

GAN Inversion with Editable StyleMap

So Honda, Ryohei Orihara^a, Yuichi Sei^b, Yasuyuki Tahara^c and Akihiko Ohsuga^d

The University of Electro-Communications, Tokyo, Japan


Keywords: GAN Inversion, StyleGAN, StyleMap, Editability, Local Editing.


Abstract: Recently, the field of GAN Inversion, which estimates the latent code of a GAN to reproduce the desired image, has attracted much attention. Once a latent variable that reproduces the input image is obtained, the image can be edited by manipulating the latent code. However, it is known that there is a trade-off between reconstruction quality, which is the difference between the input image and the reproduced image, and editability, which is the plausibility of the edited image. In our study, we attempted to improve reconstruction quality by extending latent code that represents the properties of the entire image in the spatial direction. Next, since such an expansion significantly impairs the editing quality, we performed a GAN Inversion that realizes both reconstruction quality and editability by imposing an additional regularization. As a result, the proposed method yielded a better trade-off between the reconstruction quality and the editability against the baseline from both quantitative and qualitative perspectives, and is comparable to state-of-the-art(SOTA) methods that adjust the weights of the generators.


1 INTRODUCTION

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are generative models consisting of a Generator, which generates data similar to the training data, and a Discriminator, which distinguishes whether the input is true data or generated data. StyleGAN(Karras et al., 2019, 2020), which has made great achievements as unconditional image generation application, can not only generate high-quality images but also control the semantic properties of the images because in the latent space the properties are disentangled and become independently manipulatable. In StyleGAN, noise \mathbf{z} , which follows a standard normal distribution, is transformed into latent code \mathbf{w} by a mapping network and used for image generation. In pre-trained StyleGAN, the latent space \mathcal{W} , which is the space where latent code \mathbf{w} is distributed, is known to have the disentanglement properties. In other words, various image editing can be realized by controlling the \mathcal{W} space. However, in order to edit an arbitrary image, latent code is needed to generate such an image. The task of “inverting” the input image to the latent code is called GAN Inversion. Usu-

ally, inversion into \mathcal{W} space is the last resort due to its poor reconstruction quality. Therefore, many studies use $\mathcal{W}+$ space with as many different latent codes as the number of convolution layers. At a resolution of 1024×1024 , an arbitrary input image is represented in $\mathcal{W}+ \subset \mathbb{R}^{18 \times 512}$ space using 18 latent codes. However, there are two problems: (1) $\mathcal{W}+$ space is not always sufficient for reconstruction, and (2) Inversion in $\mathcal{W}+$ space impairs editability. To solve the former problem, we propose to perform GAN Inversion using StyleMap, which is a spatial extension of the latent code. The extension frees the StyleGAN Generator from the need to represent the features of the entire image as vectors and allows detailed reconstruction of each segmented image region. We found that this method yields high reconstruction quality but poor editability. Several studies (Tov et al., 2021; Zhu et al., 2020b) have shown that editability decreases as the estimated latent code deviates from the region of the latent space used in unconditional image generation. Therefore, we incorporated simple regularizations to improve editability. The regularizations make our GAN Inversion competitive with existing work in the trade-off between reconstruction quality and editability. Finally, to confirm that the editing results were qualitatively satisfactory, we actually edited images with StyleGAN using well-known methods as shown in Figure 1.

^a  <https://orcid.org/0000-0002-9039-7704>

^b  <https://orcid.org/0000-0002-2552-6717>

^c  <https://orcid.org/0000-0002-1939-4455>


^d  <https://orcid.org/0000-0001-6717-7028>



Figure 1: GAN Inversion by the proposed method and image editing results using it. From left to right: input image, reconstructed image, aged image, smiling image, and toonified image. The last two rows, from top to bottom: input image, reconstructed image, and local editing with neighboring images. Local editing mixes the original image at the top of the image and the neighboring image at the bottom.

This paper is organized as follows. Section 2 describes related works on GAN Inversion and its editability, and latent codes extended in the spatial direction. Section 3 explains the architecture of the proposed method and the regularization method. Section 4 presents the experimental setup and results. Section 5 provides a quantitative and qualitative evaluation of the proposed method. Section 6 discusses the limitations of our study and the validity of our assumptions. Finally, Section 7 presents conclusions and future research.

2 RELATED WORKS

2.1 GAN Inversion

GAN Inversion is a task to estimate the latent code of a GAN such that the desired image can be generated. The basic policy is to find the latent code that minimizes the difference between the input image and the generated image. The simplest way to achieve this is to optimize the latent code directly (Abdal et al., 2019) using, for example, gradient descent. While this method yields high reconstruction quality, it suffers from very long inference times. On the other hand, encoder-based methods (Richardson et al., 2021a; Tov et al., 2021), while inferior in reconstruction quality, have the significant advantage of faster inference time. Intermediate between these two methods are (1) separately optimizing a latent code inferred by a pre-trained encoder as an initial value and (2) progressively obtaining a better latent code us-

ing the encoder multiple times (Alaluf et al., 2021a).

2.2 Editability of GAN Inversion

Pre-trained StyleGANs can reconstruct arbitrary images due to their generative power. For example, a StyleGAN trained on a human face image can reconstruct an image of a cat's face or even a bedroom (Zhu et al., 2020a). On the other hand, there is no guarantee that the estimated latent code has good editability. Several studies have attributed this phenomenon to the difference in the distribution of the latent code at the time of generation and those of the estimated latent code.

Tov et al. (2021) proposed Encoder for Editing (e4e), which improves editability at the expense of reconstruction quality by inverting input image to a latent code distributed in a space close to the \mathcal{W} space at image generation. Here, the closeness to the distribution in \mathcal{W} space is defined as (1) the closeness of each of the 18 latent codes and (2) the closeness of the distribution of each latent code to the distribution in \mathcal{W} space. For the former, the encoder is trained to minimize the L_2 norm between the latent code controlling the coarsest scale and other latent codes. For the latter, a Discriminator is used to determine if the latent code is sampled from \mathcal{W} space.

Zhu et al. (2020b) qualitatively showed that $\mathbf{p} = \text{LeakyReLU}_{5.0}(\mathbf{w})$, the output of the mapping network \mathbf{w} before passing the final activation function, is multivariate normally distributed. They then performed a highly editable GAN Inversion by adding the Mahalanobis' distance between the estimated la-

latent code and the mean of \mathbf{p} as a penalty term and optimizing it by gradient descent. Note that the policy by Zhu et al. (2020b) allows for high reconstruction quality and editability, while the inference time is still long.

2.3 Spatially Extended Latent Code

In general, GAN uses a latent code to represent the properties of the entire image. However, there is also a demand to generate images by specifying different properties for each region. Examples include methods that generate images by providing semantic segmentation masks as conditions (Park et al., 2019; Isola et al., 2017), GAN architectures that simply extend the latent code in the spatial direction (Kim et al., 2021), and local editing methods for the generated image of a trained StyleGAN (Hong et al., 2020).

In particular, StyleMapGAN (Kim et al., 2021) is an image generation architecture that allows local editing of images by simply extending the latent code in the spatial direction, and also performs GAN Inversion by learning additional encoders. However, it is a modification of the StyleGAN1 architecture, making it difficult to take advantage of the rich resource of pre-trained StyleGAN2 weights.

2.4 Pivotal Tuning Inversion

Pivotal Tuning Inversion (PTI) (Roich et al., 2021) is a novel method of GAN Inversion that has attracted much attention in recent years. PTI is a two-stage method where in the first stage images are inverted to \mathcal{W} space, which has high editability but low reconstruction quality, and later the generator weights are tuned to minimize the difference between its output and the input image. It is reported that such a strategy achieves fine results in editability and reconstruction quality. The SOTA methods in encoder-based GAN Inversion are HyperStyle (Alaluf et al., 2021b) and HyperInverter (Dinh et al., 2022), which are methods that employ the PTI strategy.

3 METHOD

3.1 Architecture of Encoder

We modified the pSp encoder (Richardson et al., 2021a) to be able to output StyleMap. The pSp encoder transforms each of the three levels of intermediate feature maps into latent code using Map2Style blocks. We defined the Map2Map block by replacing some of the stride 2 convolution layers of the

Map2Style block with stride 1 convolution layers. The architecture of the proposed method is shown in Figure 2.

Each Map2Map block downsamples the feature map by three stride-2 convolutions. Since the resolution of the feature map is halved with each downsampling, a StyleMap of 2×2 is estimated for the coarse scale and 4×4 for the medium scale. The estimated StyleMap is upsampled to the same size when the StyleGAN feature map is convolved. On the fine scale, a regular Map2Style network is used to save memory. Upon receiving a StyleMap, the pre-trained StyleGAN transforms the feature map using an operation called Spatially Modulated Convolution. Spatially Modulated Convolution is a generalized convolution operation with Weight Demodulation:

$$\text{SpModConv}_w(\mathbf{x}, \mathbf{s}) = \frac{w * (\mathbf{s} \odot \mathbf{x})}{\sqrt{\frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W (w^2 * \mathbf{s}^2)_{i,j}}} \quad (1)$$

where \mathbf{s} is the StyleMap transformed by pointwise convolution using the weights of the affine transformation layer of StyleGAN, \mathbf{x} is the feature map and w is the convolution weight.

These extensions make few fundamental changes to the architecture of the StyleGAN generator, therefore abundant pre-trained models can be used without modification.

3.2 Reconstruction Loss

In our method, the reconstruction loss is the same as the one used in pSp. The loss is composed of L_2 loss, LPIPS loss (Zhang et al., 2018), and ID loss. LPIPS loss is a measure known to be close to human perception. ID loss is defined by $\mathcal{L}_{ID} = 1 - R(\mathbf{x}) \cdot R((G \circ E)(\mathbf{x}))$ for the ArcFace (Deng et al., 2019) network R that outputs face similarity.

3.3 Regularization of Latent Code

Since we found that Inversion to StyleMap improves reconstruction quality but significantly impairs editability, we examined the latent space. We first investigated the singular values of the outputs of the mapping network, inspired by the insights of Zhu et al (Zhu et al., 2020b). The singular values are shown in Figure 3.

Assuming that small singular values contribute little to the properties of the image, we performed dimensionality reduction and had the Map2Map and Map2Style blocks output a normalized latent code of 128 dimensions. The encoder restores the latent code to 512 dimensions. That is, for example, the output \mathbf{v}_i

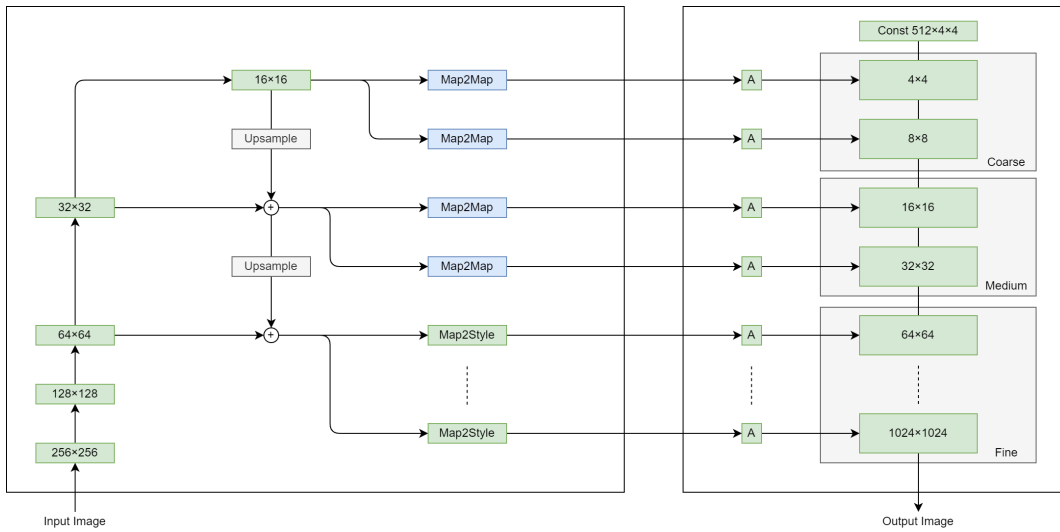


Figure 2: Our architecture. The left frame of the figure shows the encoder and the right frame shows the StyleGAN generator. Similar to the pSp encoder, the feature pyramid in the ResNet backbone converts the input image into a 3-scale feature map. However, the coarse and medium scale feature maps are converted to a StyleMap by the Map2Map block, and the fine scale feature map is converted to a Style by the original Map2Style block.

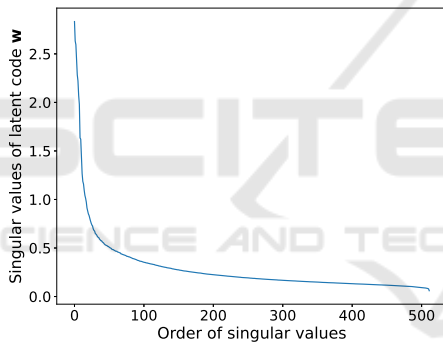


Figure 3: Visualization of the singular values of the output of the mapping network. The 512 singular values are side by side. The large singular values are only a small percentage of the total, and after that they are much smaller.

of Map2Style is transformed as $\mathbf{w}_i = A\mathbf{v}_i + \bar{\mathbf{w}}$. where $A \in \mathbb{R}^{512 \times 128}$ and $\bar{\mathbf{w}}$ is the average of the mapping network output.

Two regularization terms were added to encourage the encoder output to be distributed closer to \mathcal{W} space.

The first is a term to bring each latent code closer to the distribution of the output of the mapping network. The outputs of the Map2Map and Map2Style blocks should be standardized and uncorrelated. Therefore, we used the KL divergence with the standard normal distribution for the output \mathbf{v}_i of each block as the loss function:

$$\mathcal{L}_{\text{KLD}}(\mathbf{v}_i) = \frac{1}{D} \sum_{d=0}^{D-1} \left(\ln(\sigma_{i,d}) + \frac{1 + \mu_{i,d}^2}{2\sigma_{i,d}^2} - \frac{1}{2} \right) \quad (2)$$

where D is the number of dimensions of \mathbf{v}_i , this time 128, and $\mu_{i,d}$ and $\sigma_{i,d}^2$ are the mean and variance for the d th element of \mathbf{v}_i , respectively.

The second is the difference minimization of the latent code at each scale. The output of Map2Map for the coarsest scale is downsampled to 1×1 and the L_2 norm with the other outputs is used for the loss function.

The first constraint implicitly assumes that the \mathcal{W} space is multivariate normal. However, as Zhu et al. (2020b) showed, the mapping network of the StyleGAN is generally closer to the distribution transformed by the activation function than to the multivariate normal distribution. However, we experimentally confirmed that the constraint works well despite this fact.

4 EXPERIMENTS

4.1 Dataset

For GAN Inversion of human face images, The encoder is trained with FFHQ(Karras et al., 2019), a face image dataset. StyleGAN, pre-trained on the same dataset, is used as the generator. CelebA-HQ(Karras et al., 2018), a face image dataset different from FFHQ, is used for evaluation.

4.2 Experimental Results

The inversion results of the existing method and the inversion results with StyleMap without regularization and with regularization are shown in Figure 4. Input images were randomly selected from the CelebA-HQ test set. Although all methods are able to reconstruct the input image reasonably well, the proposed method without regularization and HyperInverter have higher reconstruction quality.

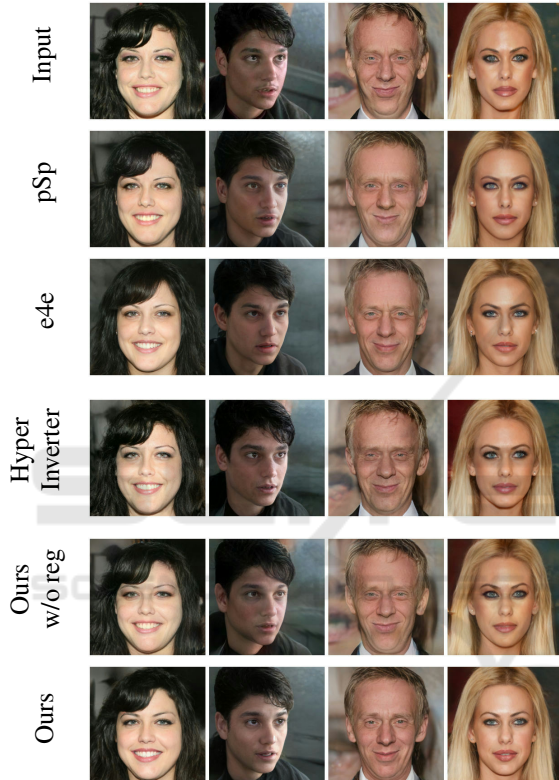


Figure 4: Inversion results of CelebA-HQ for existing and proposed methods.

5 EVALUATION

5.1 Quantitative Evaluation of Reconstruction Quality

The evaluations by LPIPS and MSE of our method as well as pSp and e4e are shown in Table 1. The proposed method is better than the baselines. In particular, the proposed method without regularization even outperforms the SoTA, HyperInverter.

Table 1: Quantitative evaluation of reconstruction quality.

Method	LPIPS↓	MSE↓
pSp	0.16	0.03
e4e	0.20	0.05
HyperInverter	0.11	0.02
Ours w/o reg	0.10	0.02
Ours	0.15	0.03

5.2 Qualitative Evaluation of Editability

5.2.1 Latent Code Editing

The results of image editing by adding Age vectors by InterfaceGAN(Shen et al., 2020) for the existing and proposed methods are shown in Figure 5. HyperInverter, pSp and the proposed method without regularization have low plausibility of the edited image. Especially in the second column, HyperInverter’s editing even changed the gender of the subject.

5.2.2 Toonify

Toonify(Pinkney and Adler, 2020) is a method that performs impressive image transformations by switching layers midway between a pre-trained StyleGAN and a fine-tuned version of it on a different data set. The results of Toonify editing for each method are shown in Figure 6. Although pSp reports better toonification than the default by learning with different settings in an additional report(Richardson et al., 2021b), we used the default encoder for all methods for the sake of fair comparison. In other words, the encoders used in Figure 6 are the same as in Figure 4, respectively. The proposed method with regularization is as plausible as e4e, while pSp and the proposed method without regularization are less visually plausible. Although HyperInverter’s toonification is plausible, its editing effect is insignificant

5.2.3 Local Editing

Local editing of images is possible by editing the StyleMap. Figure 7 shows how local editing was performed by gently interpolating the StyleMap.

5.3 Quantitative Evaluation of Editability

Tov et al. (2021) proposed Latent Editing Consistency (LEC) as a measure of editability in GAN Inversion:

$$LEC(f_{\theta}) = \mathbb{E}_{\mathbf{x}}[\|E(\mathbf{x}) - (f_{\theta}^{-1} \circ E \circ G \circ f_{\theta} \circ E)(\mathbf{x})\|_2] \quad (3)$$

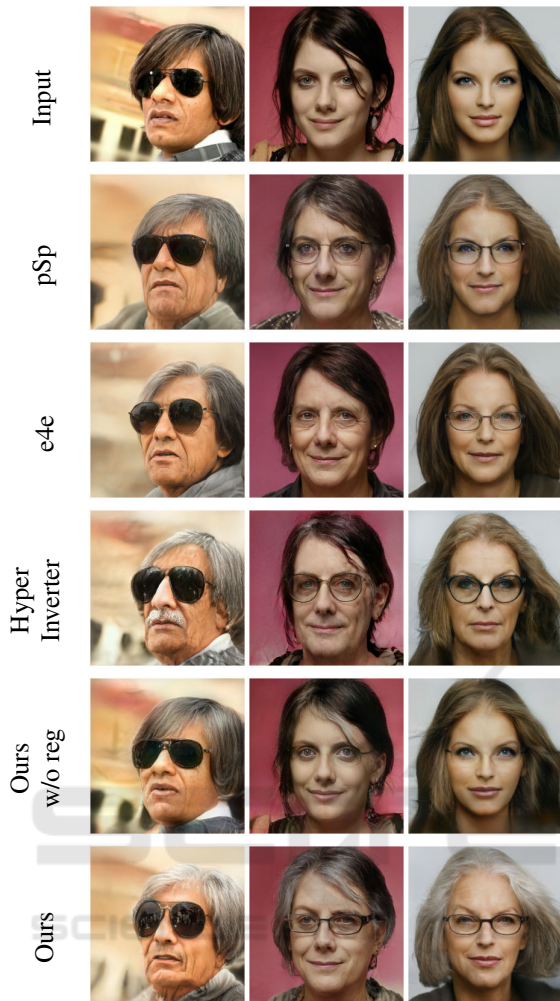


Figure 5: The inversion results of CelebA-HQ for existing and proposed methods.

where E and G are the encoder and StyleGAN generator, respectively, and $f_{\theta}(\mathbf{w}) = \mathbf{w} + \alpha \mathbf{w}_{dir}$. LEC measures the distance between the latent code estimated from an image and the latent code estimated from an image that has been edited and reverse-edited. The Age and Smile vectors produced by InterfaceGAN are used as the edit vector \mathbf{w}_{dir} . We defined avg, min, and max as extensions of LEC to StyleMap. Each is measured by calculating the mean, maximum, and minimum pixel values for the pixel-wise sum of squares of the difference between two StyleMaps. We evaluated LEC with $\alpha = 3, -3$ for the Age vector as Old and Young, respectively, and with $\alpha = 2, -2$ for the Smile vector as Smile and No Smile, respectively. The evaluation results are shown in Table 2.

The proposed method without regularization is, as expected, hardly editable. On the other hand, the proposed method with regularization shows better ed-

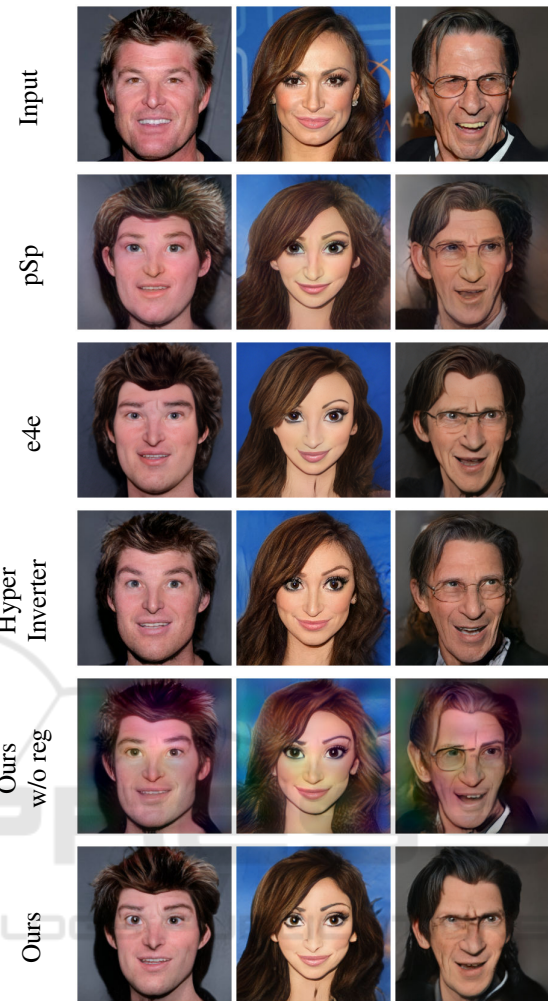


Figure 6: The results of toonification using existing and proposed methods.

Table 2: Editability Evaluation by LEC \downarrow .

Method	Old	Young	Smile	No Smile
pSp	63.03	59.43	48.09	48.15
e4e	24.33	24.59	19.92	20.45
HyperInverter	34.04	34.81	24.99	26.00
Ours w/o reg(avg)	295.07	285.80	264.90	264.88
Ours w/o reg(min)	201.18	193.42	178.17	178.38
Ours w/o reg(max)	425.15	413.76	386.07	384.96
Ours (avg)	32.42	30.57	22.94	23.09
Ours (min)	29.99	27.84	20.95	21.10
Ours (max)	36.12	34.65	26.21	26.34

itability than pSp, even if not as good as e4e. Note that e4e is a method that improves editability at the expense of reconstruction quality from pSp, and our method improves editability without sacrificing the reconstruction quality of pSp. See Figure 8 for the trade-off between reconstruction quality and editability.

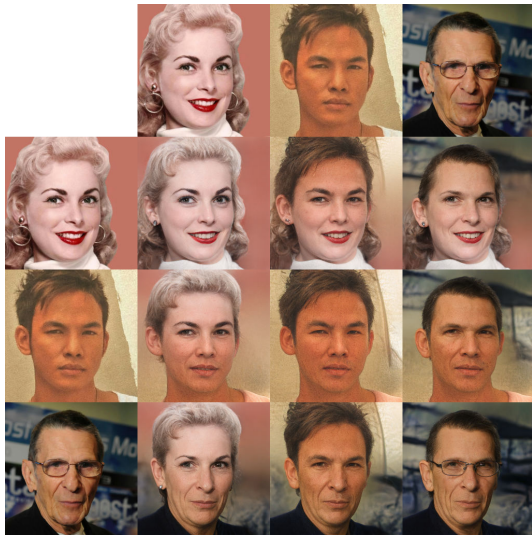


Figure 7: Local editing results. Each edited image is a composite of the top of the image in the first row and the bottom of the image in the first column.

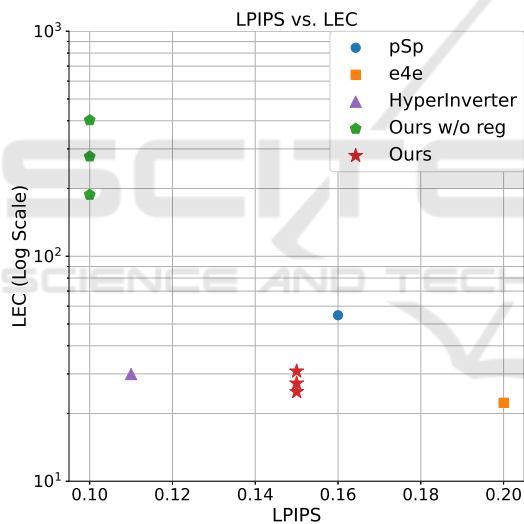


Figure 8: Trade-off between reconstruction quality (LPIPS) and editability (LEC). As editability, the average of four values of LEC is plotted. Avg, min, and max are plotted as the LEC of the proposed method, respectively. The proposed method with regularization, plotted in \star , achieves an excellent trade-off.

6 DISCUSSION

6.1 Input Image Challenging to Reproduce

It is still challenging to reconstruct specific images as well as the existing methods, while the proposed method has qualitatively and quantitatively better re-



Figure 9: Examples of images that are difficult to reconstruct and comparison of actual reconstruction results.

construction quality than the existing methods. Disappointing results can be seen as shown in Figure 9, where arms, clothes, or ornaments cover the face (rows 1-2), or where the face orientation is extreme (rows 3). However, while pSp vaguely describes the garment in the second line, the proposed method reconstructs it better. In the third line, the proposed method can reconstruct the extreme angles of the face in the input image, while the baselines are not able to reproduce it. In addition, the proposed method also tries to reconstruct the background, which the baselines have given up on. Overall, even with challenging inputs, the proposed method is able to reconstruct them as well as HyperInverter.

6.2 Regularization Assuming Normal Distribution

In this study, we designed the architecture assuming that each element of the output of the Map2Map and Map2Style blocks (1) has 0 expected value, (2) has 1 variance, and (3) is uncorrelated. However, we used the KL divergence to the standard normal distribution as a penalty term to encourage its output to satisfy the above conditions. This leads to the following two problems:

- KL divergence to each element does not guarantee uncorrelatedness
- \mathcal{W} space is different from a normal distribution

Despite these problems, the proposed method shows good results. For the former, if each element of the output of the block is correlated, the distribution of the estimated latent code will fall within the distribution, although it will not match the desired distribution. For the latter, although the shape of the distribution is not taken into account, it is considered to be a sufficient constraint in that standardization of each element is encouraged.

7 CONCLUSIONS

In this study, we proposed a GAN Inversion using StyleMap, a spatial extension of the latent code that controls image properties in StyleGAN. We found that a simple extension of existing encoders to StyleMap improves reconstruction quality, but significantly degrades editability. Therefore, we added regularization to improve editability. Even though the use of StyleMap is out of consideration in the design of StyleGAN, we confirmed that our method is comparable to existing methods in image editing. In addition, we showed that StyleMap allows local editing of arbitrary images. Notably, our method is comparable in performance to SOTA methods, even though it employs a strategy independent of PTI. In other words, performance can be improved by incorporating PTI's strategy into our method. In the future, we would like to adopt a PTI strategy and experiment with a wide range of data sets.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP21H03496, JP22K12157.

REFERENCES

- Abdal, R., Qin, Y., and Wonka, P. (2019). Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*.
- Alaluf, Y., Patashnik, O., and Cohen-Or, D. (2021a). Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Alaluf, Y., Tov, O., Mokady, R., Gal, R., and Bermano, A. H. (2021b). Hyperstyle: Stylegan inversion with hypernetworks for real image editing.
- Deng, J., Guo, J., Niannan, X., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *CVPR*.
- Dinh, T. M., Tran, A. T., Nguyen, R., and Hua, B.-S. (2022). Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc.
- Hong, S., Arjovsky, M., Barnhart, D., and Thompson, I. (2020). Low distortion block-resampling with spatially stochastic networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4441–4452. Curran Associates, Inc.
- Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *CVPR*.
- Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*.
- Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*.
- Kim, H., Choi, Y., Kim, J., Yoo, S., and Uh, Y. (2021). Exploiting spatial dimensions of latent in gan for real-time image editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Pinkney, J. and Adler, D. (2020). Resolution dependant gan interpolation for controllable image synthesis between domains.
- Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., and Cohen-Or, D. (2021a). Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., and Cohen-Or, D. (2021b). Encoding in style: a stylegan encoder for image-to-image translation. <https://github.com/eladrich/pixel2style2pixel#additional-applications>.
- Roich, D., Mokady, R., Bermano, A. H., and Cohen-Or, D. (2021). Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*
- Shen, Y., Gu, J., Tang, X., and Zhou, B. (2020). Interpreting the latent space of gans for semantic face editing. In *CVPR*.
- Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., and Cohen-Or, D. (2021). Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.
- Zhu, J., Shen, Y., Zhao, D., and Zhou, B. (2020a). In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- Zhu, P., Abdal, R., Qin, Y., Femiani, J., and Wonka, P. (2020b). Improved stylegan embedding: Where are the good latents?