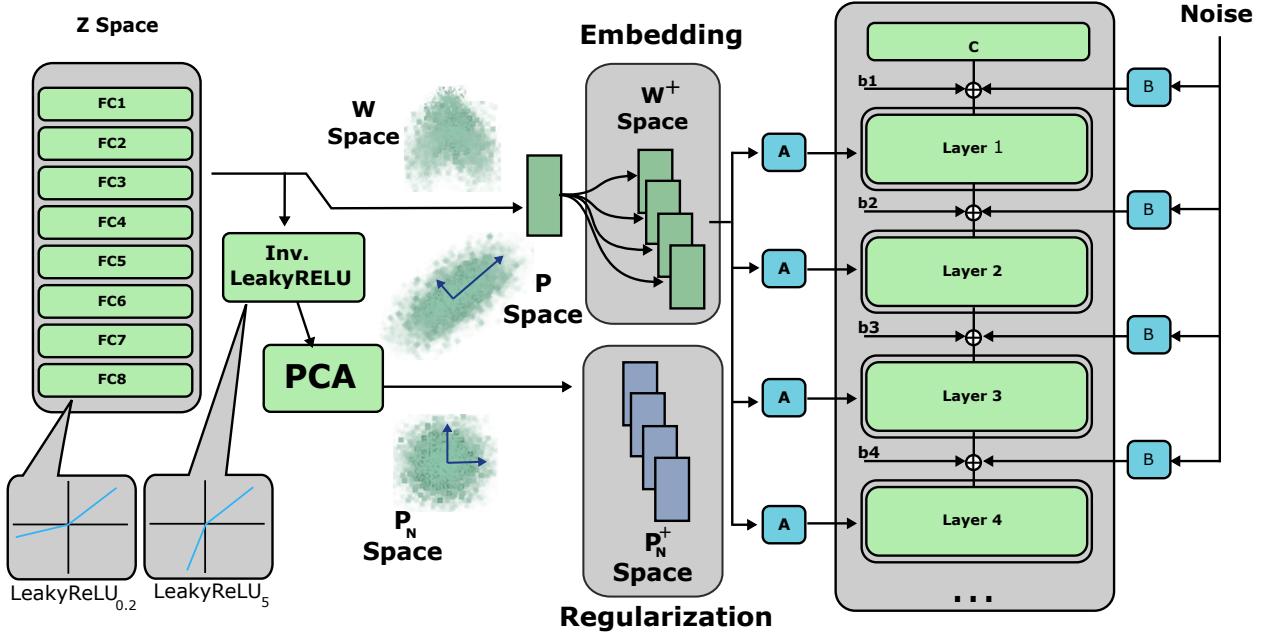


Fig. 2. We show six different spaces into which projection of a given image is possible. These are: Z space, W space, W^+ space, P space, P_N space and P_N^+ space. Notice that translation from W to P or P to W space goes through an invertible function (Leaky ReLU).

We tested the marginal distribution of the data along each principal component axis, and found that 219 of the 512 marginal are normal at 5% significance according to the Mardia test. We show two marginal distributions where the Mardia statistic indicates that the data is not normal in Fig 3. Since the distribution visually appears to have a unimodal bell-shaped marginal distribution when we look at a single axis, we used a Dip test [Hartigan and Hartigan 1985] for unimodality and confirmed that the multivariate distribution does indeed have a single peak. Although the distribution is not a multivariate normal distribution, it is unimodal and has correlated features. Visual inspection of Fig 1 and Fig 3 indicates that even if it is not normal, the data could be well-approximated by a normal distribution. Another reason why the data cannot be a true normal distribution is because the mapping network uses an MLP architecture. Such an architecture creates a distribution with compact support, and the generated latent variables cannot be arbitrarily large. 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 is inspired by principal component analysis (PCA) whitening, and aims to eliminate the dependencies among latent variables in P space. We define a transformation from P to our P_N space as:

$$\hat{v} = \Lambda^{-\frac{1}{2}} \cdot U^T(x - \mu) \quad (2)$$

where Λ is a diagonal matrix of singular values, U is an orthogonal matrix of principal component directions, and μ is a mean vector.

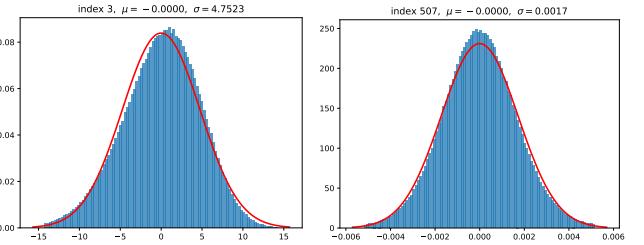


Fig. 3. Examples of marginal distributions of the latent-codes, with a best-approximating normal distribution indicated in red. The distribution is clearly not normal based on skew (principle axis 3, left subplot) and kurtosis (principle axis 507, right subplot), however even the marginals that fail a normality test are still visually quite similar to a normal distribution.

Intuitively, this whitening, or spherling, transform maps 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 $\|\hat{v}\|$ is related to the Mahalanobis distance $d_M(\cdot)$ of points $x \sim N(\mu, \Sigma)$ with the singular value decomposition of

symmetric $\Sigma = \mathbf{U}\Lambda\mathbf{U}^T$, that is

$$d_M^2(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad (3)$$

$$= \left(\Lambda^{-\frac{1}{2}} \mathbf{U}^T (\mathbf{x} - \boldsymbol{\mu}) \right)^2 \quad (4)$$

$$= \hat{\mathbf{v}}^T \hat{\mathbf{v}} \quad (5)$$

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 (regularization has more effect) to directions of low variance in the space used when training the GAN. The effect of this space on latent codes is illustrated in Fig 4

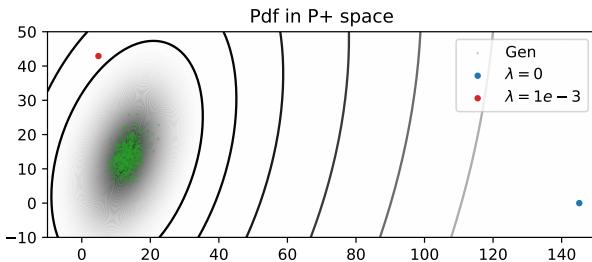


Fig. 4. The relationship of latent codes generated by the GAN (green) and two latent codes found by GAN inversion of the same image. The distribution is projected onto the plane containing the two latent-codes and the center of the distribution. We observe that increasing regularization does not shift the latent code straight towards the center, the reconstruction is less regularized along the long axis of the ellipse, and it is closer to the center in the short leg of the ellipse.

We are inspired by generalized Tikhonov regularization, which biases latent codes to more dense portions of the latent space. Unlike truncation, this regularization approach has a much greater effect on subspaces where the input distribution has higher precision, and it has less effect in subspaces where the input distribution has high variance. Simply shifting a latent code towards the origin in $W+$ space would, for example, also change the components of the latent code that are important for reconstruction. On the other hand, regularizing based on d_M (i.e. in P_N space) has the desirable effect that it preserves components that also have high variation in the training corpus.

Without regularization, editing tasks are affected because images contain elements that are not modeled by the GAN, so that their embedding can end up in a less-explored region of the embedding space, where the GAN may be more likely to produce low quality results even though the particular image being inverted seems plausible. This effect can be observed by comparing the negative log-density (the Mahalanobis distance, $d_M^2 = \|\hat{\mathbf{v}}\|^2$) vs the number of optimization steps for GAN inversion with and without regularization, shown in Fig 5. Reconstruction quality is improved by a negligible amount, whereas the log-density becomes arbitrarily high. An edit operation shifts the embedding to a new location, so if the original embedding was in a high density part of the embedding space the perturbed embedding is also likely to be nearby in a high density region. However, in the low-density portion of the space, a perturbed image will also have a low-density and since it is not the

result of inverting a ‘real world’ image it may be perceived as low quality or unrealistic.

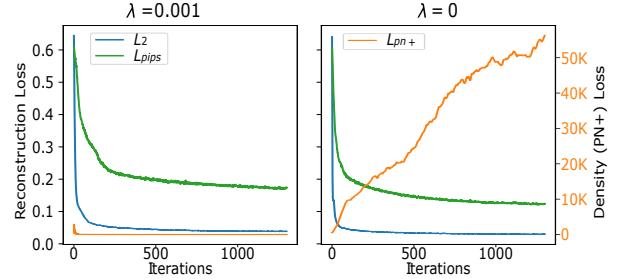


Fig. 5. The effect of regularization in P_N^+ space vs reconstruction quality. The orange curves are proportional to the negative log-density ($\|\mathbf{v}\|^2$) of points based on a normal approximation to the distribution of latent codes, the green and blue curves are reconstruction losses. Observe that P_N^+ loss can be kept very low with little effect on reconstruction, whereas ignoring the loss causes $\|\mathbf{v}\|^2$ to be arbitrarily large (note that the scale of the y -axis on the right is different in order to fit the curve into the plot). Shifting the latent codes towards higher-density regions has very little effect on reconstruction quality.

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 [Karras et al. 2020b]. P_N space sampling yields an FID of 3.28 compared to an FID of 2.81. Such a 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 [Abdal et al. 2019].

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 [Abdal et al. 2019],

$$\mathbf{w}^+ = \{\mathbf{w}_i\}_{i=1}^{18} \quad (6)$$

we extend P_N space to P_N^+ space as

$$\mathbf{v} = \{\Lambda^{-1} \mathbf{V}^T (\mathbf{x}_i - \boldsymbol{\mu})\}_{i=1}^{18} \quad (7)$$

where $\mathbf{x}_i = \text{LeakyReLU}_{5.0}(\mathbf{w}_i)$. Each of the latent-codes 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 Reconstruction Losses

One important aspect of I2S is to use a perceptual regularizer based on VGG-perceptual loss [Simonyan and Zisserman 2014]. However, we find that using LPIPS [Zhang et al. 2018] results in a noticeable improvement. Following PULSE [Menon et al. 2020] we also propose to use bicubic downsampling on the generated images and Lanczos downsampling [Turkowski [n.d.]] 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.

4.2 P_N^+ Density Based Loss

Based on our results, it seems most promising to devise a regularizer that penalizes an embedding if it goes too far from the center in P_N^+ . Let I be the input image, $L = L_{LPIPS} + L_2$ be a loss function consisting of LPIPS and pixel-wise L2 loss terms, w^+ and v be the latent codes in W^+ and P_N^+ space respectively, and λ is a hyperparameter controlling the amount of regularization. We have:

$$w^{+*} = \arg \min_{w^+} L(I, G(w^+)) + \lambda \|v\|^2 \quad (8)$$

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 EVALUATION

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.

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 [Abdal et al. 2019] is the baseline method we would like to improve upon which embeds images into the W^+ space.
- PULSE [Menon et al. 2020] proposed embedding onto the surface of a hypersphere in Z space.
- StyleGAN2 [Karras et al. 2020b] embeds images into the W space with the help of noise regularization.
- StyleGAN2Encoder [robertluxemburg 2020] 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 [Richardson et al. 2020] 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 hyperparameter settings for I2S to limit overfitting. We do not compare to the IDInvert method [Zhu et al. 2020] and Multi-code embedding [Gu et al. 2020] because they only work on the original StyleGAN architecture [Karras et al. 2019] at a low resolution of 256×256 . We also do not use noise embedding as proposed by I2S++ [Abdal et al. 2020a] because it is too hard to control with respect to editing quality.

For the evaluation, it is important to use a set of test images that were not part of the set used to train StyleGAN2. 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 [Heusel et al. 2017] 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. 6, 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.

Method	SSIM	RMSE	PSNR	VGG	LPIPS	FID
pSp [Richardson et al. 2020]	0.81	0.09	21.35	0.91	0.30	54.80
PUL [Menon et al. 2020]	0.81	0.09	21.63	1.05	0.37	64.63
S2E [robertluxemburg 2020]	0.84	0.08	22.50	0.93	0.32	70.70
SG2 [Karras et al. 2020b]	0.79	0.12	19.47	0.80	<u>0.21</u>	46.49
I2S [Abdal et al. 2019]	0.83	0.07	23.45	0.63	<u>0.21</u>	36.60
$\lambda = 0.01$	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 [Simonyan and Zisserman 2014], LPIPS [Zhang et al. 2018], and FID [Heusel et al. 2017]. The two best results are underlined; the best result is bold.

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 [Abdal et al. 2020b; Härkönen et al. 2020; Shen et al. 2020; Tewari et al. 2020b]. Note that these editing frameworks are orthogonal to our work, as they do not propose new embedding algorithms. We choose StyleFlow [Abdal et al. 2020b] 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 the viewer’s right. In Fig. 7, 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 not fully accomplish the editing task. For example, when performing an edit to change the age, PULSE produces results

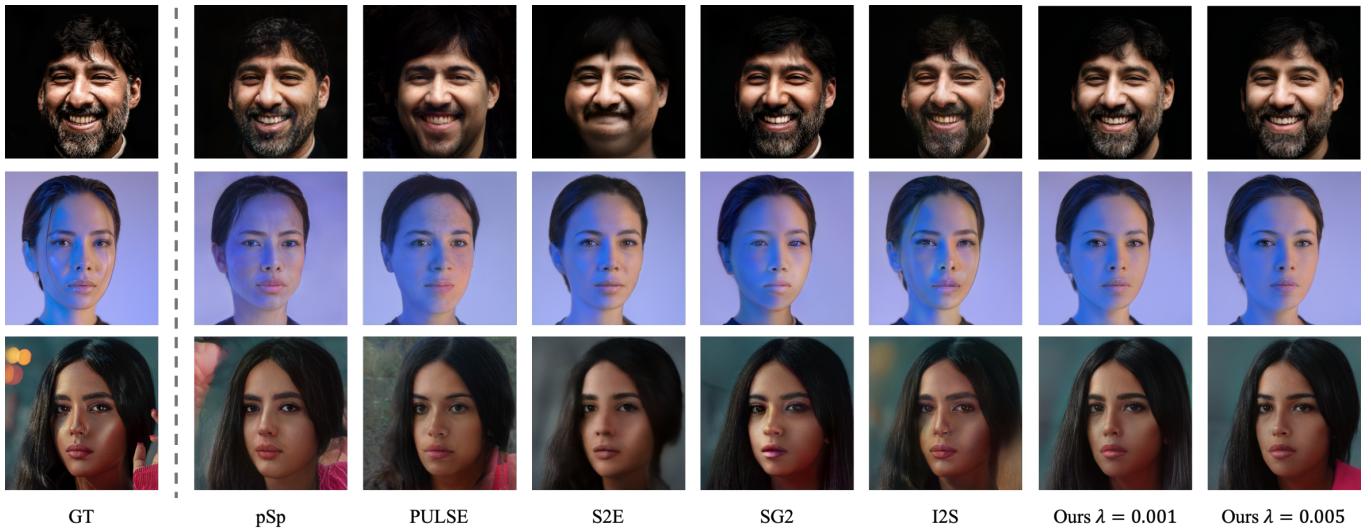


Fig. 6. Reconstruction quality of different embedding methods using the StyleGAN2 generator. Close inspection will show that none of the methods can cope with specular reflections. The most accurate reconstruction results are from I2S, which is equivalent to II2S (ours) with $\lambda = 0$. However, by increasing λ we are able to find a compromise between high reconstruction quality and also realism. For example, the nose-ring on the bottom row is outside the space of the GAN, I2S generates an artifact that II2S does not.



Fig. 7. Four selected examples each for editing age, pose, lighting, and style transfer. Close inspection shows that II2S (Ours) consistently accomplishes the edit and also maintains the highest level of realism and also similarity to the source image after edits; this is consistent with a user study.

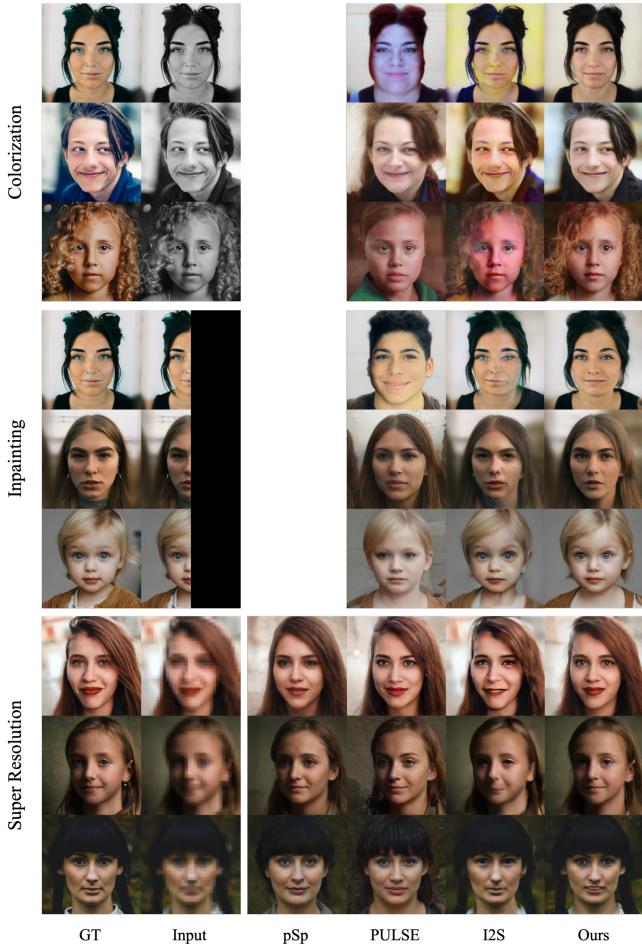


Fig. 8. Examples of conditional embedding results. Three example of each task are shown from top to bottom: Image colorization, inpainting, and super resolution. For pSp we only include their super-resolution results because they only included model weights for that task. Close inspection will show that II2S results satisfy the conditions with most realistically reconstructed images.

that look noticeably younger than the target age. We further study this phenomenon using the state-of-the-art classifier Microsoft Face API [Microsoft 2020] 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 w^{+*} in W^+ space so that the embedded image $G(w^{+*})$ i) faithfully captures the “condition” of I and ii) is a realistic face image. Accordingly, we define conditional embedding as:

$$w^{+*} = \arg \min_{w^+} L(I, f(G(w^+))) + \lambda \cdot R(w^+) \quad (9)$$

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 L_2 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 converts 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 generators, G is trained on the FFHQ dataset and G' is a variant of G fine-tuned on the MetFace [Karras et al. 2020a] 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. 8 and results for style transfer are shown in Fig. 7. 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 ground truth 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 loses to the 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 II2S over a competing method, and a number above 50% means that the II2S solution was selected more frequently. Each survey received 120 responses, so any number below 40% or above 60% 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. The pSp approach for editing age was favored by more users, however the results were not statistically significant. 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 applications.

	GT	I2S	SG2	S2E	PUL	pSp
Reconstruction	-	39.2	58.3	71.7	83.3	55.0
Edit:Age	-	80.0	57.5	97.5	64.2	47.08
Edit:Pose	-	69.2	55.0	79.2	69.2	58.4
Edit:Light	-	79.2	64.2	88.3	76.7	52.94
Style Trans.	-	77.5	84.2	76.7	73.3	90.83
Colorization	64.4	95.0	-	-	79.4	-
Inpainting	28.3	65.8	-	-	75.8	-
Super Res.	43.3	94.17	-	-	65.8	52.5

Table 2. A user study comparing II2S to the Ground Truth (GT), pSp, StyleGAN2 (SG2), StyleGAN2 Encoder (S2E), Pulse (PUL) and I2S. Cases where II2S is preferred are in bold. Excluding ground-truth, there are only two competing methods that outperform II2S for any task.

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 so that a visual difference is noticeable between images generated from the embeddings. In Fig. 6 we show the visual difference between $\lambda = 0.005$ and $\lambda = 0.001$. In Tab. 1 we show how λ affects reconstruction metrics. However, reconstruction metrics are not as informative as human perception. A user study to evaluate different choices of λ is shown in Table 3, and based on the average user response 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 from 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 20th 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. 9 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.

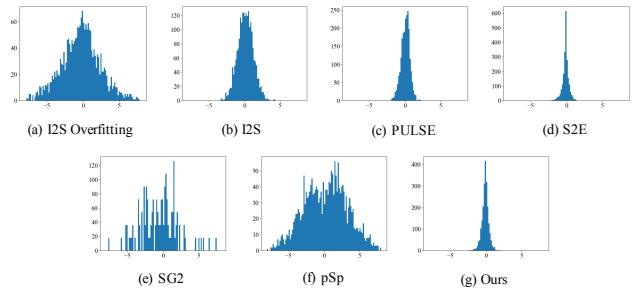


Fig. 9. Histograms showing the distribution of latents in an arbitrary (the twentieth) dimension of the P_N^+ Space.

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 L_2 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 competing methods for downstream applications.

REFERENCES

- Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2stylegan: How to embed images into the stylegan latent space?. In *Proceedings of the IEEE international conference on computer vision*. IEEE, Long Beach, CA, 4432–4441.
- Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020a. Image2StyleGAN++: How to Edit the Embedded Images?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8296–8305.
- Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. 2020b. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv e-prints* (2020), arXiv–2008.
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).
- David Bau, Hendrik Strobelt, William Peebles, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, et al. 2020. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727* (2020).
- David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. 2019. Seeing what a gan cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*. 4502–4511.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
- Jinjin Gu, Yujun Shen, and Bolei Zhou. 2020. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3012–3021.
- Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. 2020. Collaborative Learning for Faster StyleGAN Embedding. *arXiv preprint arXiv:2007.01758* (2020).
- Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. *arXiv preprint arXiv:2004.02546* (2020).
- J. A. Hartigan and P. M. Hartigan. 1985. The Dip Test of Unimodality. *The Annals of Statistics* 13, 1 (1985), 70 – 84. <https://doi.org/10.1214/aos/1176346577>
- N Henze and B Zirkler. 1990. A class of invariant consistent tests for multivariate normality. *Communications in statistics-Theory and Methods* 19, 10 (1990), 3595–3617.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*. 6626–6637.
- Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).
- Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. In *Proc. NeurIPS*.
- Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4401–4410.
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020b. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.
- Kanti V Mardia. 1970. Measures of multivariate skewness and kurtosis with applications. *Biometrika* 57, 3 (1970), 519–530.
- Kanti V Mardia. 1974. Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies. *Sankhyā: The Indian Journal of Statistics, Series B* (1974), 115–128.
- Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. 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*. 2437–2445.
- Microsoft. 2020. Azure Face. <https://azure.microsoft.com/en-in/services/cognitive-services/face/>
- Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).
- Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
- Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2020. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. *arXiv preprint arXiv:2008.00951* (2020).
- robertluxemburg. 2020. GitHub repository: stylegan2encoder. <https://github.com/robertluxemburg/stylegan2encoder>.
- Yujun Shen, Ceyuan Yang, Xiaou Tang, and Bolei Zhou. 2020. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. *arXiv preprint arXiv:2005.09635* (2020).
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, Christian Theobalt, et al. 2020a. PIE: Portrait Image Embedding for Semantic Control. *arXiv preprint arXiv:2009.09485* (2020).
- Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. 2020b. StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6142–6151.
- Ken Turkowski. [n.d.]. Turkowski Filters for Common Resampling Tasks 10 April 1990 Filters for Common Resampling Tasks.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.
- Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W. Jacobs. 2019. Deep Single Portrait Image Relighting. In *International Conference on Computer Vision (ICCV)*.
- Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain GAN Inversion for Real Image Editing. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*. Springer, 597–613.

A METRICS

A.1 Comparison of Editing Quality

We choose StyleFlow [Abdal et al. 2020b] 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 [Härkönen et al. 2020] 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 the 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. 10, 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 Microsoft’s face API [Microsoft 2020]. Lighting is measured by a different network [Zhou et al. 2019]. Overall, our method has the best results. pSp is very slightly better than ours in two edits, but clearly worse in two others.

In addition to using StyleFlow [Abdal et al. 2020b] as the evaluation method of editing quality, for the pose change experiments, we use the right-left pose edit from GANSpace [Härkönen et al. 2020]. Specifically, for a latent code in the GANSpace coordinate system, we can set the coordinate corresponding to pose to five different values: -2σ , $-\sigma$, 0 , σ , and $+2\sigma$, where σ is the eigenvalue for the direction. Fig. 11 shows the qualitative results of pose change edits. Quantitative results are presented in Table 5. We observe that the strength of an edit depends on the initial latent code and the

Method	Age	Pose	Lighting	Gender	Eyeglasses
	50	20	R→L	Swap	Add
pSp [Richardson et al. 2020]	38.25	<u>18.2</u>	0.87	0.78	1.00
PUL [Menon et al. 2020]	33.8	18.3	0.88	0.71	0.76
S2E [robertluxemburg 2020]	60.1	25.7	<u>0.91</u>	<u>0.93</u>	0.83
SG2 [Karras et al. 2020b]	29.4	13.2	0.85	0.33	0.38
I2S [Abdal et al. 2019]	<u>53.4</u>	18.1	0.90	0.85	0.71
Ours	50.2	23.5	0.94	0.99	0.98

Table 4. Quantitative comparison of editing quality.

embedding method. This poses a challenge for the evaluation of different embedding methods using GANSpace.

Method	-2σ	-1σ	0	1σ	2σ
pSp	19.40	10.48	-0.58	-11.58	-20.63
PUL	20.12	10.16	-0.31	-9.66	-17.99
S2E	<u>25.75</u>	11.66	-1.34	-13.52	-26.35
SG2	12.49	4.84	-2.04	-8.36	-14.75
I2S	18.42	8.29	-0.81	-9.27	-18.49
Ours	26.23	13.16	-0.01	-12.07	<u>-23.42</u>

Table 5. Pose metric for testing the editing quality using GANspace [Härkönen et al. 2020]. We mark the highest ranges. These numbers by themselves are not conclusive with regard 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. 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 the W^+ 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. Qualitative results are shown in Fig. 15. We propose the use of P_N^+ as a regularizer, however it is a nonlinear mapping from W^+ space, which can negatively impact entanglement. However, we show the results of linear interpolation in Fig. 16, 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 good a latent disentangles the style and contents at different layers of the StyleGAN

Method	Style Mixing	Interp. in W^+	Interp. in P_N^+
pSp	65.45	57.93	58.74
PUL	65.05	63.73	63.39
S2E	66.22	67.84	69.91
SG2	58.48	49.40	<u>54.19</u>
I2S	<u>57.36</u>	43.03	55.25
Ours	55.69	<u>48.61</u>	49.15

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

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. 17. 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 is 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 may overemphasize reconstruction quality in the hyperparameter settings, when the

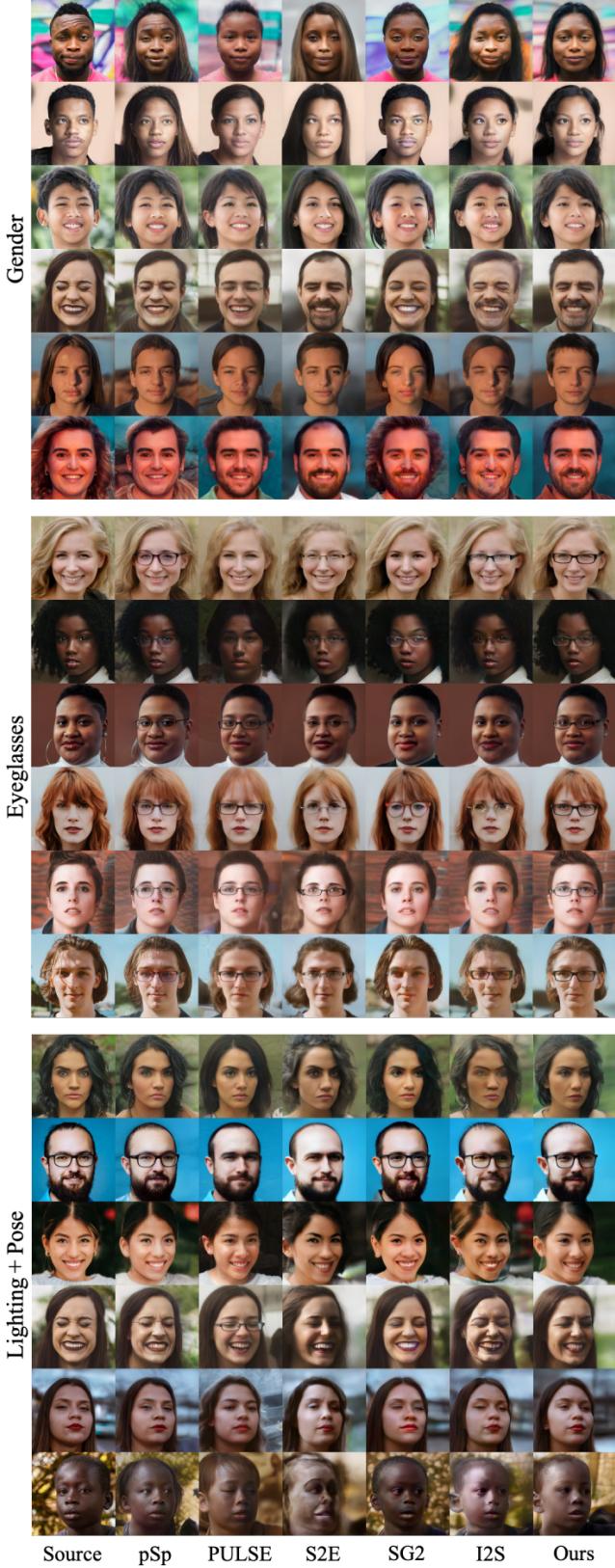


Fig. 10. Examples of editing gender, eyeglasses and lighting + pose. The results of edits using the proposed P_N^+ regularizer are shown on the right. Compared to other methods, our results tend to preserve non-edited attributes/identity of the input, generate realistic images, and also accomplish the editing task.

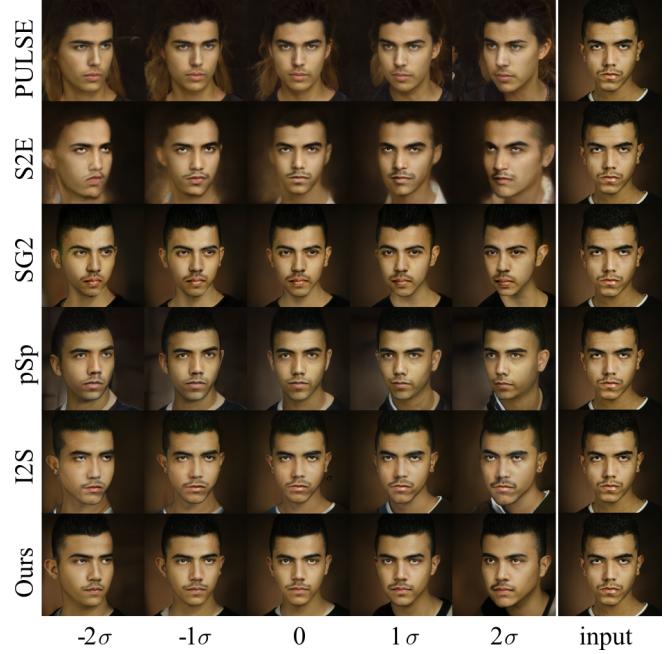


Fig. 11. Examples of pose editing using GANspace [Häkkinen et al. 2020]. Our approach does a better of reconstructing the input than many methods (notice the facial hair) and it also accomplished the editing task. Compared to I2S, which also does well at reconstructing the image near the input pose, our approach maintains realism at the extreme poses. For example, the nose position in 3D, and the shape of the iris are preserved well in ours.



Fig. 12. LPIPS perceptual loss vs. VGG perceptual loss. The original I2S used VGG-loss, but LPIPS encourages better reconstruction of details.

goal it editing. For example, we could observe that Zhu et al. [Zhu et al. 2020] and Tewari et al. [Tewari et al. 2020a] use suboptimal I2S [Abdal et al. 2019] hyperparameter settings for some 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. 18 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



Fig. 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 hypersphere constraint creates. Our approach has no such constraint.

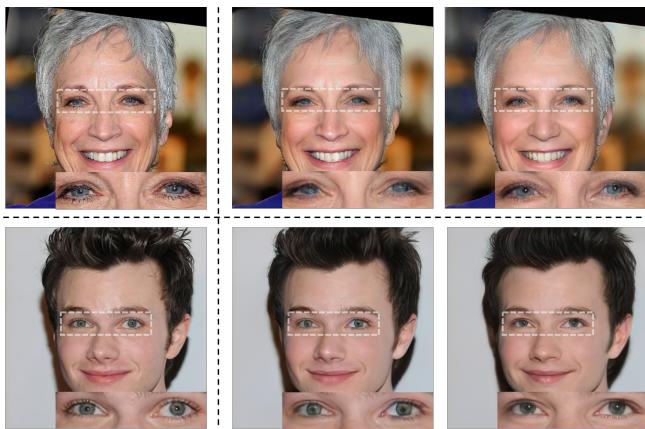


Fig. 14. Eye correction obtained by applying the regularizer in the P_N^+ space on W^+ embedding. Left: original image, middle: W^+ embedding, right: W^+ embedding with P_N^+ regularizer. Without our regularization, the generator is not able to preserve the shapes of the pupil and iris.

time-consuming to do this for our complete dataset, we believe that I2S results would improve using per image fine-tuning. 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 experiment 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.

Interpolation in W^+ Space



Fig. 15. Examples of Interpolation in W^+ space. This shows two linear interpolations, interpolating from column one to column five, and then from column five to column ten. The last row uses our embeddings - notice that the initial reconstruction quality (columns one, five, ten) is high, and that in-between images remain plausible.

Interpolation in P_N^+ Space



Fig. 16. Examples of Interpolation in P_N^+ space. This shows two linear interpolations, one from the top-left image to the center, and the other interpolation from top-center to the top-right image. We do not recommend actually performing edits in this space – P_N^+ space is useful for *finding* latents in dense regions of the mapping network, however the actual edits produce more intuitive results if they are carried out in a disentangled space such as W^+ .

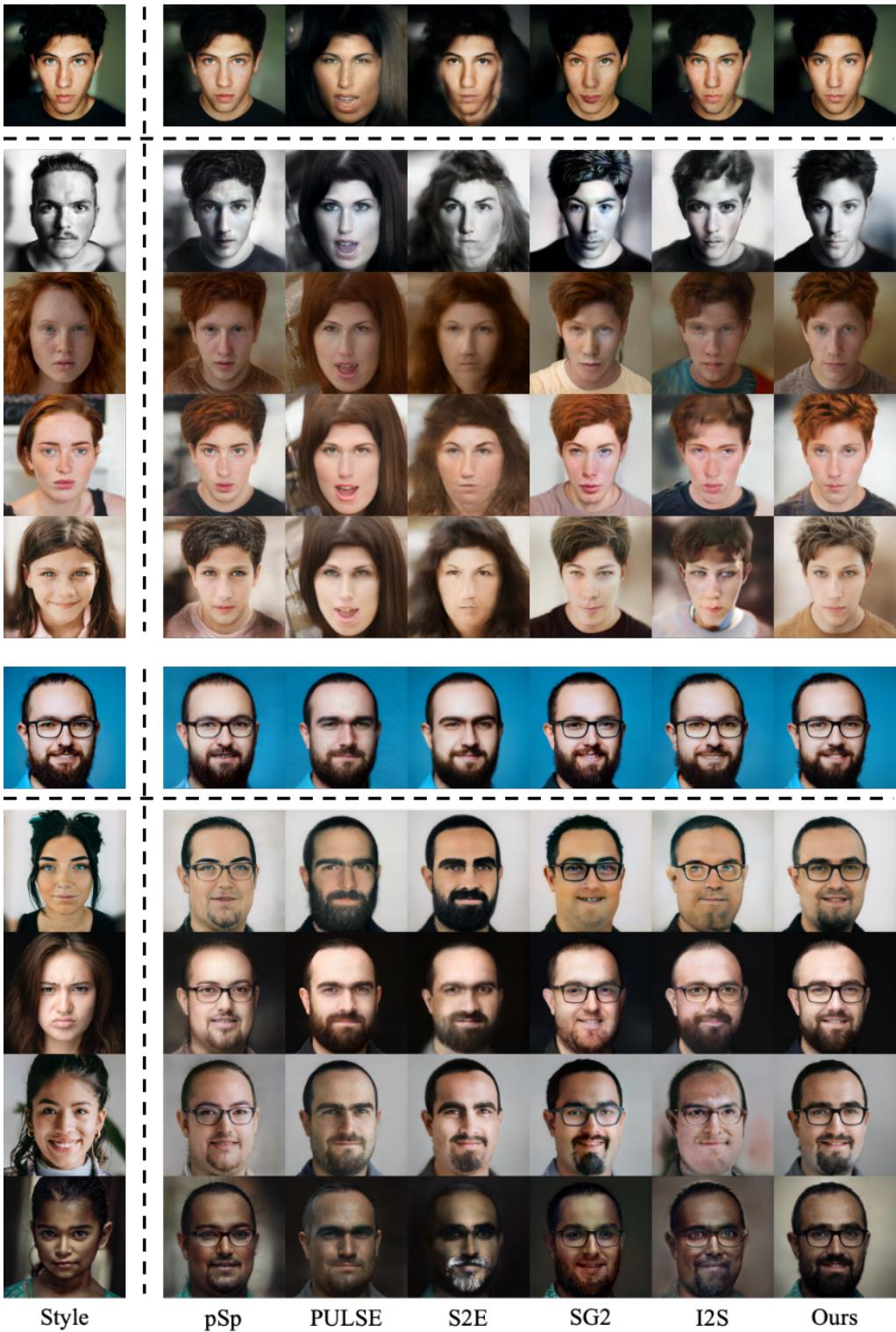


Fig. 17. Examples of style mixing.

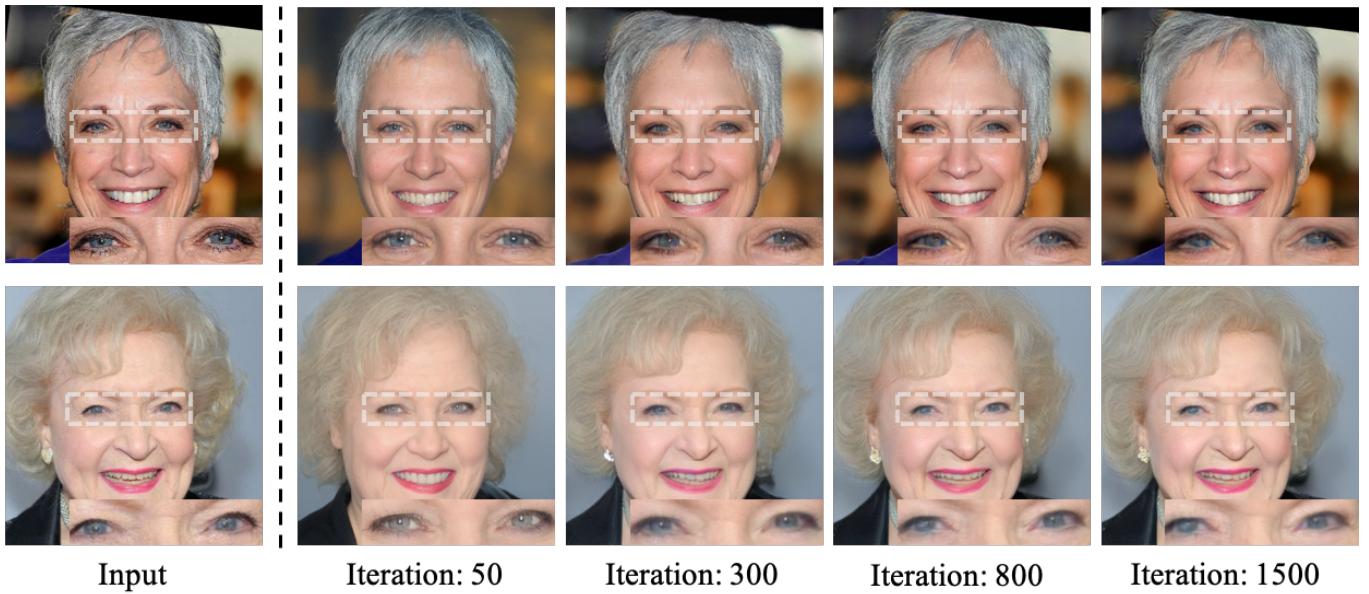


Fig. 18. Progression of embedding using I2S. Notice that the eyes eventually lose their realism with 800 iterations. The original image is shown to the left.

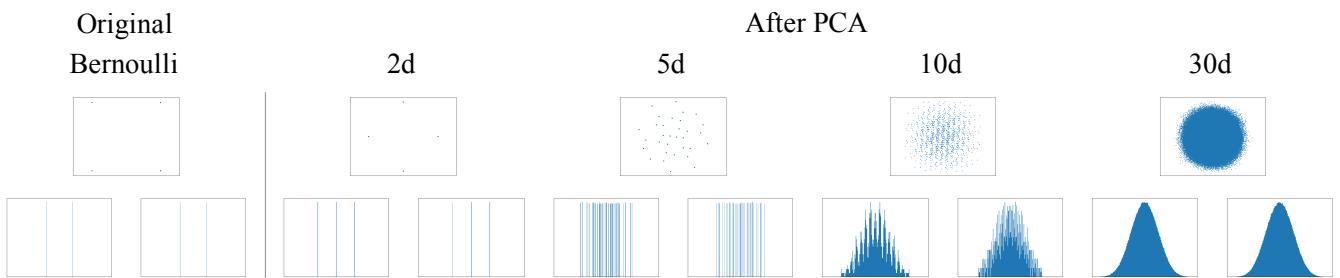


Fig. 19. Marginal and pair-wise distributions of Bernoulli distribution in different dimensions before and after PCA. Only show the first two dimensions after PCA.