Style transfer, GANs

Vlad Shakhuro

4 December 2025

Outline

1. Metrics in image generation

2. Style transfer

3. Unconditional generation with GANs

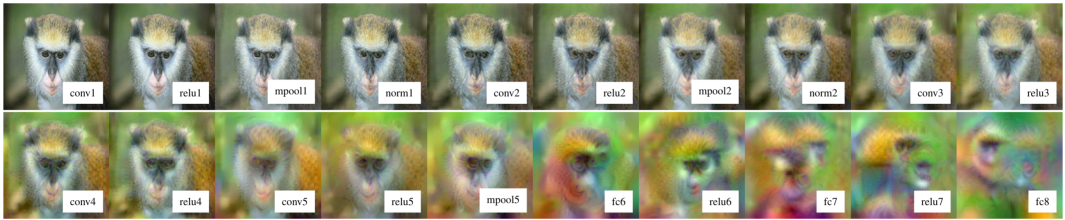
4. Conditional generation with GANs

Reconstructing images from neural features

Initialize x with white noise
Optimize x using following loss:

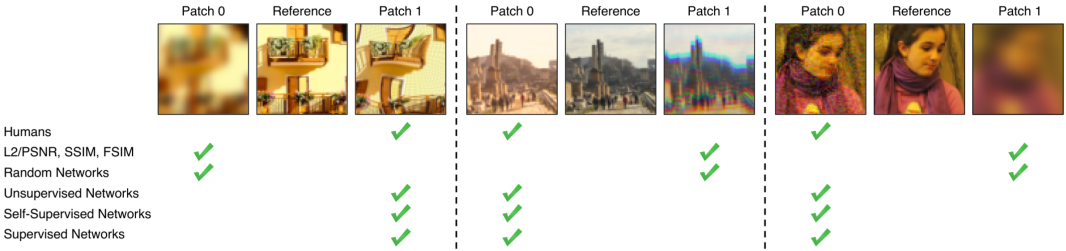
$$x^* = \arg\min_{x \in \mathbb{R}^{H \times W \times C}} \|\Phi(x) - \Phi_0\|^2 + \lambda R(x)$$

$$R(x) = \sum_{i,j} (x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2$$



Mahendran, Vedaldi. Understanding Deep Image Representations by Inverting Them. CVPR 2015

LPIPS: comparing two images



Zhang et al. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. CVPR 2018

Human comparison

Q: Which one is a real artwork?

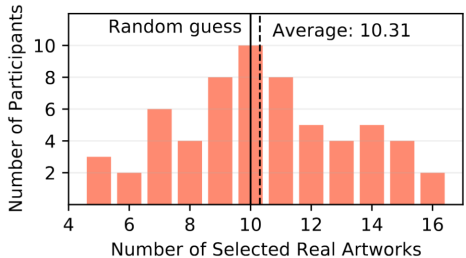


Figure 9. **(Left)** An example of our Artistic Style Transfer Confusion Test. Only 40.6% participants successfully distinguished the real artwork in this example. The answer can be found in our supplementary material. **(Right)** The statistical results with a total of 61 participants, where each participant is asked 20 questions.

Inception score (IS)

For generated image x:

- $p(y \mid x)$ should have low entropy
  (generated image is somehow confidently classified)
- $\int p(y \mid x = G(z))dz$ should have high entropy
  (generated images are varied)

Inception score:

$$\mathrm{IS}(X_s) = \mathop{\mathbb{E}}_{x \in X_s} KL\big(p(y \mid x) \parallel p(y)\big)$$

Label prediction $p(y \mid x)$ is computed using Inception model

Drawback: real images aren't used for computing metric

Salimans et al. Improved Techniques for Training GANs. NIPS 2016

Fréchet Inception Distance (FID)

Assume that image features computed with Inception model have normal distribution. Compute Fréchet (also called Wasserstein-2) distance between two gaussians:

$$\mathrm{FID}(X_r, X_s) = \left\| \mu_{X_r} - \mu_{X_s} \right\|^2 - \mathrm{Tr}\left(\Sigma_{X_r} + \Sigma_{X_s} - 2\sqrt{\Sigma_{X_r}\Sigma_{X_s}} \right)$$

Drawbacks:

1. Inception embeddings aren't normally distributed
2. Estimating (2048×2048)-dimensional covariance matrices from a small sample can lead to large errors
3. Has a bias that depends on the X_s model

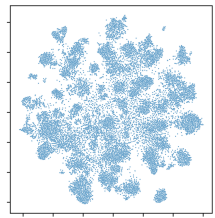


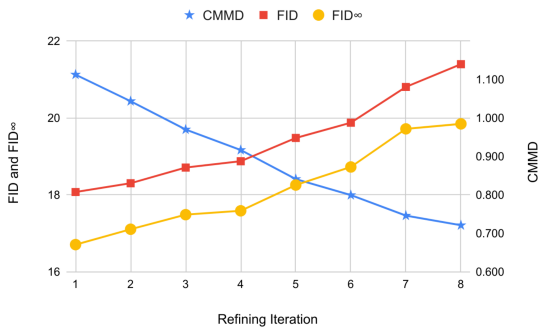
Figure 2. *t-SNE visualization of Inception embeddings of the COCO 30K dataset. Note that even in the reduced-dimensional 2D representation, it is easy to identify that embeddings have multiple modes and do not follow a multivariate normal distribution.*

Heusel et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. NIPS 2017
Chong, Forsyth. Effectively Unbiased FID and Inception Score and where to find them. CVPR 2020

CLIP and Maximum Mean Discrepancy (CMMD)

Use CLIP embeddings (more diverse than Inception embs) and MMD distance:

$$\text{CMMD}(X_r, X_s) = \mathop{\mathbb{E}}_{x_r, x_r'} k(x_r, x_r') + \mathop{\mathbb{E}}_{x_s, x_s'} k(x_s, x_s') - 2 \mathop{\mathbb{E}}_{x_r, x_s} k(x_r, x_s)$$



Jayasumana et al. Rethinking FID: Towards a Better Evaluation Metric for Image Generation. CVPR 2024

Outline

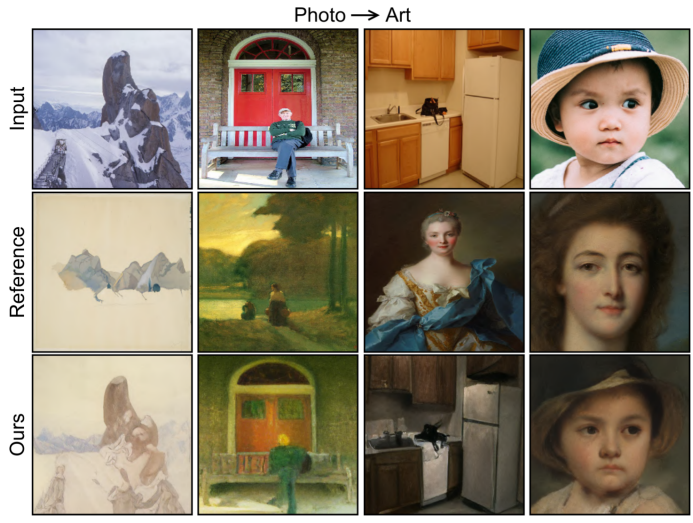
1. Metrics in image generation

2. Style transfer

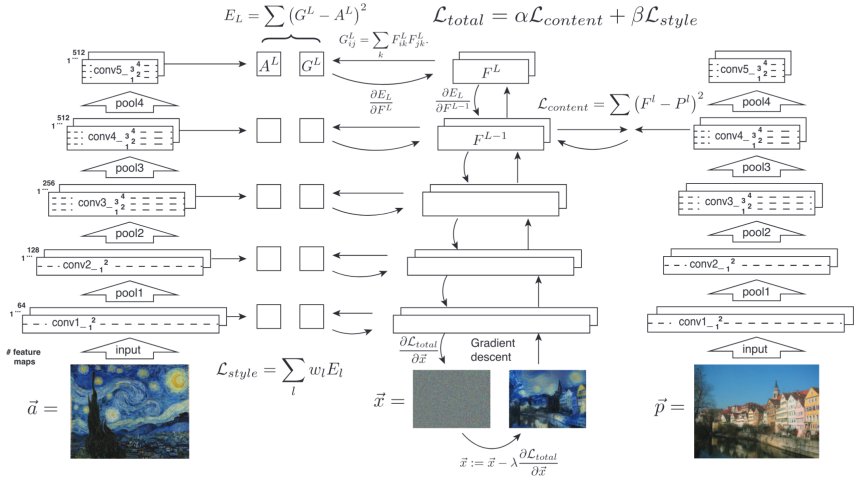
3. Unconditional generation with GANs

4. Conditional generation with GANs

Style transfer

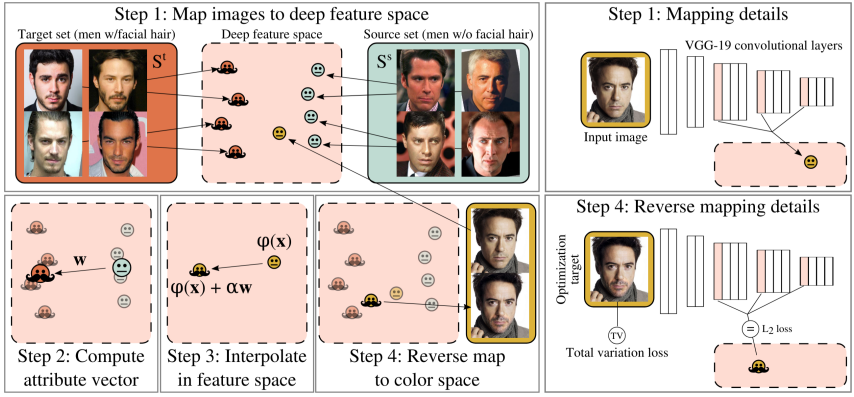


Generating stylized images from noise



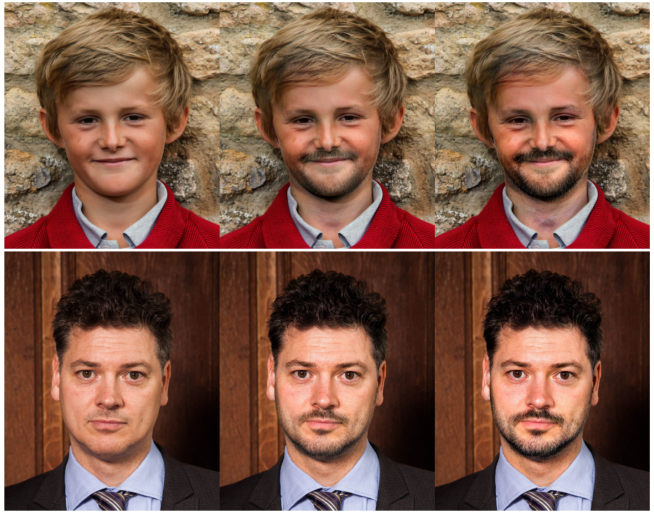
Gatys et al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Deep Feature Interpolation



Upchurch et al. Deep Feature Interpolation for Image Content Changes. CVPR 2017

Deep Feature Interpolation results



Training a neural network for a single style

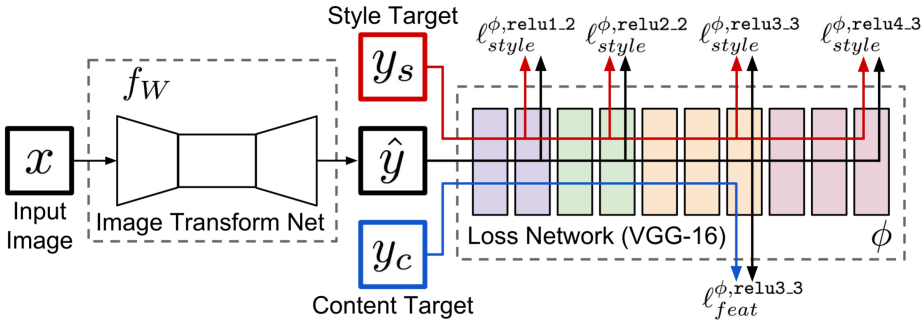


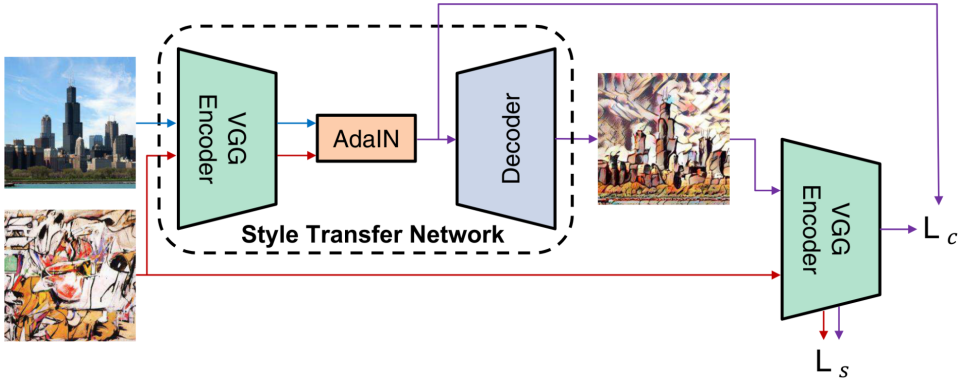
Fig. 2. System overview. We train an *image transformation network* to transform input images into output images. We use a *loss network* pretrained for image classification to define *perceptual loss functions* that measure perceptual differences in content and style between images. The loss network remains fixed during the training process.

Johnson et al. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ECCV 2016

Instance Normalization

Ulyanov et al. Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis. ICCV 2017

Adaptive Instance Normalization



$$AdaIN(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

Huang, Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. ICCV 2017

Outline

1. Metrics in image generation

2. Style transfer

3. Unconditional generation with GANs

4. Conditional generation with GANs

Datasets: FFHQ

70k images of people with permissive license
1024 × 1024 resolution

Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

Datasets: LandscapeHQ

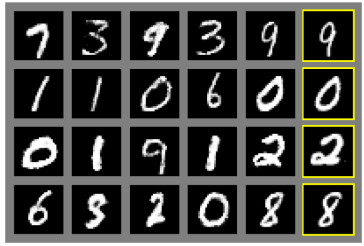
90k images of landscapes from Unsplash and Flickr with high resolution

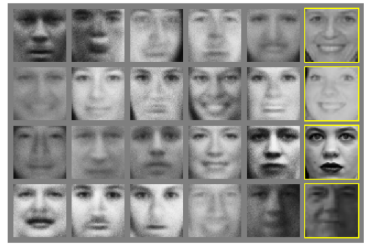
Skorokhodov et al. Aligning Latent and Image Spaces to Connect the Unconnectable. ICCV 2021

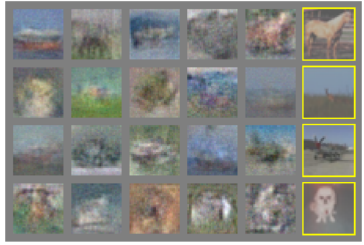
GAN

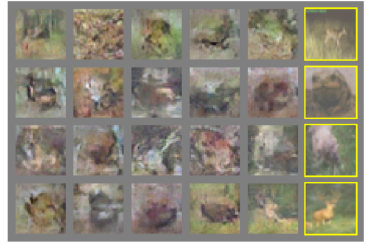
Goodfellow et al. Generative Adversarial Nets. NIPS 2014

GAN results

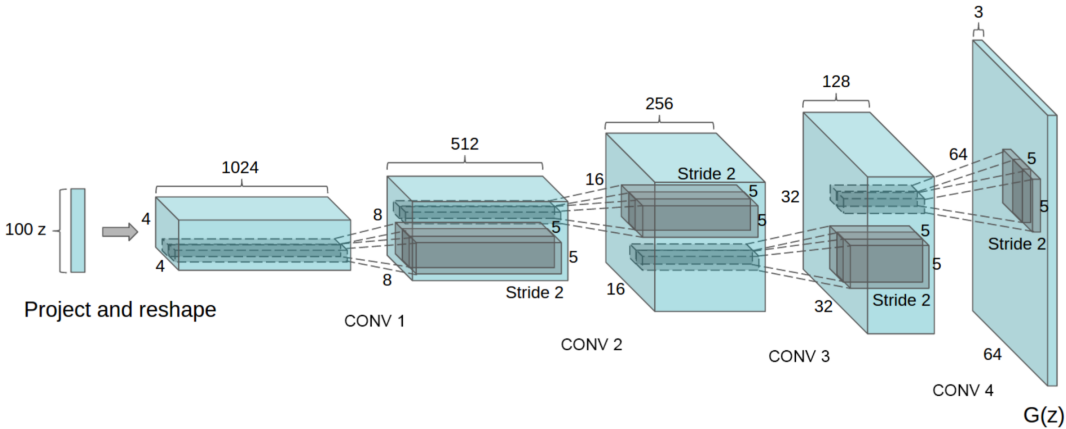








DCGAN



Radford, Metz. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ICLR 2016

Wasserstein loss function

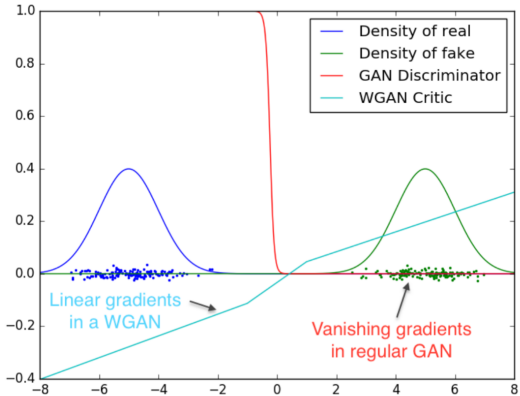


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the discriminator of a minimax GAN saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

Arjovsky et al. Wasserstein Generative Adversarial Networks. ICML 2017

Wasserstein GAN training procedure

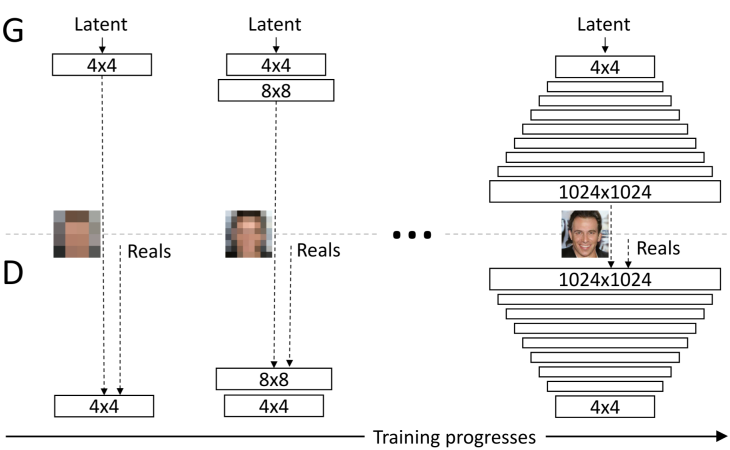
Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m} \sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)})) \right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Progressive GAN



Karras et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018

Progressive GAN

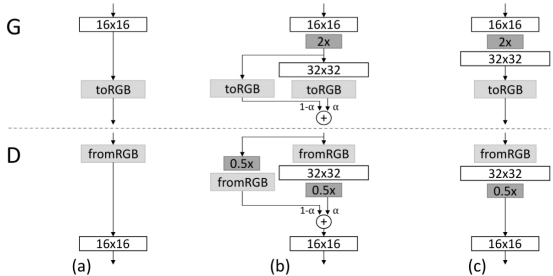
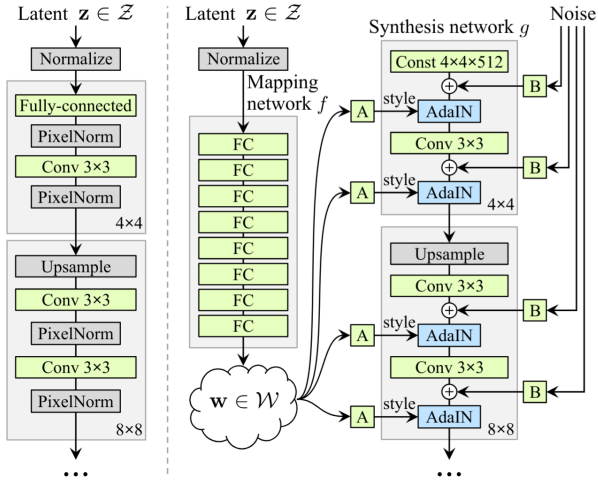


Figure 2: When doubling the resolution of the generator (G) and discriminator (D) we fade in the new layers smoothly. This example illustrates the transition from 16×16 images (a) to 32×32 images (c). During the transition (b) we treat the layers that operate on the higher resolution like a residual block, whose weight α increases linearly from 0 to 1. Here $\boxed{2\times}$ and $\boxed{0.5\times}$ refer to doubling and halving the image resolution using nearest neighbor filtering and average pooling, respectively. The $\boxed{\text{toRGB}}$ represents a layer that projects feature vectors to RGB colors and $\boxed{\text{fromRGB}}$ does the reverse; both use 1×1 convolutions. When training the discriminator, we feed in real images that are downscaled to match the current resolution of the network. During a resolution transition, we interpolate between two resolutions of the real images, similarly to how the generator output combines two resolutions.

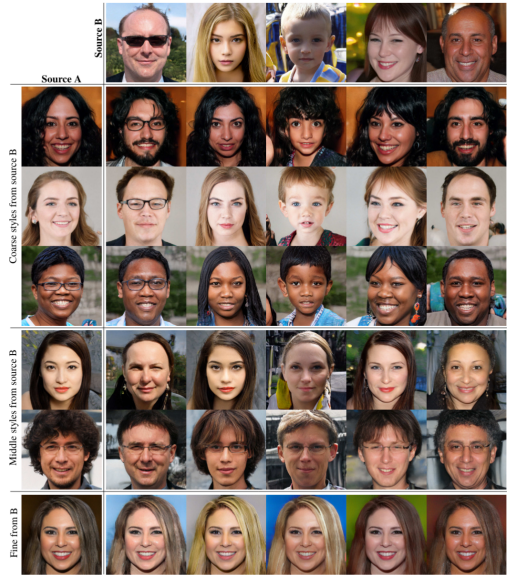
StyleGAN



(a) Traditional (b) Style-based generator

Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

Style mixing in StyleGAN



Outline

1. Metrics in image generation

2. Style transfer

3. Unconditional generation with GANs

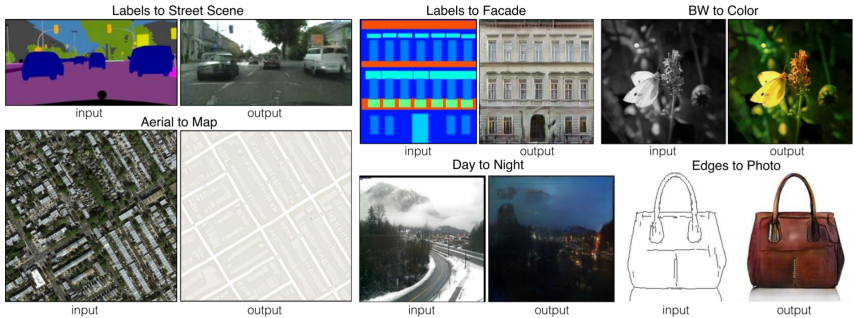
4. Conditional generation with GANs

Superresolution

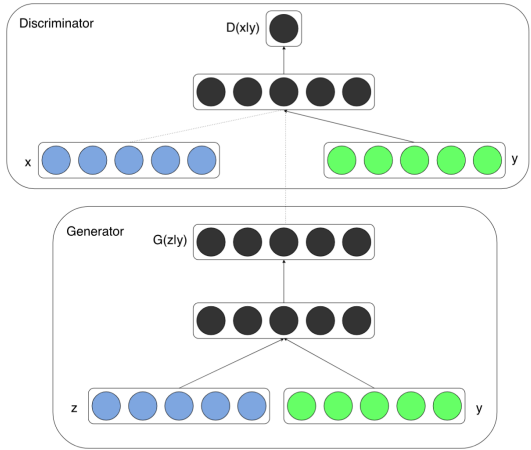
Ground Truth **Bicubic** **Ours** (ℓ_{pixel}) **SRCNN** [11] **Ours** (ℓ_{feat})

Inpainting

Domain adaptation

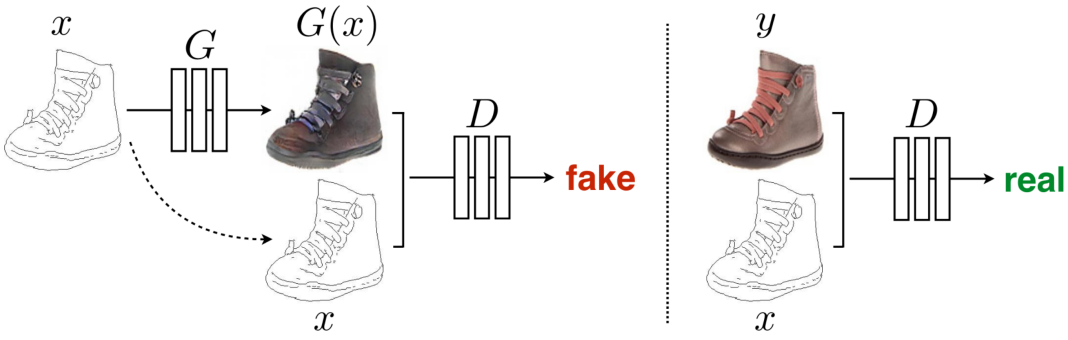


cGAN



Mirza, Osindero. Conditional Generative Adversarial Nets. arXiv: 1411. 1784

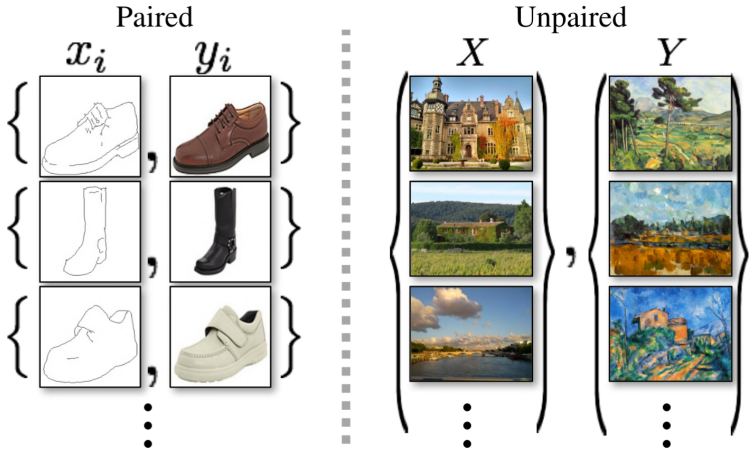
pix2pix



Isola et al. Image-to-Image Translation with Conditional Adversarial Networks. CVPR 2017

pix2pix results

Unpaired data



CycleGAN

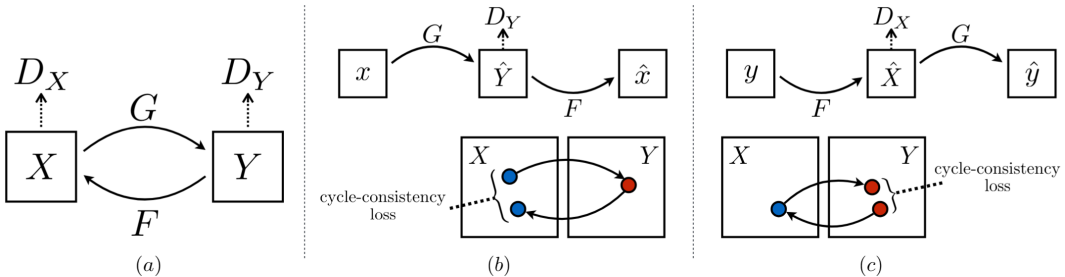
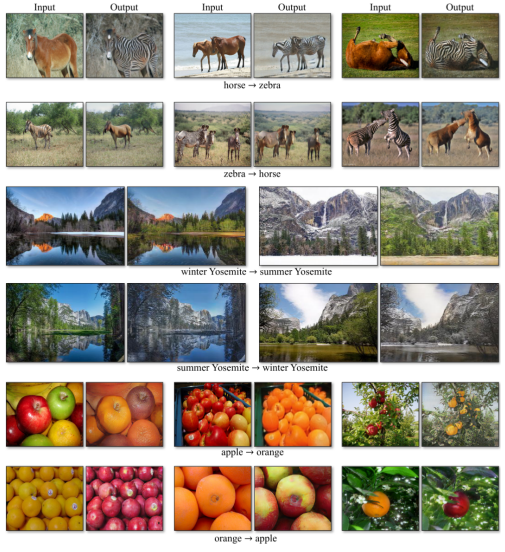


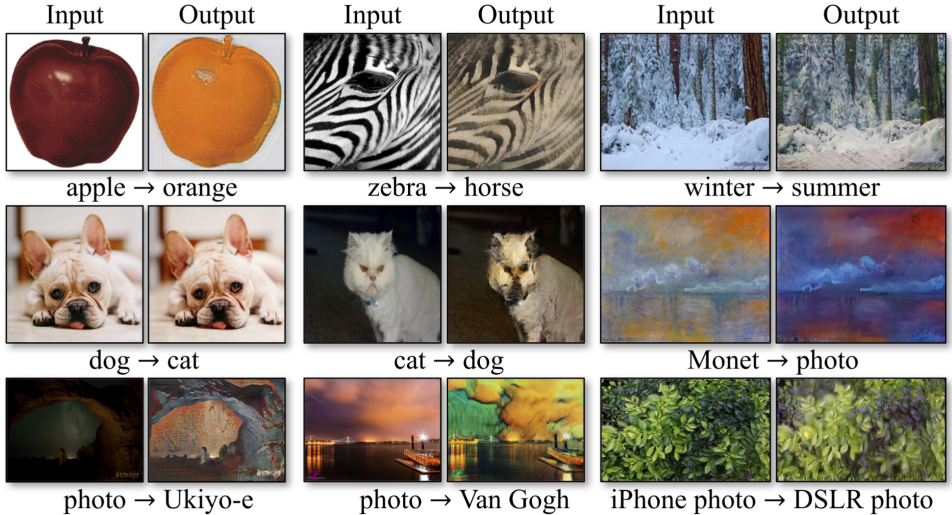
Figure 3: (a) Our model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV 2017

CycleGAN results



CycleGAN failures



SPADE

Park et al. Semantic Image Synthesis with Spatially-Adaptive Normalization. CVPR 2019

AdaIN in SPADE

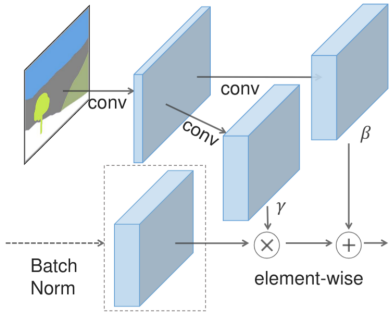


Figure 2: In the SPADE, the mask is first projected onto an embedding space and then convolved to produce the modulation parameters γ and β. Unlike prior conditional normalization methods, γ and β are not vectors, but tensors with spatial dimensions. The produced γ and β are multiplied and added to the normalized activation element-wise.

SPADE

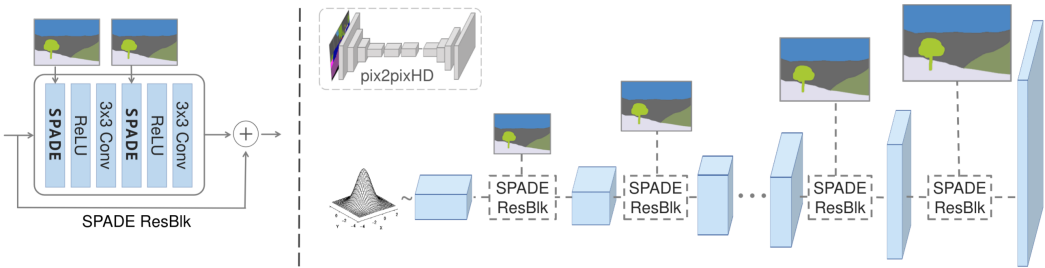
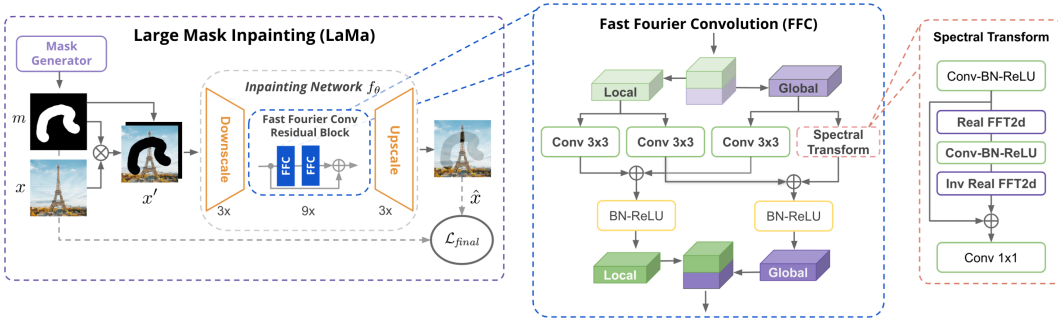


Figure 4: In the SPADE generator, each normalization layer uses the segmentation mask to modulate the layer activations. *(left)* Structure of one residual block with the SPADE. *(right)* The generator contains a series of the SPADE residual blocks with upsampling layers. Our architecture achieves better performance with a smaller number of parameters by removing the downsampling layers of leading image-to-image translation networks such as the pix2pixHD model [48].

LAMA



Suvorov et al. Resolution-robust Large Mask Inpainting with Fourier Convolutions. WACV 2022

LAMA results

Original	LaMa-Regular 45M	LaMa-Regular (deep) 78M	**LaMa-Fourier** **27M**	CoModGAN [64] 109M	MADF [67] 85M

LAMA high-res results

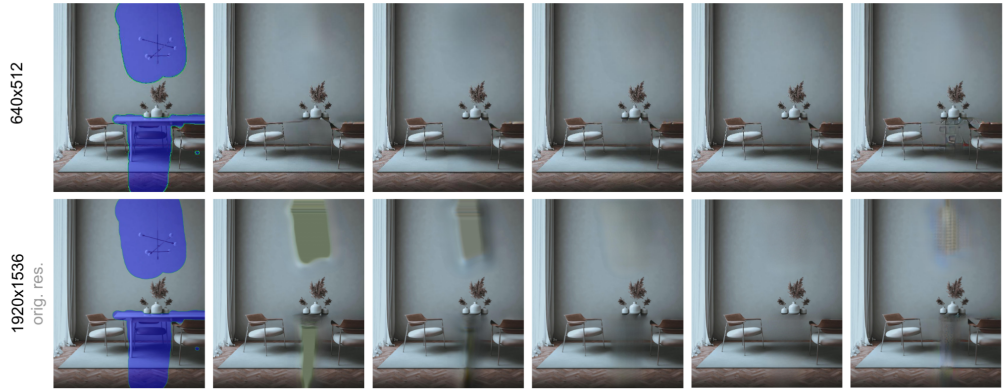
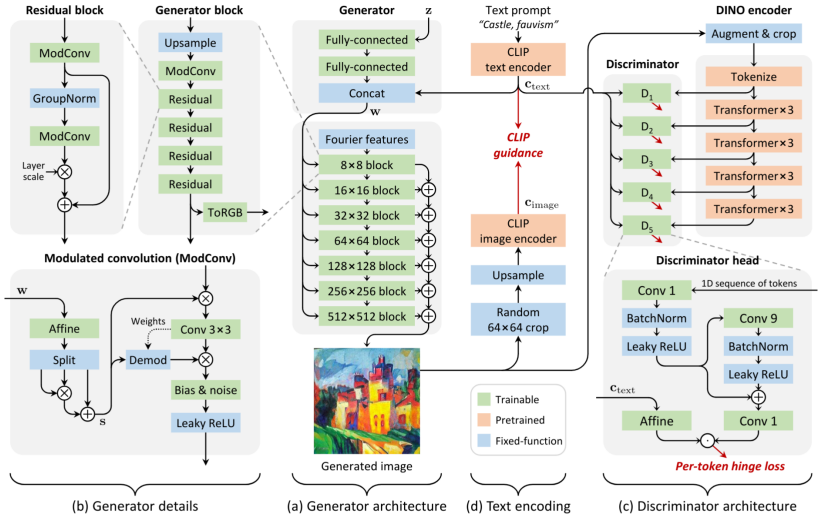


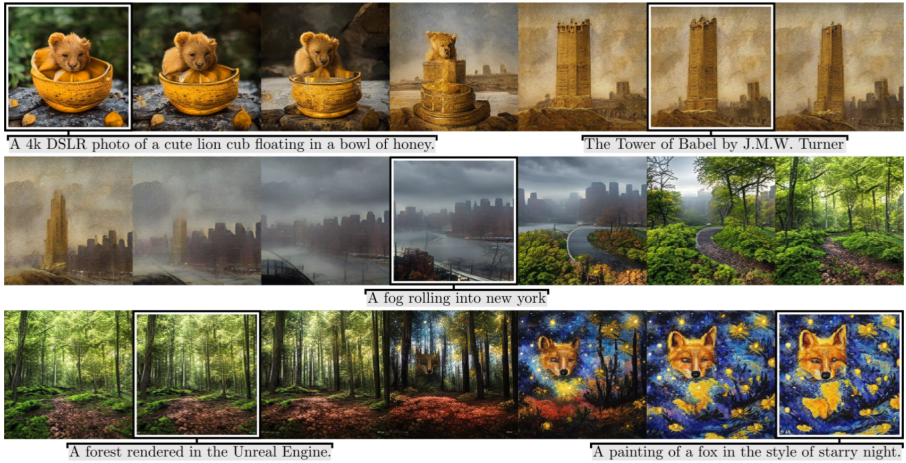
Figure 5: Transfer of inpainting models to a higher resolution. All LaMa models were trained using 256×256 crops from 512×512, and MADF [67] was trained on 512×512 directly. As the resolution increases, the models with regular convolutions swiftly start to produce critical artifacts, while FFC-based models continue to generate semantically consistent image with fine details. More negative and positive examples of our 51M model can be found at `bit.ly/3k0gaIK`.

StyleGAN-T



Sauer et al. StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis. ICML 2023

StyleGAN-T



Sauer et al. StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis. ICML 2023

Comparison with diffusion models

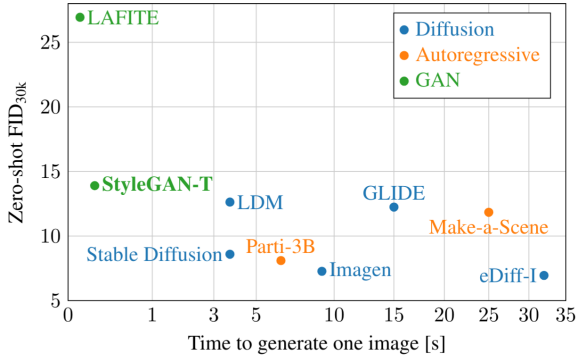


Figure 1. **Quality vs. speed** in large-scale text-to-image synthesis. StyleGAN-T greatly narrows the quality gap between GANs and other model families while generating samples at a rate of 10 FPS on an NVIDIA A100. The y-axis corresponds to zero-shot FID on MS COCO at 256×256 resolution; lower is better.

Conclusion

We reviewed following topics:

- reconstructing images from neural features
- various metrics on top of neural features used for image comparison
- style transfer: optimization-based and training networks for single and several styles
- unconditional image generation with GANs
- conditional image generation with GANs