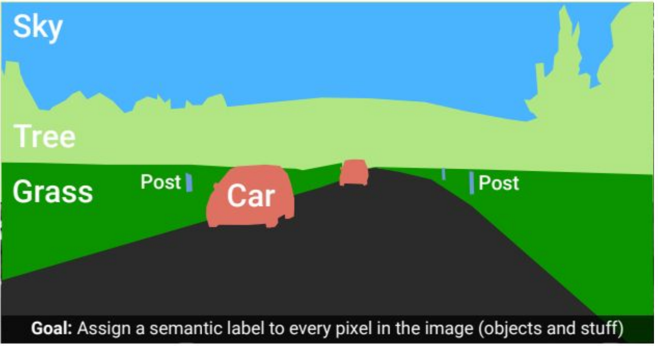
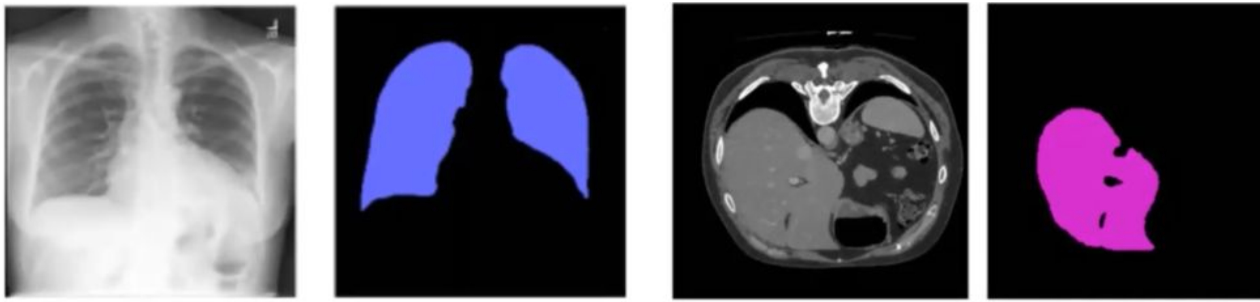
Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization

Semantic Segmentation Challenges



Computer Vision Lecture 10 A. Geiger, University of Tübingen



Medical Image Annotation

- Very time consuming
- Expensive domain experts
- Bad Out-of-domain generalization

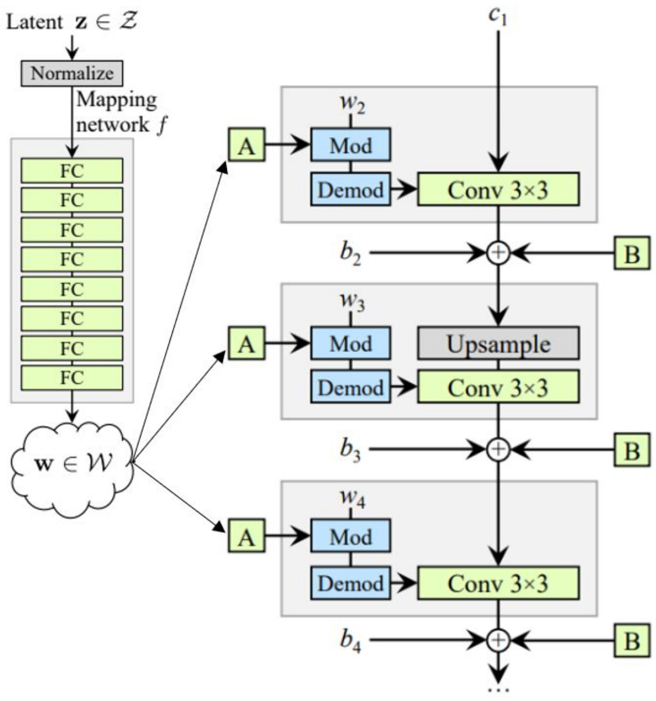
Solutions

- Learn and model *joint* image-label distribution
- Fully generative approach (StyleGAN2)
- Using Unlabeled data

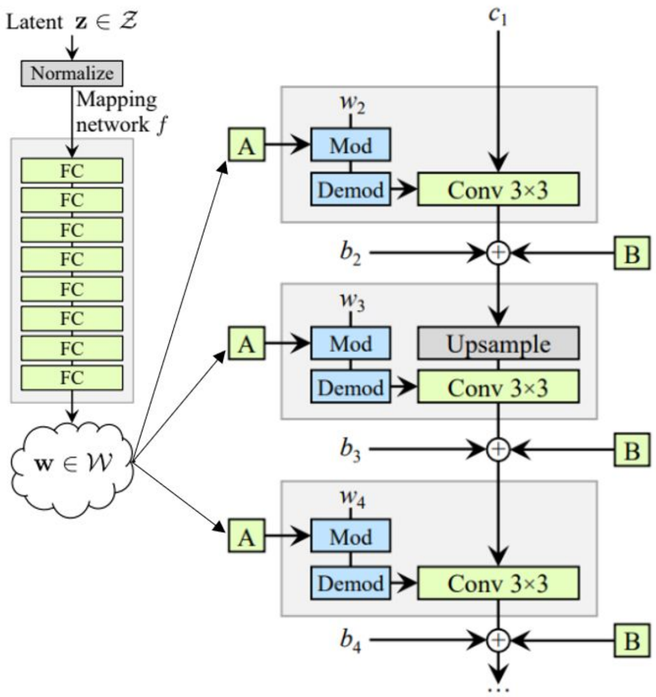
Formel definition

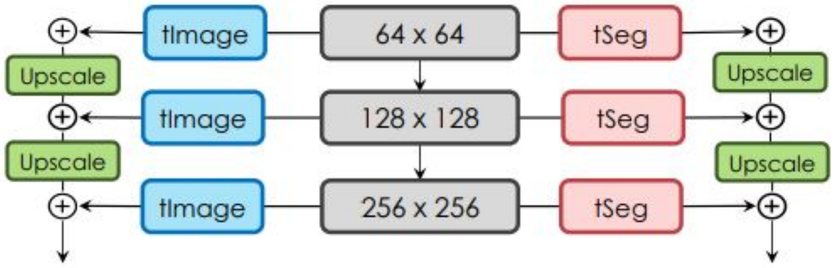
- Traditional: $f : \mathcal{X} \rightarrow \mathcal{Y}$ maximize $p(y|x)$
- Generative approach: $p(z)$ $G(z) : \mathcal{Z} \rightarrow (\mathcal{X}, \mathcal{Y})$

Generator - StyleGan2



Generator - StyleGan2



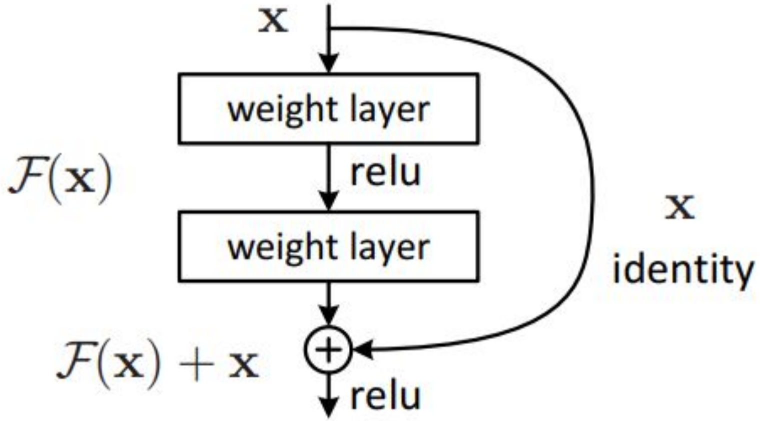


$$G : \mathbb{Z} \rightarrow \mathcal{W} \rightarrow (\mathcal{X}, \mathcal{Y})$$

output images $x \in \mathcal{X}$, pixel-wise labels $y \in \mathcal{Y}$

Discriminators

-

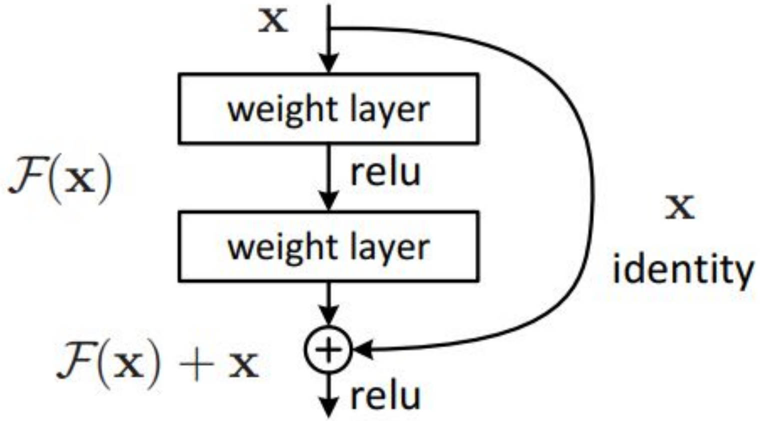


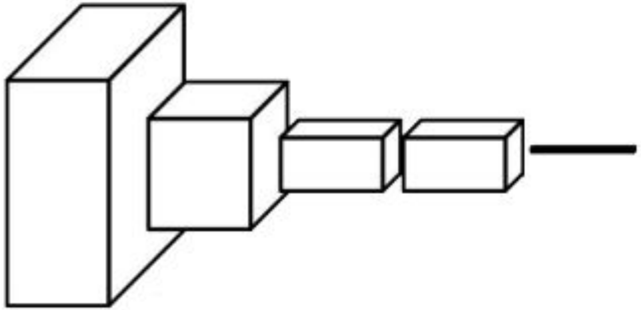
He et al.: Deep Residual Learning for Image Recognition. CVPR, 2016.

$$D_r : \mathcal{X} \rightarrow \mathbb{R}$$

Discriminators

-

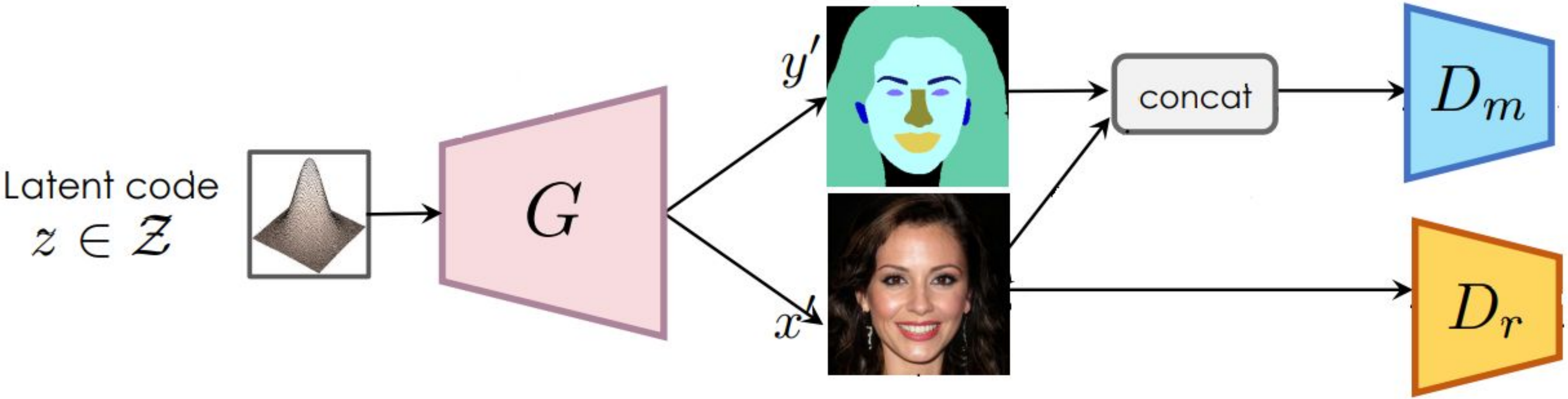




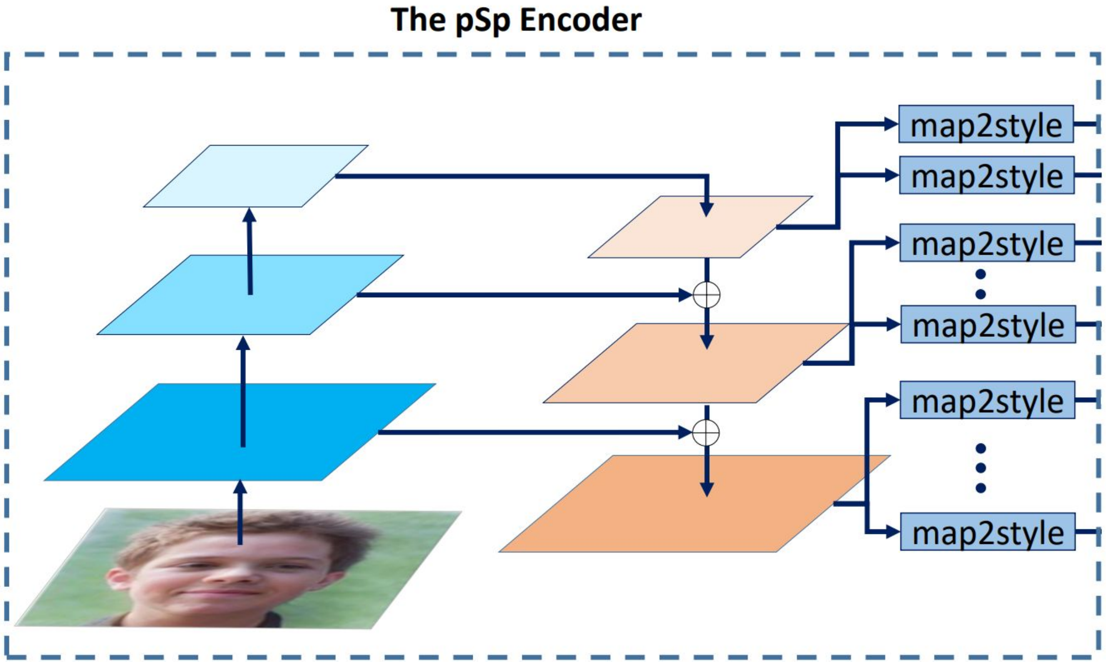
He et al.: Deep Residual Learning for Image Recognition. CVPR, 2016.

$$D_r : \mathcal{X} \to \mathbb{R} \qquad D_m : (\mathcal{X}, \mathcal{Y}) \to \mathbb{R}$$

Generator and Discriminator



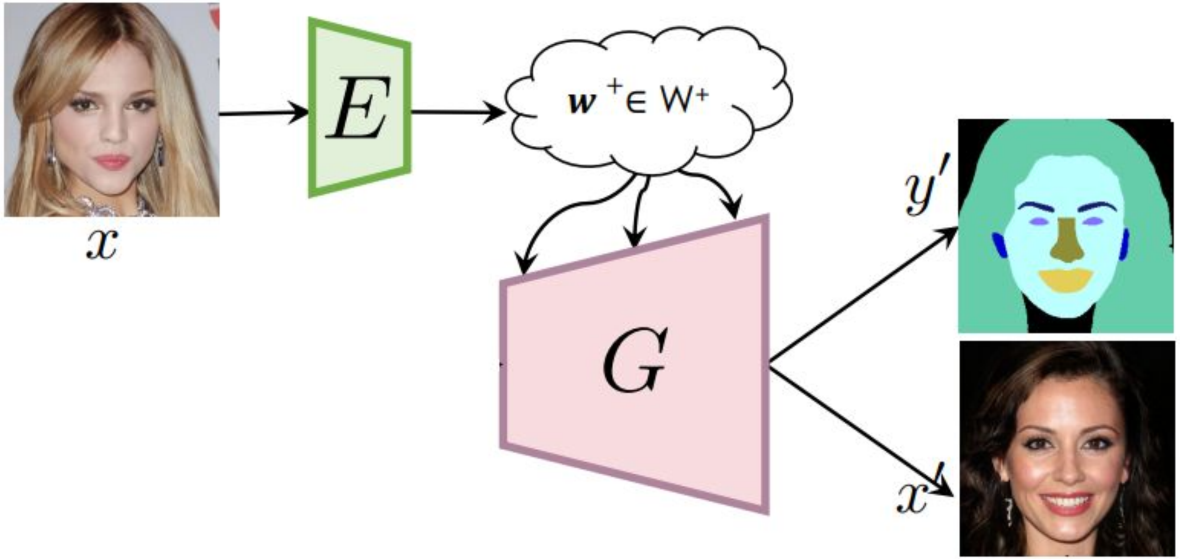
Encoder



Richardson, Elad, et al. "Encoding in style: a stylegan encoder for image-to-image translation." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

$$E : \mathcal{X} \rightarrow \mathcal{W}^+$$

Inference



Training - Datasets

- Unlabeled: $D_u = \{x_1, \ldots, x_n\}$
- Labeled: $D_l = \{(x_1, y_1), \ldots, (x_k, y_k)\}$
- $k \ll n$

Training Generator and Discriminators - Loss Functions

- $L_{D_r} = \mathbb{E}_{x_r \sim D_u}[\log D_r(x_r)] + \mathbb{E}_{(x_f,\cdot)=G(z), z \sim p(z)}[\log(1 - D_r(x_f))]$

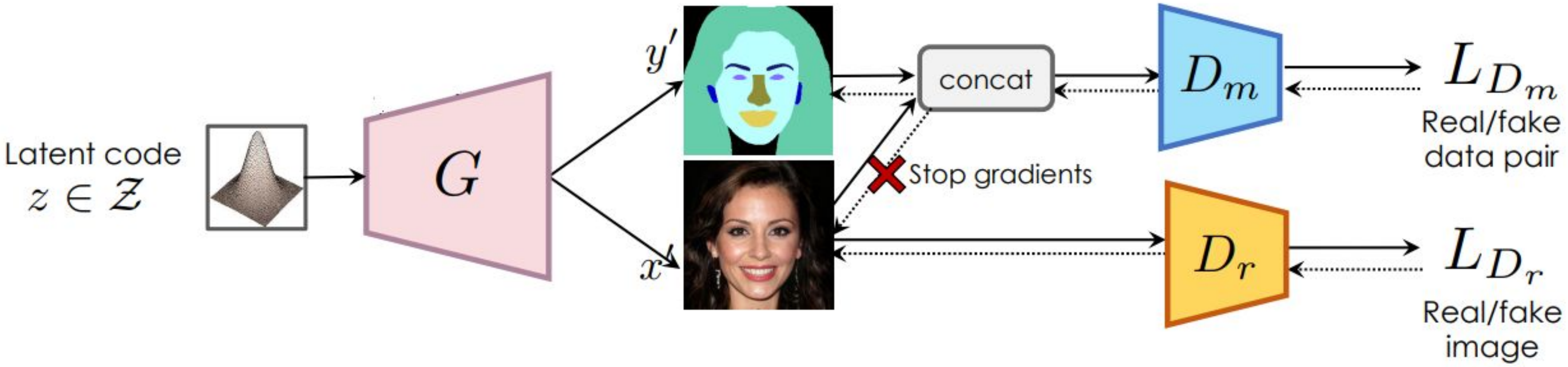
Training Generator and Discriminators - Loss Functions

- $L_{D_r} = \mathbb{E}_{x_r \sim D_u}[\log D_r(x_r)] + \mathbb{E}_{(x_f, \cdot)=G(z), z\sim p(z)}[\log(1 - D_r(x_f))]$
- $L_{D_m} = \mathbb{E}_{(x_r, y_r)\sim D_l}[\log D_m(x_r, y_r)] + \mathbb{E}_{(x_f, y_f)=G(z), z\sim p(z)}[\log(1 - D_m(x_f, y_f))]$

Training Generator and Discriminators - Loss Functions

- $L_{D_r} = \mathbb{E}_{x_r \sim D_u}[\log D_r(x_r)] + \mathbb{E}_{(x_f, \cdot)=G(z), z\sim p(z)}[\log(1 - D_r(x_f))]$
- $L_{D_m} = \mathbb{E}_{(x_r, y_r)\sim D_l}[\log D_m(x_r, y_r)] + \mathbb{E}_{(x_f, y_f)=G(z), z\sim p(z)}[\log(1 - D_m(x_f, y_f))]$
- $L_G = \mathbb{E}_{(x_f, \cdot)=G(z), z\sim p(z)}[\log(1 - D_r(x_f))] + \mathbb{E}_{(x_f, y_f)=G(z), z\sim p(z)}[\log(1 - D_m(x_f, y_f))]$

Training Generator and Discriminators



Training Encoder - Loss Functions

- $\mathcal{L}_E = \mathcal{L}_s + \mathcal{L}_u$

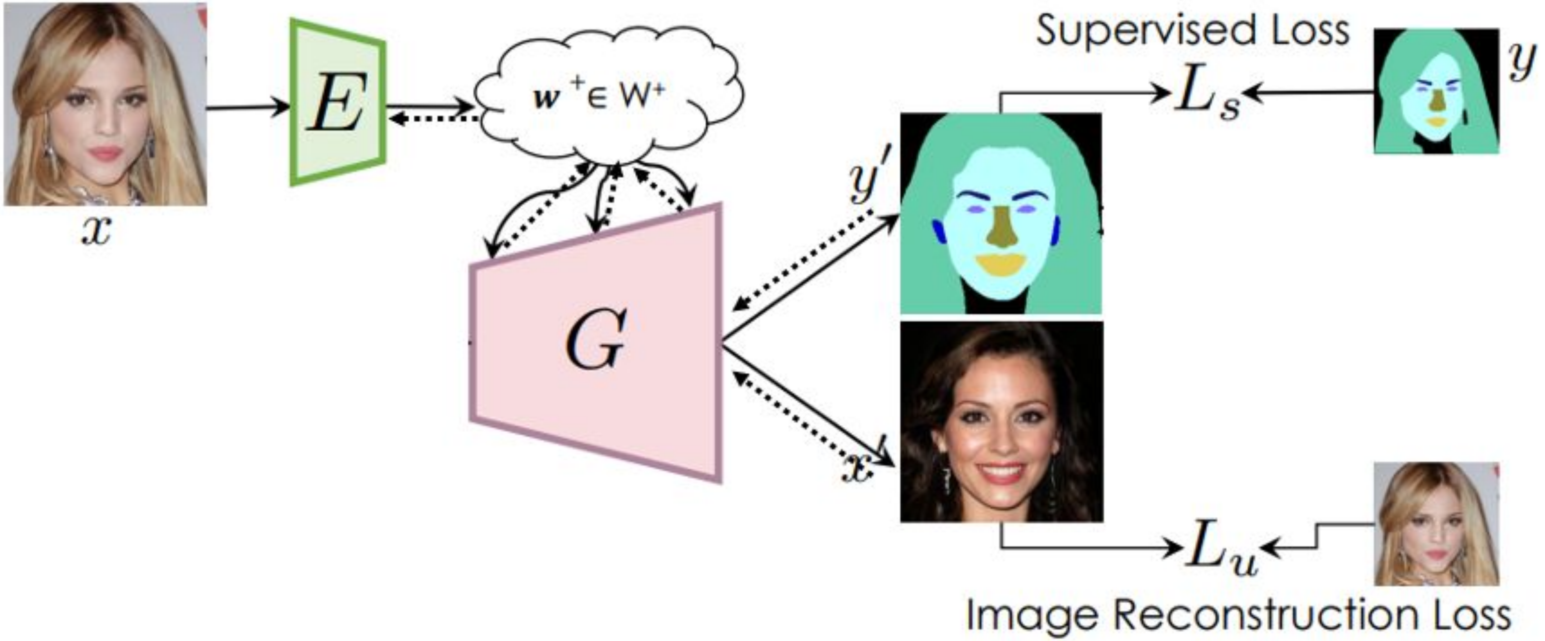
Training Encoder - Loss Functions

- $\mathcal{L}_E = \mathcal{L}_s + \mathcal{L}_u$
- $\mathcal{L}_s = \mathbb{E}_{(x,y)\sim D_l}\mathbf{H}(y, G_y(E(x))) + \mathbf{DC}(y, G_y(E(x)))$

Training Encoder - Loss Functions

- $\mathcal{L}_E = \mathcal{L}_s + \mathcal{L}_u$
- $\mathcal{L}_s = \mathbb{E}_{(x,y)\sim D_l}\mathbf{H}(y, G_y(E(x))) + \mathbf{DC}(y, G_y(E(x)))$
- $\mathcal{L}_u = \mathbb{E}_{x\sim D_l \cup D_u}\mathcal{L}_{LPIPS}(x, G_x(E(x))) + \lambda_1 \|x - G_x(E(x))\|_2^2$

Training Encoder



Inference - test-time optimization

- given target image x^* find optimal pixel-wise labels y^*
- $w^* = \underset{w^+ \in \mathcal{W}^+}{\arg\min} [\mathcal{L}_{\text{reconst}}(x^*, G_x(w^+)) + \lambda_2 \|w^+ - E(G(w^+))\|_2^2]$
- $\mathcal{L}_{\text{reconst}}(x, x^*) = \mathcal{L}_{\text{LPIPS}}(x, x^*) + \lambda_3 \|x - x^*\|_2^2$

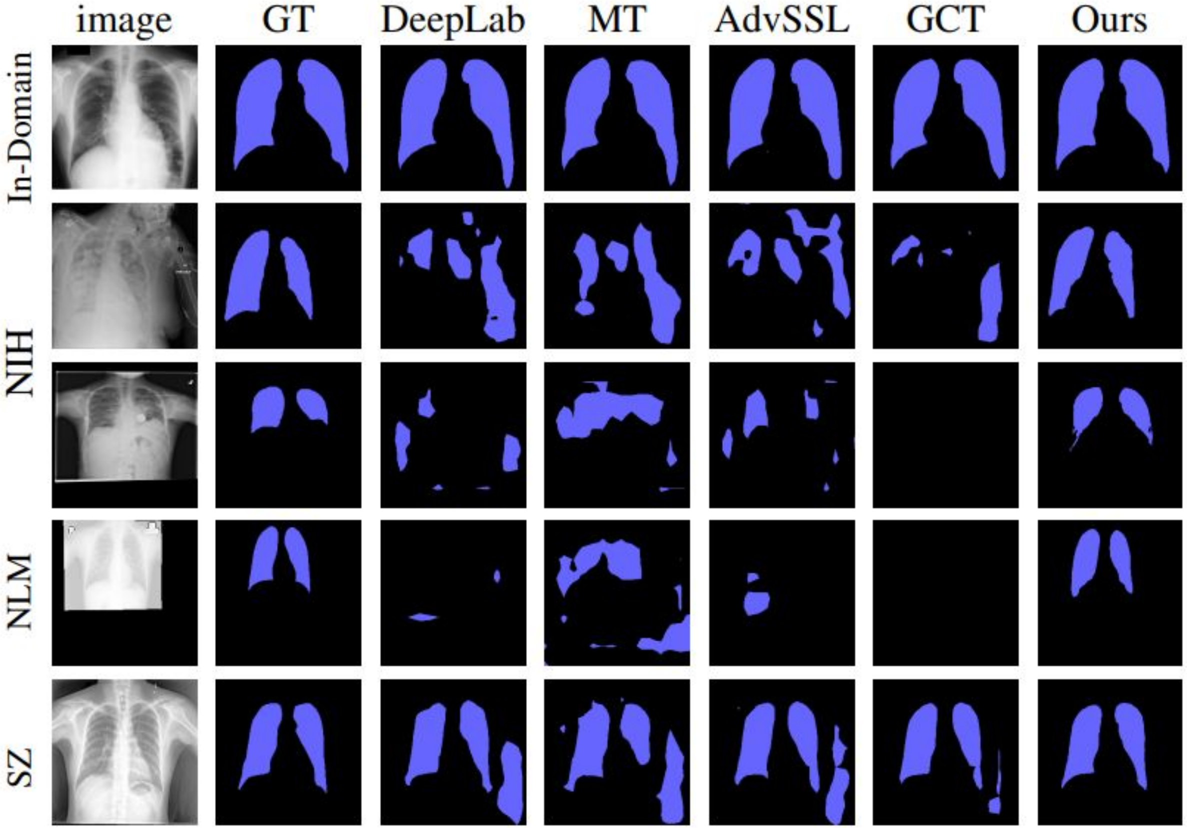
Chest X-ray Lung Segmentation

	Trained with **9** labeled data samples				Trained with **35** labeled data samples				Trained with **175** labeled data samples			
Method	JSRT	NLM	NIH	SZ	JSRT	NLM	NIH	SZ	JSRT	NLM	NIH	SZ
U-Net	0.9318	0.8605	0.6801	0.9051	0.9308	0.8591	0.7363	0.8486	0.9464	0.9143	0.7553	0.9005
DeepLab	0.9006	0.6324	0.7361	0.8124	0.9556	0.8323	0.8099	0.9138	0.9666	0.8175	0.8093	0.9312
MT	0.9239	0.8287	0.7280	0.8847	0.9436	0.8239	0.7305	0.8306	0.9604	0.8626	0.7893	0.8846
AdvSSL	0.9328	0.8500	0.7720	0.8901	0.9552	0.8191	0.5298	0.8968	**0.9684**	0.8344	0.7627	0.8846
GCT	0.9235	0.6804	0.6731	0.8665	0.9502	0.8327	0.7527	0.9184	0.9644	0.8683	0.7981	0.9393
Ours-NO	0.9464	0.9303	0.9097	0.9334	0.9471	0.9294	0.9223	0.9409	0.9465	0.9232	0.9204	0.9403
Ours	**0.9591**	**0.9464**	**0.9133**	**0.9362**	**0.9668**	**0.9606**	**0.9322**	**0.9485**	0.9669	**0.9509**	**0.9294**	**0.9469**

Chest X-ray Lung Segmentation

Method	Trained with **9** labeled data samples				Trained with **35** labeled data samples				Trained with **175** labeled data samples			
	JSRT	NLM	NIH	SZ	JSRT	NLM	NIH	SZ	JSRT	NLM	NIH	SZ
U-Net	0.9318	0.8605	0.6801	0.9051	0.9308	0.8591	0.7363	0.8486	0.9464	0.9143	0.7553	0.9005
DeepLab	0.9006	0.6324	0.7361	0.8124	0.9556	0.8323	0.8099	0.9138	0.9666	0.8175	0.8093	0.9312
MT	0.9239	0.8287	0.7280	0.8847	0.9436	0.8239	0.7305	0.8306	0.9604	0.8626	0.7893	0.8846
AdvSSL	0.9328	0.8500	0.7720	0.8901	0.9552	0.8191	0.5298	0.8968	**0.9684**	0.8344	0.7627	0.8846
GCT	0.9235	0.6804	0.6731	0.8665	0.9502	0.8327	0.7527	0.9184	0.9644	0.8683	0.7981	0.9393
Ours-NO	0.9464	0.9303	0.9097	0.9334	0.9471	0.9294	0.9223	0.9409	0.9465	0.9232	0.9204	0.9403
Ours	**0.9591**	**0.9464**	**0.9133**	**0.9362**	**0.9668**	**0.9606**	**0.9322**	**0.9485**	0.9669	**0.9509**	**0.9294**	**0.9469**

Chest X-ray Lung Segmentation



Number of Labeled vs Unlabeled Examples

		Unlabeled		
		3K	10K	28K
Labeled	30	0.6786	0.6845	0.6902
	150	0.7046	0.7438	0.7600
	1500	0.7566	0.7710	0.7810

(a) CelebA-Mask (In-Domain)

		Unlabeled		
		3K	10K	28K
Labeled	30	0.5410	0.5799	0.5883
	150	0.5871	0.6152	0.6336
	1500	0.6011	0.6204	0.6633

(b) MetFaces-40 (Out-Domain)

CT-MRI Transfer

	Trained with **8** labeled examples			Trained with **20** labeled examples			Trained with **118** labeled examples		
Method	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out
U-Net	0.7610	0.2568	0.3293	0.8229	0.3428	0.2310	0.8680	0.4453	0.4177
Ours-NO	0.8036	0.4811	0.5135	0.8462	**0.5538**	0.4511	0.8603	0.5055	**0.5633**
Ours	**0.8747**	**0.5565**	**0.5678**	**0.8961**	0.4989	**0.4575**	**0.9169**	**0.5097**	0.5243

CT-MRI Transfer

Method	Trained with **8** labeled examples			Trained with **20** labeled examples			Trained with **118** labeled examples		
	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out
U-Net	0.7610	0.2568	0.3293	0.8229	0.3428	0.2310	0.8680	0.4453	0.4177
Ours-NO	0.8036	0.4811	0.5135	0.8462	**0.5538**	0.4511	0.8603	0.5055	**0.5633**
Ours	**0.8747**	**0.5565**	**0.5678**	**0.8961**	0.4989	**0.4575**	**0.9169**	**0.5097**	0.5243

CT-MRI Transfer

	Trained with **8** labeled examples			Trained with **20** labeled examples			Trained with **118** labeled examples		
Method	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out
U-Net	0.7610	0.2568	0.3293	0.8229	0.3428	0.2310	0.8680	0.4453	0.4177
Ours-NO	0.8036	0.4811	0.5135	0.8462	**0.5538**	0.4511	0.8603	0.5055	**0.5633**
Ours	**0.8747**	**0.5565**	**0.5678**	**0.8961**	0.4989	**0.4575**	**0.9169**	**0.5097**	0.5243

CT-MRI Transfer

	Trained with **8** labeled examples			Trained with **20** labeled examples			Trained with **118** labeled examples		
Method	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out	CT	MRI T1-in	MRI T1-out
U-Net	0.7610	0.2568	0.3293	0.8229	0.3428	0.2310	0.8680	0.4453	0.4177
Ours-NO	0.8036	0.4811	0.5135	0.8462	**0.5538**	0.4511	0.8603	0.5055	**0.5633**
Ours	**0.8747**	**0.5565**	**0.5678**	**0.8961**	0.4989	**0.4575**	**0.9169**	**0.5097**	0.5243

Conclusion

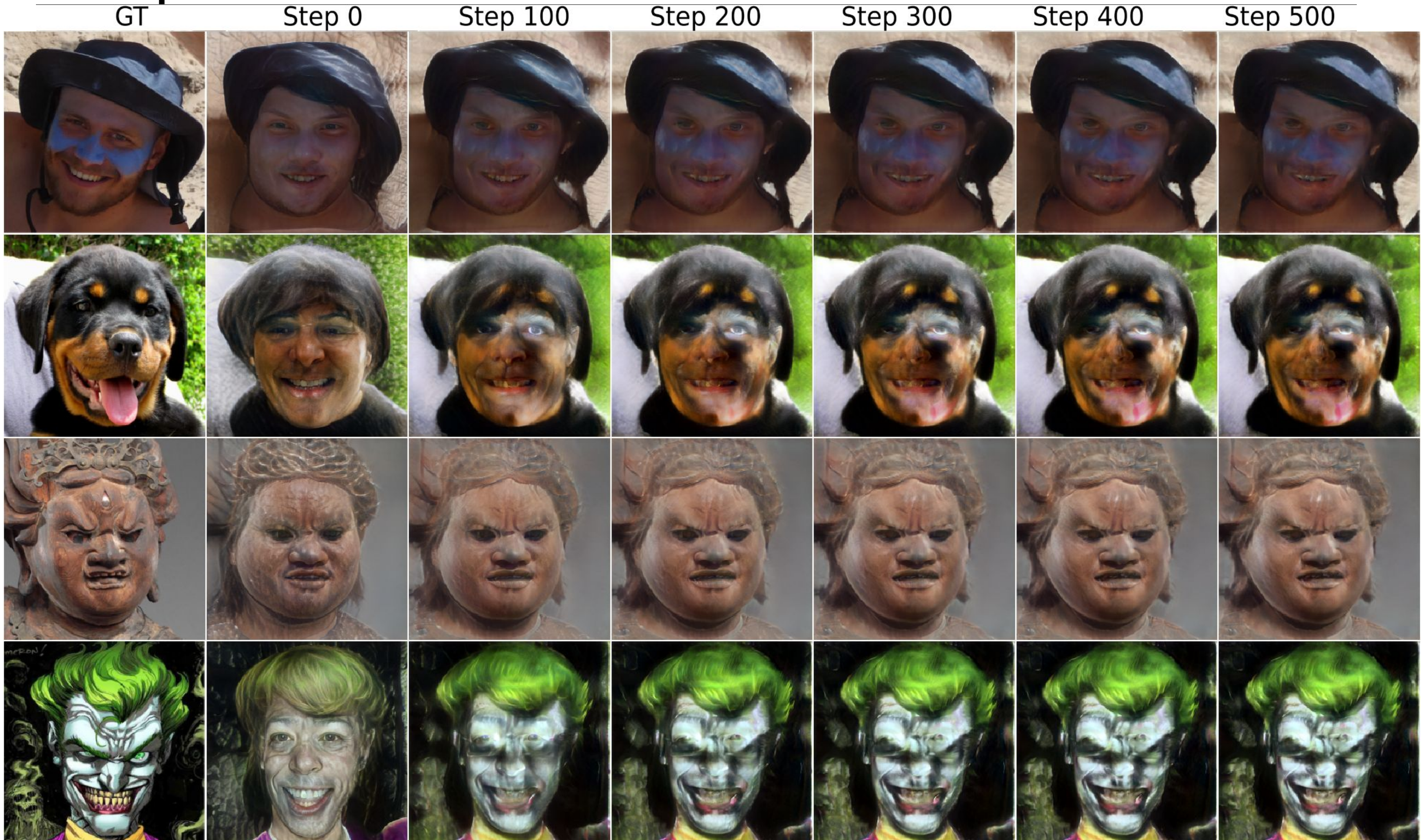
- Really good Out-of-Domain Performance
- Computationally expensive
    - Training
    - Inference

Hackathon

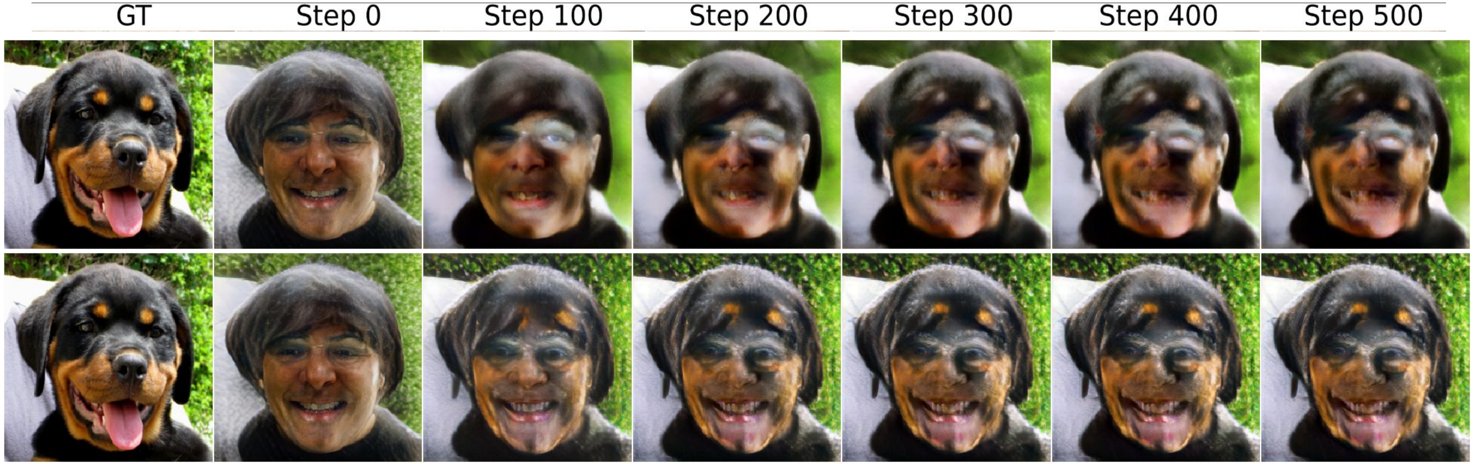
Setup

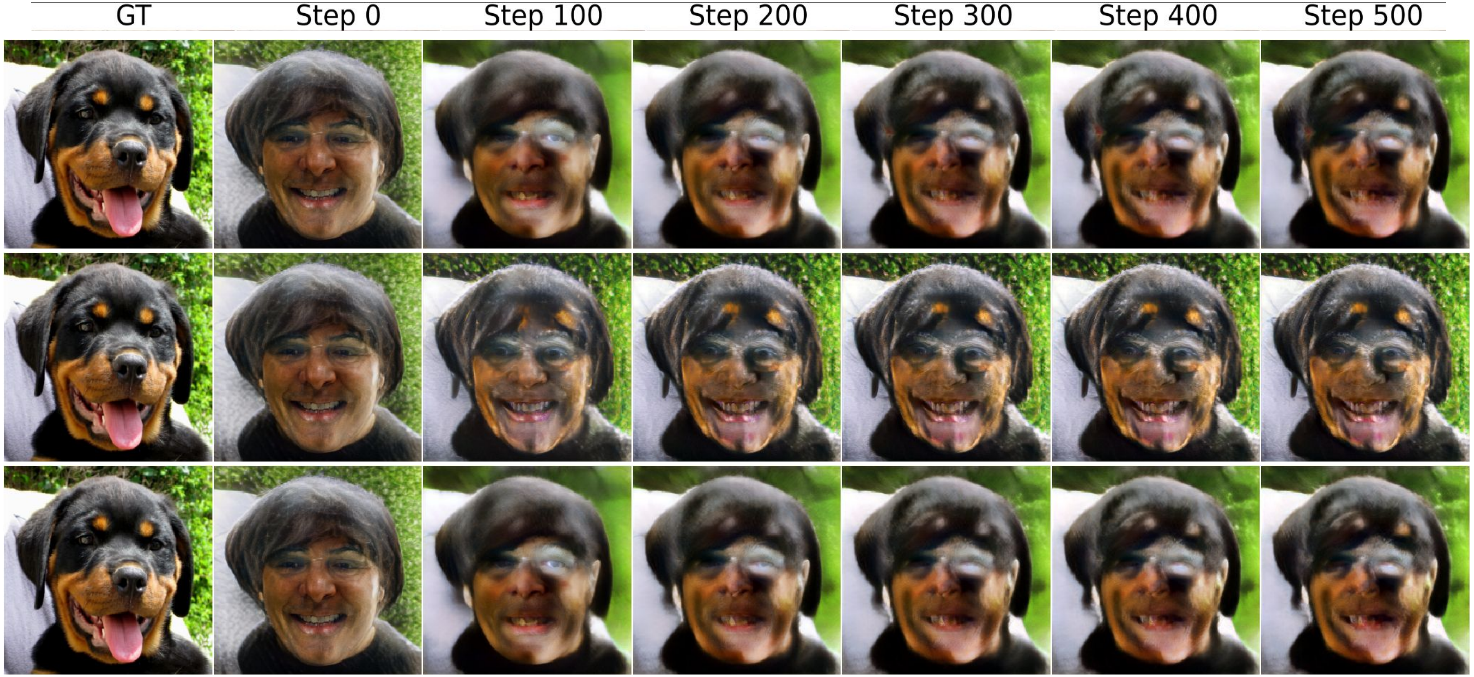
- No models published used in this paper
- *Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation*
- Pretrained pSp Encoder/StyleGAN2
- Test-time optimization experiments with Adam (lr=0.01)

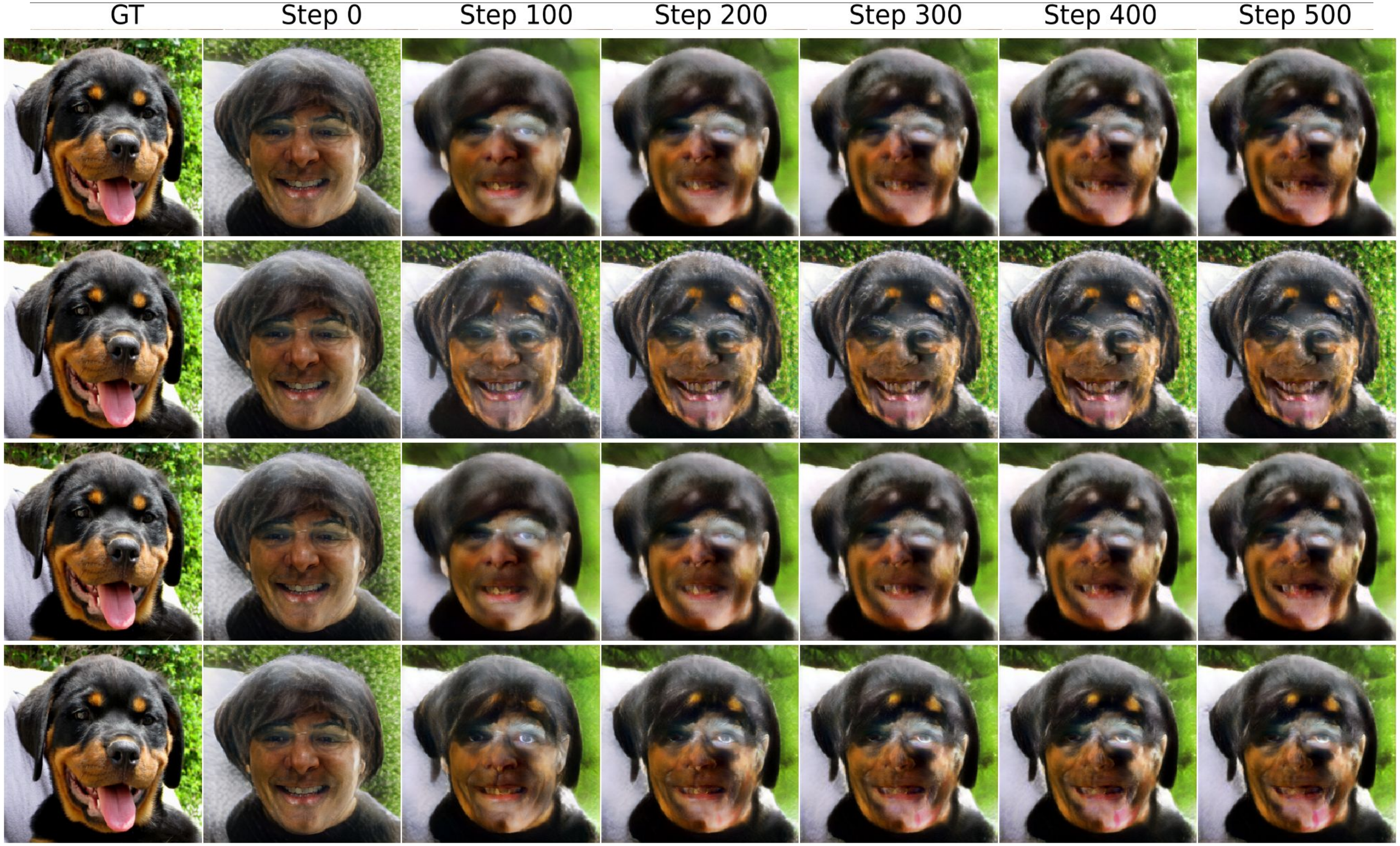
Test-time optimazaiton




GT
Step 0
Step 100
Step 200
Step 300
Step 400
Step 500


GT
Step 0
Step 100
Step 200
Step 300
Step 400
Step 500


GT
Step 0
Step 100
Step 200
Step 300
Step 400
Step 500


GT
Step 0
Step 100
Step 200
Step 300
Step 400
Step 500

Sources

KARRAS, Tero, et al. Analyzing and improving the image quality of stylegan. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020. S. 8110-8119.

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

run1

run2

run3