Deep Learning and Generative Models

Paper assignment:

In-Domain GAN Inversion for Real Image Editing

Summary

- Paper's introduction
- GAN inversion problem
- Current GAN inversion state of art
- *In domain* GAN inversion
- Optimization phase
- Real image editing
- Results
- References

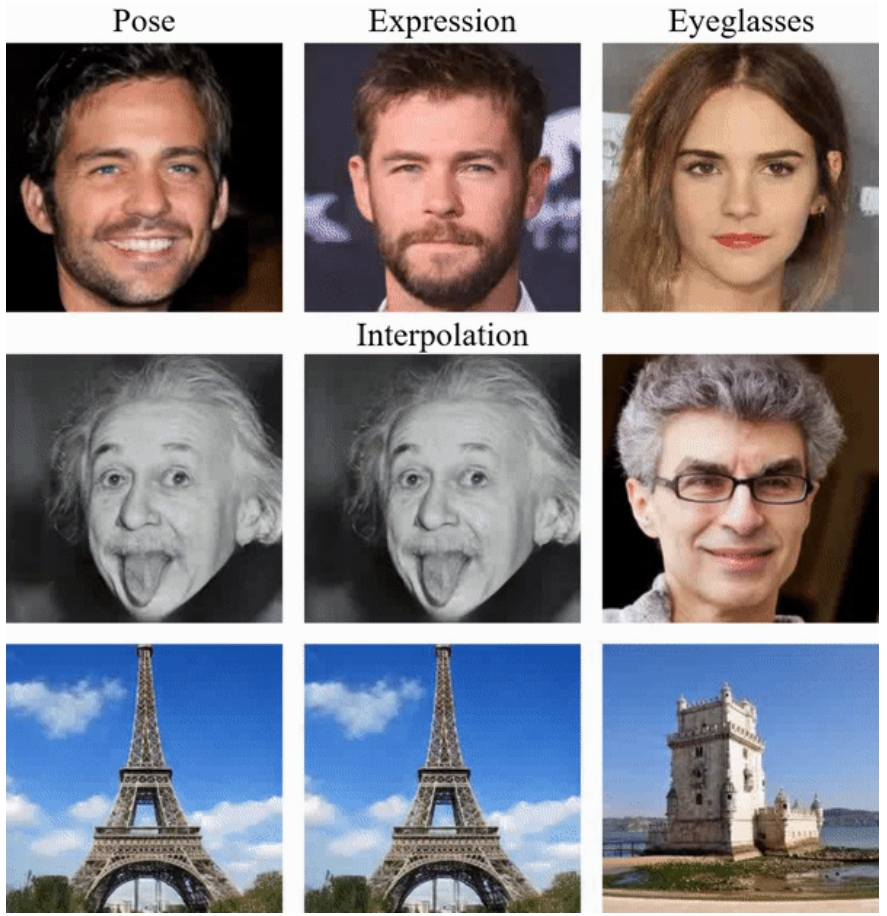


Fig. 1. Illustration of the In-Domain GAN inversion [12]

Introduction

- A variety of semantics emerge in the latent space of GANs
- Main idea: use these details in order to edit real images by changing some semantic details (like changing attributes in human faces) in the latent code of the image

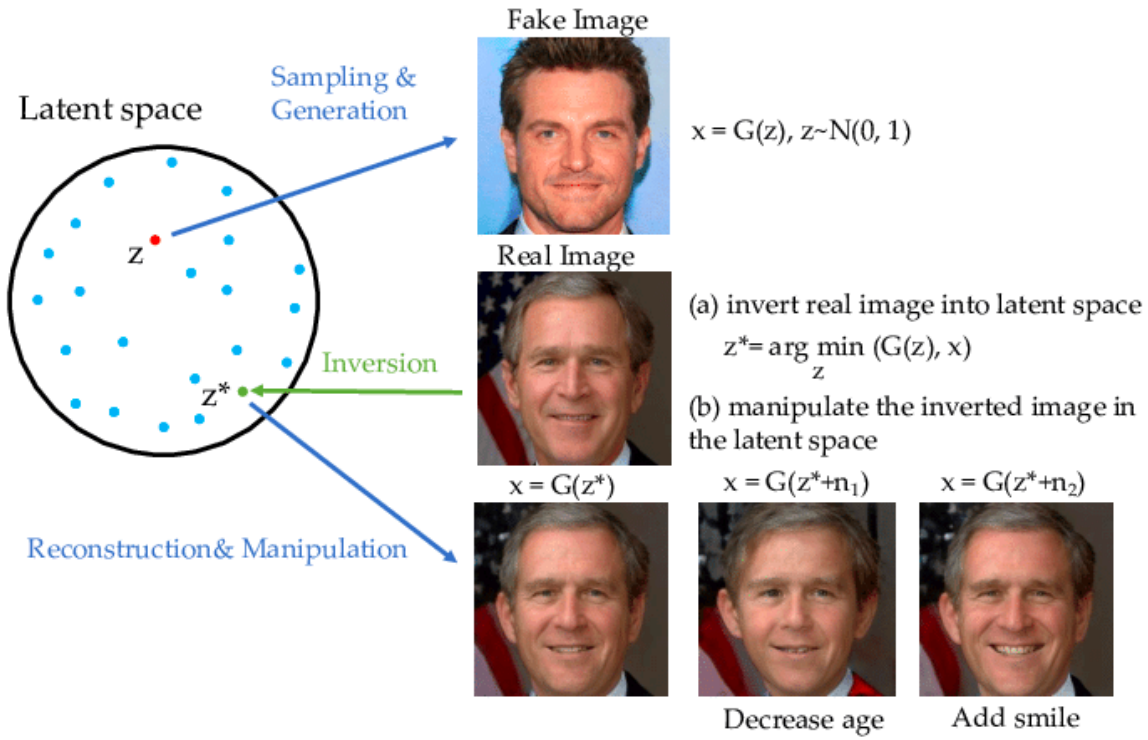


Fig. 2. Illustration of GAN inversion from Xia et al. [1]

GAN inversion problem

- Existing inversion methods typically focus on reconstructing the target image by pixel values
- During these inversions we loose all of the semantic informations
- Editing images by using the inverted code produced by these inversions is almost impossible due to lack of semantic details inside the inverted code
- A good GAN inversion method should not only reconstruct the target image at the pixel level, but also align the inverted code with the semantic knowledge encoded in the latent space

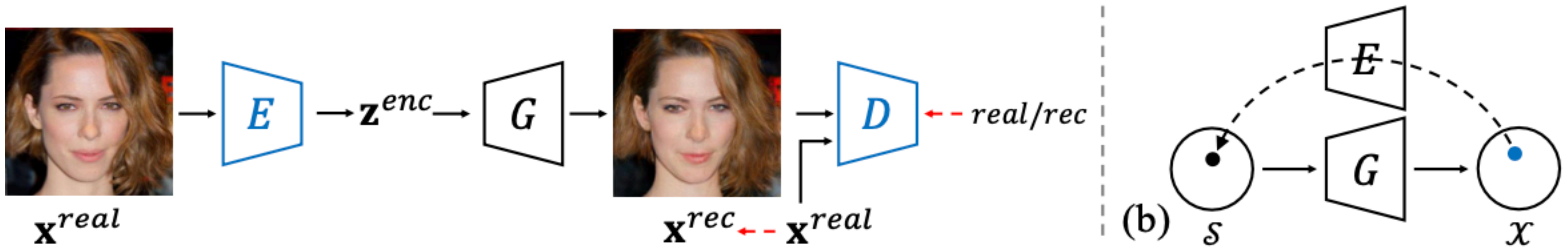
GAN inversion goals

- Does the inverted code lie in the original latent space of GANs?
- Can the inverted code semantically represent the target image?
- Does the inverted code support image editing by reusing the knowledge learned by GANs?

Solution to GAN inversion problem

- Zhu et al. [12] proposes an *in-domain* GAN inversion approach to recover the input image at both the pixel level and the semantic level
- First, training a novel domain-guided encoder is required in order to map the image space to the latent space such that all codes produced by the encoder are in-domain
- Then, performing instance-level domain-regularized optimization by involving the encoder as a regularizer to better reconstruct the pixel values without affecting the semantic property of the inverted code is used to produce better results

Solution to GAN inversion problem

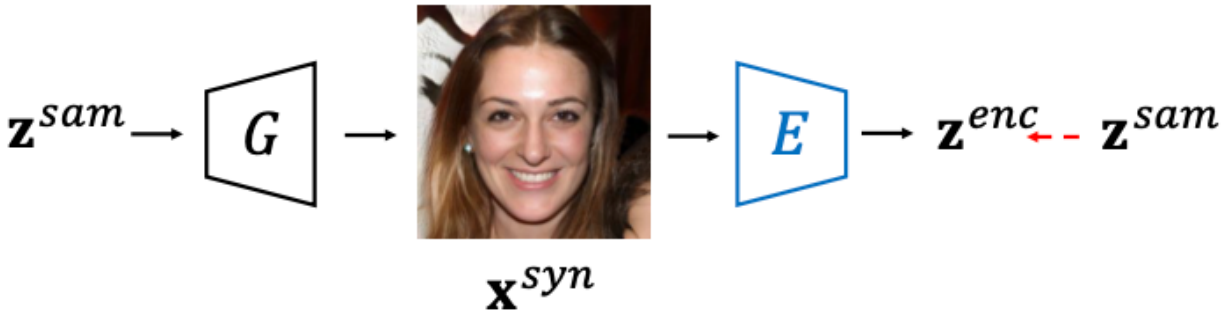


GAN inversion state of art

- Existing inversion approaches typically fall into two types:
- One is learning-based, which first synthesizes a collection of images with randomly sampled latent codes and then uses the images and codes as inputs and supervisions respectively to train a deterministic model
- The other is optimization-based, which deals with a single instance at one time by directly optimizing the latent code to minimize the pixel-wise reconstruction loss

Learning-based models

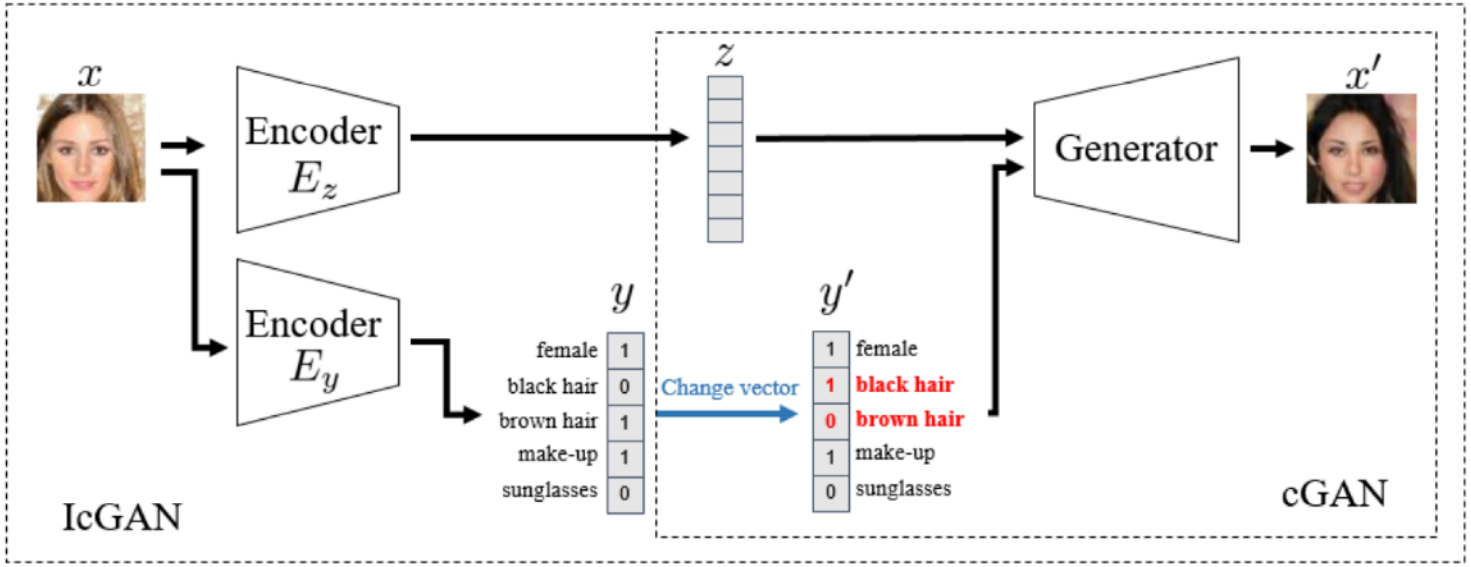
- A learning-based inversion method aims to learn an encoder network to map an image into the latent space such that the reconstructed latent code is as similar to the sampled one



$$\min_{\Theta_E} \mathcal{L}_E = ||\mathbf{z}^{sam} - E(G(\mathbf{z}^{sam}))||_2$$

Encoders state of art

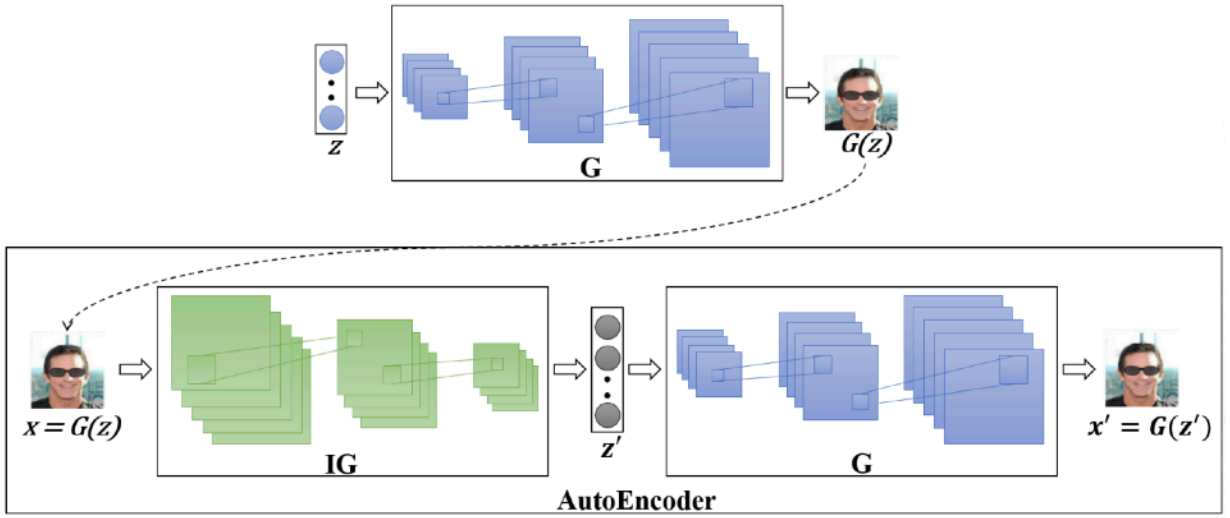
- Perarnau et al. [2] (Invertible conditional GAN) encode a real image x into a latent representation z and attribute information y, and then apply variations on it to generate a new modified



$$\mathcal{L}_{ez} = \mathbb{E}_{z\sim p_z, y'\sim p_y} \|z - E_z(G(z, y'))\|_2^2$$
$$\mathcal{L}_{ey} = \mathbb{E}_{x,y\sim p_{data}} \|y - E_y(x)\|_2^2$$

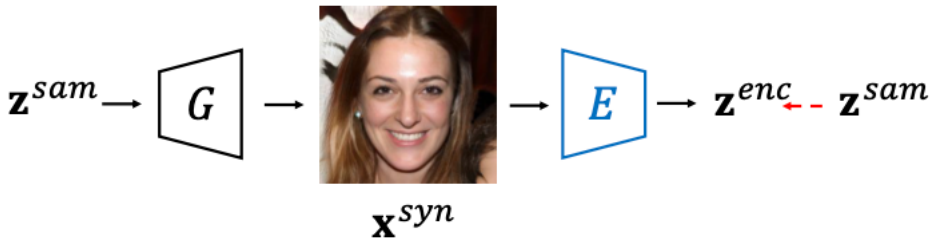
Encoders state of art

- Luo et al. [3] idea of AEGAN is inspired from AutoEncoder
- The innovation of AEGAN is that they focus on the reconstructed images instead of the reconstructed noise vectors. This model does not directly minimize the difference between z and z' , but try to minimize the difference between the x and x'.



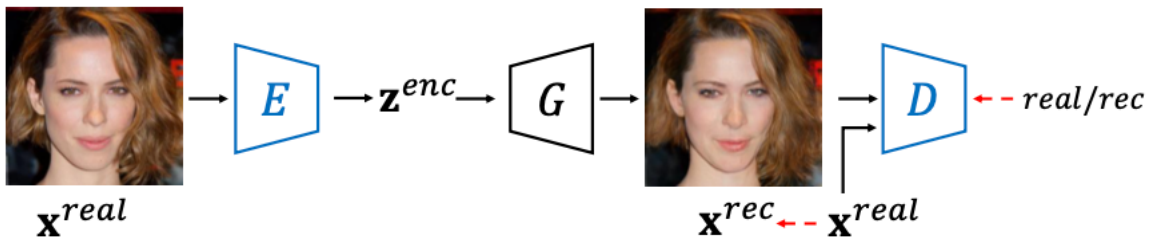
Domain-Guided Encoder

- We argue that the supervision by only reconstructing z^{sam} is not powerful enough to train an accurate encoder
- Also, the generator is actually omitted and cannot provide its domain knowledge to guide the training of encoder



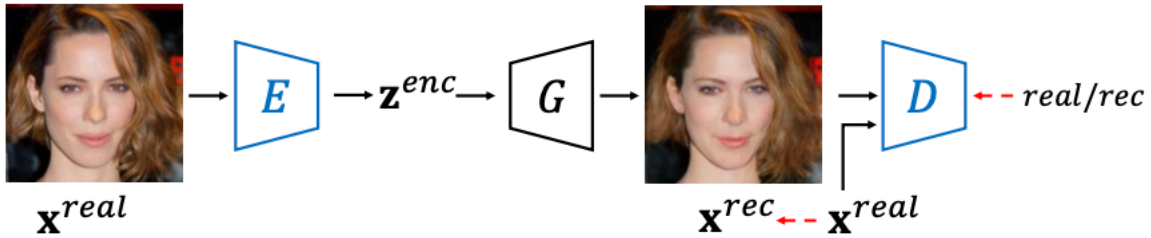
Domain-Guided Encoder

- The output of the encoder is fed into the generator to reconstruct the input image such that the objective function comes from the image space instead of latent space
- This involves semantic knowledge from the generator in training and provides more informative and accurate supervision. The output code is therefore guaranteed to align with the semantic domain of the generator



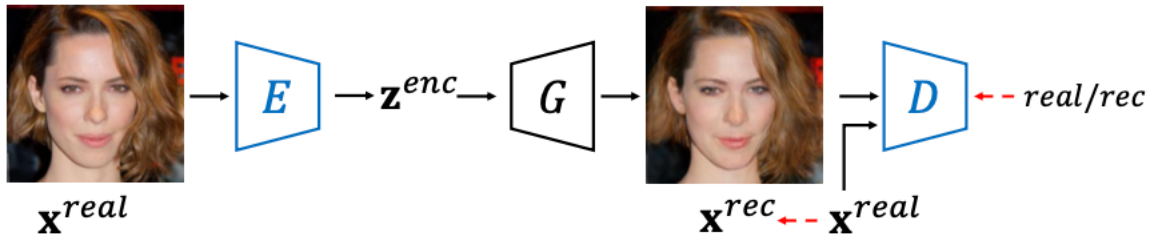
Domain-Guided Encoder

- Instead of being trained with synthesized images, the domain-guided encoder is trained with real images, making our encoder more applicable to real applications.



Domain-Guided Encoder

- To make sure the reconstructed image is realistic enough, Zhu et al. [12] employ the discriminator to compete with the encoder. In this way, as much information as possible can be acquired from the GAN model (i.e., both two components of GAN are used).
- The adversarial training manner also pushes the output code to better fit the semantic knowledge of the generator



Domain-Guided Encoder

- Here's the training process where P_{data} denotes the distribution of real data and γ is the hyper-parameter for the gradient regularization. λ_{VGG} and λ_{adv} are the perceptual and discriminator loss weights. F(·) denotes the VGG feature extraction model

$$\min_{\Theta_E} \mathcal{L}_E = ||\mathbf{x}^{real} - G(E(\mathbf{x}^{real}))||_2 + \lambda_{vgg}||F(\mathbf{x}^{real}) - F(G(E(\mathbf{x}^{real})))||_2 - \lambda_{adv} \mathop{\mathbb{E}}_{\mathbf{x}^{real} \sim P_{data}} [D(G(E(\mathbf{x}^{real})))],$$

$$\min_{\Theta_D} \mathcal{L}_D = \mathop{\mathbb{E}}_{\mathbf{x}^{real} \sim P_{data}} [D(G(E(\mathbf{x}^{real})))] - \mathop{\mathbb{E}}_{\mathbf{x}^{real} \sim P_{data}} [D(\mathbf{x}^{real})] + \frac{\gamma}{2} \mathop{\mathbb{E}}_{\mathbf{x}^{real} \sim P_{data}} [||\nabla_{\mathbf{x}} D(\mathbf{x}^{real})||_2^2],$$

Optimization-based algorithm

- An optimization-based inversion approach directly solves the objective function through back-propagation to find a latent code that minimizes pixel-wise reconstruction-loss

$$z^* = argmin_z ||x - G(z)||_2$$

$$z^* \leftarrow z^* - \eta \nabla_{z^*} ||x - G(z^*)||_2$$

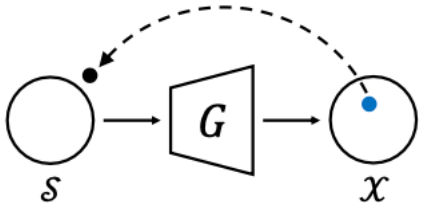
- All of the semantic knowledge is loose and the code also can be outside of the latent space

Optimization-based algorithm

- Various approaches perform optimization using gradient descent [4]. The optimization problem is highly non-convex and requires a good initialization, else risks being stuck in local minima
- It is important to note that the optimization procedure is performed during inference and can be computationally expensive, requiring many passes through the generator each time a new image is to be mapped to its latent vector.

Domain-Regularized Optimization

- It is hard to learn a perfect reverse mapping with an encoder alone due to its limited representation capability
- Even though the inverted code from the proposed domain-guided encoder can well reconstruct the input image based on the pre-trained generator and ensure the code itself to be semantically meaningful, we still need to refine the code to make it better fit the target individual image at the pixel values



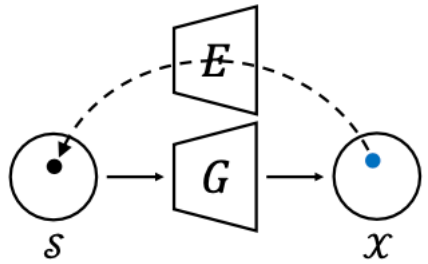
Domain-Regularized Optimization

- Previous methods propose to gradient descent algorithm to optimize the code
- It is critical to choose the optimizer since a good optimizer helps alleviate the local minima problem
- Another important issue for optimization-based GAN inversion is the initialization of latent code. Since is an highly nonconvex problem, the reconstruction quality strongly relies on a good initialization of z

$$z* = argmin_z ||x - G(z)||_2$$

Domain-Regularized Optimization

- The solution is using the output of the domain-guided encoder as an ideal starting point which avoids the code from getting stuck at a local minimum and also significantly shortens the optimization process
- Also, including the domain-guided encoder as a regularizer to preserve the latent code within the semantic domain of the generator.



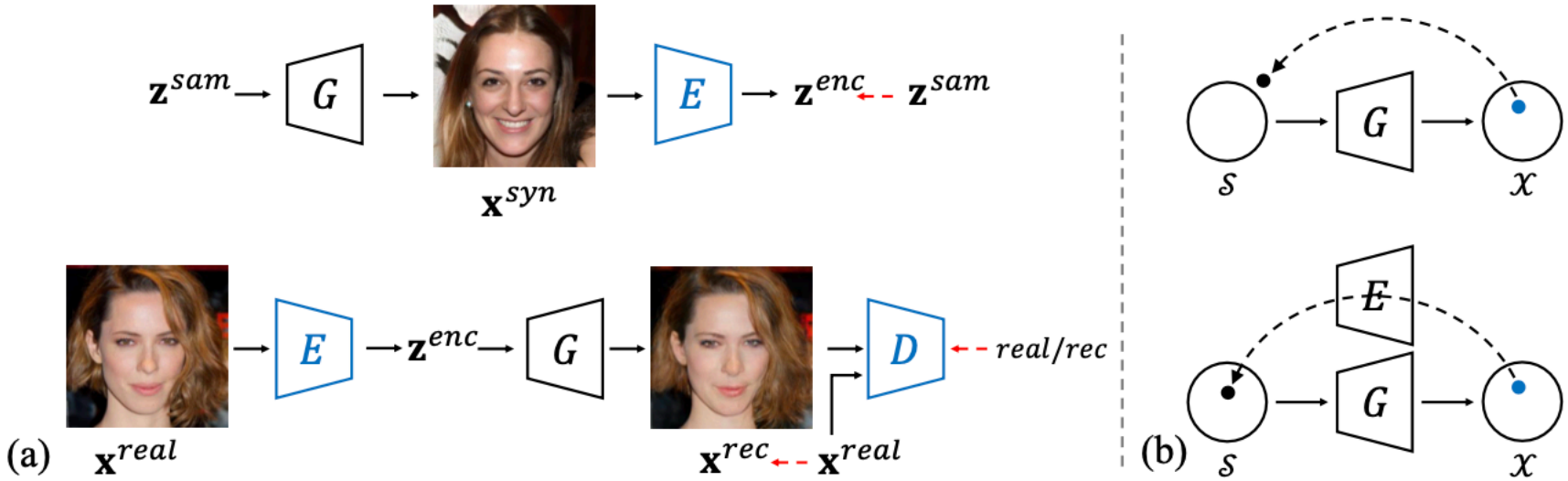
Domain-Regularized Optimization

- The objective function for the optimization phase will be:

$$\mathbf{z}^{inv} = \arg\min_{\mathbf{z}} \; ||\mathbf{x} - G(\mathbf{z})||_2 \; + \lambda_{vgg}||F(\mathbf{x}) - F(G(\mathbf{z}))||_2 + \lambda_{dom}||\mathbf{z} - E(G(\mathbf{z}))||_2,$$

Final structure

- Conventional Encoder vs Domain-Guided Encoder
- Conventional Optimization vs Domain-Regularized optimization

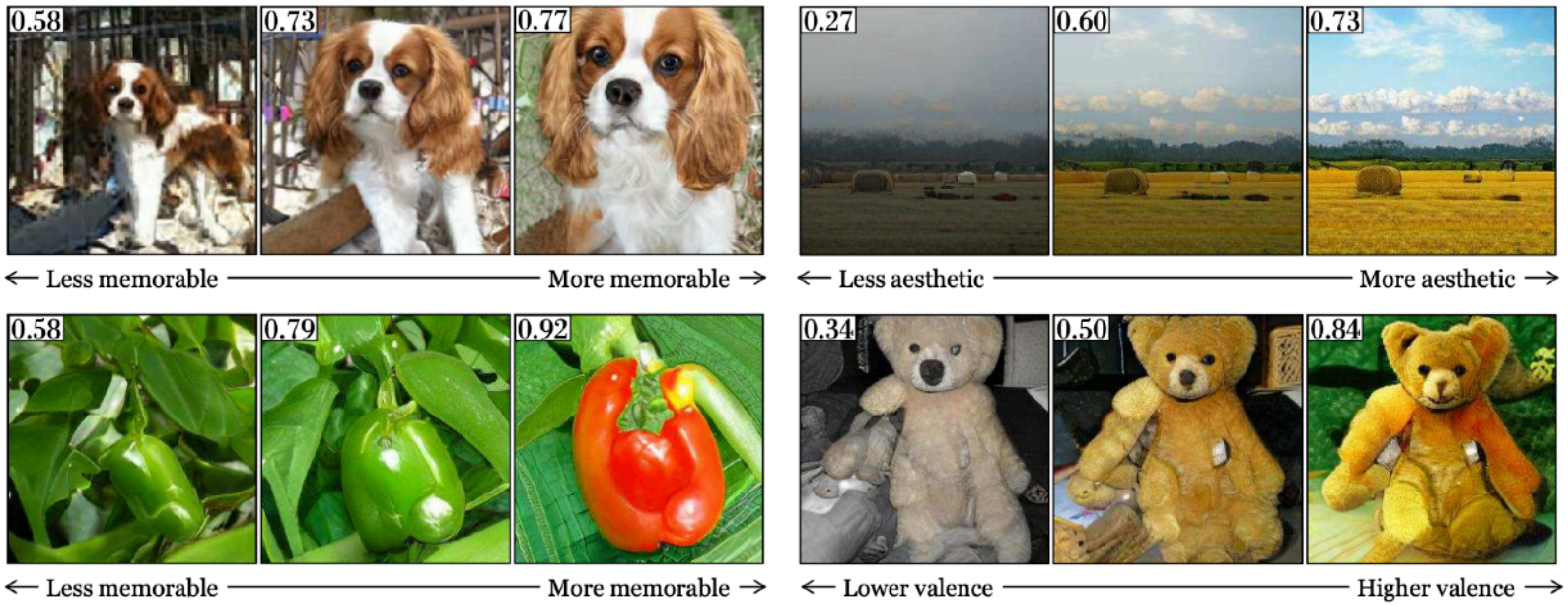


Semantics inside the latent space

- GAN inversion is not the end goal
- A lot of studies [5,6,7] show that varying the latent code inside the latent space can modify the attributes of the generated image
- So, the reason that we invert a real image into the latent space of a trained GAN model is that it allows us to manipulate the image by varying the inverted code in the latent space for a certain attribute

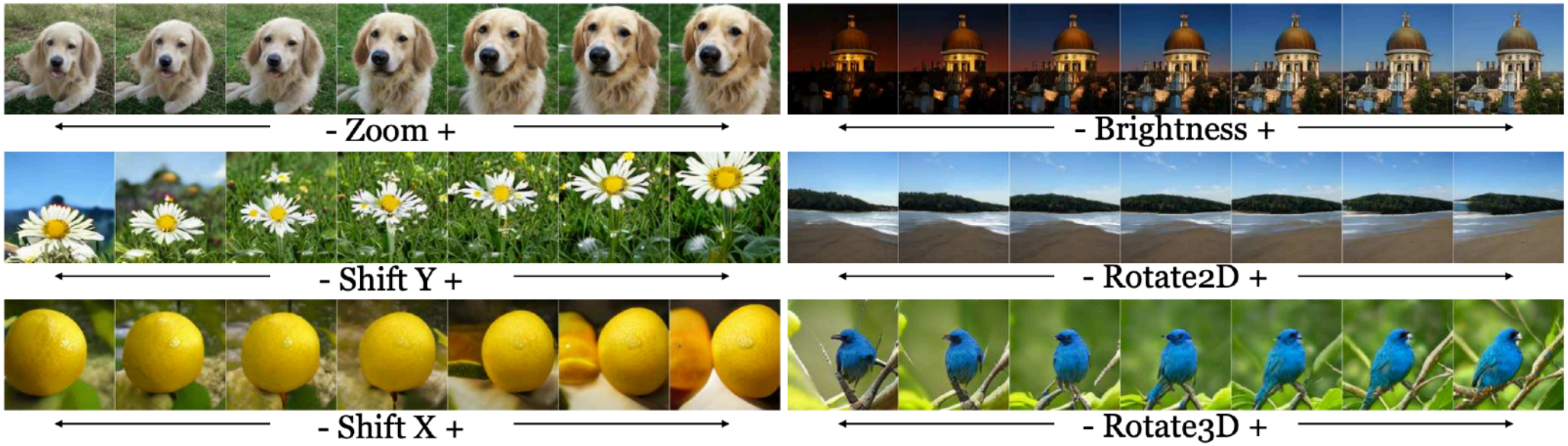
Semantics inside the latent space

- Goetschalckx et al. [5] showed how to make the synthesis from GANs more memorable



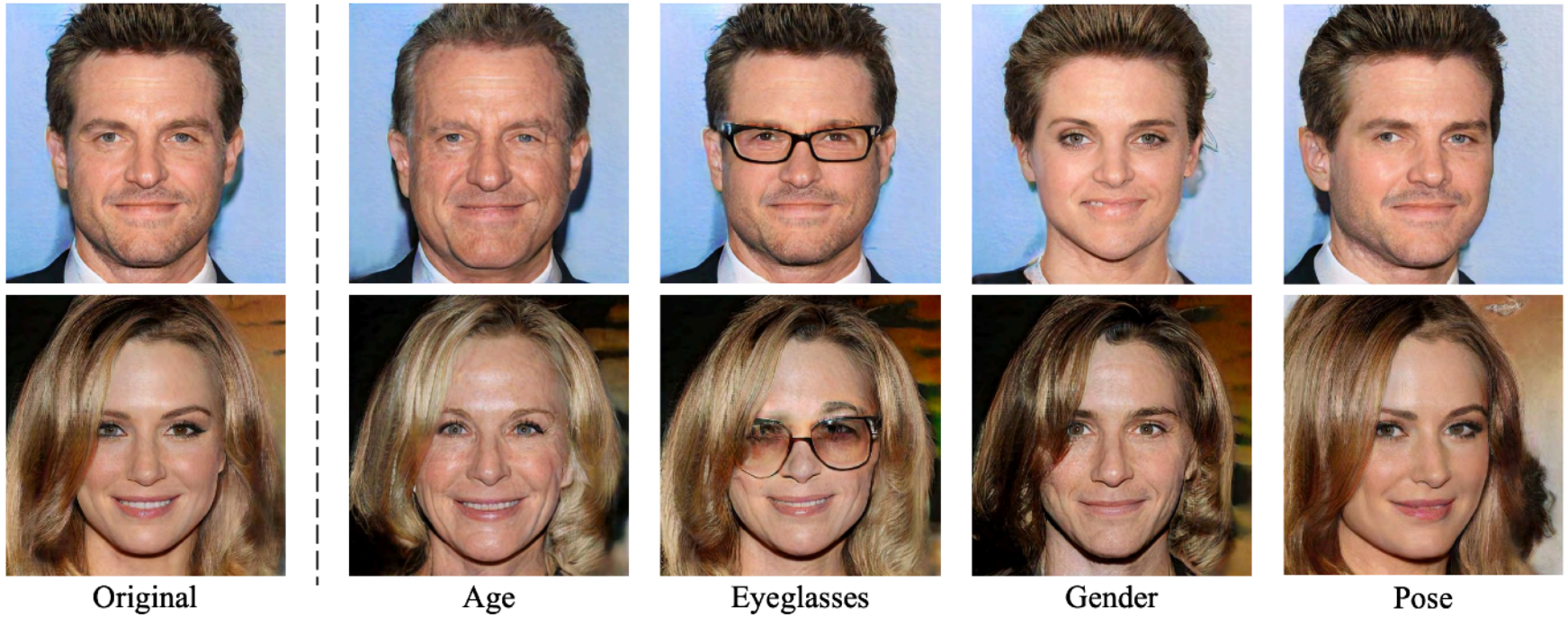
Semantics inside the latent space

- Jahanian et al. [6] achieved camera movements and color changes by shifting the latent distribution



Semantics inside the latent space

- Shen et al. [7] interpreted the latent space of GANs for semantic face editing



Choice of latent space

- StyleGAN [8] model proposes to first map the initial latent space Z to a second latent space W with Multi-Layer Perceptron (MLP), and then feed the codes $w \in W$ to the generator for image synthesis
- Such additional mapping has already been proven to learn more disentangled semantics
- We focus on the semantic (i.e., in-domain) property of the inverted codes, making W space more appropriate for analysis
- Inverting to W space achieves better performance than Z space

Discovering Interpretable Directions

- To modifying latent code z we need to choose a direction n and then move on that direction with α step

$$z = z + \alpha n$$

- We obviously want to choose a direction that will modify a selected semantic detail (e.g. we want to convert a male face into a female)
- Such directions can be identified inside the latent space through supervised, unsupervised, or self-supervised manners

Moving inside the latent space

- For binary attributes (e.g. male or female faces) there exists a hyperplane in the latent space serving as the separation boundary [7]
- Semantic remains the same when the latent code walks within the same side of the hyperplane yet turns into the opposite when across the boundary
- We can define the distance from z to this hyperplane $d(n, z) = n^T z$
- When z lies near the boundary and is moved toward and across the hyperplane, both the distance and the semantic *vary* accordingly. And it is just at the time when the distance changes its numerical sign that the semantic attribute reverses.

Moving inside the latent space

- When there is more than one attribute, editing one may affect another since some semantics can be coupled with each other.
- Shen et al. finds a projected direction, such that moving samples along this new direction can change “attribute 1” without affecting “attribute 2”.

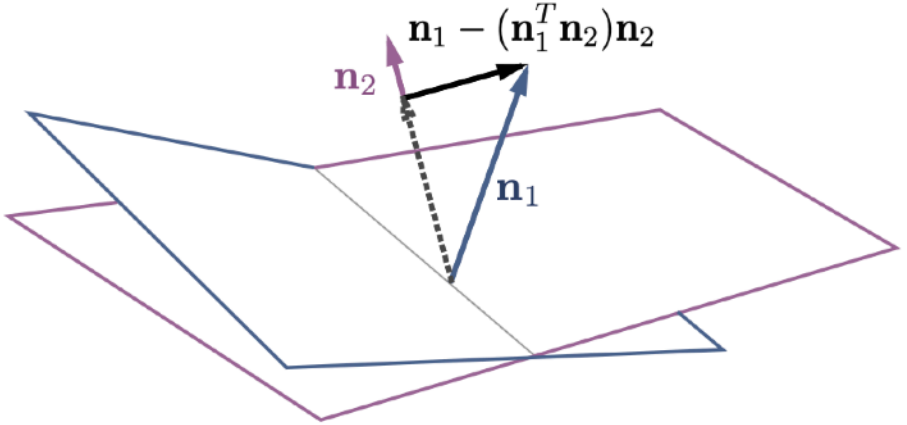
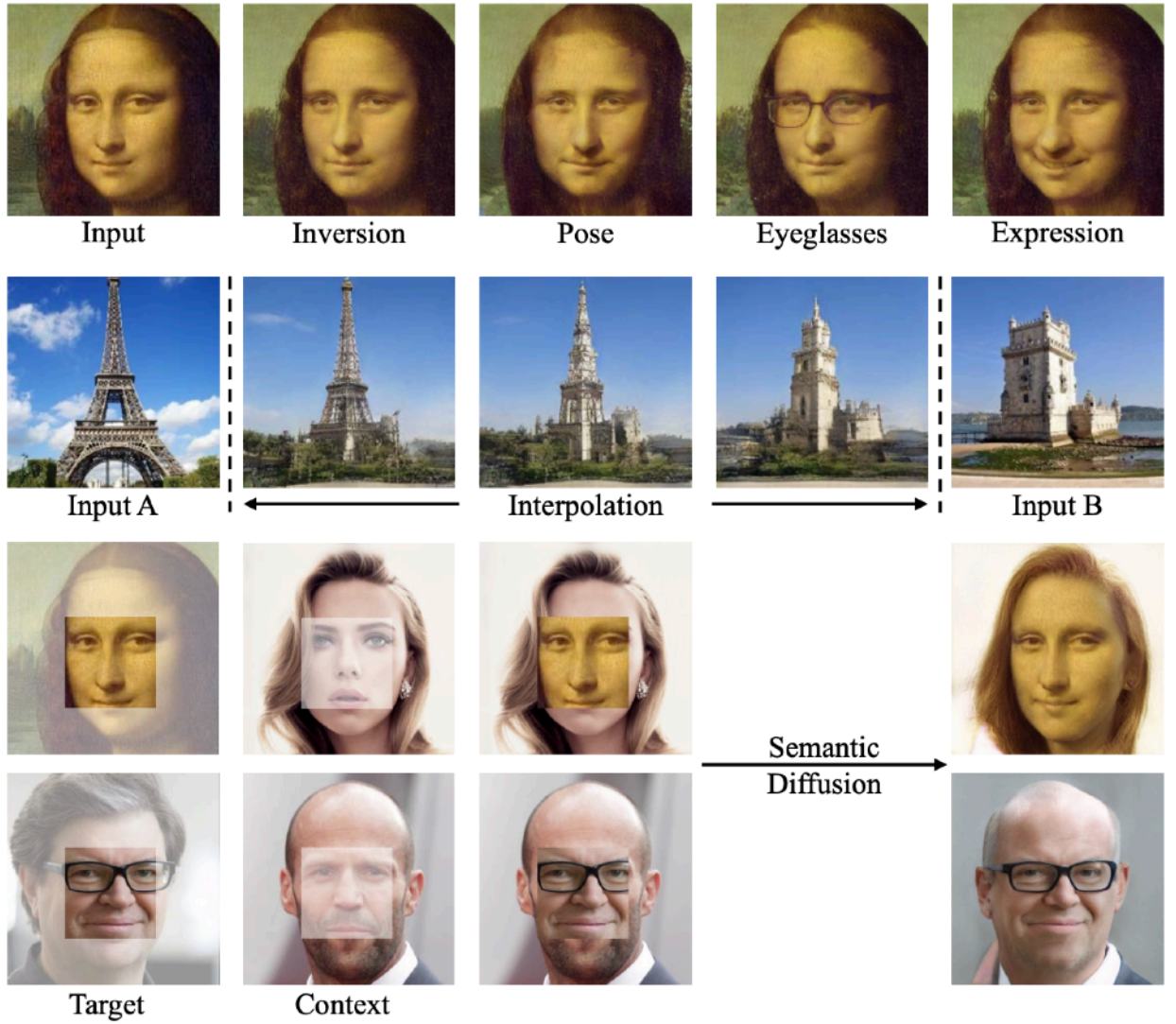


Fig. 3. Illustration of discovering disentangled directions for multiple attributes [7]

Results


Input
Inversion
Pose
Eyeglasses
Expression
Input A
Interpolation
Input B
Semantic
Diffusion
Target
Context

Experimental settings

- Experiments was conducted on FFHQ dataset [8], which contains 70,000 high-quality face images, and LSUN dataset [9], which consists of images from 10 different scene categories
- The GANs to invert are pre-trained following StyleGAN
- As for the perceptual loss, we take $conv4\_3$ 3 as the VGG output. Loss weights are set as $\lambda_{VGG} = 5e^{-5}$, $\lambda_{adv} = 0.1$, and $\gamma = 10\ \lambda_{dom} = 10$
- We set $\lambda_{dom} = 2$. in for the domain-regularized optimization

Image2StyleGAN

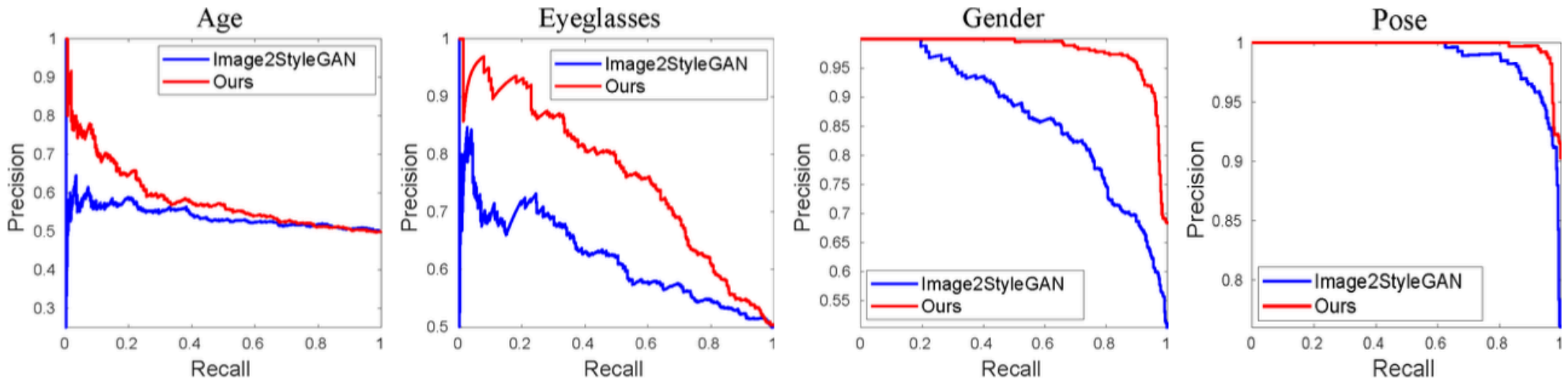
- **Input:** An image $I \in R^{n \times m \times 3}$ to embed; a pre-trained generator G(.)
- **Output:** The embedded latent code w^* and the embedded image $G(w^*)$ optimized via F'.

1. Initialize latent code $w^* = w$;
2. **While** not converged **do**
3. $L \leftarrow L_{percept} G(w^*, I) + \frac{\lambda}{N} ||G(w^*) - I||_2^2$;
4. $w^* \leftarrow w^* - \eta F'(\nabla_{w^*} L)$;
5 **end**

Semantic Analysis of the Inverted Codes

- 7k real face images and a off-the-shelf attribute classifiers to predict age (young v.s. old), gender (female v.s. male), eyeglasses (absence v.s. presence), and pose (left v.s. right)
- State-of-the-art GAN inversion method, Image2StyleGAN [11], and the proposed in-domain GAN inversion to invert these images back to the latent space of a fixed StyleGAN model trained on FFHQ dataset
- InterFaceGAN [7] is used to search the semantic boundaries for the aforementioned attributes in the latent space. Then, these boundaries as well as the inverted codes are used to evaluate the attribute classification performance.

Semantic Analysis of the inverted codes

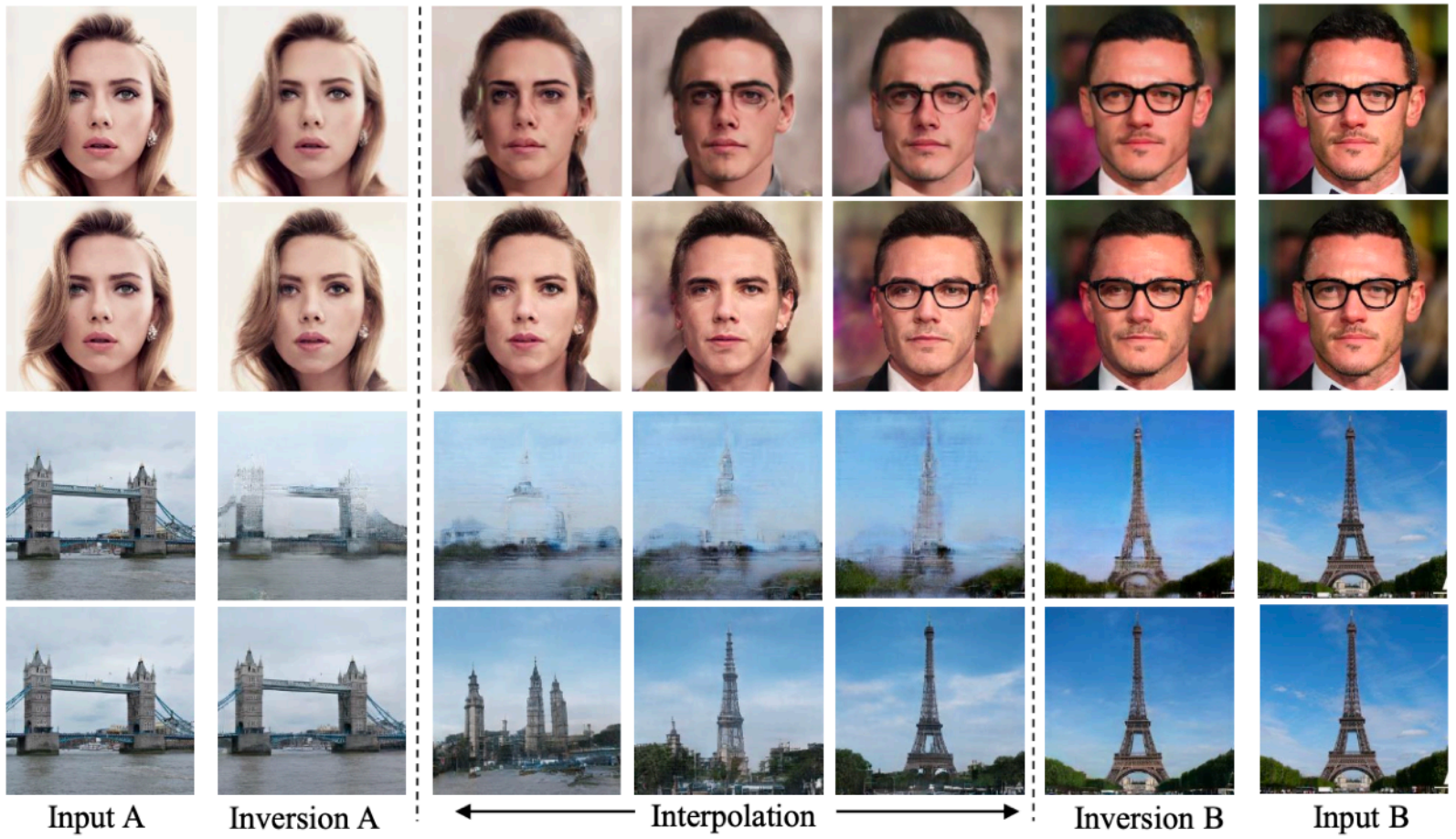


Inversion Quality and Speed

		Face			Tower		
Method	**Speed**	**FID↓**	**SWD↓**	**MSE↓**	**FID↓**	**SWD↓**	**MSE↓**
Traditional Encoder [36]	**0.008s**	88.48	100.5	0.507	73.02	69.19	0.455
MSE-based Optimization [29]	290s	58.04	29.19	**0.026**	69.16	55.35	0.068
Domain-Guided Encoder (Ours)	0.017s	52.85	**13.02**	0.062	46.81	27.13	0.071
In-Domain Inversion (Ours)	8s	**42.64**	13.44	0.030	**44.77**	**26.44**	**0.052**

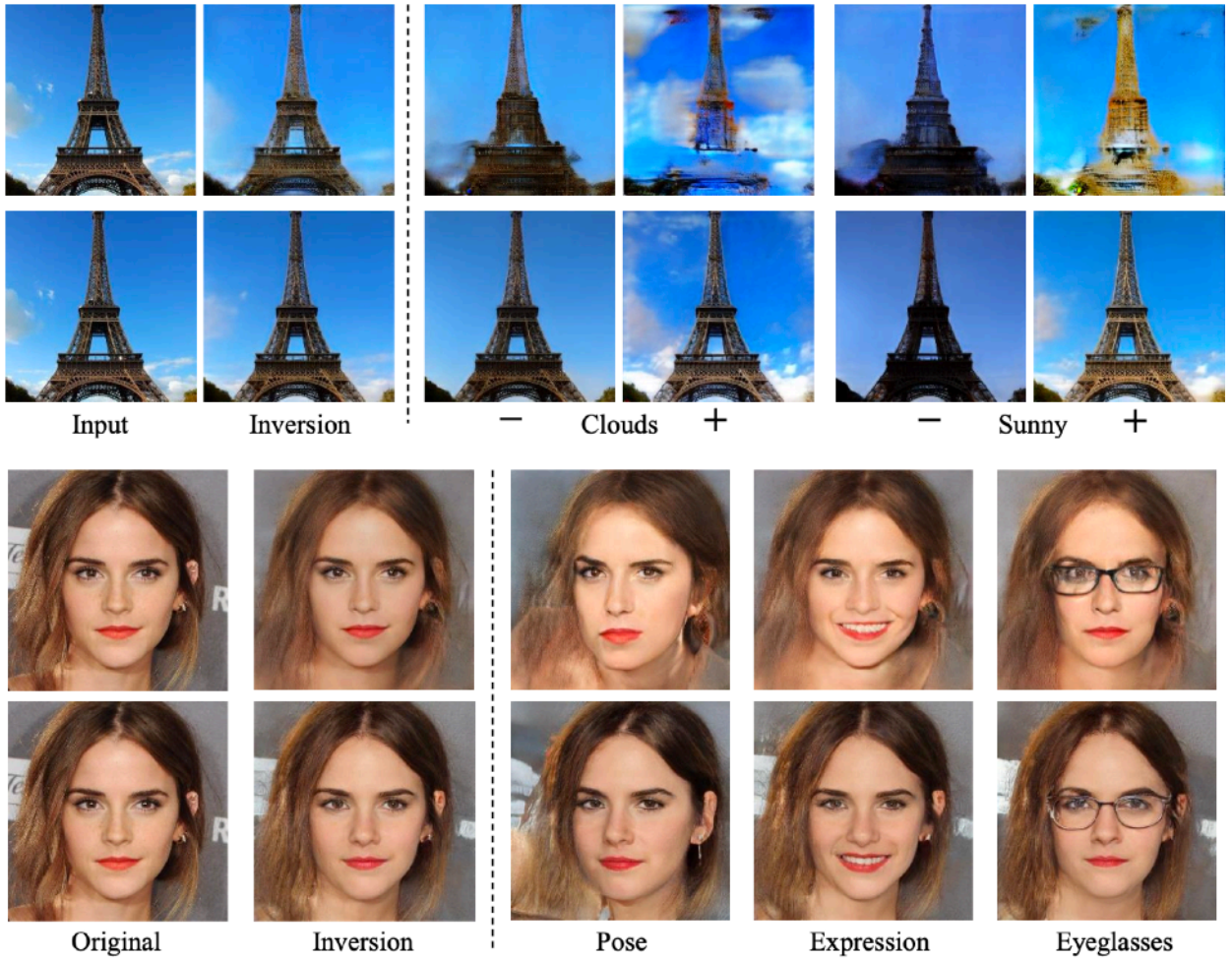
The domain-guided encoder can produce much better reconstruction results compared to the traditional encoder with comparable inference time. It also provides a better initialization for further the domain-regularized optimization, leading to a significantly faster speed (~35X faster) than the state-of-the-art optimization-based method

Image Interpolation



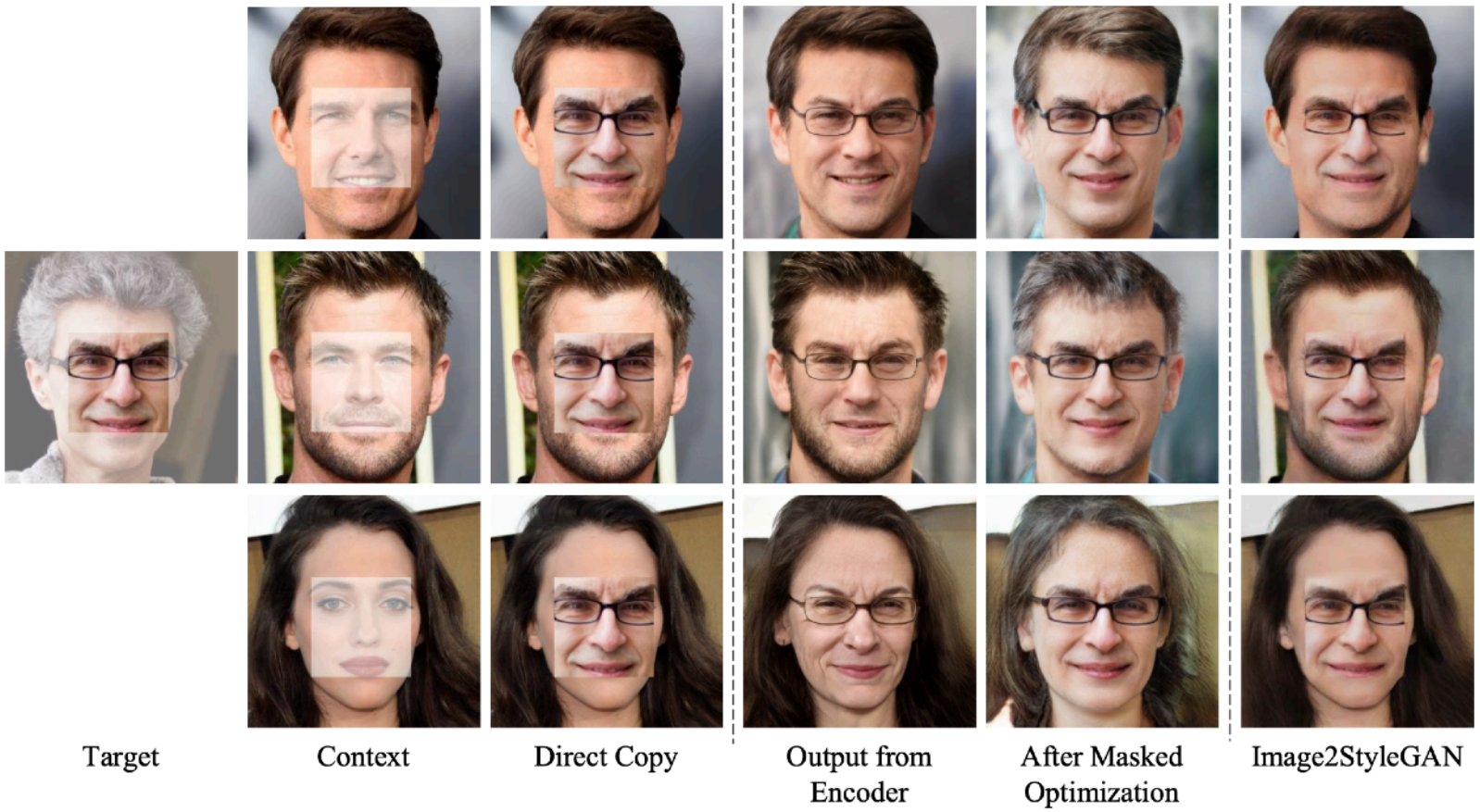
Interpolation				Manipulation			
Face		Tower		Face		Tower	
FID↓	SWD↓	FID↓	SWD↓	FID↓	SWD↓	FID↓	SWD↓
112.09	38.20	121.38	67.75	83.69	28.48	113	52.91
91.18	**33.91**	**57.22**	**28.24**	**76.43**	**17.99**	**57.92**	**31.50**

Semantic Manipulation

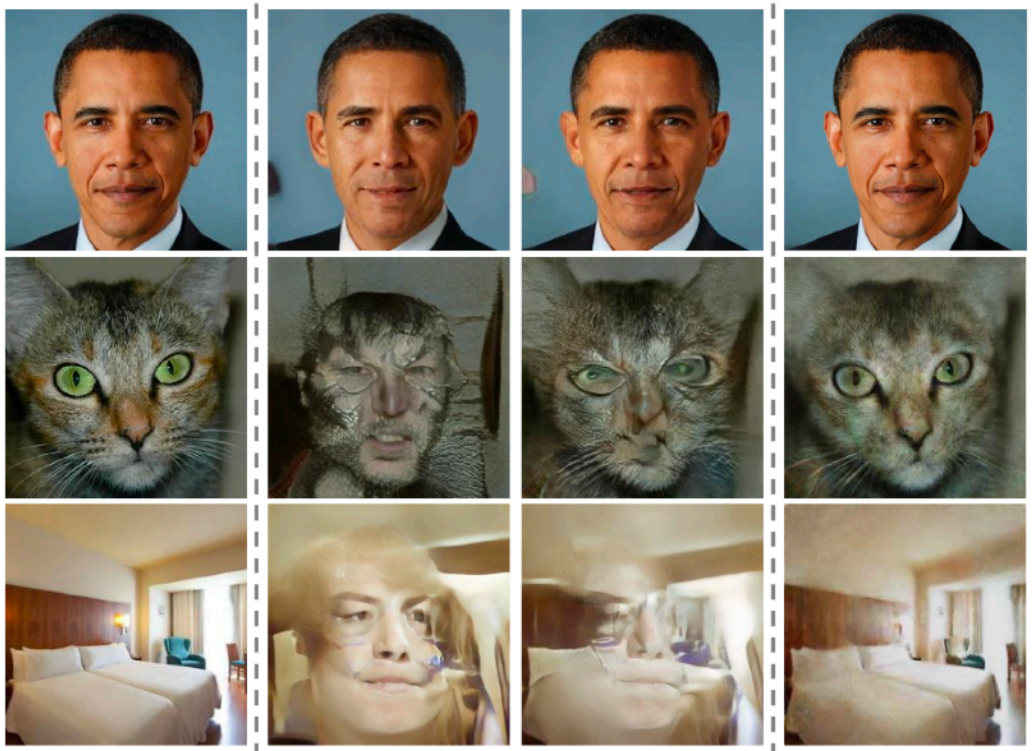


Semantic Diffusion

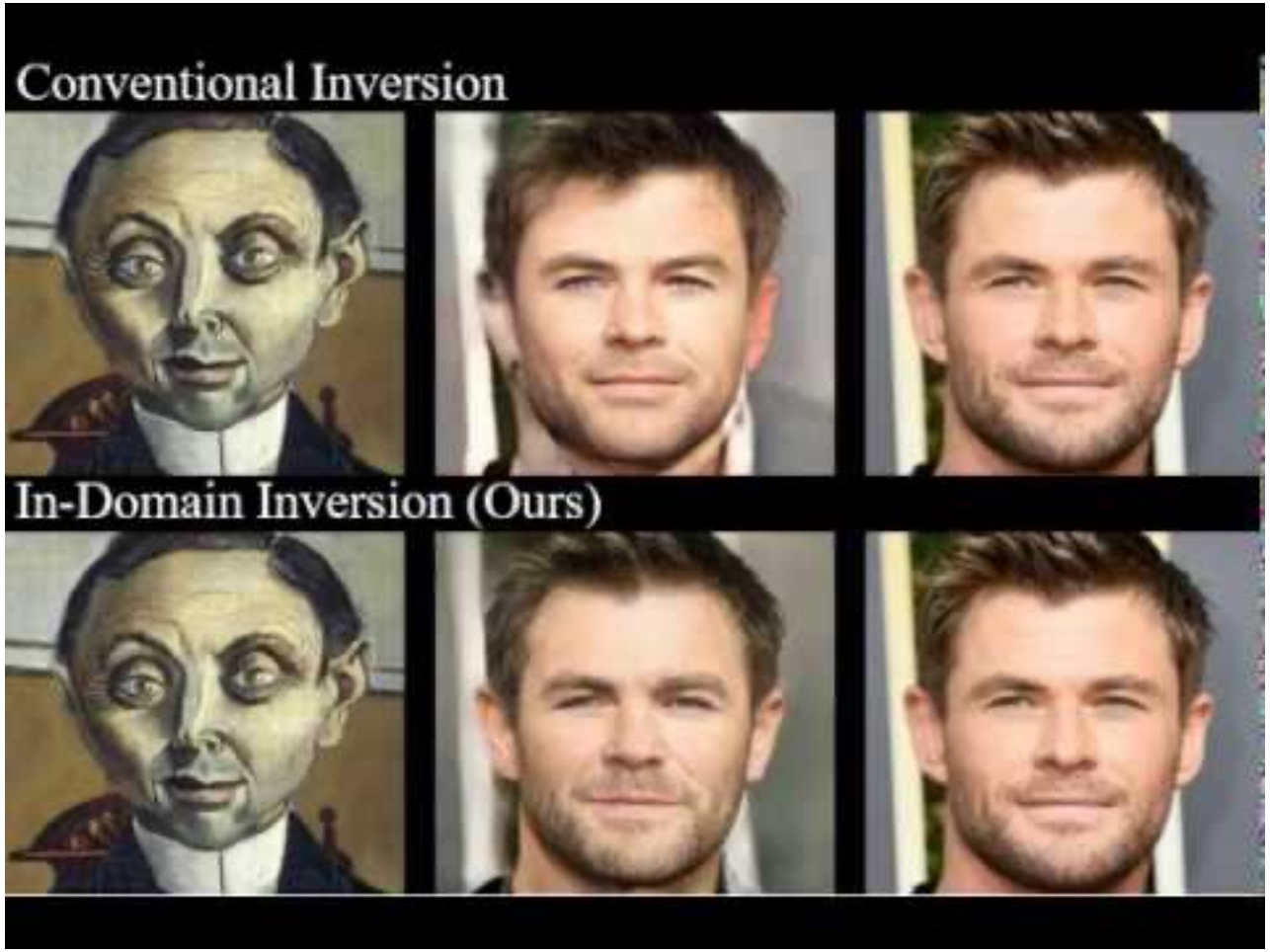
Semantic Diffusion



Semantically meaningful codes



More results



References

- Xia, Weihao & Zhang, Yulun & Yang, Yujiu & Xue, Jing-Hao & Zhou, Bolei & Yang, Ming-Hsuan. (2022). GAN Inversion: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 1-17. 10.1109/TPAMI.2022.3181070
- Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, José M. Álvarez: Invertible Conditional GANs for image editing. CoRR abs/1611.06355 (2016)
- Junyu Luo: Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets. CoRR abs/1703.10094 (2017)
- Antonia Creswell, Anil Anthony Bharath: Inverting The Generator Of A Generative Adversarial Network. CoRR abs/1611.05644 (2016)
- Lore Goetschalckx, Alex Andonian, Aude Oliva, Phillip Isola: Lore Goetschalckx, Alex Andonian, Aude Oliva, Phillip Isola: GANalyze: Toward Visual Definitions of Cognitive Image Properties. CoRR abs/1906.10112 (2019)

References

- Ali Jahanian, Lucy Chai, Phillip Isola: On the "steerability" of generative adversarial networks. CoRR abs/1907.07171 (2019)
- Yujun Shen, Jinjin Gu, Xiaoou Tang, Bolei Zhou: Interpreting the Latent Space of GANs for Semantic Face Editing. CoRR abs/1907.10786 (2019)
- Tero Karras, Samuli Laine, Timo Aila: A Style-Based Generator Architecture for Generative Adversarial Networks. CoRR abs/1812.04948 (2018)
- Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)

References

- Rameen, A., Yipeng, Q., Peter, W.: Image2stylegan: How to embed images into the stylegan latent space? In: ICCV (2019)
- Jiapeng Zhu, Yujun Shen, Deli Zhao, Bolei Zhou: In-Domain GAN Inversion for Real Image Editing. CoRR abs/2004.00049 (2020)

Questions time

Thanks