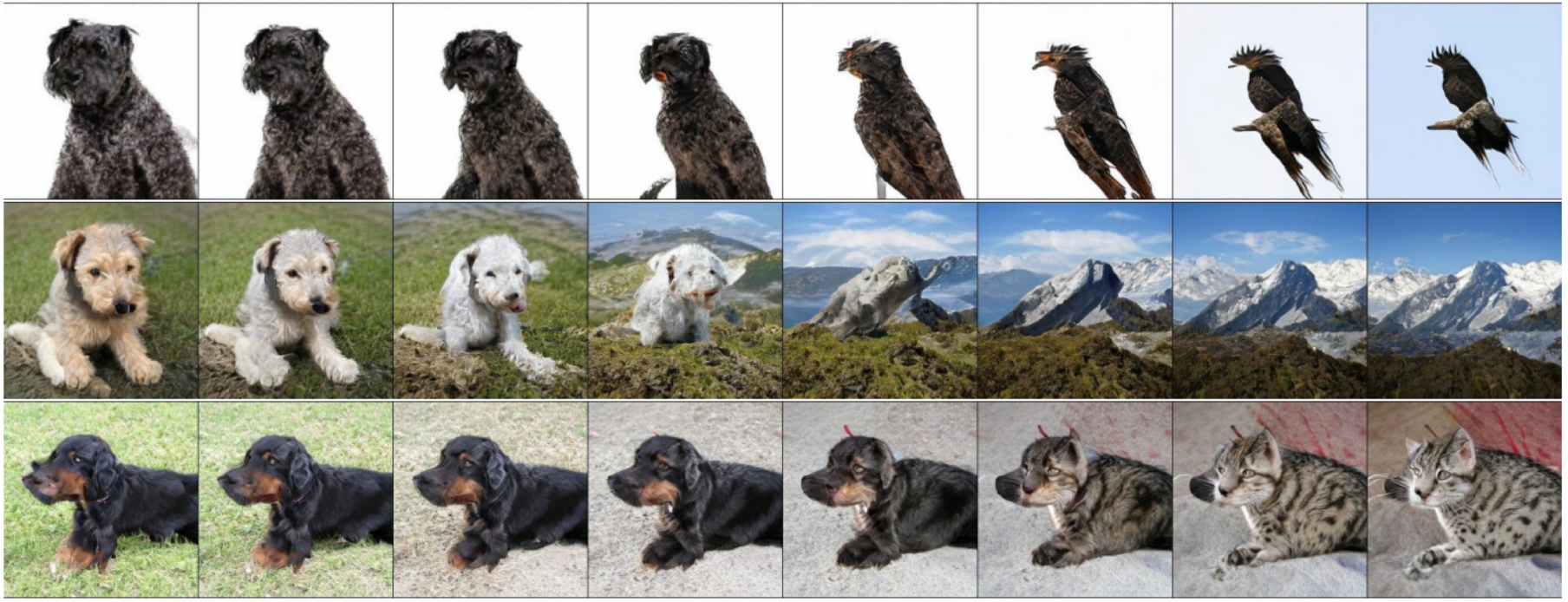
Advanced GAN models and applications



BigGAN (2018)

“The early years”

“The early years”

EBGAN (2017)

BigGAN (2018)

Advanced GAN models and applications

- Progressive GAN
- StyleGAN
- GAN-based image editing applications
- Self-attention GAN, BigGAN
- StyleGAN-XL, GigaGAN, StyleGAN-T

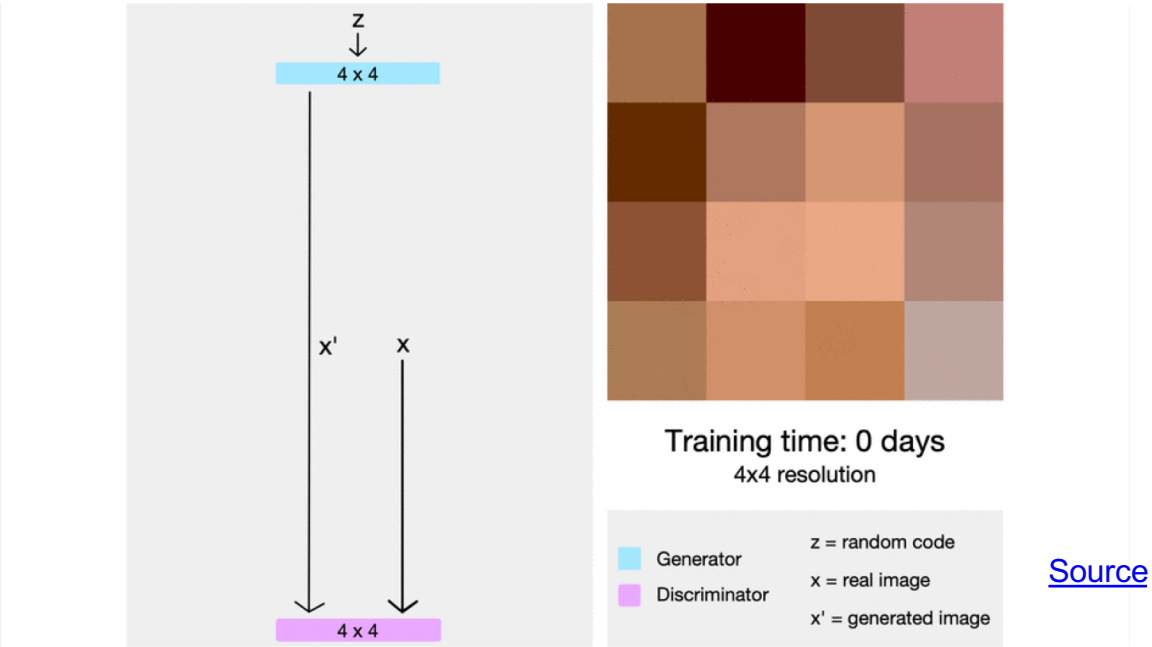
Progressive GANs

Realistic face images up to 1024 x 1024 resolution

T. Karras, T. Aila, S. Laine, J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018

Progressive GANs

- Key idea: train lower-resolution models, gradually add layers corresponding to higher-resolution outputs



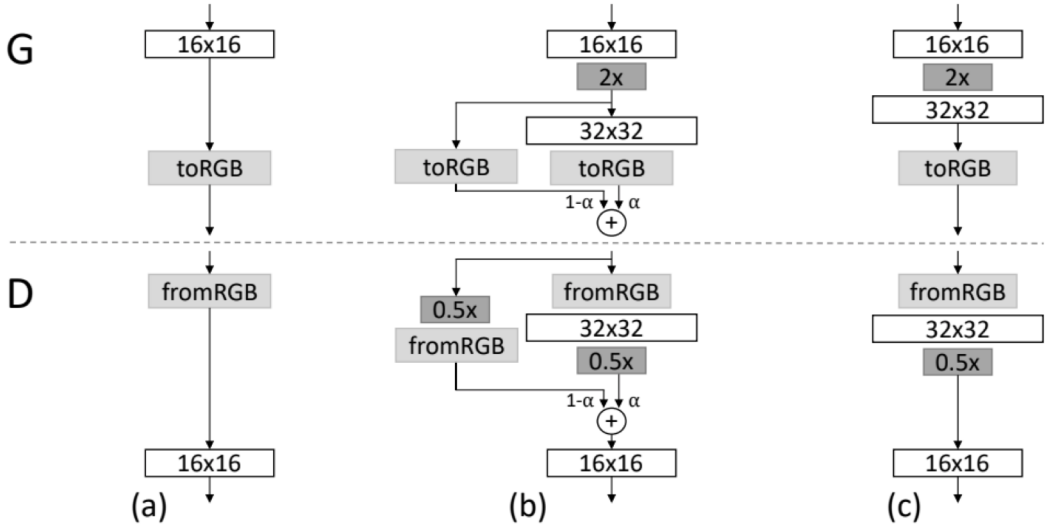
Source

T. Karras, T. Aila, S. Laine, J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018

Progressive GANs

- Key idea: train lower-resolution models, gradually add layers corresponding to higher-resolution outputs

Transition from 16x16 to 32x32 images



T. Karras, T. Aila, S. Laine, J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018

Progressive GANs: Implementation details

- Loss: WGAN-GP loss (preferred) or LSGAN
- Architectures:
  - Nearest neighbor upsampling (2x2 replication) followed by regular convolutions instead of transposed conv layers
  - Average pooling instead of striding for downsampling in discriminator
  - Leaky ReLUs used in discriminator and generator
  - Per-pixel response normalization in generator: rescale feature vector in each pixel to unit length after each conv layer
- Use of minibatch standard deviation in discriminator (append to feature map)
- Exponential moving average of generator weights for display

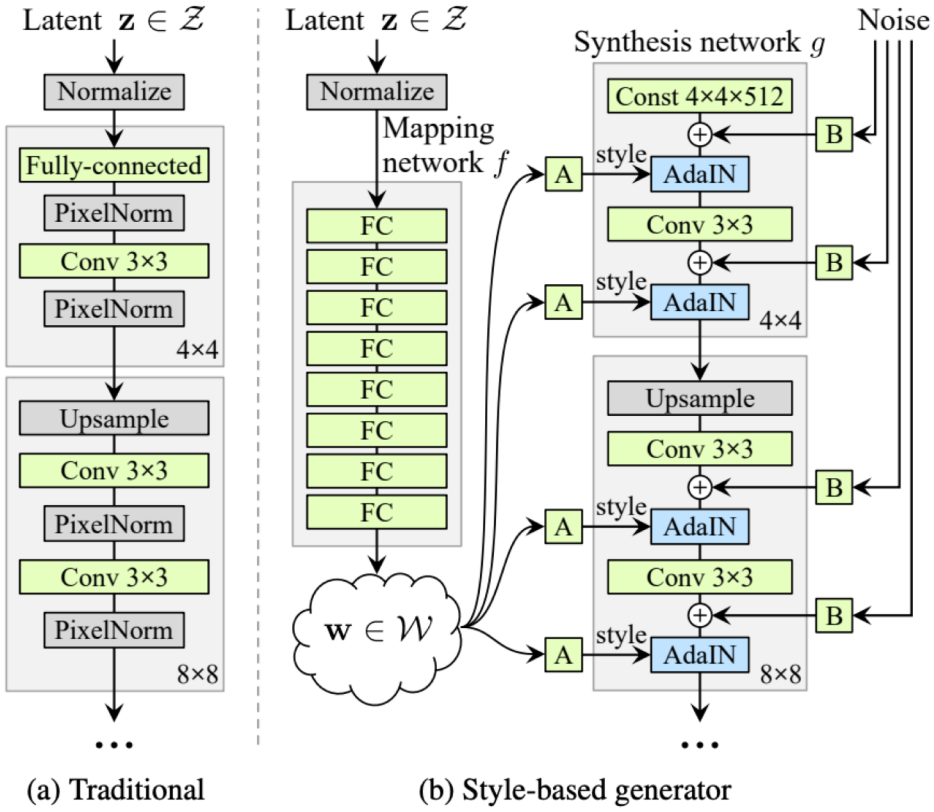
T. Karras, T. Aila, S. Laine, J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018

Progressive GANs: Results

256 x 256 results for LSUN categories

StyleGAN

- Built on top of Progressive GAN
- Start generation with constant (instead of noise vector)
- Noise vector is transformed to latent vector *w* that is later specialized to *style codes*
- Style codes control *adaptive instance normalization* (AdaIN) or scaling and biasing of each feature map
- Add noise after each convolution and before nonlinearity (enables stochastic detail)

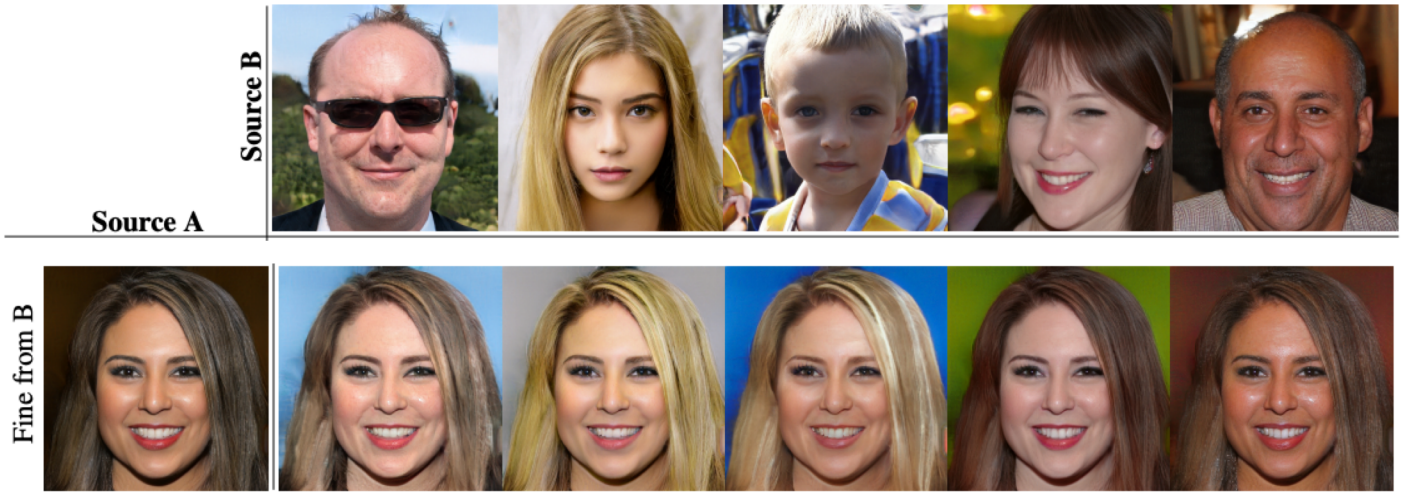


T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

StyleGAN: Results

T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

Mixing styles



“Two sets of images were generated from their respective latent codes (sources A and B); the rest of the images were generated by copying a specified subset of styles from source B and taking the rest from source A.”

Mixing styles

Mixing styles

DeepFakes

TECHNOLOGY

That smiling LinkedIn profile face might be a computer-generated fake

March 27, 2022 · 7:00 AM ET

https://www.npr.org/2022/03/27/1088140809/fake-linkedin-profiles

StyleGAN: Bedrooms

T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

StyleGAN: Cars

T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

StyleGAN2

- Change normalization, remove progressive growing to address StyleGAN artifacts

Figure 1. Instance normalization causes water droplet -like artifacts in StyleGAN images. These are not always obvious in the generated images, but if we look at the activations inside the generator network, the problem is always there, in all feature maps starting from the 64x64 resolution. It is a systemic problem that plagues all StyleGAN images.

Figure 6. Progressive growing leads to "phase" artifacts. In this example the teeth do not follow the pose but stay aligned to the camera, as indicated by the blue line.

T. Karras et al. Analyzing and Improving the Image Quality of StyleGAN. CVPR 2020

StyleGAN3

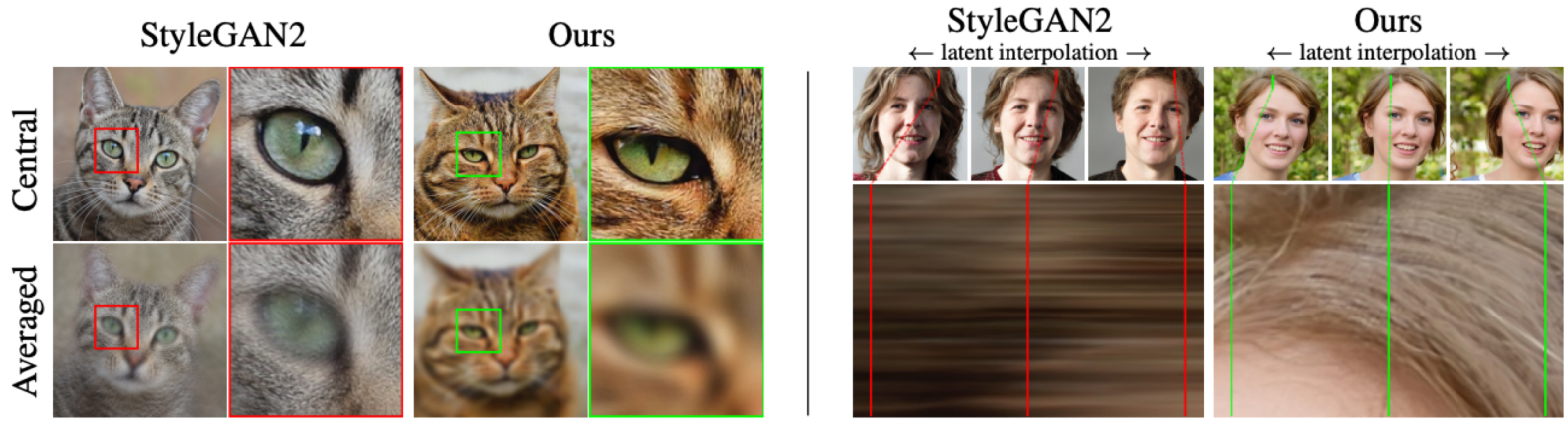
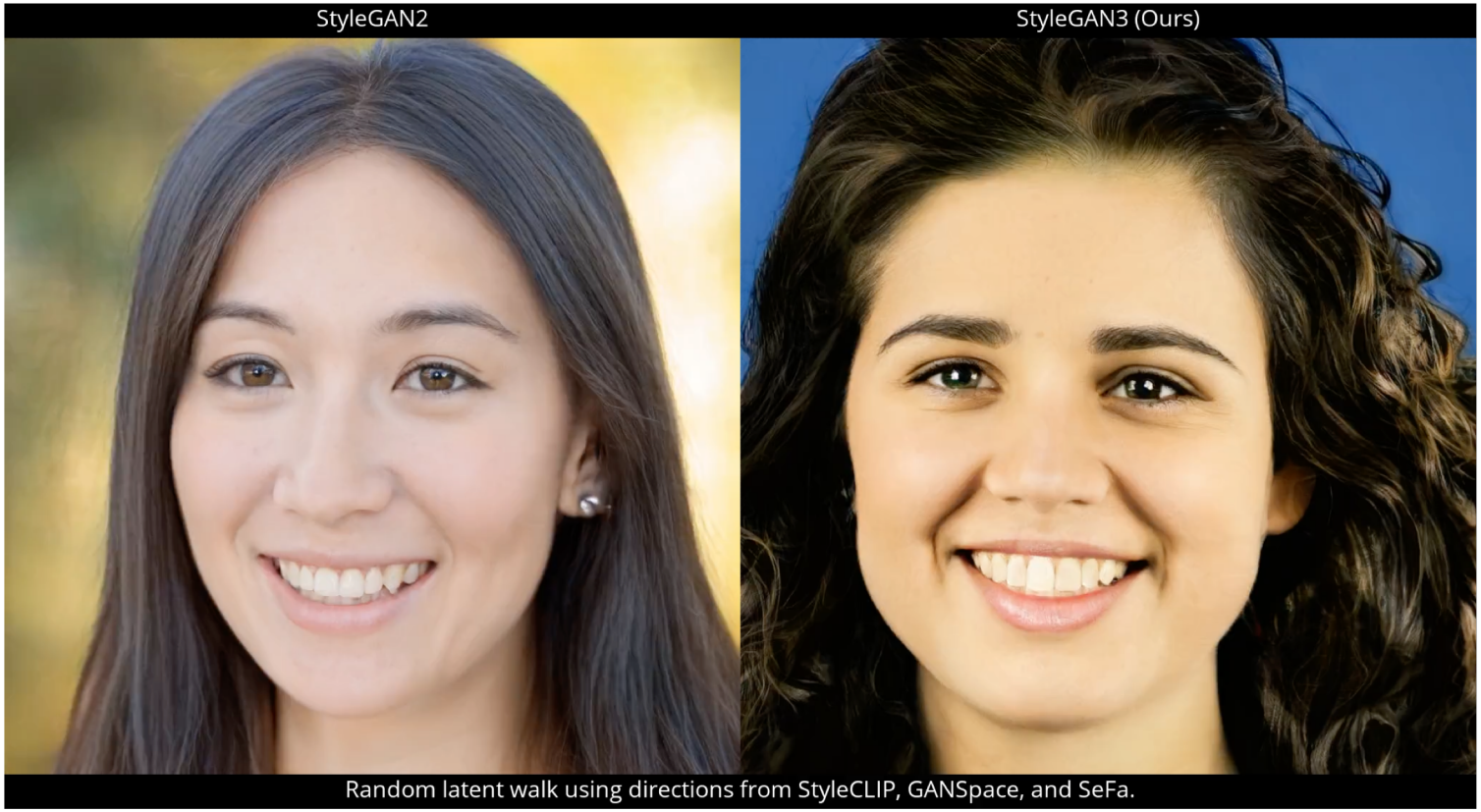


Figure 1: Examples of "texture sticking". **Left:** The average of images generated from a small neighborhood around a central latent (top row). The intended result is uniformly blurry because all details should move together. However, with StyleGAN2 many details (e.g., fur) stick to the same pixel coordinates, showing unwanted sharpness. **Right:** From a latent space interpolation (top row), we extract a short vertical segment of pixels from each generated image and stack them horizontally (bottom). The desired result is hairs moving in animation, creating a time-varying field. With StyleGAN2 the hairs mostly stick to the same coordinates, creating horizontal streaks instead.

T. Karras et al. Alias-Free Generative Adversarial Networks. NeurIPS 2021

StyleGAN3



T. Karras et al. Alias-Free Generative Adversarial Networks. NeurIPS 2021

Videos

Outline

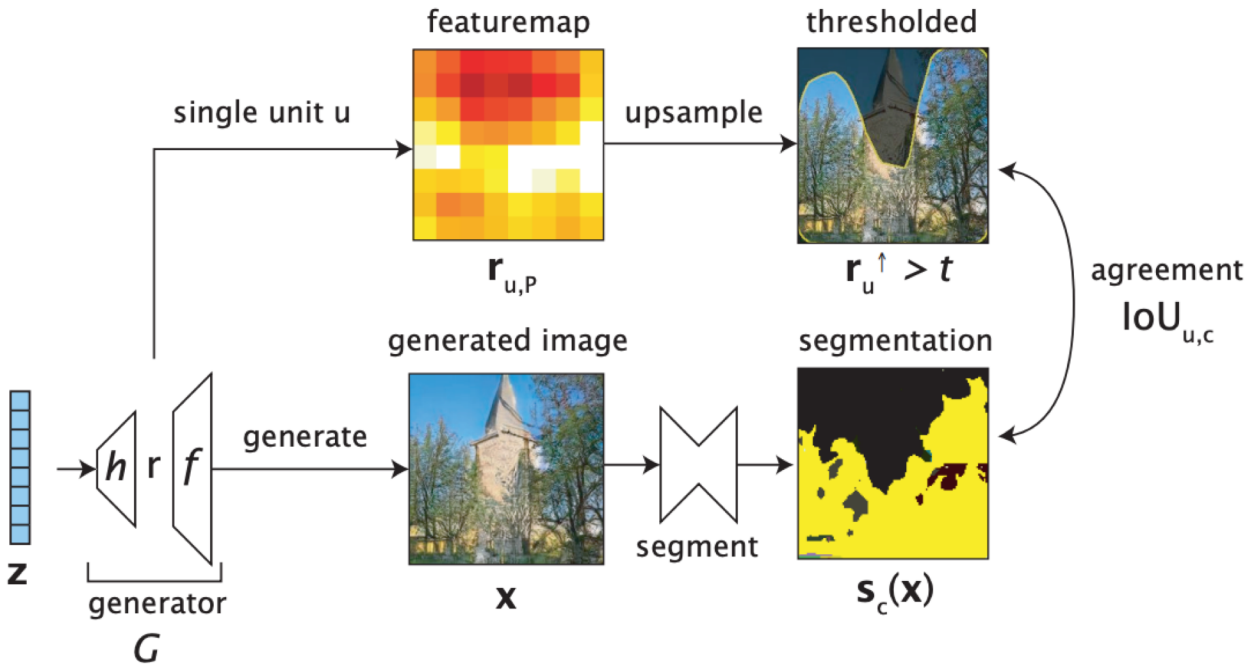
- Progressive GAN
- StyleGAN
- GAN-based image editing applications

GAN Dissection

D. Bau et al. GAN Dissection: Visualizing and understanding generative adversarial networks. ICLR 2019

GAN Dissection

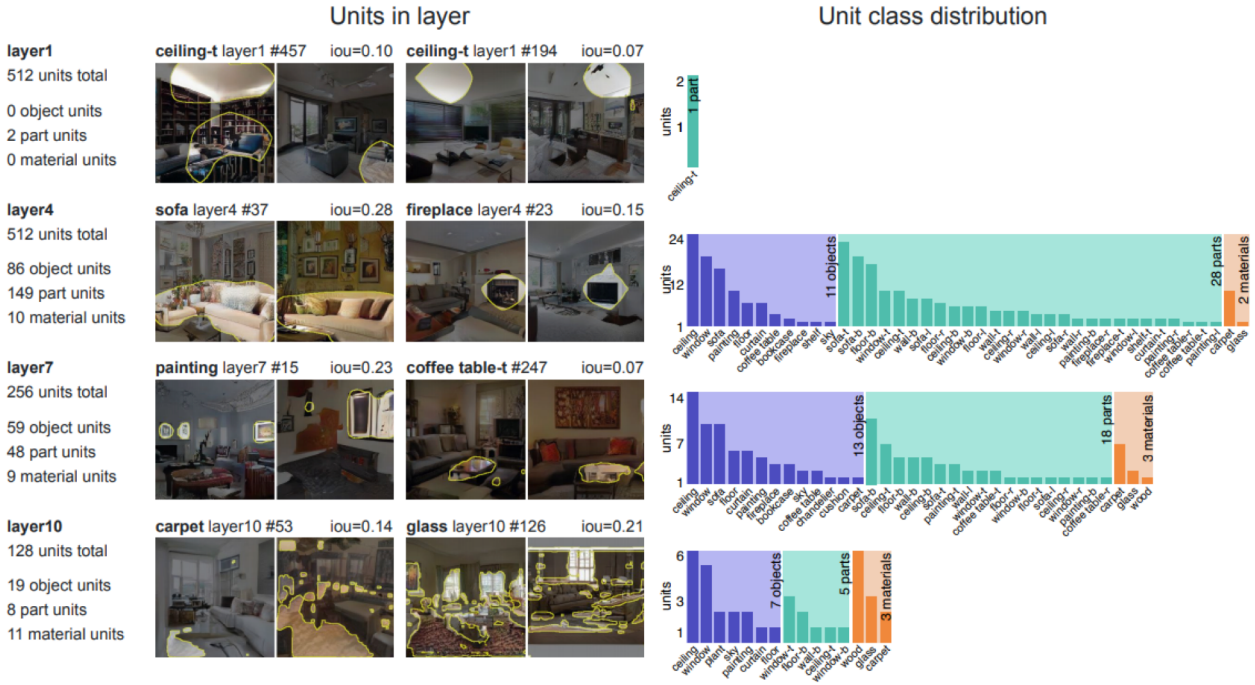
- Dissection:



D. Bau et al. [GAN Dissection: Visualizing and understanding generative adversarial networks](). ICLR 2019

GAN Dissection

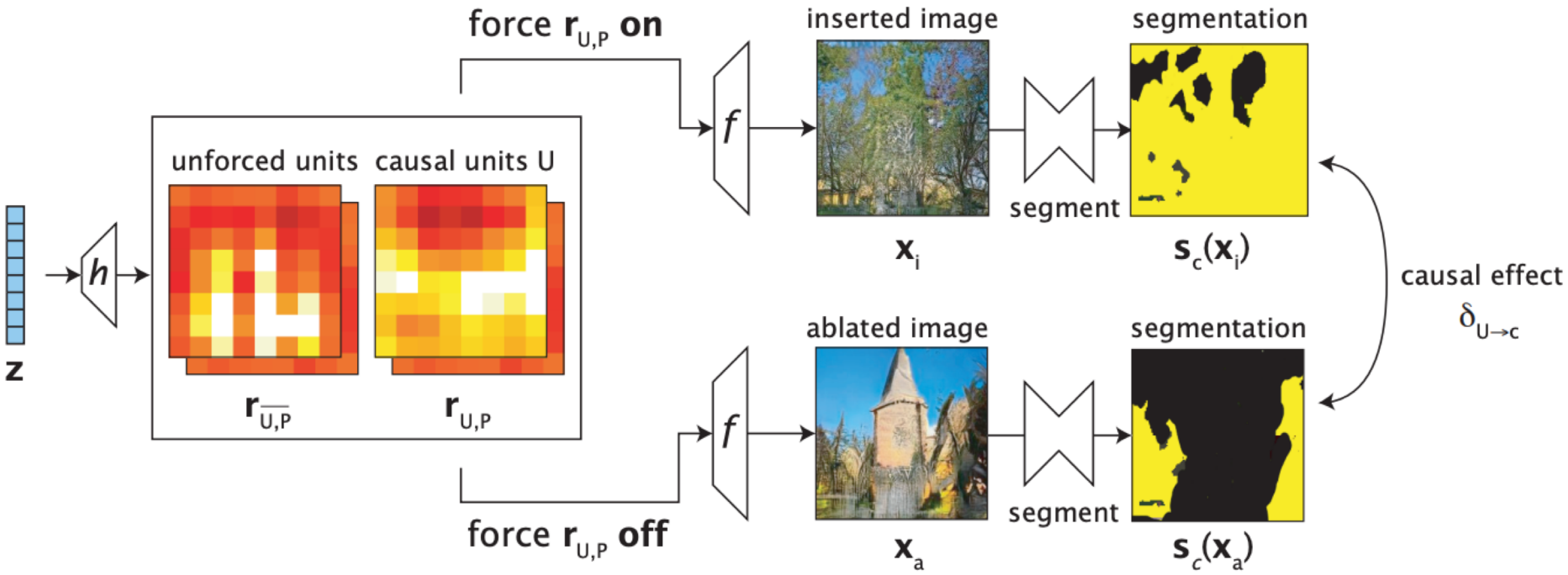
- Interpreting units at different levels of a progressive GAN (trained on “bedroom”):



D. Bau et al. [GAN Dissection: Visualizing and understanding generative adversarial networks](). ICLR 2019

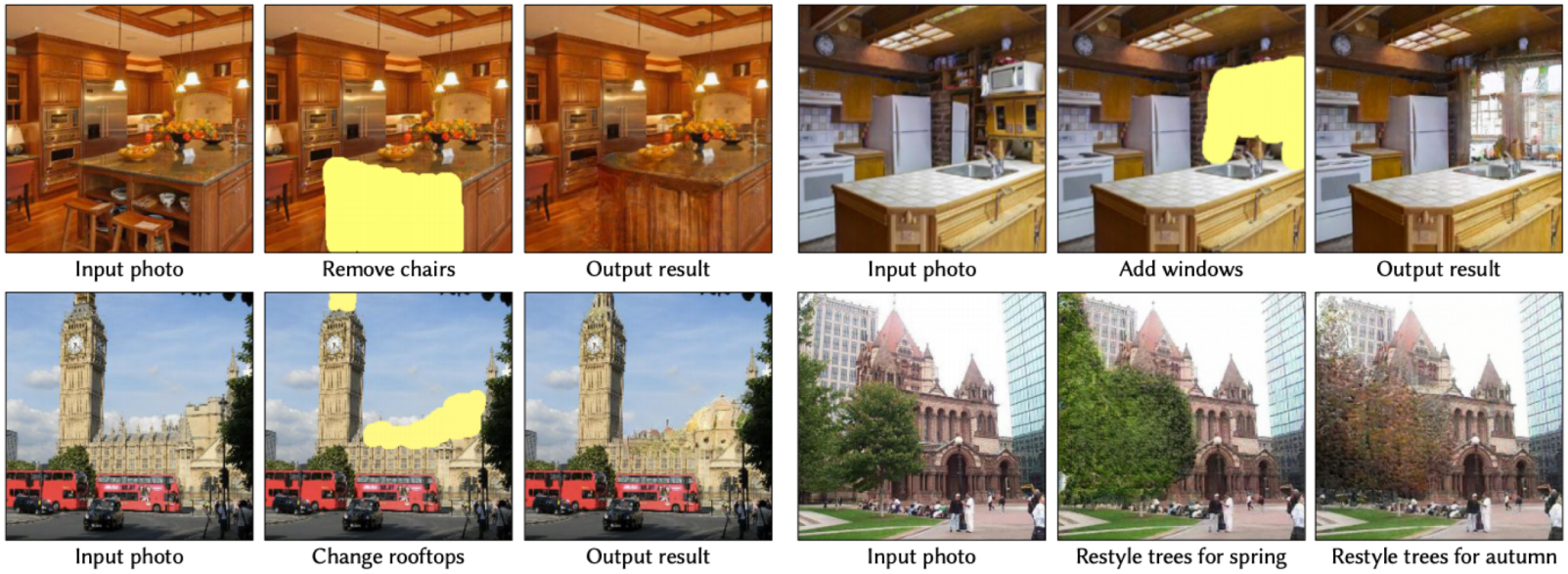
GAN Dissection

- Intervention:



D. Bau et al. GAN Dissection: Visualizing and understanding generative adversarial networks. ICLR 2019

GANPaint demo

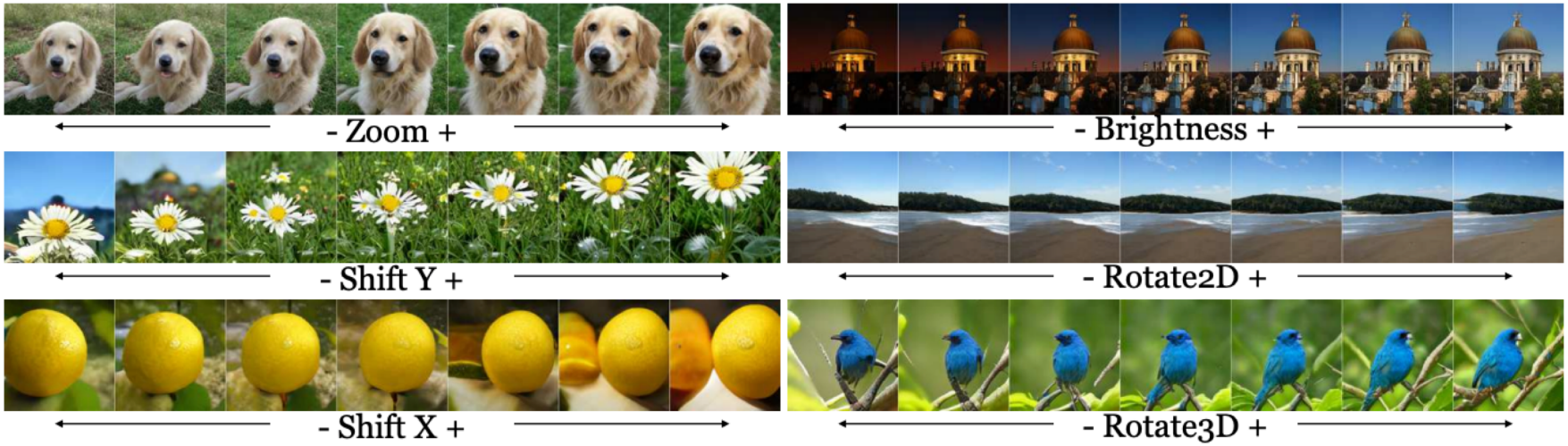


https://ganpaint.io/demo/?project=church

D. Bau et al. Semantic Photo Manipulation with a Generative Image Prior. SIGGRAPH 2019

Discovery of latent space directions

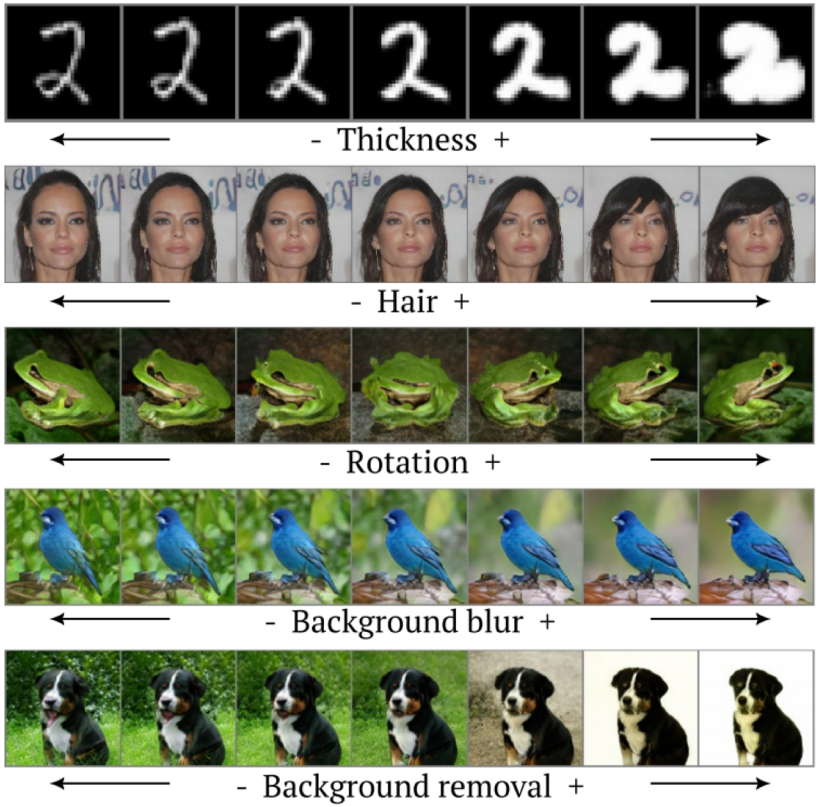
- Try to find simple “walks” in the latent space of GANs to achieve various meaningful transformations to explore structure of that space and test GANs’ ability to interpolate between training samples



A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. ICLR 2020

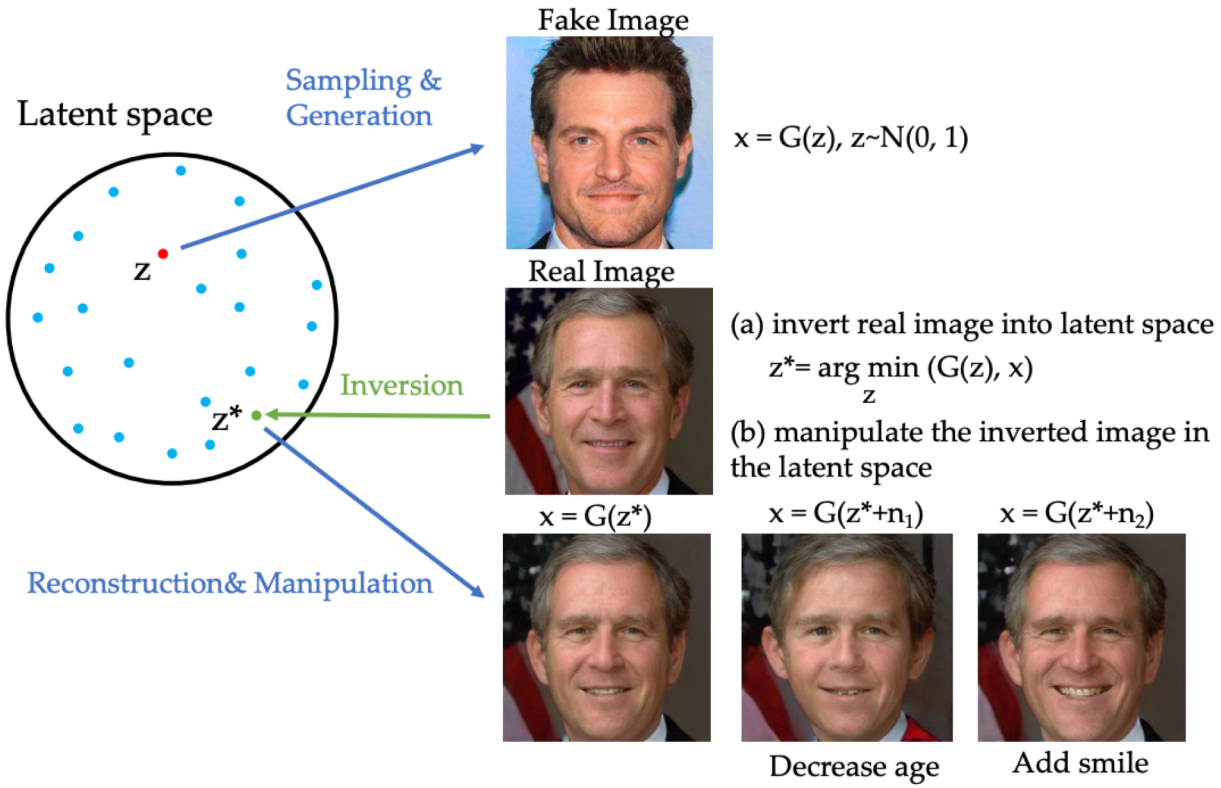
Discovery of latent space directions

- Goal: learn a set of directions inducing “disentangled” image transformations that are easy to distinguish from each other



A.Voynov and A. Babenko. Unsupervised Discovery of Interpretable Directions in the GAN Latent Space. ICML 2020

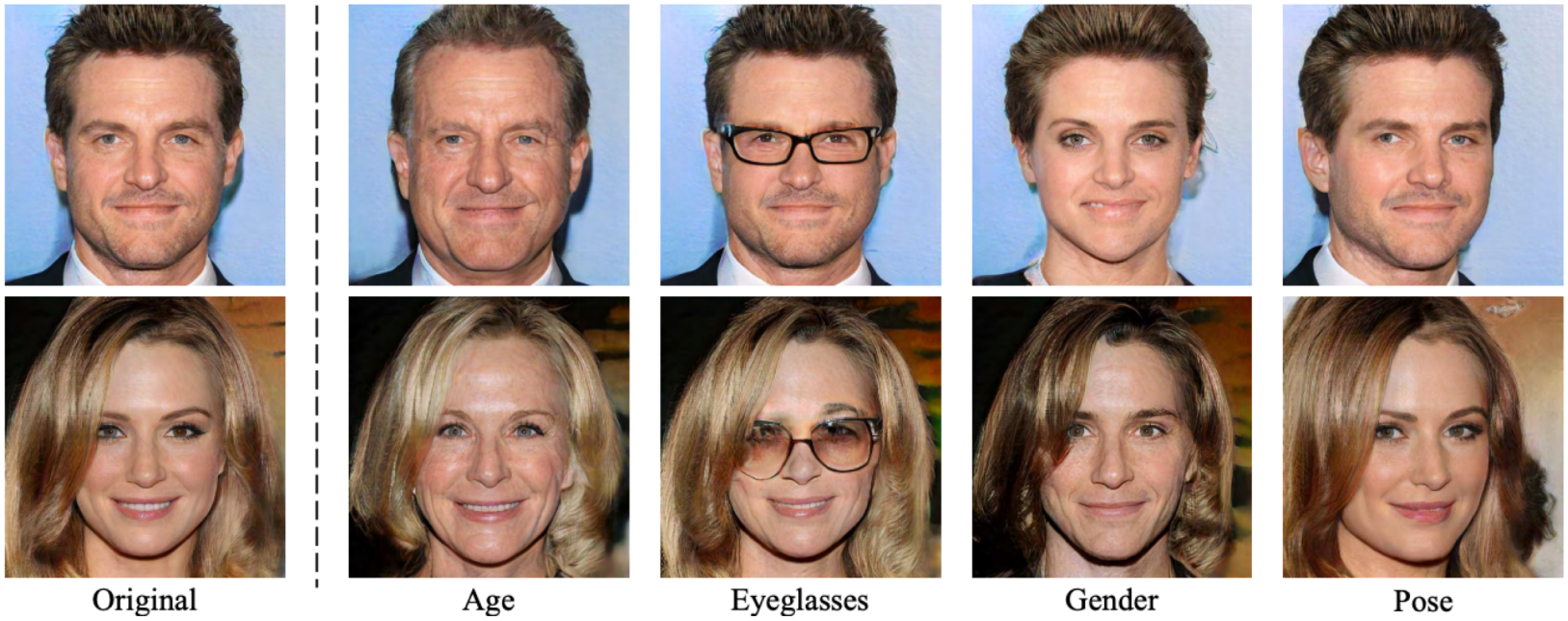
GAN inversion and image editing



W. Xia et al. GAN Inversion: A Survey. arXiv 2022

GAN-based image editing

- Use pre-trained attribute classifiers to find latent space directions corresponding to pose, smile, age, gender, eyeglasses



Y. Shen, J. Gu, X. Tang, B. Zhou. Interpreting the Latent Space of GANs for Semantic Face Editing. CVPR 2020

GAN-based image editing

Figure 1. Disentanglement in style space, demonstrated using three different datasets. Each of the three groups above shows two manipulations that occur independently inside the same semantic region (hair, bed, and car, from left to right). The indices of the manipulated layer and channel are indicated in parentheses.

Z. Wu et al. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. CVPR 2021

GAN-based image editing

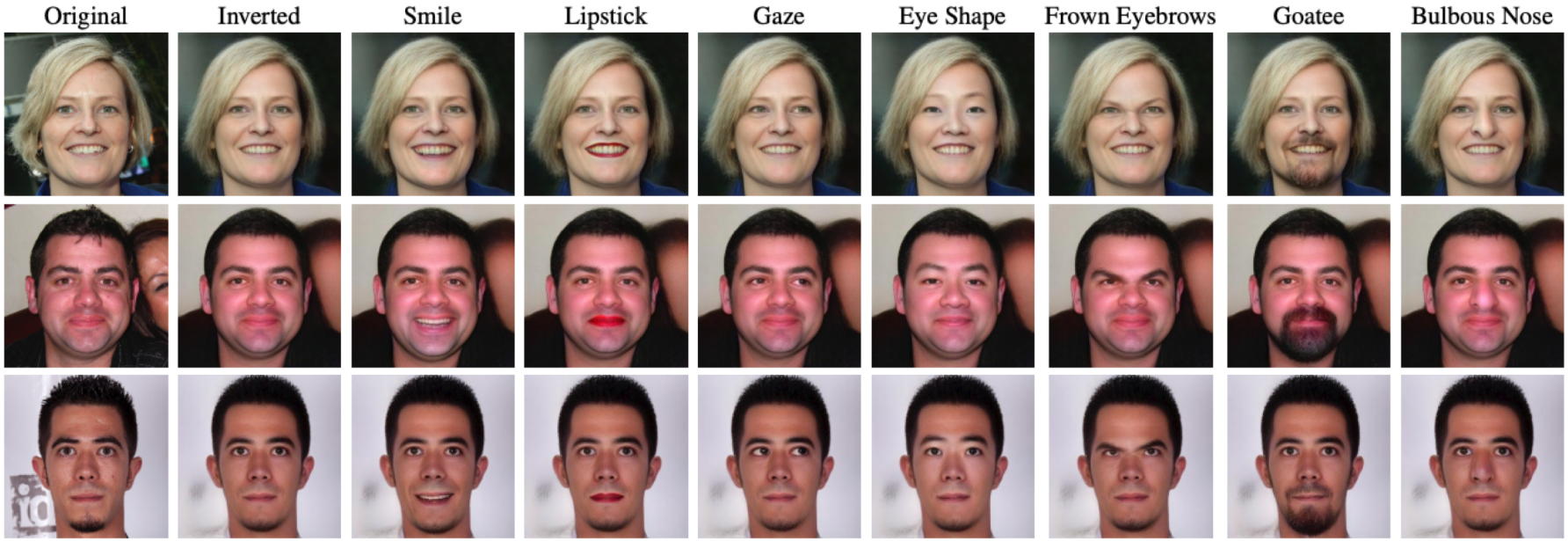
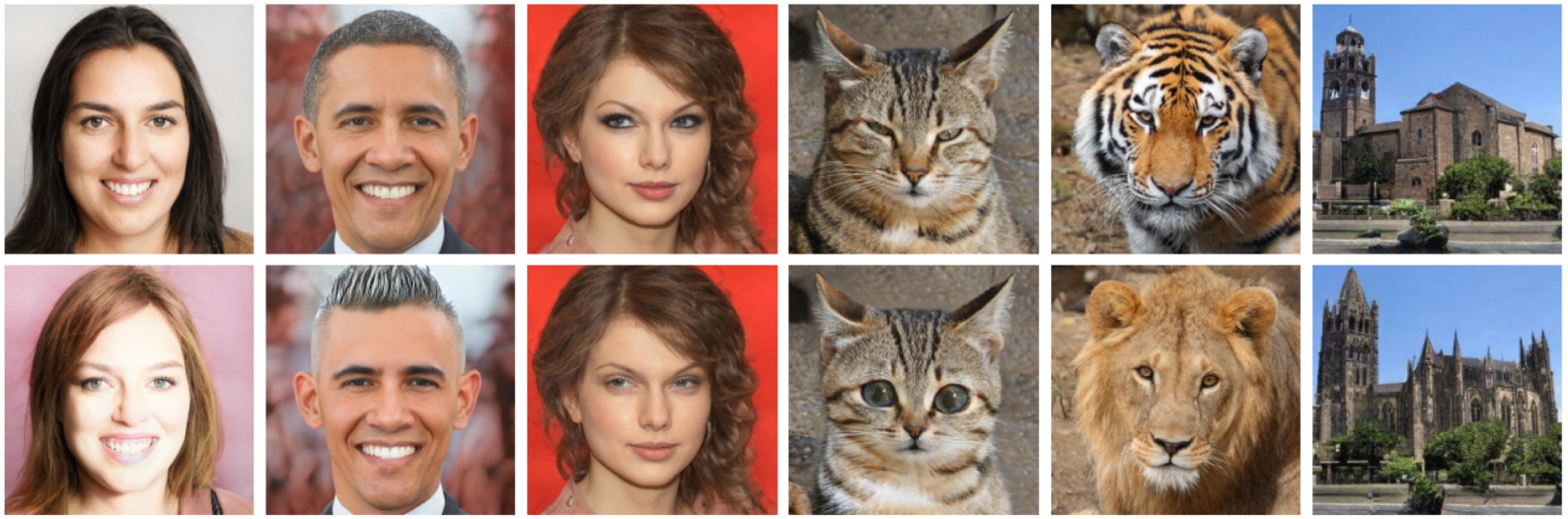


Figure 7. Manipulation of real images using encoder-based inversion. Original images are from FFHQ, and were not part of the encoder's training set.

Z. Wu et al. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. CVPR 2021

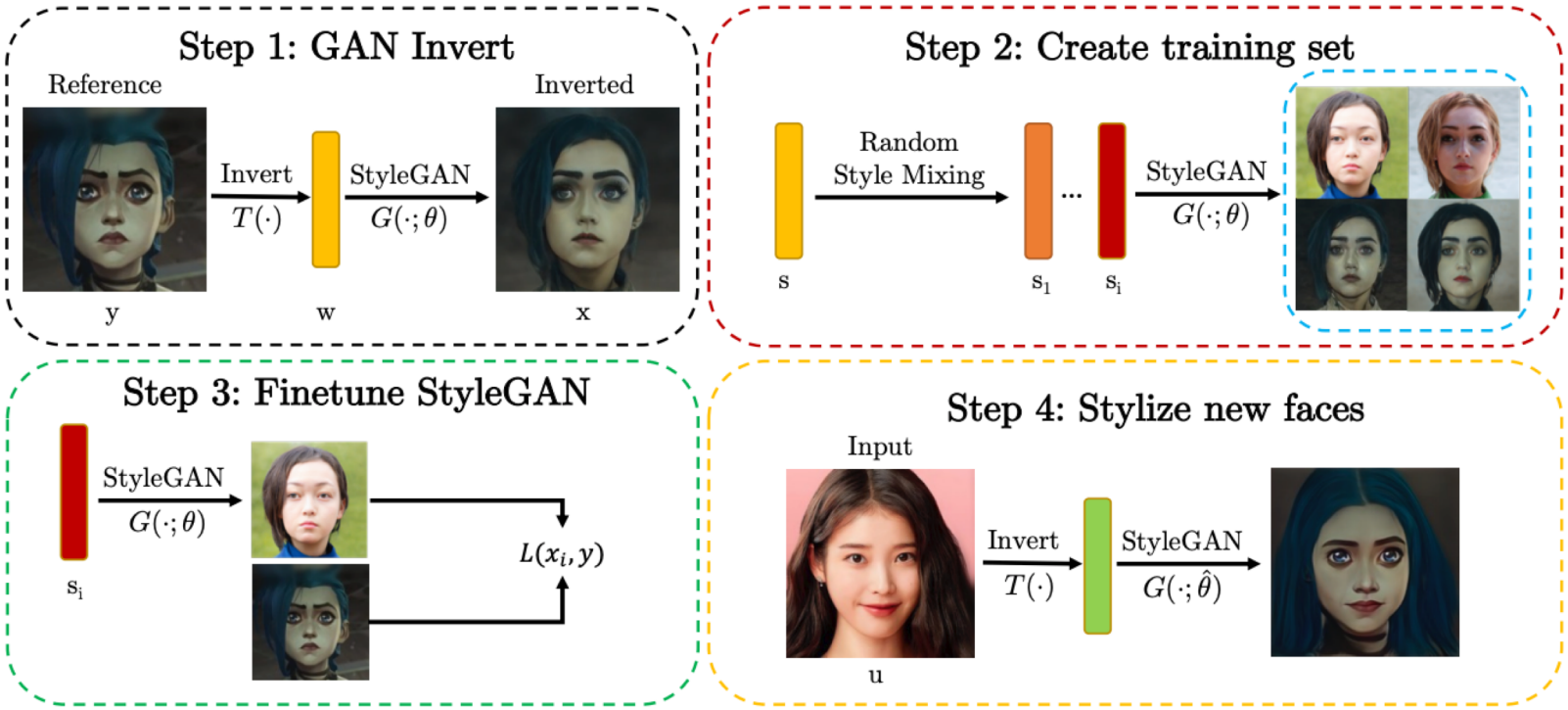
Editing using text prompts

- Combine powerful recent image-text embedding technique (CLIP) with pre-trained StyleGAN for text-based image editing



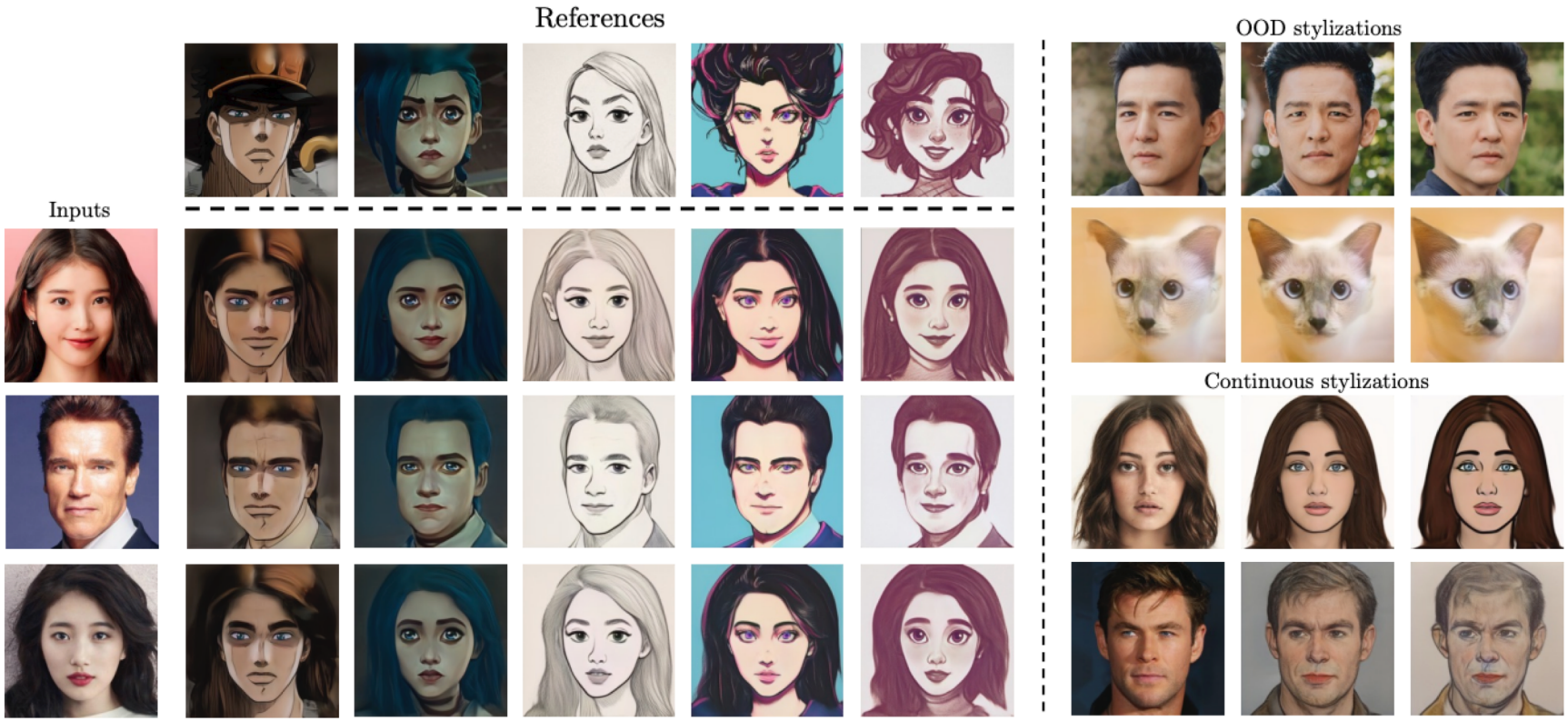
O. Patashnik et al.. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. ICCV 2021

Stylization by GAN fine-tuning



M. J. Chong and D. Forsyth. JoJoGAN: One Shot Face Stylization. ECCV 2022

Stylization by GAN fine-tuning



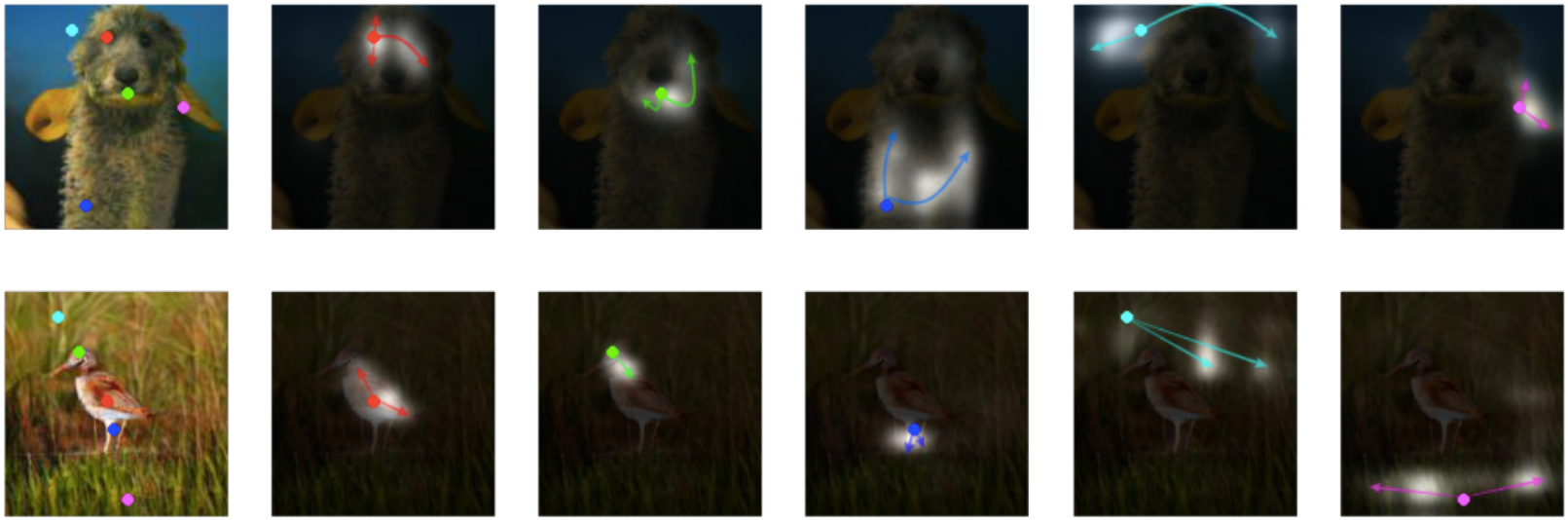
M. J. Chong and D. Forsyth. JoJoGAN: One Shot Face Stylization. ECCV 2022

Outline

- Progressive GAN
- StyleGAN
- GAN-based image editing applications
- Self-attention GAN, BigGAN

Self-attention GAN

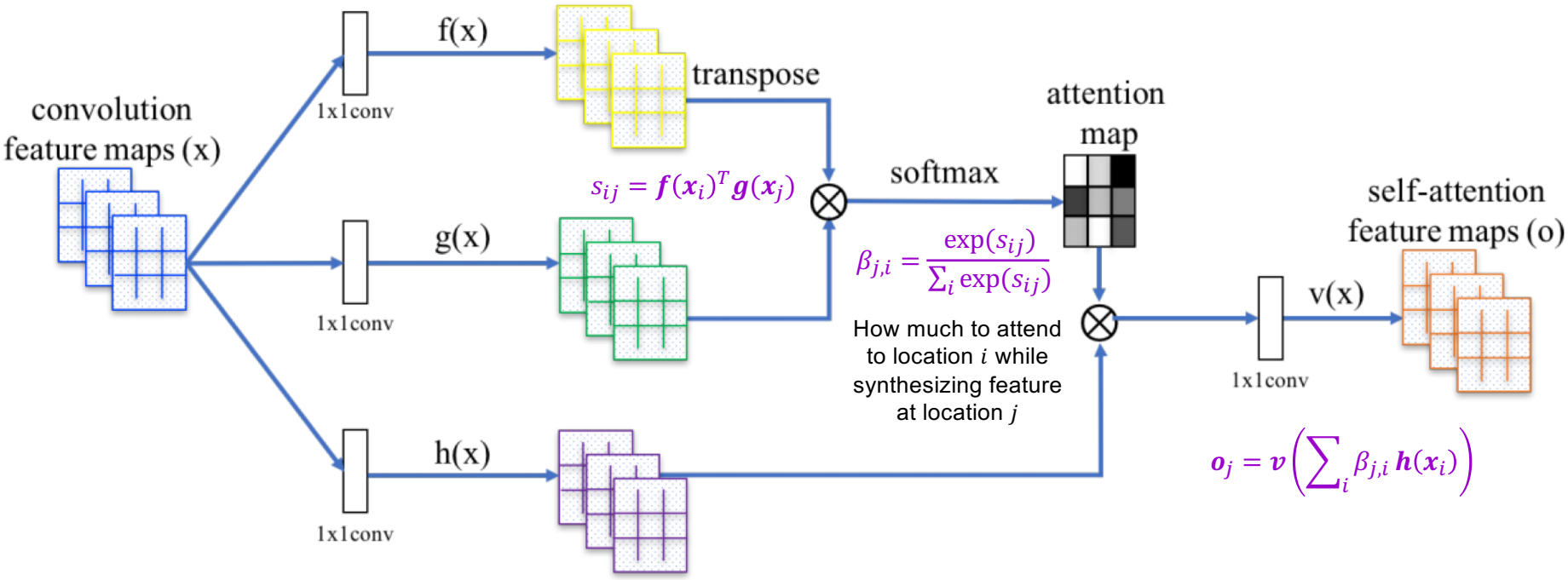
- Adaptive receptive fields to capture non-local structure



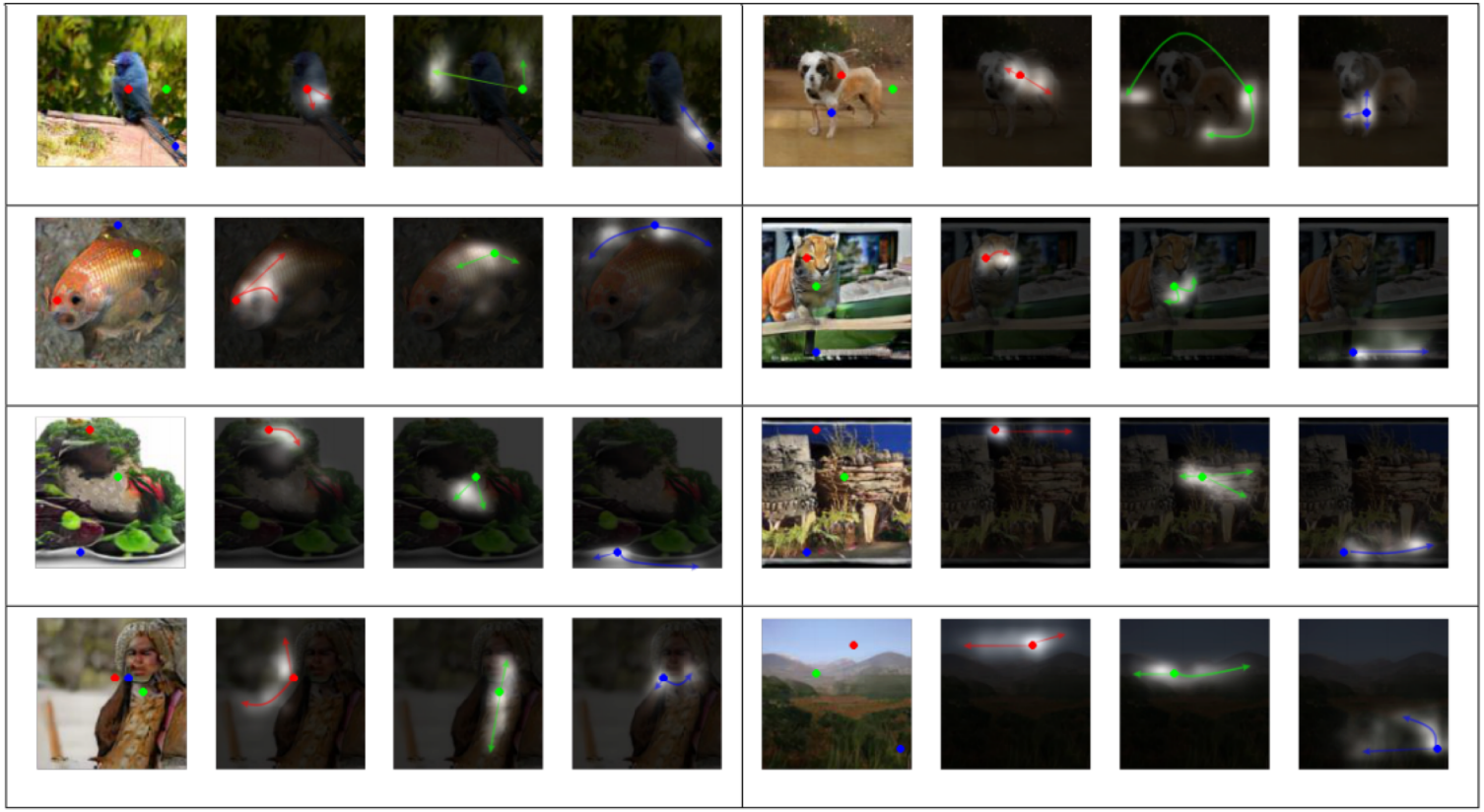
H. Zhang, I. Goodfellow, D. Metaxas, A. Odena. Self-Attention Generative Adversarial Networks. ICML 2019

Self-attention GAN

- Adaptive receptive fields to capture non-local structure (based on Wang et al., 2018)

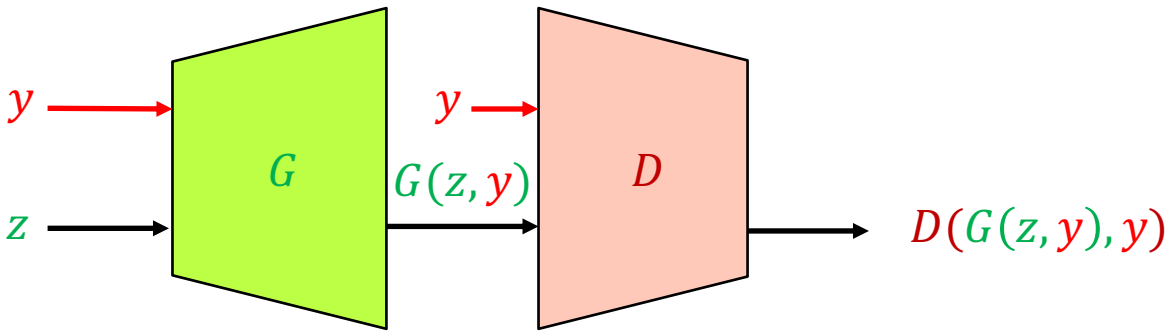


Attention map visualization



Self-attention GAN: Implementation details

- Both generator and discriminator are conditioned on class label y



Self-attention GAN: Implementation details

- Both generator and discriminator are conditioned on class label y
  - Conditioning the discriminator: *projection* (Miyato & Koyama, 2018)
  - Conditioning the generator: *conditional batch norm*

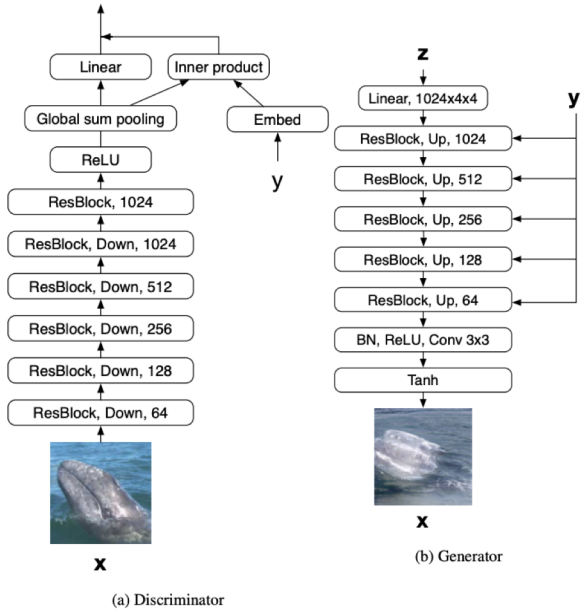


Figure source

Self-attention GAN: Implementation details

- Both generator and discriminator are conditioned on class label y
- Hinge loss formulation
  - Discriminator: Drive discriminator score on real data above 1, on generated data below -1

$$L_D = -\mathbb{E}_{(x,y)\sim p_{\text{data}}}[\min(0, D(x,y) - 1)]$$
$$-\mathbb{E}_{z\sim p_z, y\sim p_{\text{data}}}[\min(0, -D(G(z,y),y) - 1)]$$

  - Generator: maximize discriminator score on generated data

$$L_G = -\mathbb{E}_{z\sim p_z, y\sim p_{\text{data}}} D(G(z,y),y)$$

Self-attention GAN: Implementation details

- Both generator and discriminator are conditioned on class label y
- Hinge loss formulation
- *Spectral normalization* for generator and discriminator (Miyato et al., 2018) – divide weight matrices by largest singular value (estimated)
- Different learning rates for generator and discriminator (TTUR or two-timescale update rule – Heusel et al., 2017)

Self-attention GAN: Results

- 128 x 128 ImageNet

goldfish

indigo bunting

redshank

Saint Bernard

BigGAN

- Scale up SA-GAN to generate ImageNet images up to 512 x 512 resolution

A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, ICLR 2019

BigGAN: Implementation details

- 8x larger batch size, 50% more channels (2x more parameters) than baseline SA-GAN
- Hierarchical latent space: feed (transformations of) z vector into multiple layers of the generator

BigGAN: Implementation details

- 8x larger batch size, 50% more channels (2x more parameters) than baseline SA-GAN
- Hierarchical latent space: feed (transformations of) z vector into multiple layers of the generator
- Truncation trick: at test time, resample the components of the z vector whose magnitude falls above a certain threshold
  - Trade off diversity for image quality

"The effects of increasing truncation. From left to right, the threshold is set to 2, 1, 0.5, 0.04."

BigGAN: Implementation details

- 8x larger batch size, 50% more channels (2x more parameters) than baseline SA-GAN
- Hierarchical latent space: feed (transformations of) z vector into multiple layers of the generator
- Truncation trick: at test time, resample the components of the z vector whose magnitude falls above a certain threshold
- Lots of other tricks (initialization, training, etc.)
- Training observed to be unstable, but good results are achieved “just before collapse”
- Evidence that discriminator memorizes the training data, but the generator doesn’t

BigGAN: Implementation details

https://xkcd.com/1838/

Announcements and reminders

- **Assignment 3** was due **yesterday**
- **Assignment 4** is out, due **Wednesday, April 19**
- **Quiz 3** will be out **9AM tomorrow** through **9AM Monday, April 10**
- **Project progress reports** are due **next Friday, April 14**

Last time: Advanced GAN models and applications

- Progressive GAN
- StyleGAN
- GAN-based image editing applications
- Self-attention GAN, BigGAN
- StyleGAN-XL, GigaGAN, StyleGAN-T

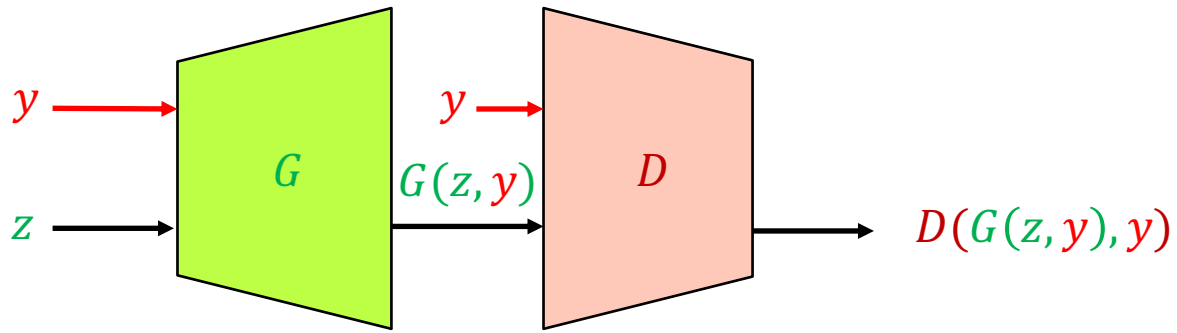
BigGAN

- Scale up SA-GAN to generate ImageNet images up to 512 x 512 resolution

A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, ICLR 2019

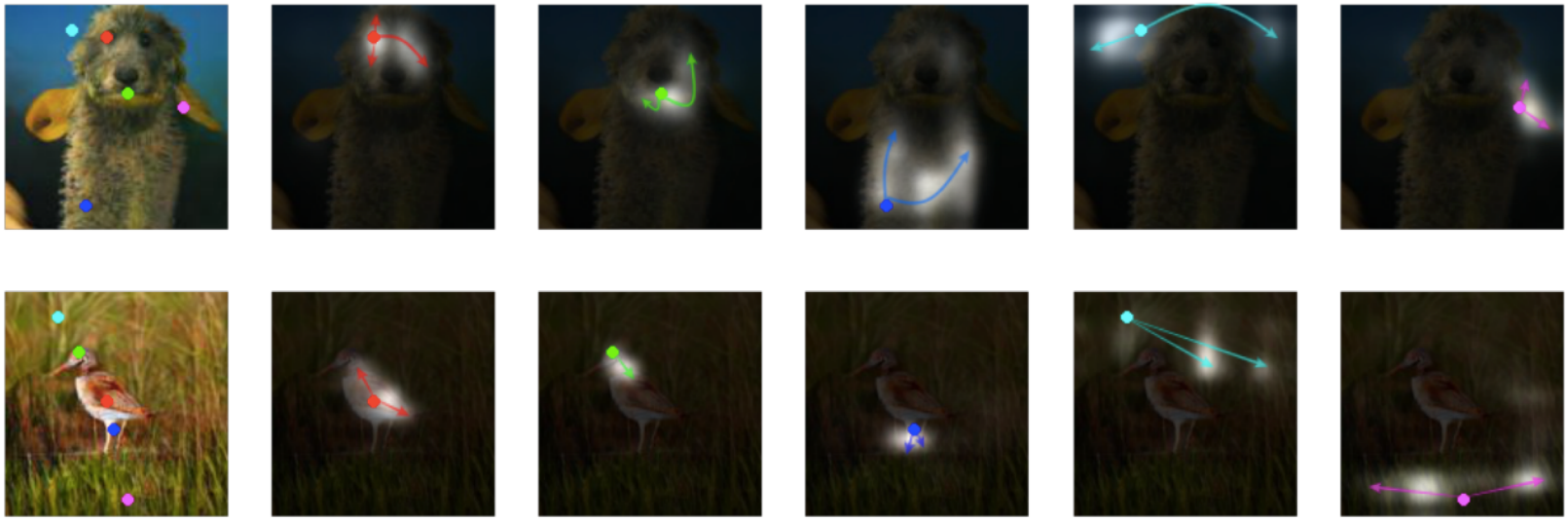
Review: Class-conditional generation

- Both generator and discriminator are conditioned on class label y



Review: Self-attention GAN

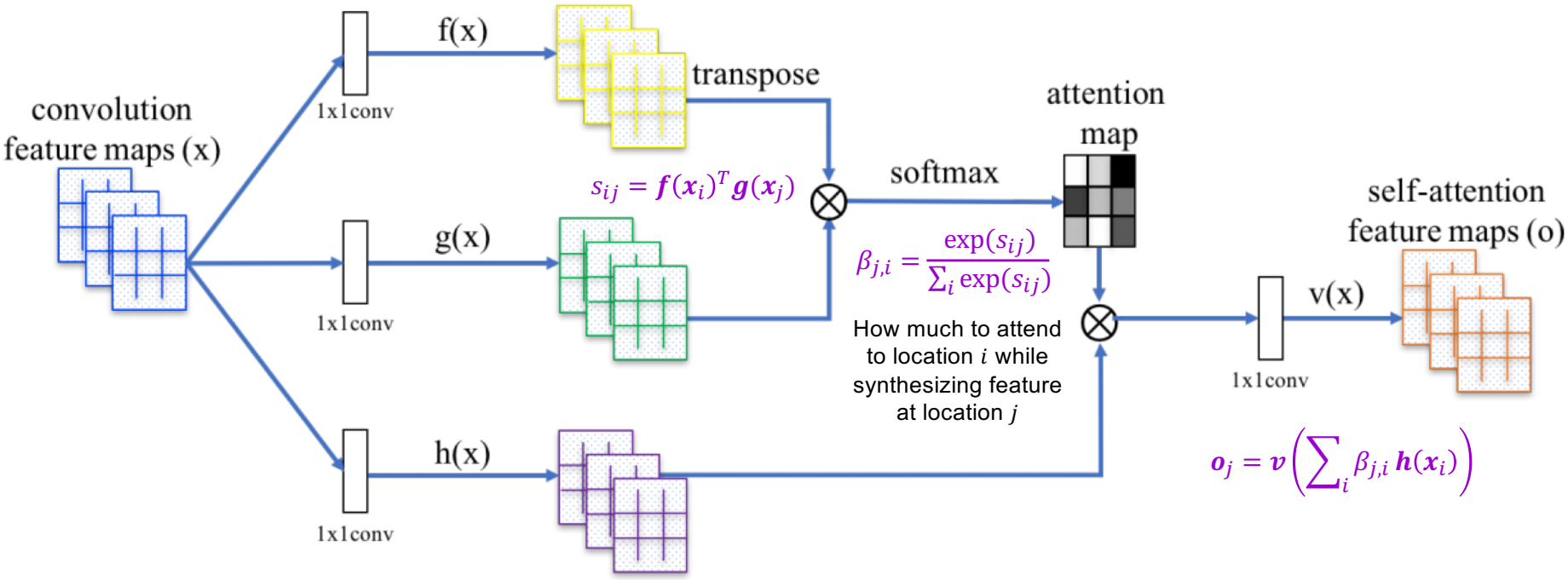
- Adaptive receptive fields to capture non-local structure



H. Zhang, I. Goodfellow, D. Metaxas, A. Odena. Self-Attention Generative Adversarial Networks. ICML 2019

Review: Self-attention GAN

- Adaptive receptive fields to capture non-local structure (based on Wang et al., 2018)



BigGAN: Results

- Samples at 256 x 256 resolution:

BigGAN: Results

- Samples at 512 x 512 resolution:

BigGAN: Results

- Truncation trick: at test time, resample the components of the z vector whose magnitude falls above a certain threshold
  - Trade off diversity for image quality

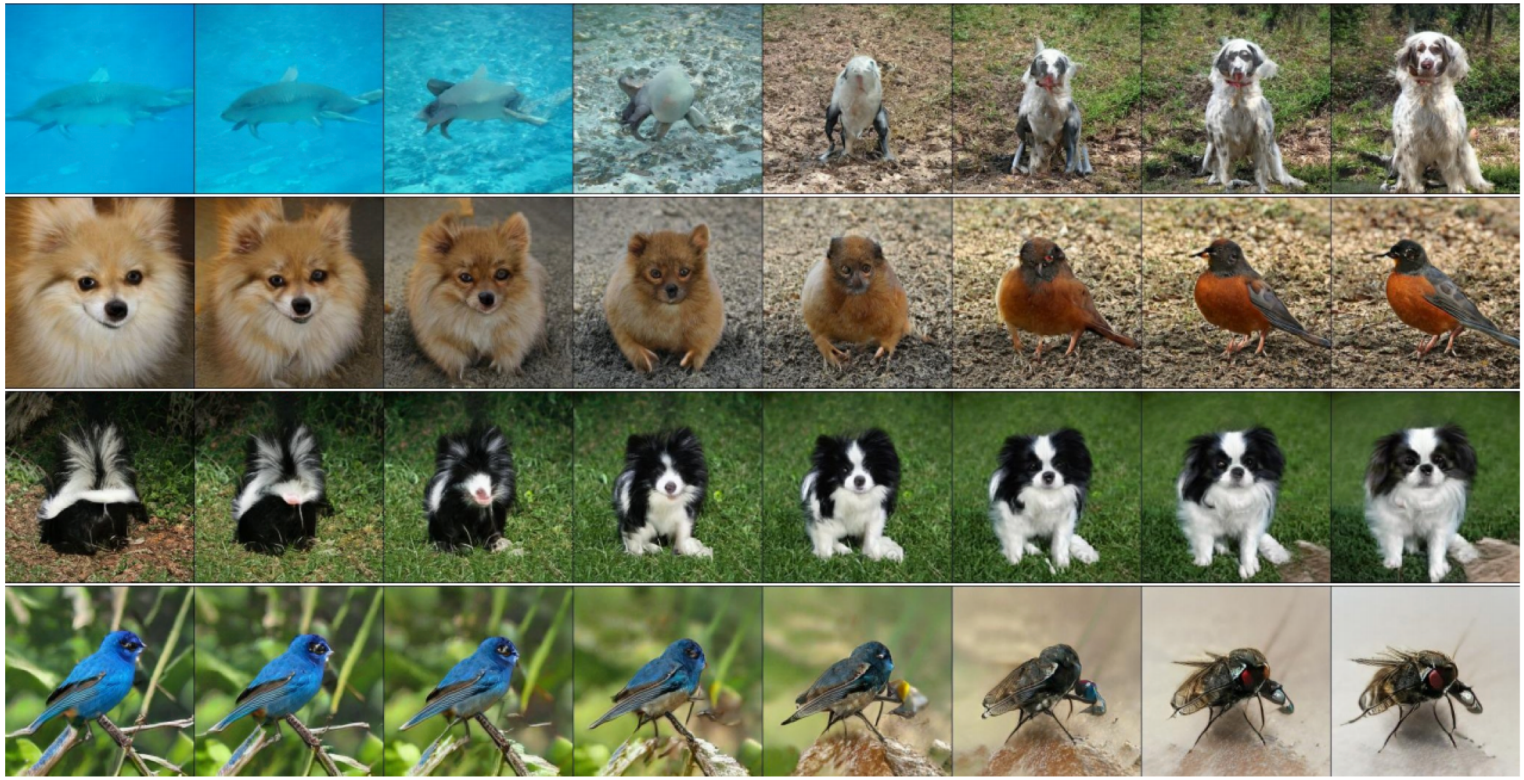
"The effects of increasing truncation. From left to right, the threshold is set to 2, 1, 0.5, 0.04."

BigGAN: Results

- Interpolation between c with z held constant:

BigGAN: Results

- Interpolation between c, z pairs:



BigGAN: Results

- Difficult classes:

Outline

- Progressive GAN
- StyleGAN
- GAN-based image editing applications
- Self-attention GAN, BigGAN
- StyleGAN-XL, GigaGAN, StyleGAN-T

StyleGAN-XL

Fig. 1. Class-conditional samples generated by StyleGAN3 (left) and StyleGAN-XL (right) trained on ImageNet at resolution 256^2.

A. Sauer et al. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. SIGGRAPH 2022

Disclaimer

https://xkcd.com/1838/

StyleGAN-XL: Details

- Generator based on StyleGAN3 with progressive growing and 3x more parameters

StyleGAN-XL: Details

- Generator based on StyleGAN3 with progressive growing and 3x more parameters
- Discriminator based on *projected GANs*
  - Recall the standard GAN objective:

$$V(G,D) = \mathbb{E}_{x\sim p_{\text{data}}} \log D(x) + \mathbb{E}_{z\sim p} \log(1 - D(G(z)))$$

  - Projected GAN objective introduces multiple discriminators operating on *projections* P_l from a fixed pre-trained feature space:

$$V(G,D) = \sum_l \left(\mathbb{E}_{x\sim p_{\text{data}}} \log D_l(P_l(x)) + \mathbb{E}_{z\sim p} \log(1 - D_l(P_l(G(z)))) \right)$$

  - Each P_l returns a feature map of a different resolution by applying random *cross-channel mixing* and *cross-scale mixing* to a pre-trained network

A. Sauer et al. Projected GANs Converge Faster. NeurIPS 2021

StyleGAN-XL: Details

Cross-channel mixing (CCM)

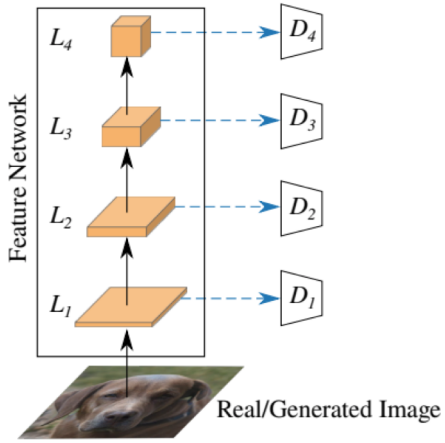


Figure 2: **CCM** (dashed blue arrows) employs 1×1 convolutions with random weights.

Cross-scale mixing (CSM)

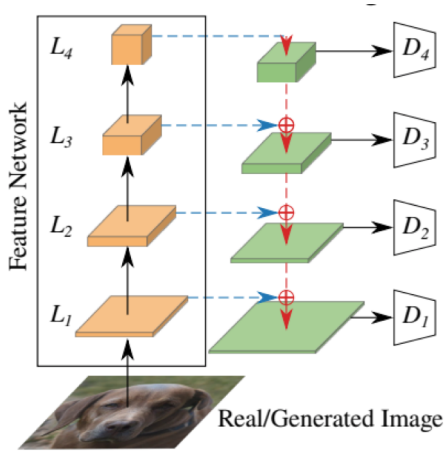


Figure 3: **CSM** (dashed red arrows) adds random 3×3 convolutions and bilinear upsampling, yielding a U-Network.

A. Sauer et al. Projected GANs Converge Faster. NeurIPS 2021

StyleGAN-XL: Details

- Generator based on StyleGAN3 with progressive growing and 3x more parameters
- Discriminator based on *projected GANs*
- Both generator and discriminator are conditioned on pre-trained *class embedding vectors*
- *Classifier guidance*: add cross-entropy loss of pre-trained classifier to the generator objective to encourage output of classifier to have high probability for correct class

StyleGAN-XL: Details

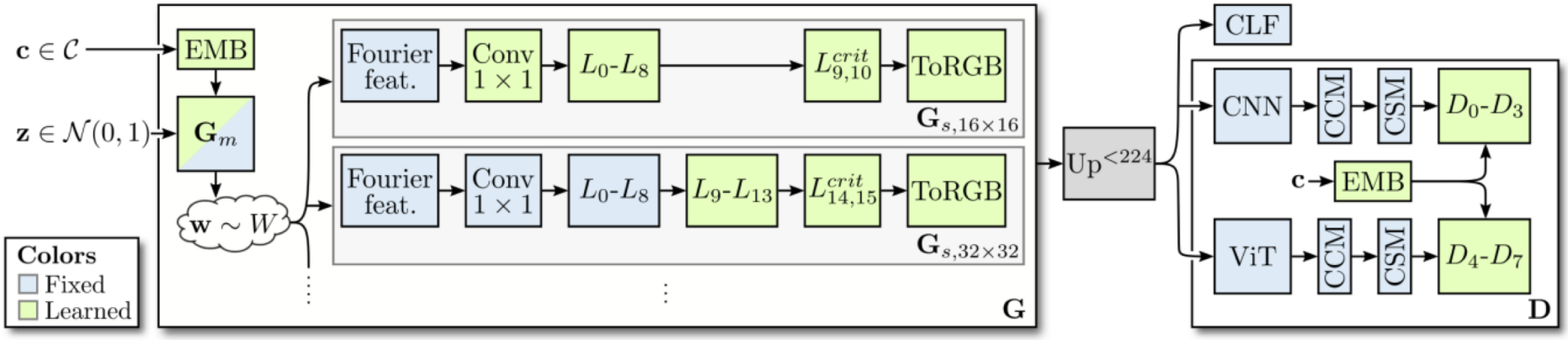


Fig. 2. **Training StyleGAN-XL**. We feed a latent code $\mathbf{z}$ and class label $\mathbf{c}$ to the pretrained embedding and the mapping network $\mathbf{G}_m$ to generate style codes $\mathbf{w}$. The codes modulate the convolutions of the synthesis network $\mathbf{G}_s$. During training, we gradually add layers to double the output resolution for each stage of the progressive growing schedule. We only train the latest layers while keeping the others fixed. $\mathbf{G}_m$ is only trained for the initial 16^2 stage and remains fixed for the higher-resolution stages. The synthesized image is upsampled when smaller than 224^2 and passed through a CNN and a ViT and respective feature mixing blocks (CCM+CSM). At higher resolutions, the CNN receives the unaltered image while the ViT receives a downsampled input to keep memory requirements low but still utilize its global feedback. Finally, we apply eight independent discriminators on the resulting multi-scale feature maps. The image is also fed to classifier CLF for classifier guidance.

StyleGAN-XL: Results

Model	FID ↓	sFID ↓	rFID ↓	IS ↑	Pr ↑	Rec ↑
Resolution 128^2						
BigGAN	6.02	7.18	6.09	145.83	**0.86**	0.35
CDM	3.52	128.80		128.80		
ADM	5.91	5.09	13.29	93.31	0.70	**0.65**
ADM-G	2.97	5.09	3.80	141.37	0.78	0.59
StyleGAN-XL	**1.81**	**3.82**	**1.82**	**200.55**	0.77	0.55
Resolution 512^2						
BigGAN	8.43	8.13	312.00	177.90	**0.88**	0.29
ADM	23.24	10.19	561.32	58.06	0.73	**0.60**
ADM-G-U	3.85	5.86	210.83	221.72	0.84	0.53
StyleGAN-XL	**2.41**	**4.06**	**51.54**	**267.75**	0.77	0.52

Model	FID ↓	sFID ↓	rFID ↓	IS ↑	Pr ↑	Rec ↑
Resolution 256^2						
StyleGAN2	49.20					
BigGAN	6.95	7.36	75.24	202.65	**0.87**	0.28
CDM	4.88	158.70		158.70		
ADM	10.94	6.02	125.78	100.98	0.69	**0.63**
ADM-G-U	3.94	6.14	11.86	215.84	0.83	0.53
StyleGAN-XL	**2.30**	**4.02**	**7.06**	**265.12**	0.78	0.53
Resolution 1024^2						
StyleGAN-XL	**2.52**	**4.12**	**413.12**	**260.14**	**0.76**	**0.51**

16^2 32^2 64^2 128^2 256^2 512^2 1024^2

Text-to-image synthesis: GigaGAN

A portrait of a human growing colorful flowers from her hair. Hyperrealistic oil painting. Intricate details.

A golden luxury motorcycle parked at the King's palace. 35mm f/4.5.

a cute magical flying maltipoo at light speed, fantasy concept art, bokeh, wide sky

M. Kang et al. Scaling up GANs for Text-to-Image Synthesis. CVPR 2023

GigaGAN: Generator

- Goal: catch up to diffusion models by training a GAN for text-to-image generation on 2B images from the LAION dataset
- 1B parameters, 6x larger than StyleGAN-XL, but smaller than diffusion models (Imagen: 3B, DALL-E 2: 5.5B, Parti: 20B)
- First generate at 64x64 resolution, then upsample to 512x512 using a separate network
- Architecture based on StyleGAN2, but with self-attention layers and sample-adaptive convolution kernel selection

GigaGAN: Generator

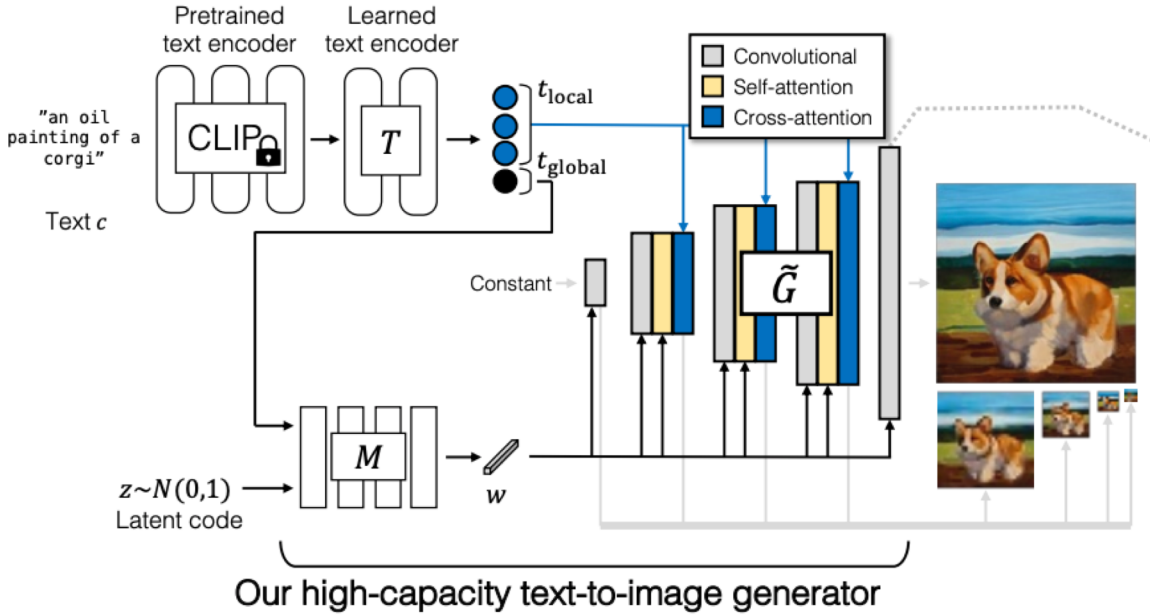
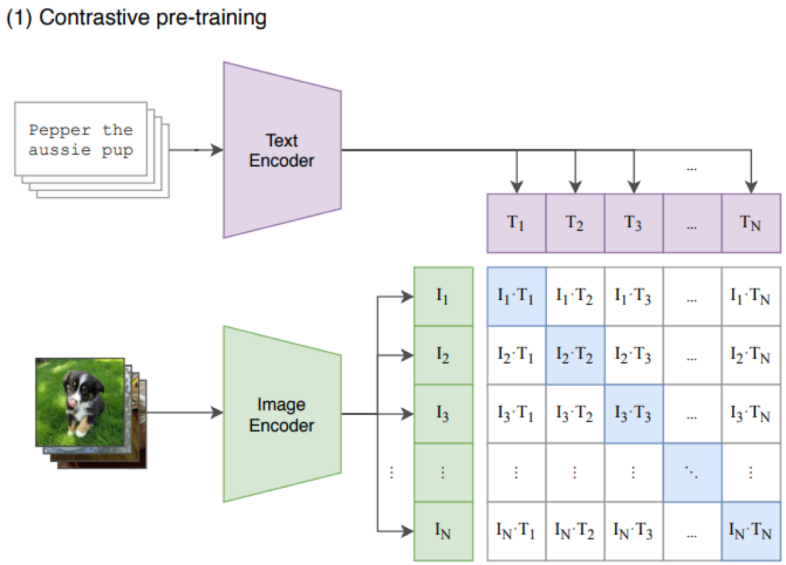


Figure 4. **Our GigaGAN high-capacity text-to-image generator.** First, we extract text embeddings using a pretrained CLIP model and a learned encoder T. The local text descriptors are fed to the generator using cross-attention. The global text descriptor, along with a latent code $\mathbf{z}$, is fed to a style mapping network M to produce style code $\mathbf{w}$. The style code modulates the main generator using our style-adaptive kernel selection, shown on the right. The generator outputs an image pyramid by converting the intermediate features into RGB images. To achieve higher capacity, we use multiple attention and convolution layers at each scale (Appendix A2). We also use a separate upsampler model, which is not shown in this diagram.

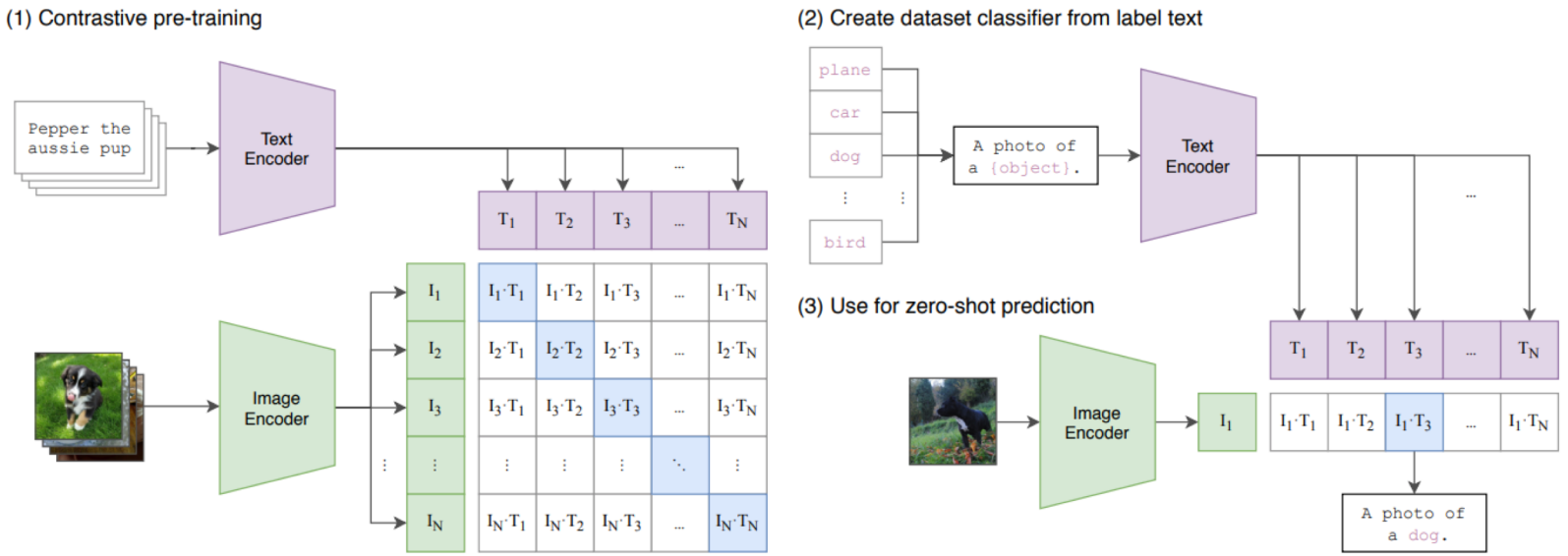
CLIP: Contrastive Language-Image Pretraining



Contrastive objective: in a batch of N image-text pairs, classify each text string to the correct image and vice versa

A. Radford et al., Learning Transferable Visual Models From Natural Language Supervision, ICML 2021
https://openai.com/blog/clip/

CLIP: Contrastive Language-Image Pretraining



A. Radford et al., [Learning Transferable Visual Models From Natural Language Supervision](), ICML 2021
https://openai.com/blog/clip/

GigaGAN: Generator

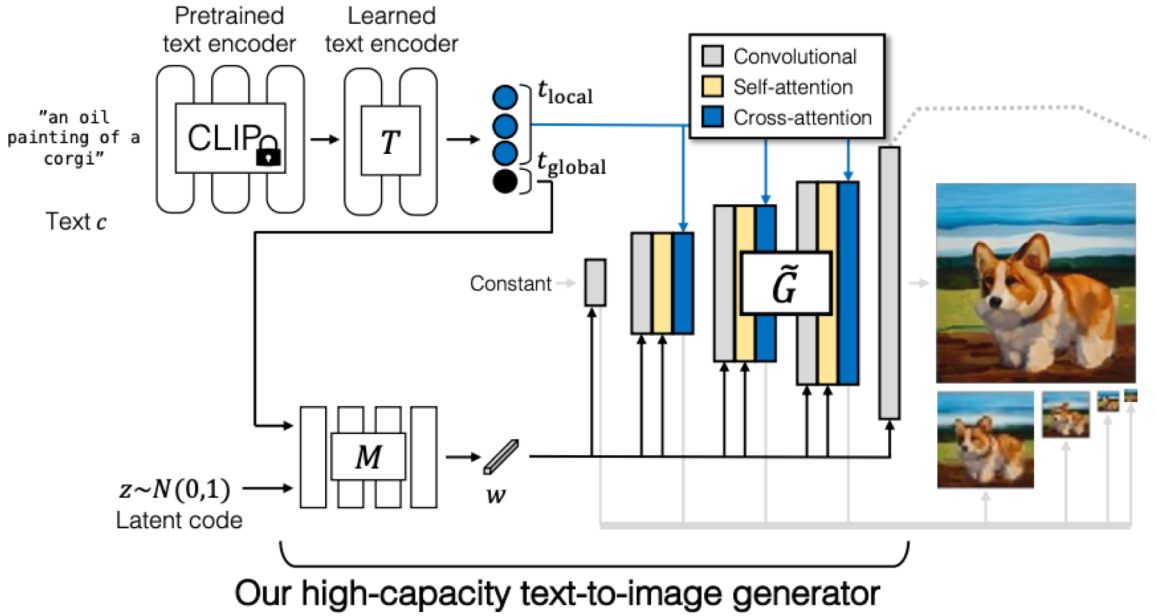


Figure 4. **Our GigaGAN high-capacity text-to-image generator.** First, we extract text embeddings using a pretrained CLIP model and a learned encoder T. The local text descriptors are fed to the generator using cross-attention. The global text descriptor, along with a latent code $\mathbf{z}$, is fed to a style mapping network M to produce style code $\mathbf{w}$. The style code modulates the main generator using our style-adaptive kernel selection, shown on the right. The generator outputs an image pyramid by converting the intermediate features into RGB images. To achieve higher capacity, we use multiple attention and convolution layers at each scale (Appendix A2). We also use a separate upsampler model, which is not shown in this diagram.

GigaGAN: Discriminator

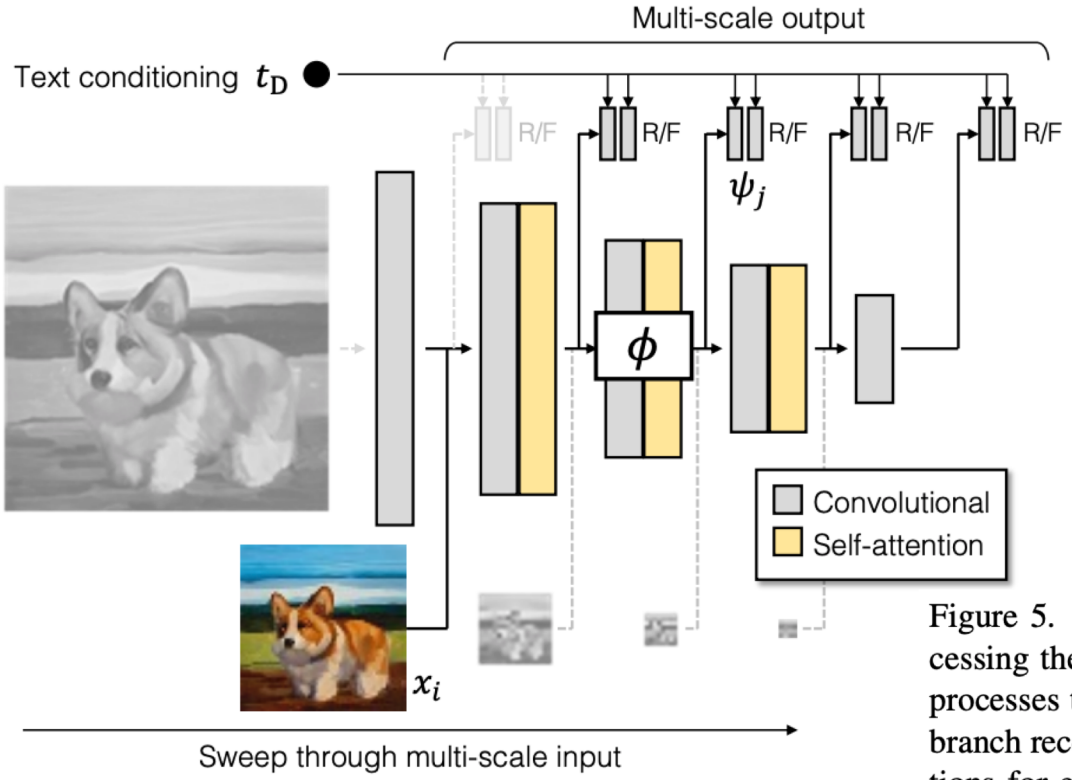


Figure 5. **Our discriminator** consists of two branches for processing the image and the text conditioning t_D. The text branch processes the text similar to the generator (Figure 4). The image branch receives an image pyramid and makes independent predictions for each image scale. Moreover, the predictions are made at all subsequent scales of the downsampling layers, making it a *multi-scale input, multi-scale output* (MS-I/O) discriminator.

GigaGAN: Discriminator

- Multiple loss terms:
  - Standard adversarial loss (NSGAN)
  - *Matching-aware loss* where image is paired with random conditioning vector and discriminator is encouraged to label the pair as “fake”
  - CLIP contrastive loss: enforce high image-text similarity using pre-trained image and text encoders
  - “Vision-aided loss” similar to projected GAN

GigaGAN: Style mixing

Figure 6. **Style mixing**. Our GAN-based architecture retains a disentangled latent space, enabling us to blend the coarse style of one sample with the fine style of another. All outputs are generated with the prompt "A Toy sport sedan, CG art." The corresponding latent codes are spliced together to produce a style-swapping grid.

GigaGAN: Prompt interpolation

Figure 7. **Prompt interpolation**. GigaGAN enables smooth interpolation between prompts, as shown in the interpolation grid. The four corners are generated from the same latent $\mathbf{z}$ but with different text prompts. The corresponding text embeddings $\mathbf{t}$ and style vectors $\mathbf{w}$ are interpolated to create a smooth transition. The same $\mathbf{z}$ results in similar layouts. See Figure 8 for more precise control.

GigaGAN: Prompt mixing

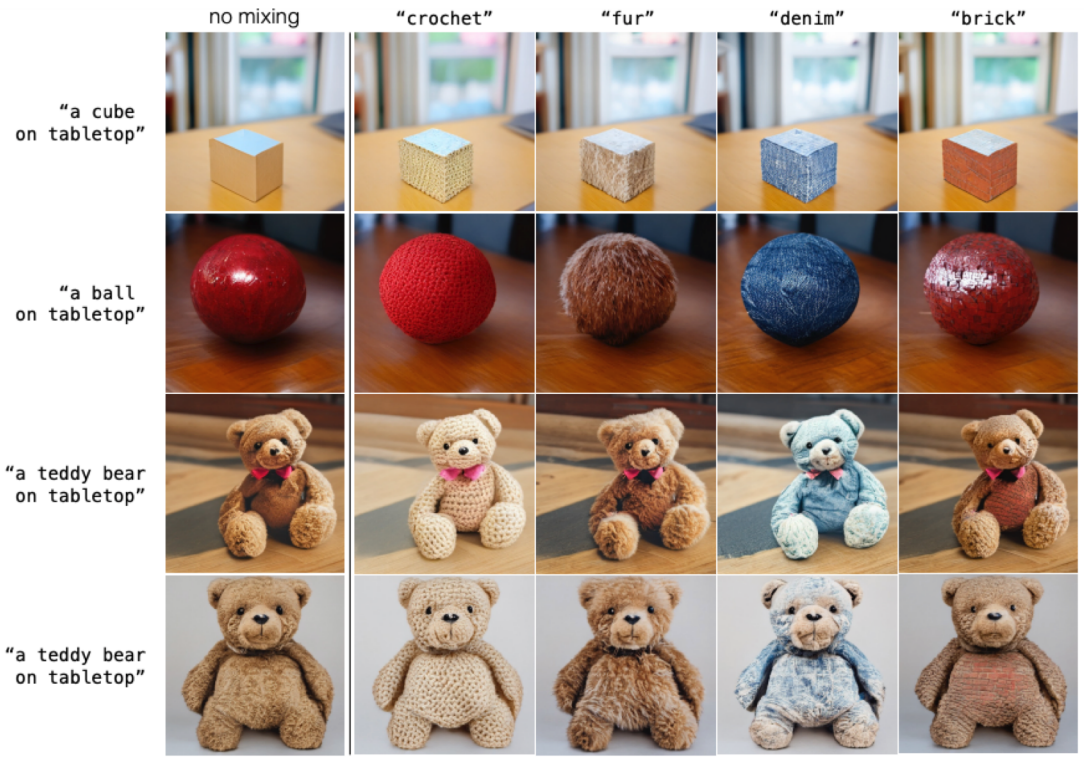
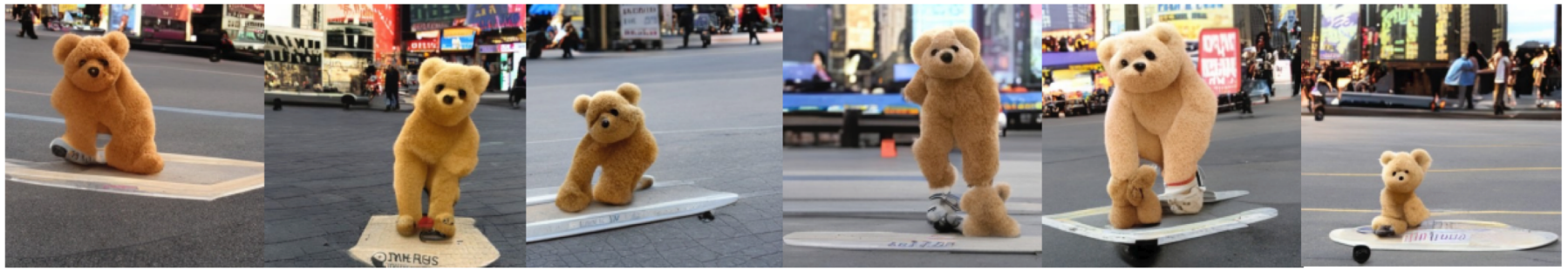


Figure 8. **Prompt mixing**. GigaGAN retains a disentangled latent space, enabling us to combine the coarse style of one sample with the fine style of another. Moreover, GigaGAN can directly control the style with text prompts. Here we generate four outputs using the prompts "a X on tabletop", shown in the "no mixing" column. Then we re-compute the text embeddings $\mathbf{t}$ and the style codes $\mathbf{w}$ using the new prompts "a X with the texture of Y on tabletop", such as "a cube with the texture of crochet on tabletop", and apply them to the second half layers of the generator, achieving layout-preserving fine style control. Cross-attention mechanism automatically localizes the style to the object of interest.

GigaGAN: Comparison with diffusion models

"A teddy bear on a skateboard in times square."

Ours (512px, 0.13s / img)



Ours (512px, 0.14s / img, truncation $\psi = 0.8$)

GigaGAN: Comparison with diffusion models

Stable Diffusion v1.5 (512px, 2.9s / img, 50 steps, guidance=7.5)

DALL·E 2 (1024px)

GigaGAN: Comparison with diffusion models

"Vibrant portrait painting of Salvador Dalí with a robotic half face."

Ours (512px, 0.13s / img)

Ours (512px, 0.14s / img, truncation $\psi = 0.8$)

GigaGAN: Comparison with diffusion models

Stable Diffusion v1.5 (512px, 2.9s / img, 50 steps, guidance=7.5)

DALL·E 2 (1024px)

GigaGAN: Quantitative evaluation

	Image quality measure	Image-text coherence measure	
Model	FID-10k ↓	CLIP Score ↑	# Param.
StyleGAN2	29.91	0.222	27.8M
+ Larger (5.7×)	34.07	0.223	158.9M
+ Tuned	28.11	0.228	26.2M
+ Attention	23.87	0.235	59.0M
+ Matching-aware D	27.29	0.250	59.0M
+ Matching-aware G and D	21.66	0.254	59.0M
+ Adaptive convolution	19.97	0.261	80.2M
+ Deeper	19.18	0.263	161.9M
+ CLIP loss	14.88	0.280	161.9M
+ Multi-scale training	14.92	0.300	164.0M
+ Vision-aided GAN	13.67	0.287	164.0M
+ Scale-up (**GigaGAN**)	9.18	0.307	652.5M

GigaGAN: Quantitative evaluation

Table 2. **Comparison to recent text-to-image models**. Model size, GPU days, total images seen during training, COCO FID-30k, and inference speed of text-image models. $*$ denotes that the model has been evaluated by us. GigaGAN achieves a lower FID than DALL·E 2 [74], Stable Diffusion [78], and Parti-750M [101], while being much faster compared to recent competitive methods.

	Model	Type	# Param.	# Images	FID-30k ↓	Inf. time
256	DALL·E [75]	Diff	12.0B	1.54B	27.50	-
	GLIDE [63]	Diff	5.0B	5.94B	12.24	15.0s
	LDM [79]	Diff	1.5B	0.27B	12.63	9.4s
	DALL·E 2 [74]	Diff	5.5B	5.63B	10.39	-
	Imagen [80]	Diff	3.0B	15.36B	7.27	9.1s
	eDiff-I [5]	Diff	9.1B	11.47B	6.95	32.0s
	Parti-750M [101]	AR	750M	3.69B	10.71	-
	Parti-3B [101]	AR	3.0B	3.69B	8.10	6.4s
	Parti-20B [101]	AR	20.0B	3.69B	7.23	-
	LAFITE [108]	GAN	75M	-	26.94	0.02s
512	SD-v1.5* [78]	Diff	0.9B	3.16B	9.62	2.9s
	Muse-3B [10]	AR	3.0B	0.51B	7.88	1.3s
	GigaGAN	GAN	1.0B	0.98B	9.09	0.13s

"While our model can be optimized to better match the feature distribution of real images than existing models, the quality of the generated images is not necessarily better… We acknowledge that this may represent a corner case of zero-shot FID on COCO2014 dataset and suggest that further research on a better evaluation metric is necessary to improve text-to-image models."

StyleGAN-T

A. Sauer et al. [StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis](). arXiv 2023

StyleGAN-T

- Attempt to extend StyleGAN-XL to text-to-image generation
- Backbone dropped down to StyleGAN2, numerous modifications to incorporate image-text conditioning
- 1B parameter model
- Achieves FID-30K of 13.9 – compared to 9.09 for GigaGAN

StyleGAN-T

"A teddy bear on a skateboard in times square"