

# Priors Guided Image Inpainting and Synthesis

Yanwei Fu

Fudan University, School of data science

yanweifu@fudan.edu.cn

Tutorial Homepage: [https://dqiaole.github.io/priors\\_guided\\_image\\_editing\\_synthesis/](https://dqiaole.github.io/priors_guided_image_editing_synthesis/)

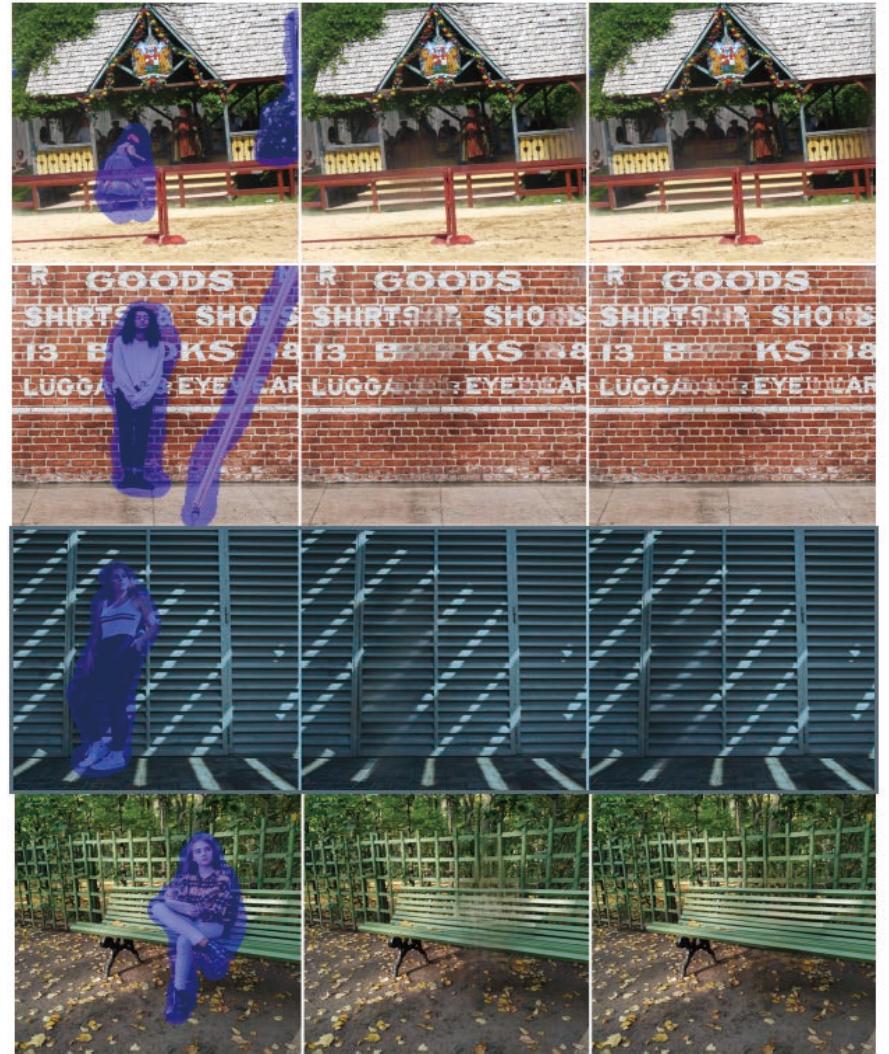
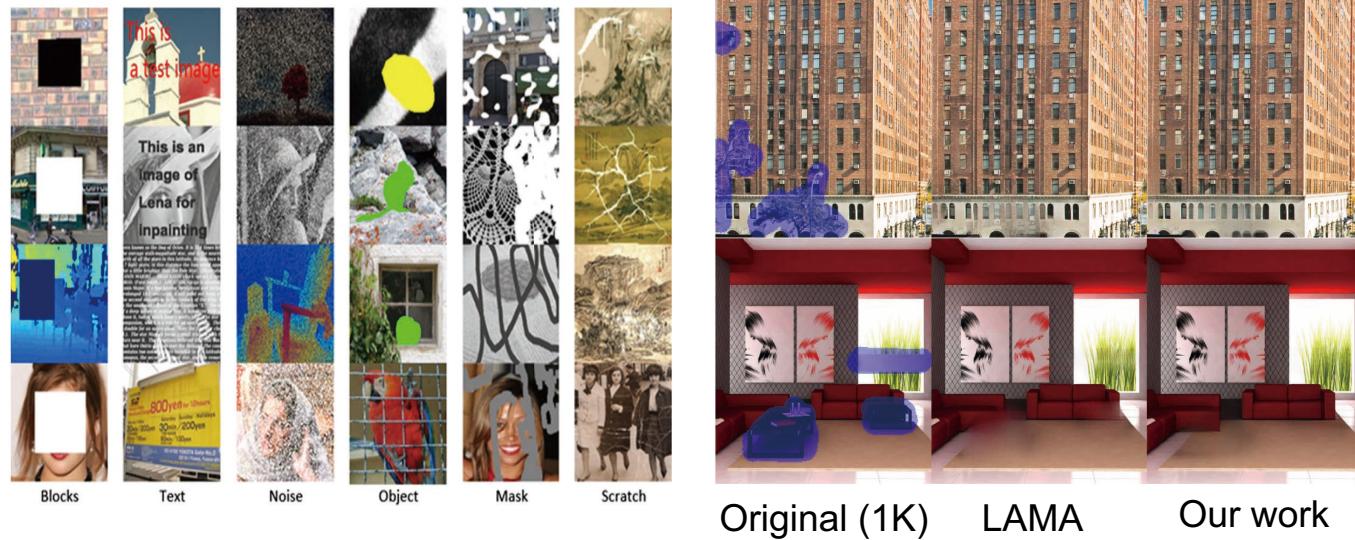
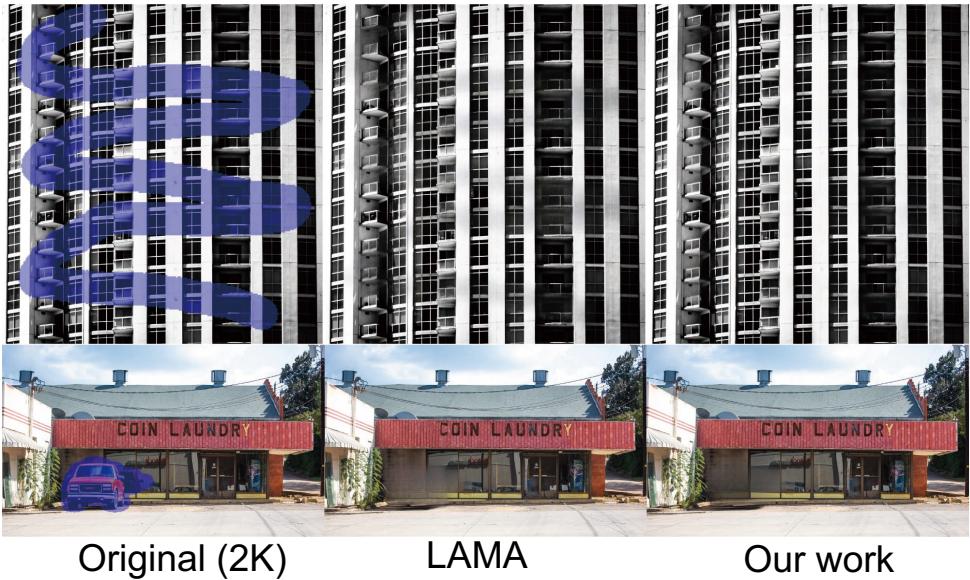


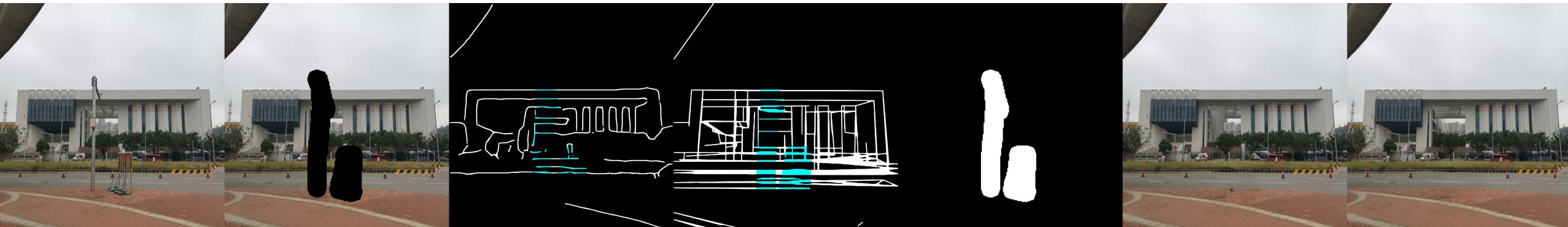
# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion

# Task1: Image Inpainting





Original  
images

Masked  
images

Edge  
maps

Lines

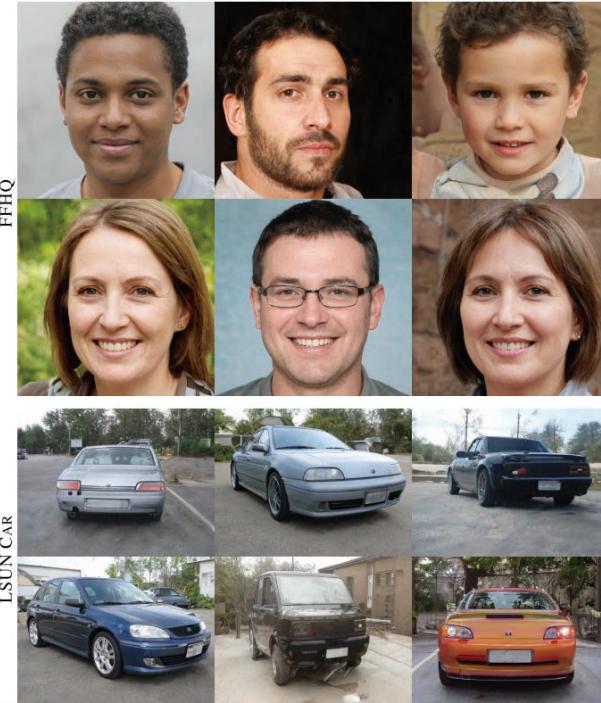
Applied Mask

LaMa

**Our ZITS++**

# Task2: Image Synthesis/Generation

StyleGAN2



StyleGAN-XL



DALLE2



Image



Unconditional

Class-conditional



Text-conditional

Prompt Engineering: A photo of {Class-label}

[StyleGAN2] Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." *CVPR*. 2020.

[StyleGAN-XL] Sauer, Axel, Katja Schwarz, and Andreas Geiger. "Stylegan-xl: Scaling stylegan to large diverse datasets." *ACM Siggraph*. 2022.

[BIGGAN] Brock, Andrew, Jeff Donahue, and Karen Simonyan. "Large scale GAN training for high fidelity natural image synthesis." *ICLR2019*.

[DALLE 2] Ramesh, Aditya, et al. "Hierarchical text-conditional image generation with clip latents." (2022).

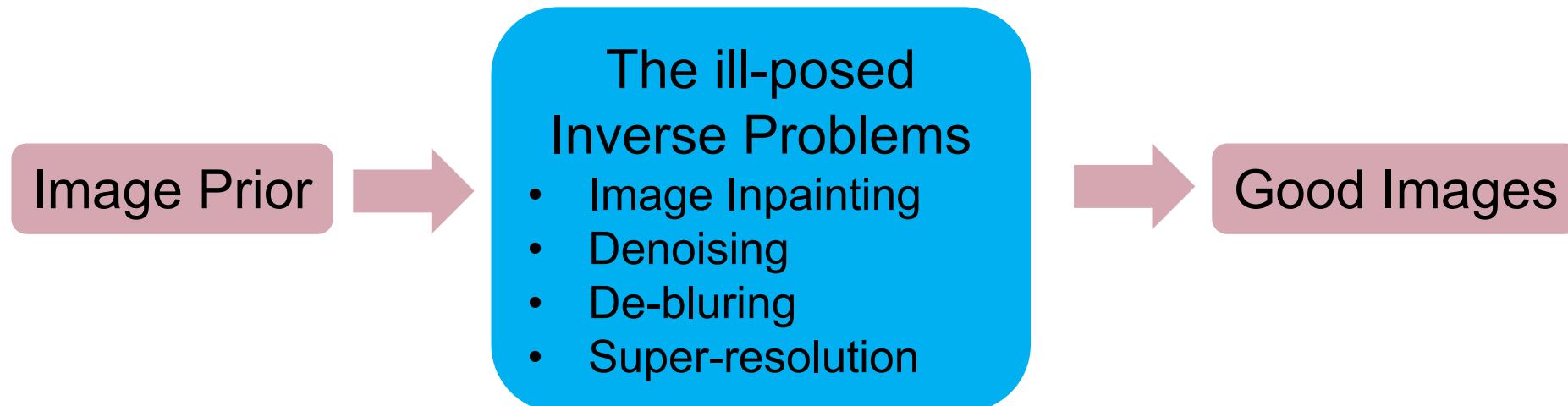
[Imagen] Saharia, Chitwan, et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding." arXiv preprint arXiv:2205.11487 (2022).

# Priors: What are the Priors?

Prior probability, of an uncertain quantity (Bayesian statistical inference)

- ▶ probability distribution *expresses one's beliefs about this quantity* before some evidence is taken into account.
- ▶ 
$$\text{Posterior} = \text{Likelihood} * \text{Prior/Evidence}$$

## ▶ Inverse Problems in Computer Vision



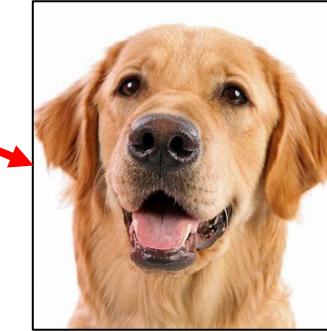
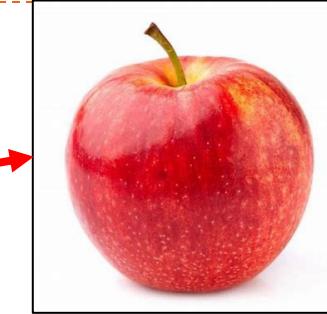
# Priors: as the Guidance to Image Tasks

High-level  
(Semantic)  
Guidance

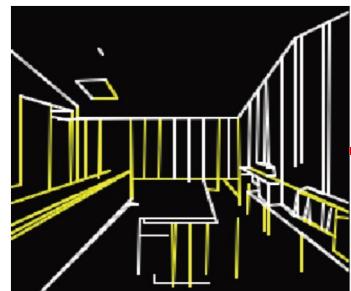
**Text:** This is an apple

**Class:** Dog

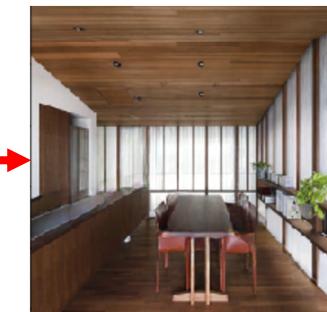
Deep  
Generative  
Model



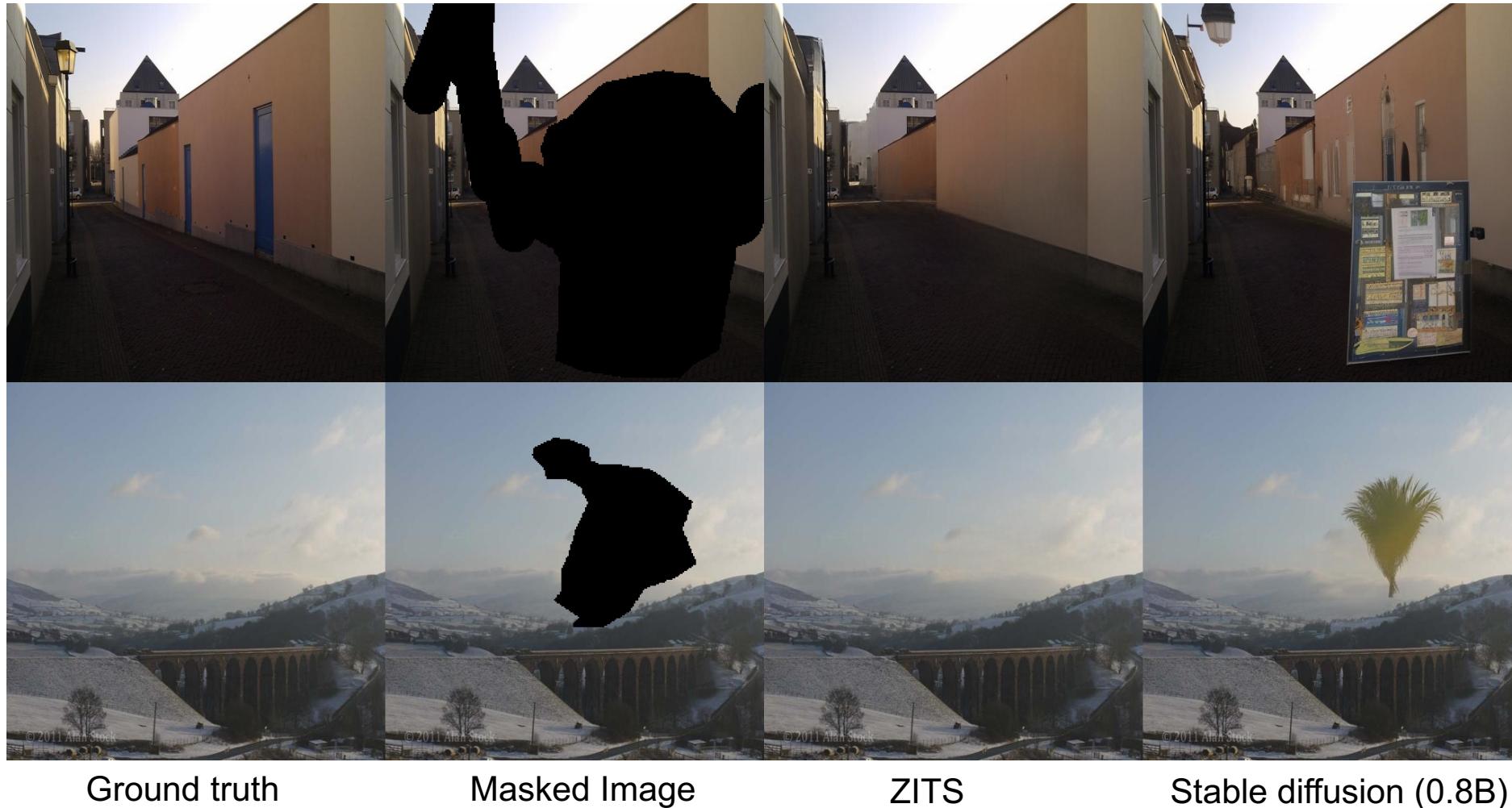
Low-level  
(Structure)  
Guidance



Deep  
Generative  
Model



# Fidelity: (high-level) Diffusion Models V.S. Inpainting Models



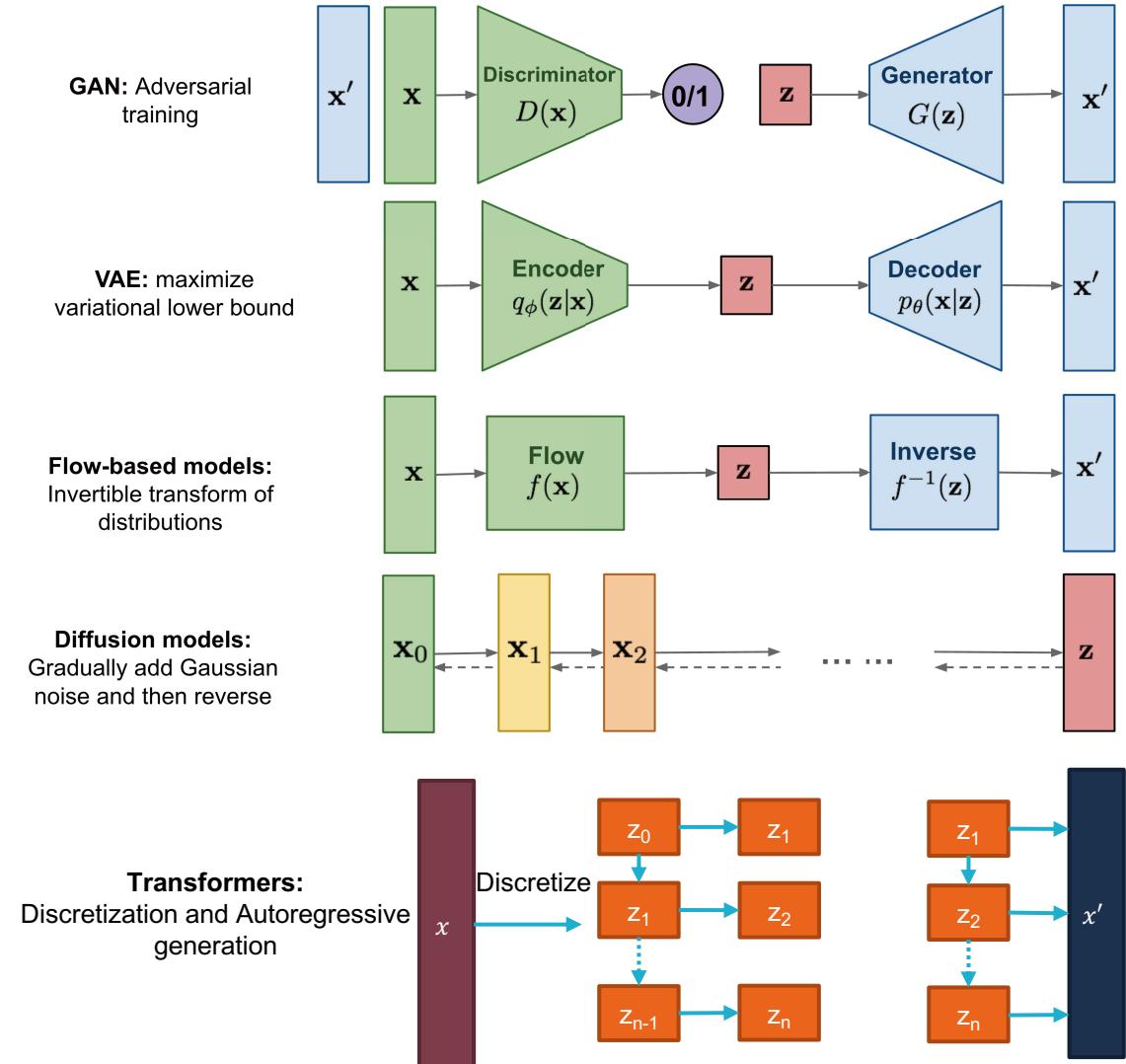
**Large capacity of model  $\not\Rightarrow$  Fidelitous and consistent inpainting results**

[ZITS] Dong Qiaole, et al. " Incremental Transformer Structure Enhanced Image Inpainting with Masking Positional Encoding." *CVPR*. 2022.

[Stable diffusion] Robin Rombach, et al. "High-Resolution Image Synthesis with Latent Diffusion Models Robin." *CVPR*. 2022.

# Contents

- ▶ Tasks and Motivation
- ▶ **Image Synthesis/Generation Methods**
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion



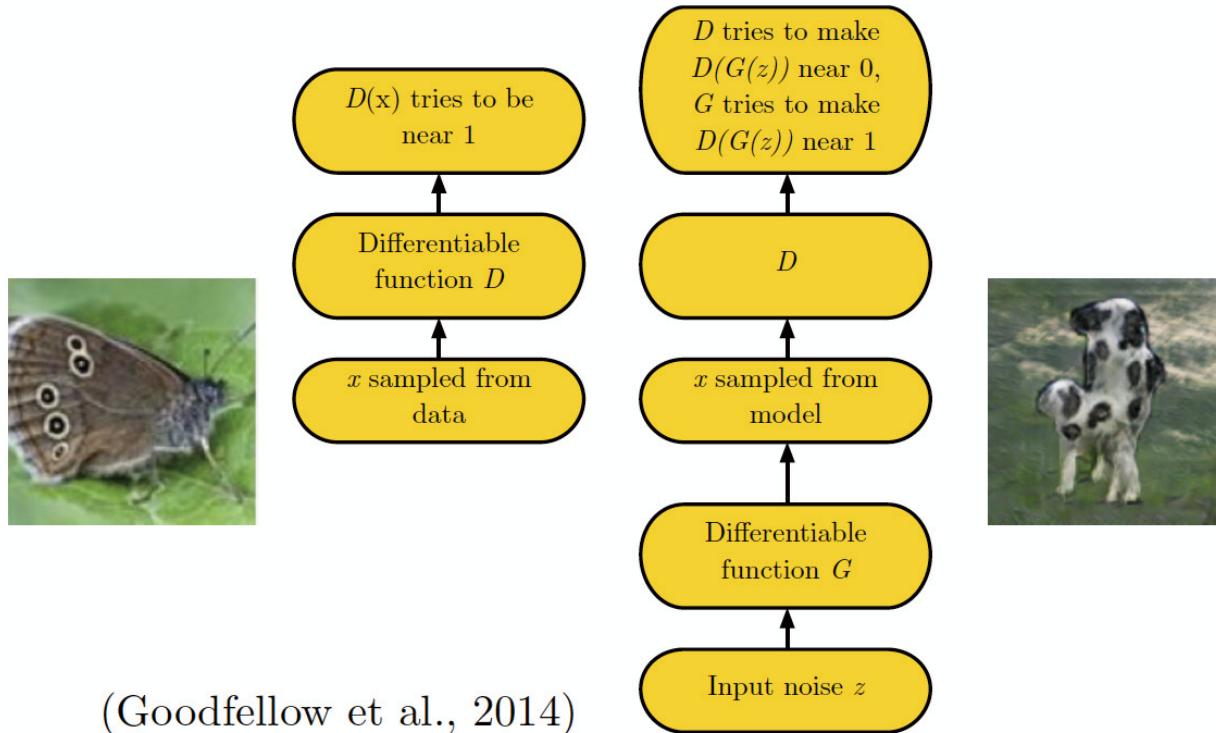


# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
  - ▶ GAN
    - ▶ Inpainting with GAN
    - ▶ GAN inversion
  - ▶ VAE and Flow
  - ▶ Transformer
  - ▶ Diffusion

# Generative Adversarial Networks (GANs)



**Improve the training of GANS:**

- WGAN[2]
- WGAN-GP[3]
- Spectral normalization[4]

...

D and G play the following two-player minimax game with value function  $V(D, G)$ [1].

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))].$$

[1] Ian J. Goodfellow et al. Generative Adversarial Nets. NeurIPS2014

[2] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." *International conference on machine learning*. PMLR, 2017.

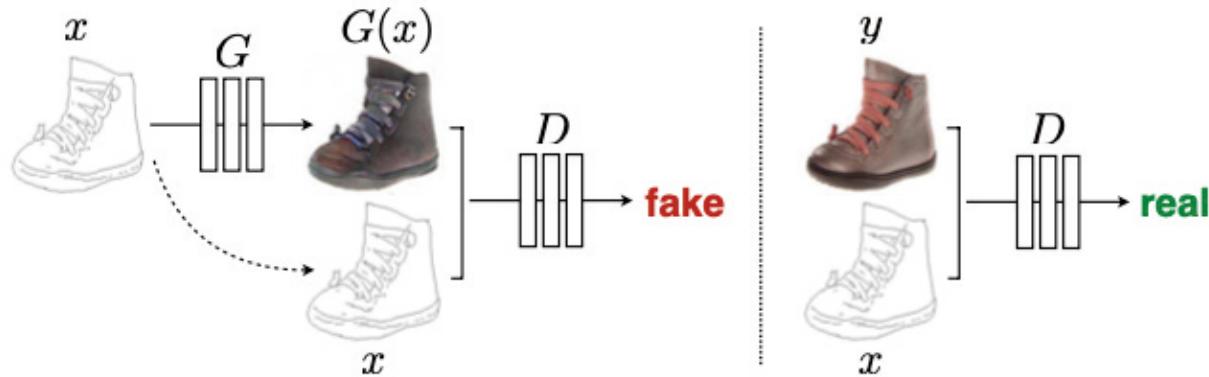
[3] Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." *Advances in neural information processing systems* 30 (2017).

[4] Miyato, Takeru, et al. "Spectral normalization for generative adversarial networks." *arXiv preprint arXiv:1802.05957* (2018).

# GANs for Image Synthesis

The adversarial training is critical for generating the high-frequency details.

Pixel2Pixel (2017)

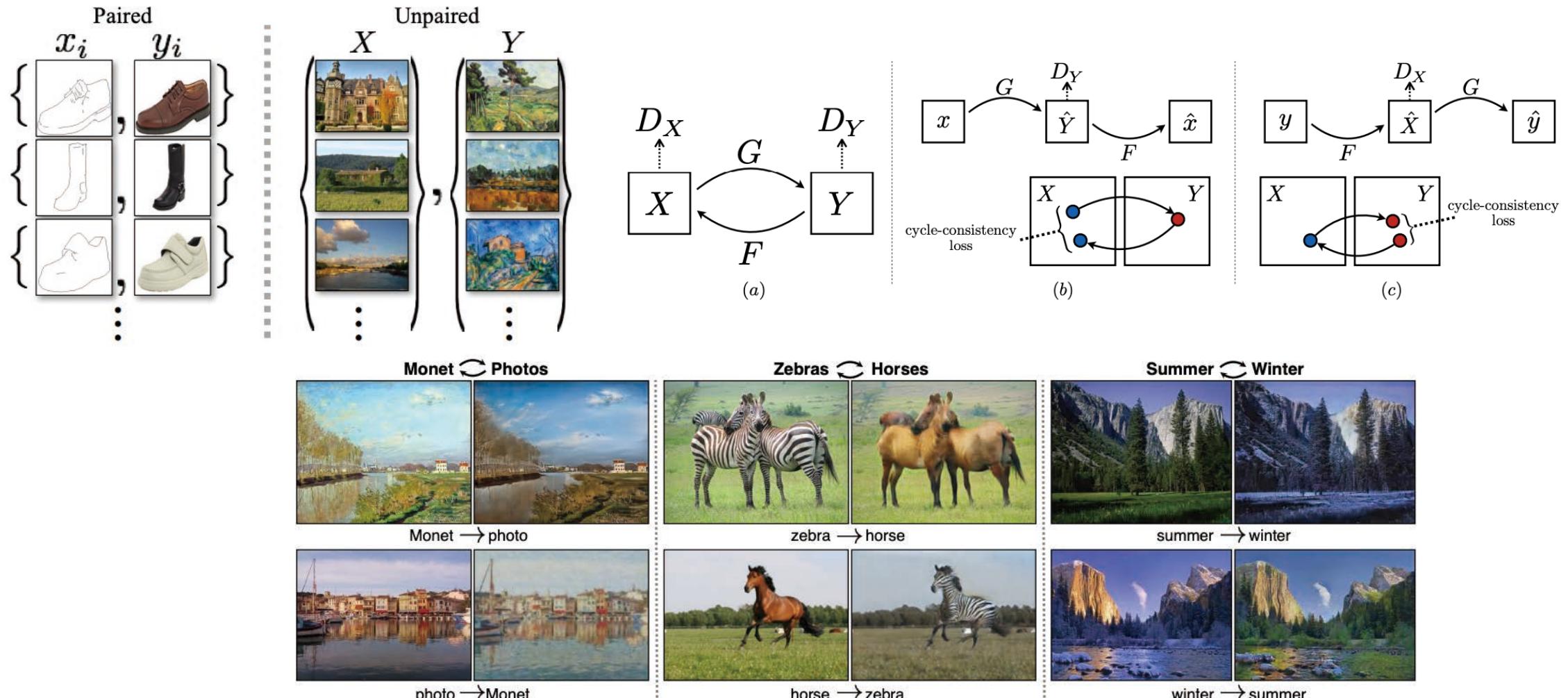


$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z)))].$$

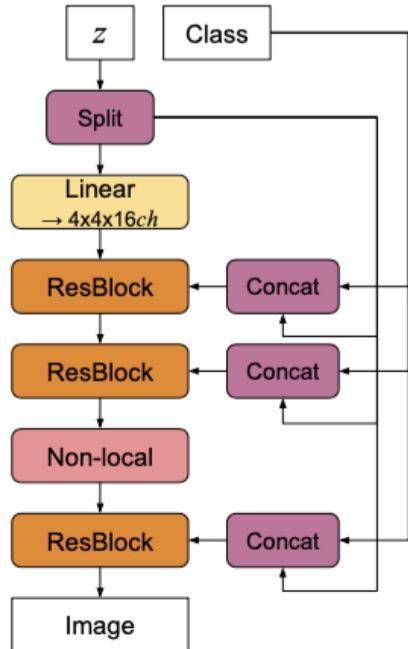


# Cycle GANs

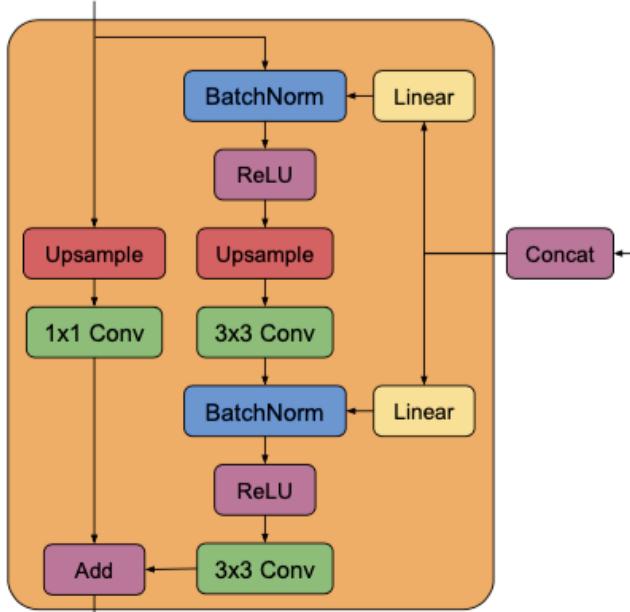
- The training can be trained with unpaired images of cycle-consistency loss.



# BigGAN: Bigger GANs



Overview



Model Block



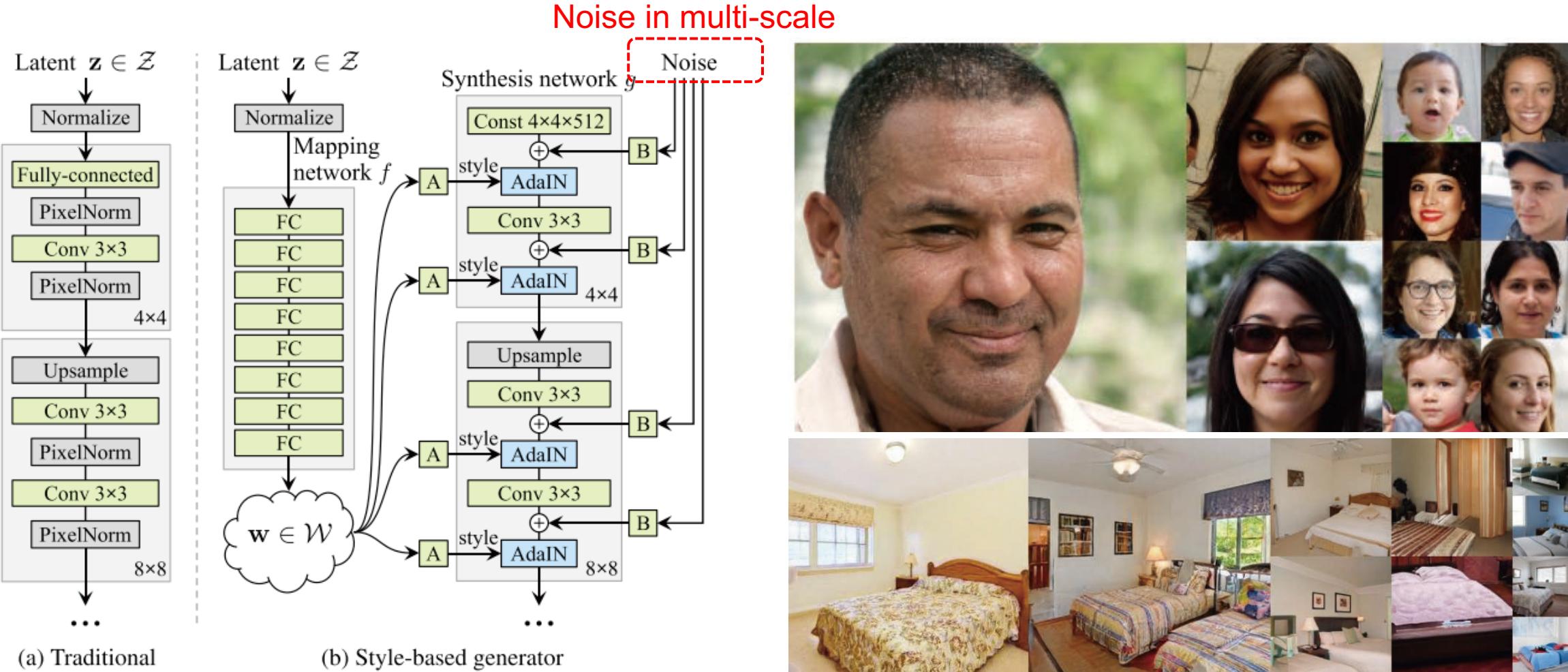
Figure 6: Samples generated by our BigGAN model at 512×512 resolution.

Scaling up the model and training for GANs

- Larger batch size (2048)
- Larger base channels (96)
- Truncation Trick

**Very good results!**

# Style-based Generator: StyleGAN



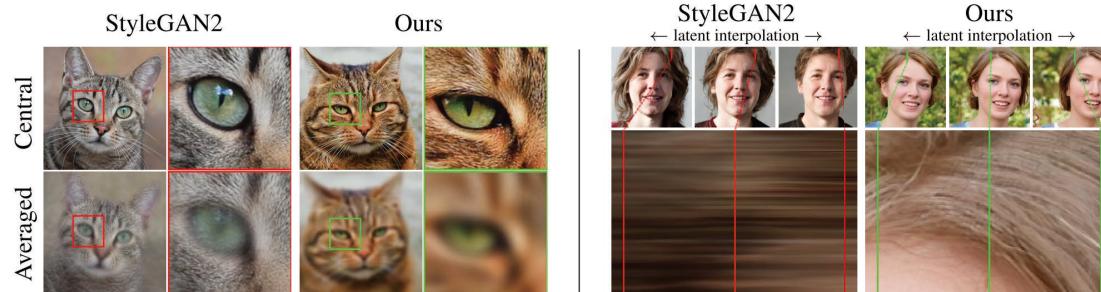
Adain for style fusion:  $\text{Adain}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}$ ,

High-quality results in FFHQ and LSUN

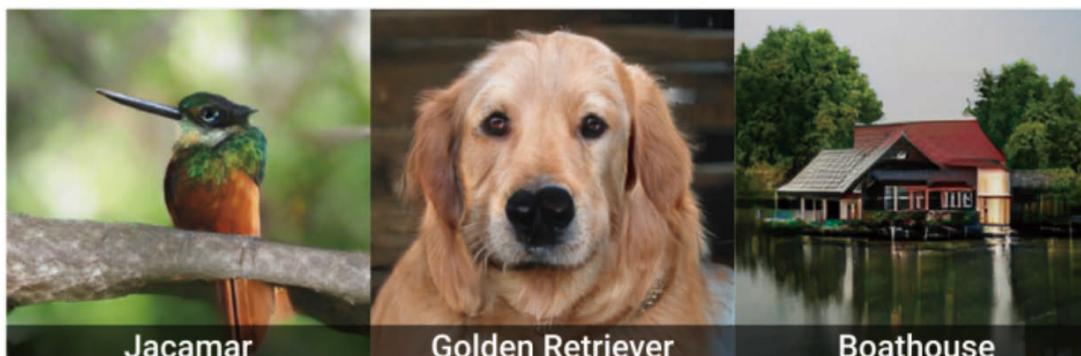
# Improved StyleGANs



StyleGAN2[1]: adjusting model architecture and training strategy to address water droplet and phase artifacts



StyleGAN3[2]: make stylegan's texture free-from image coordinates



StyleGAN-XL[3]: generalize stylegan on ImageNet

[1] Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." *CVPR* 2020.

[2] Karras, Tero, et al. "Alias-free generative adversarial networks." *NeurIPS2021*.

[3] Sauer, Axel, Katja Schwarz, and Andreas Geiger. "Stylegan-xl: Scaling stylegan to large diverse datasets." *ACM Siggraph*. 2022.

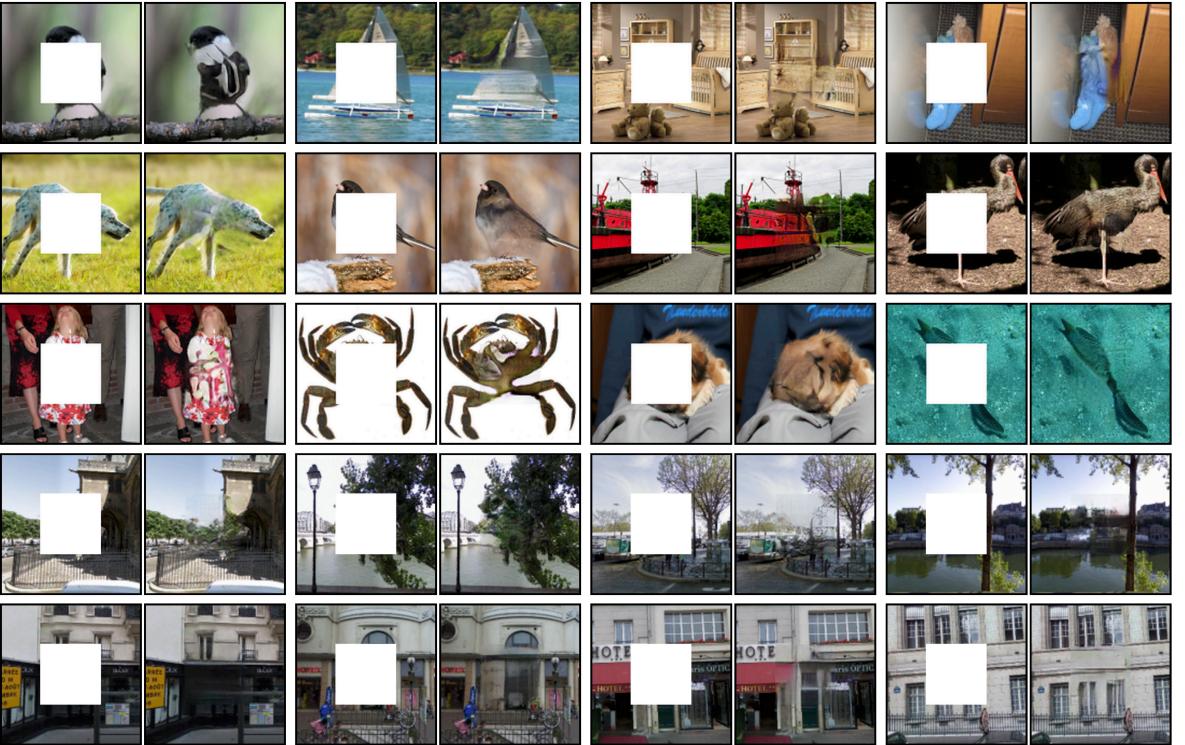
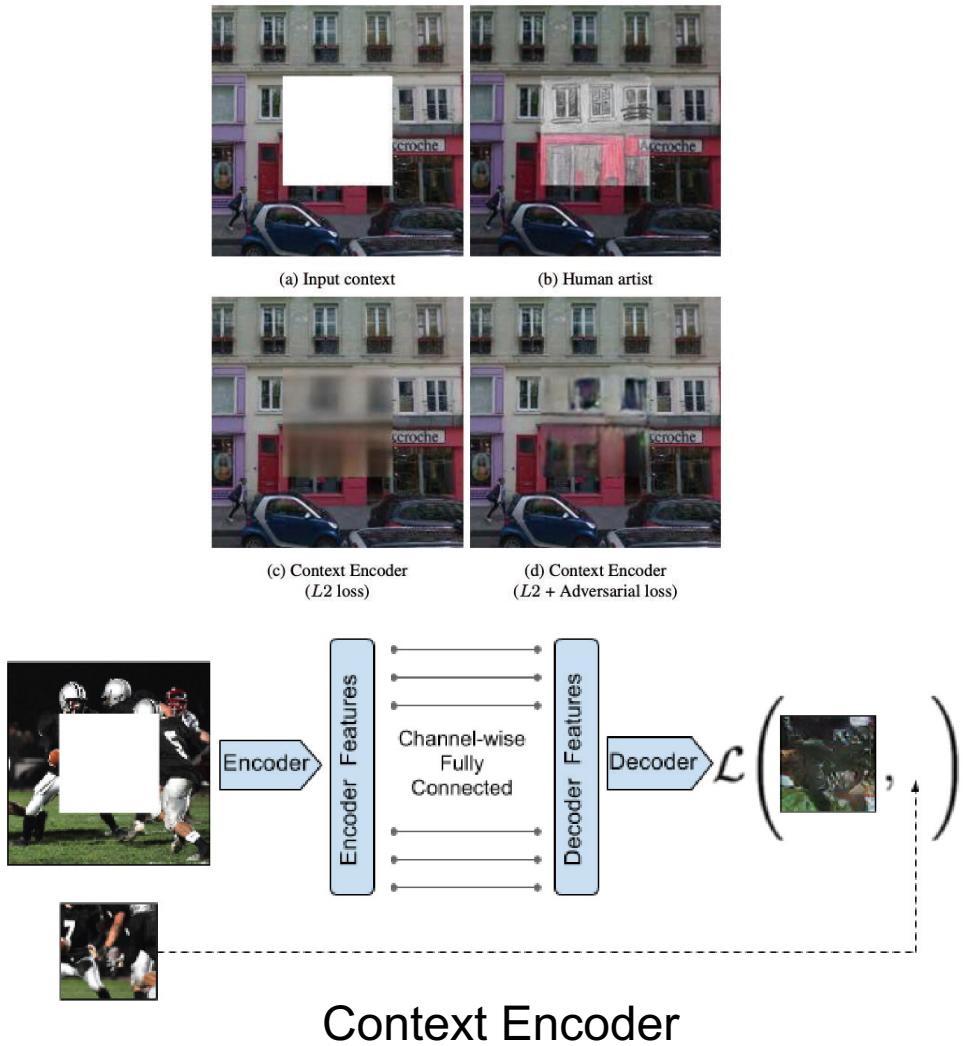


# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion

# GANs for Image Inpainting



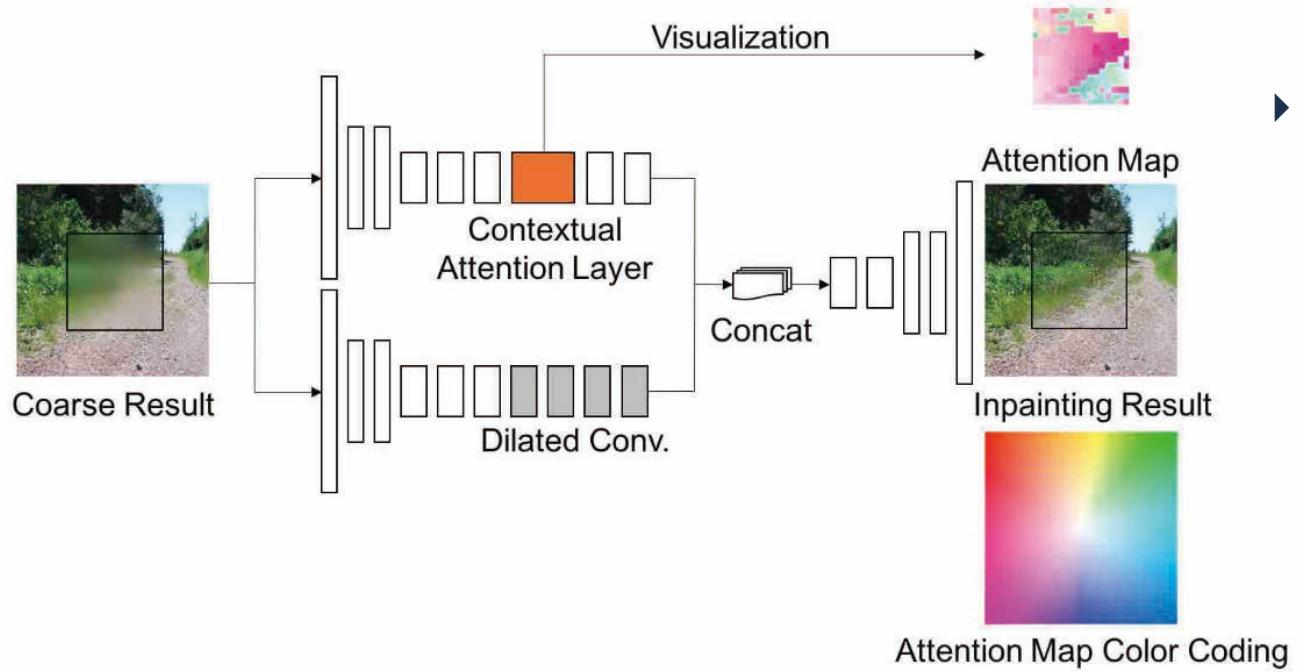
Semantic Inpainting results on held-out images by Context Encoder.  
from [https://www.cs.cmu.edu/~dpathak/context\\_encoder/#extraResults](https://www.cs.cmu.edu/~dpathak/context_encoder/#extraResults)

## Convolutional Designs for Inpainting in GANs

- Partial Convolution [2]
- Gated Convolution [3]

[1] Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." CVPR. 2016.  
 [2] Guilin Liu et al. Image Inpainting for Irregular Holes Using Partial Convolutions. ECCV2018  
 [3] Jiahui Yu et al. Free-Form Image Inpainting with Gated Convolution. ICCV2019

# Attention for Inpainting in GANs



- ▶ Aggregating features from unmasked regions for masked ones largely improves the inpainting[1].
- ▶ Guiding for the high-resolution inpainting[2,3].
- ▶ Guiding for the texture and structure feature fusion[4].

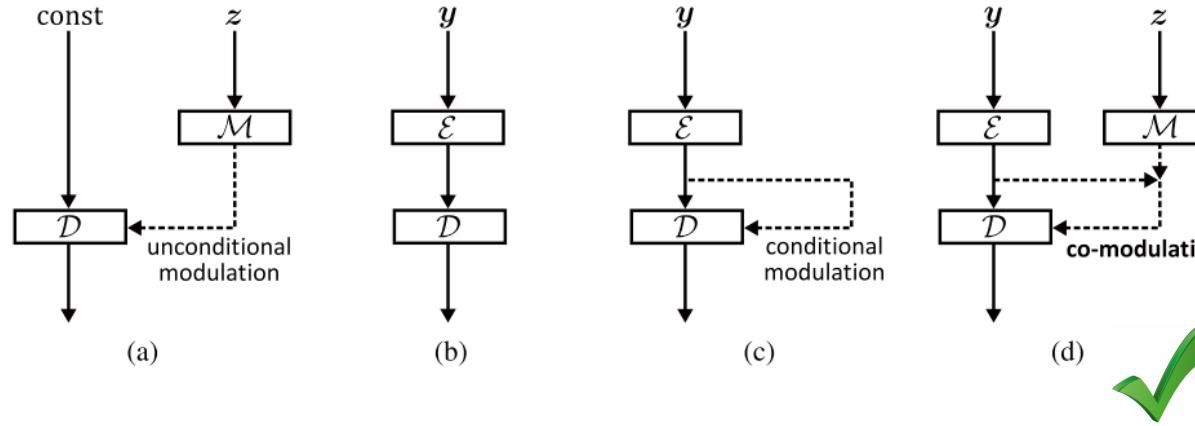
[1] Yu, Jiahui, et al. "Generative image inpainting with contextual attention." *CVPR2018*.

[2] Zeng, Yu, et al. "High-resolution image inpainting with iterative confidence feedback and guided upsampling." *ECCV2020*.

[3] Zili Yi et al. "Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting." *CVPR2020*

[4] Xiefan Guo et al. "Image Inpainting via Conditional Texture and Structure Dual Generation" *CVPR2021*

# Co-Mod: StyleGAN for Inpainting

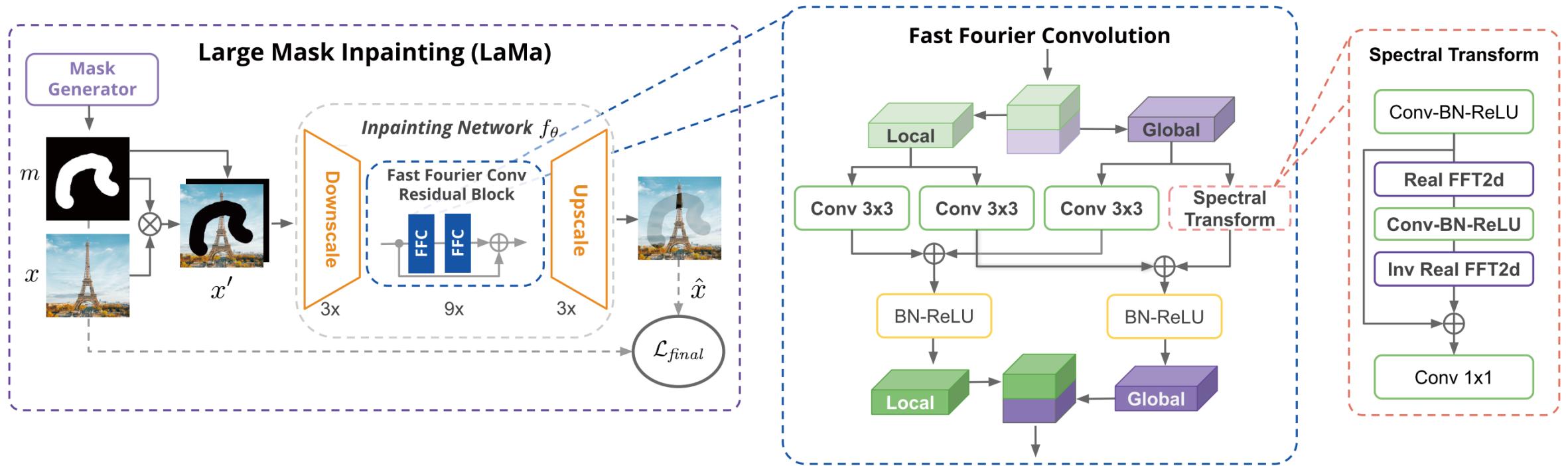


$$s = \mathcal{A}(\mathcal{E}(y), \mathcal{M}(z)),$$

- ▶ Using Conditional (unmasked image) and unconditional modulated (style feature) generator.
- ▶ Co-mod is powerful but not faithful and stable enough for some inpainting cases.



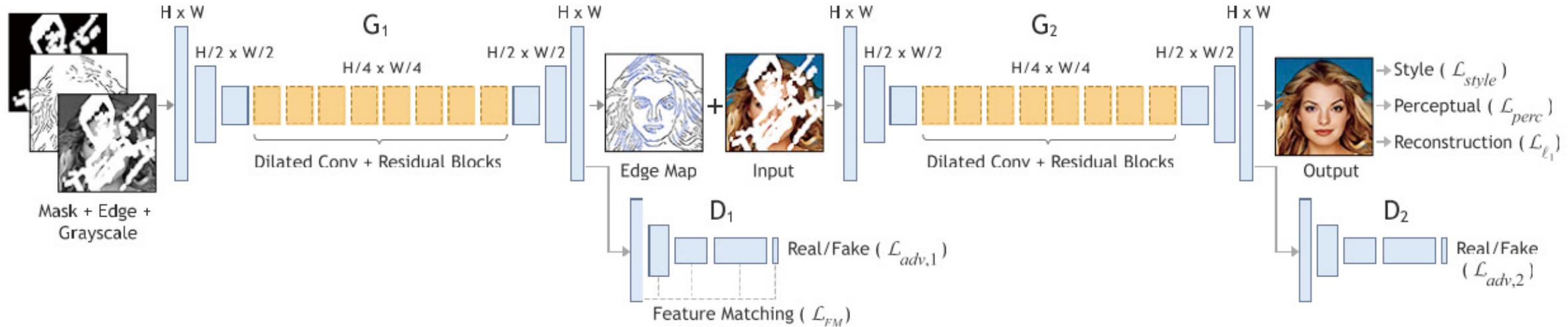
# LaMa: Resolution-robust Large Mask Inpainting with Fourier Convolutions



Fourier convolutions are used to for the high-resolution image inpainting

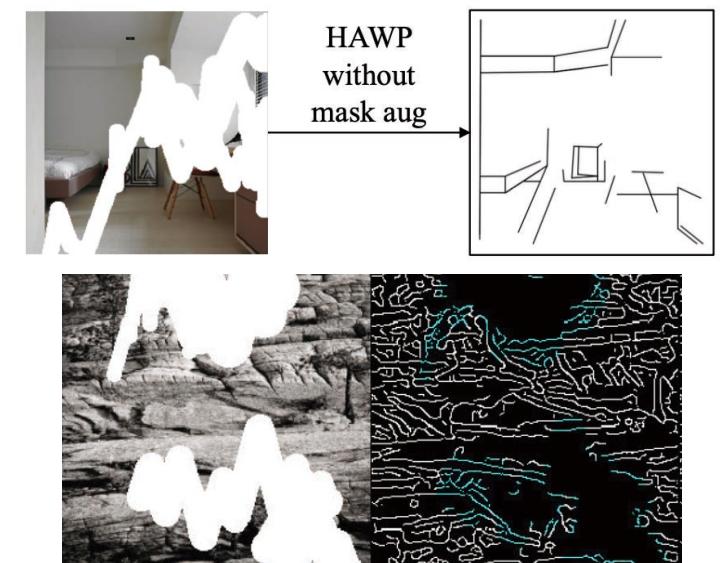
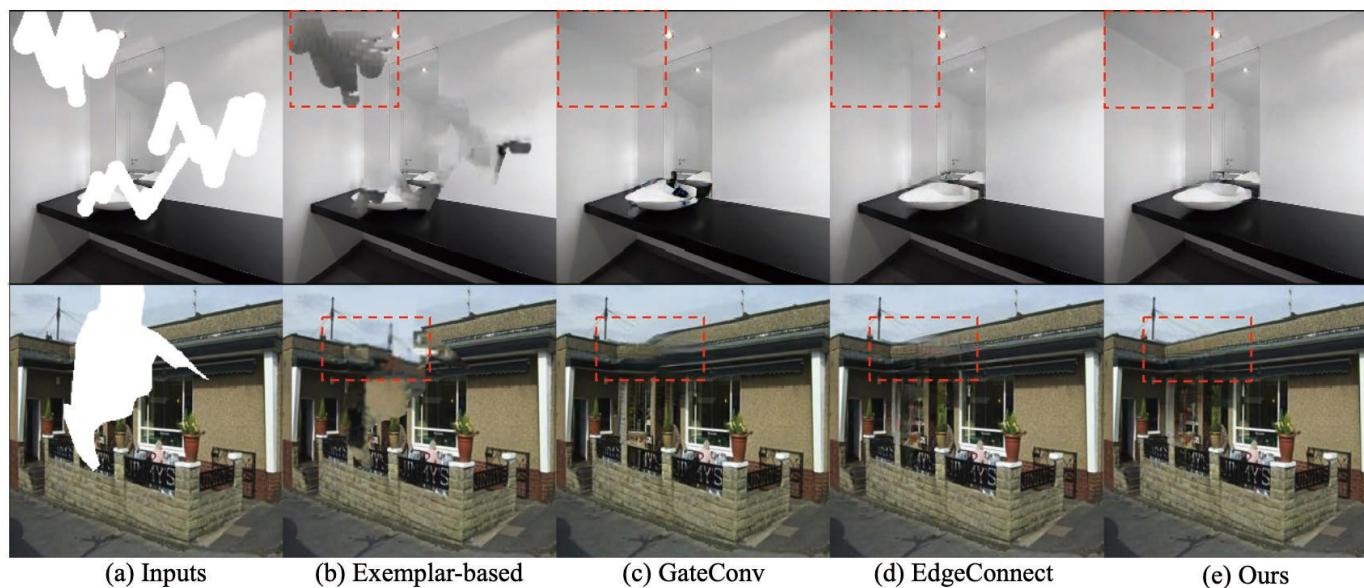
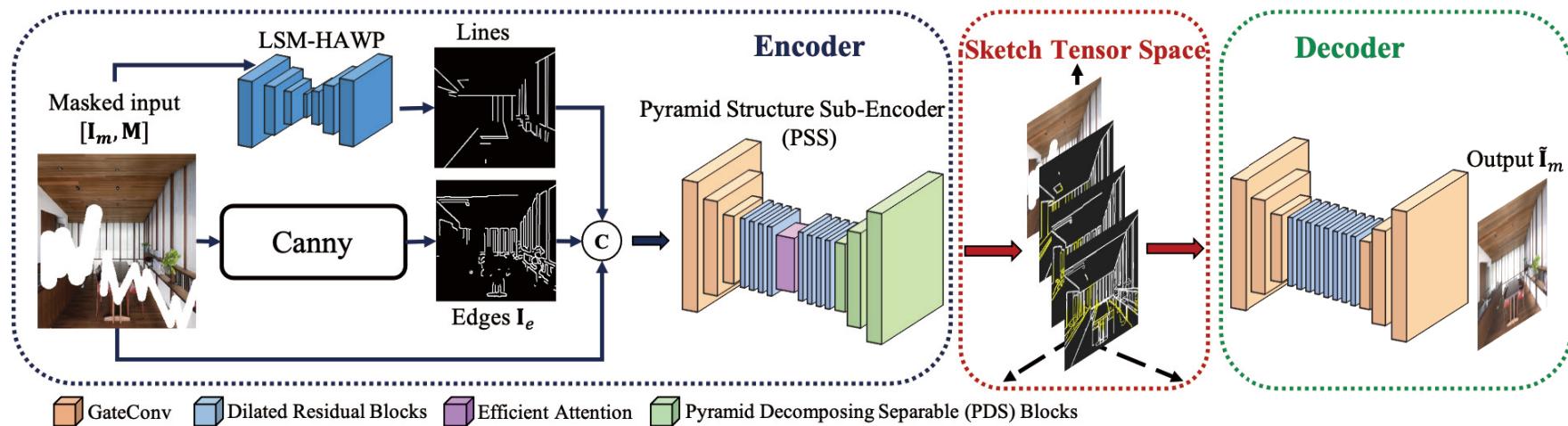
The 256x256 trained model can be generalized to high-resolution images!

# EdgeConnect: Edge Prior for Inpainting



- ▶ Recovering Canny Edges at first, then recover colored images, i.e., “lines first, color next”
- ▶ Advantage: more faithful and stable generation with structural priors
- ▶ Limitations: Limited receptive fields for both edge and color generators.

# MST: Learning a Sketch Tensor Space for Inpainting of Man-made Scenes



# Qualitative results

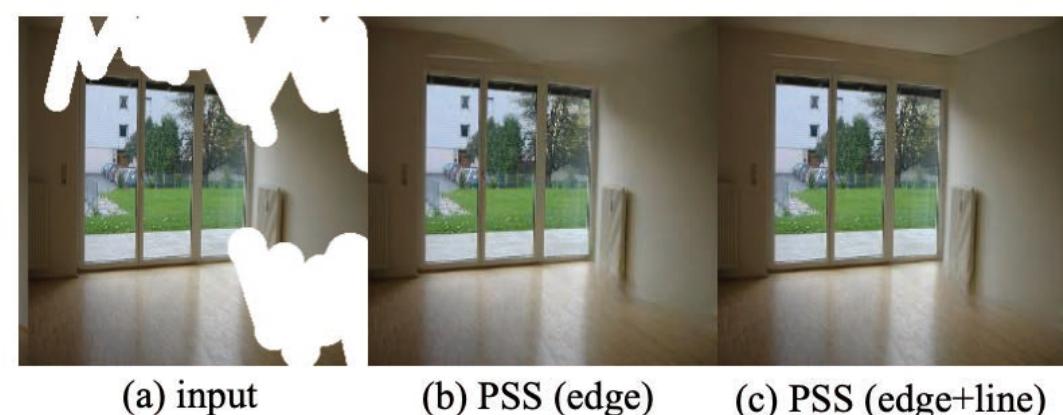
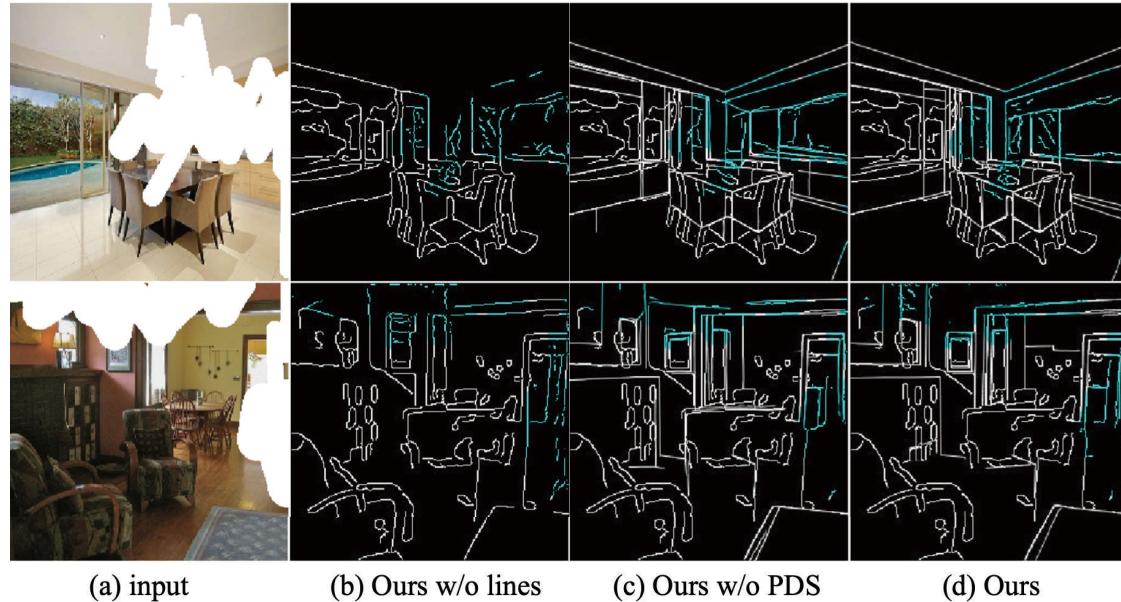
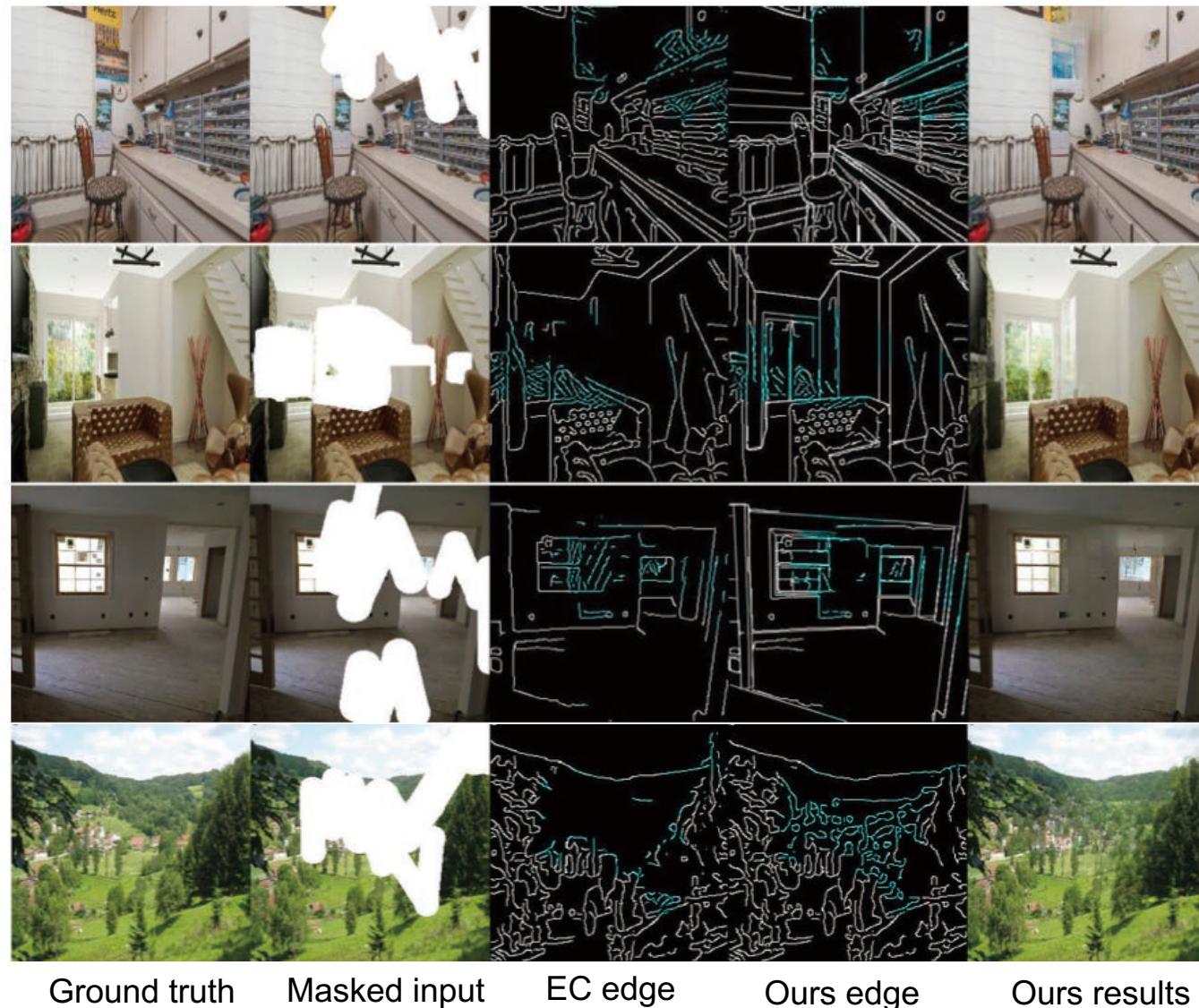
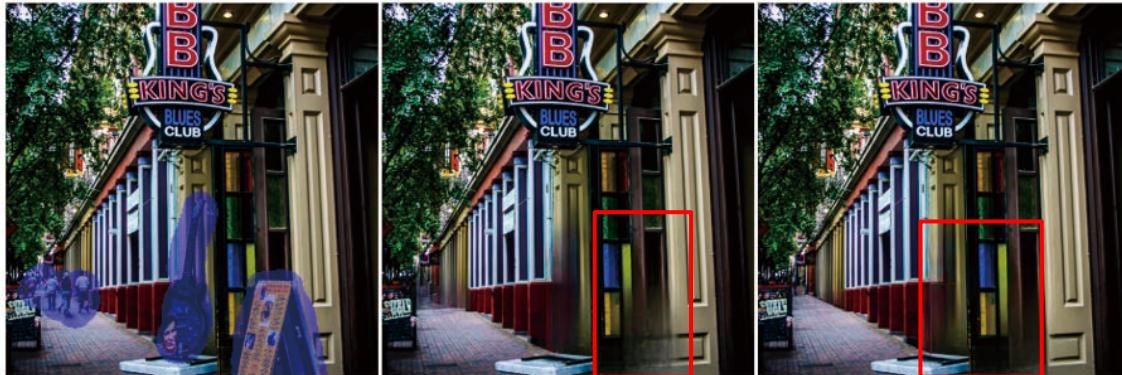
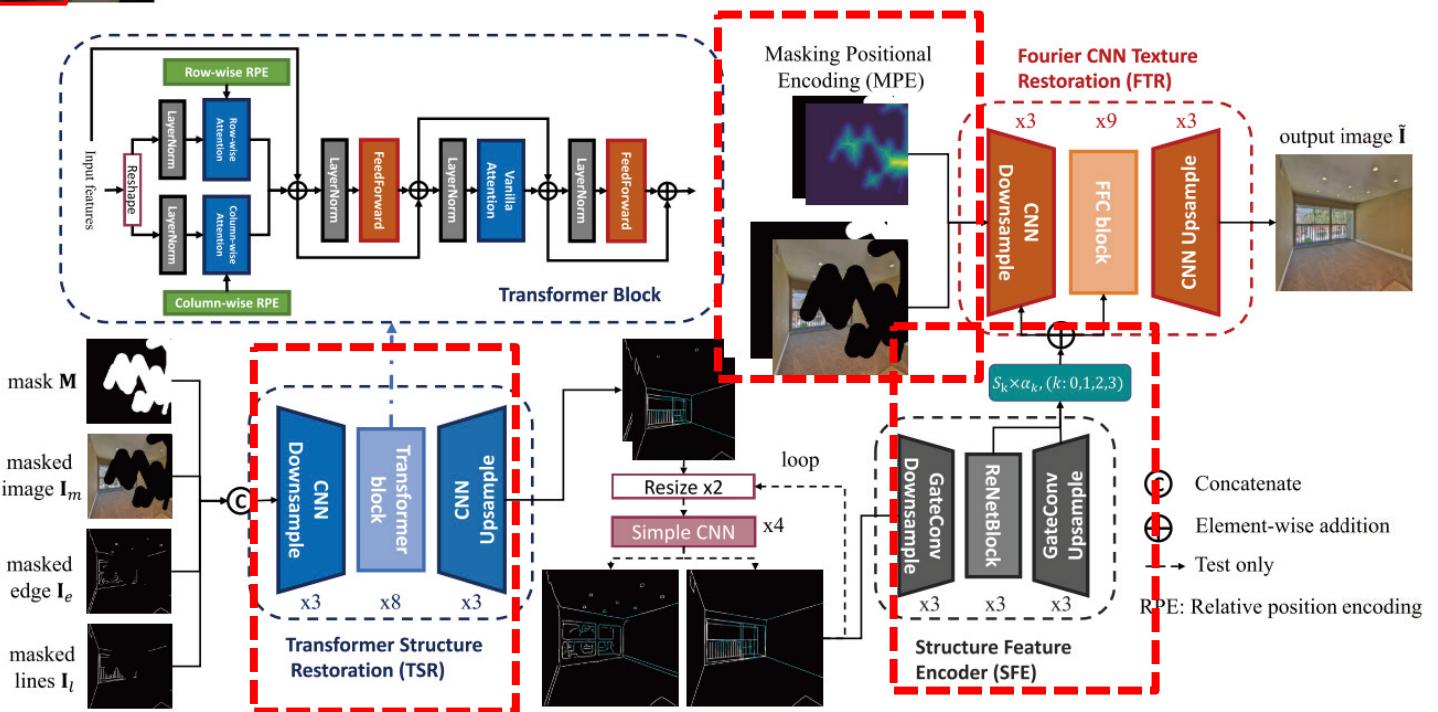
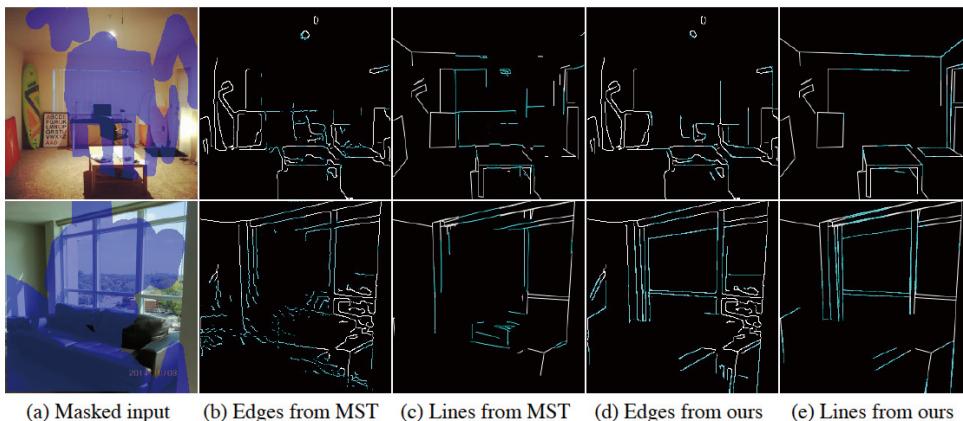


Figure 4. Qualitative results w. and w/o. lines in ShanghaiTech.

# ZITS: ZeroRA based Incremental Transformer Structure



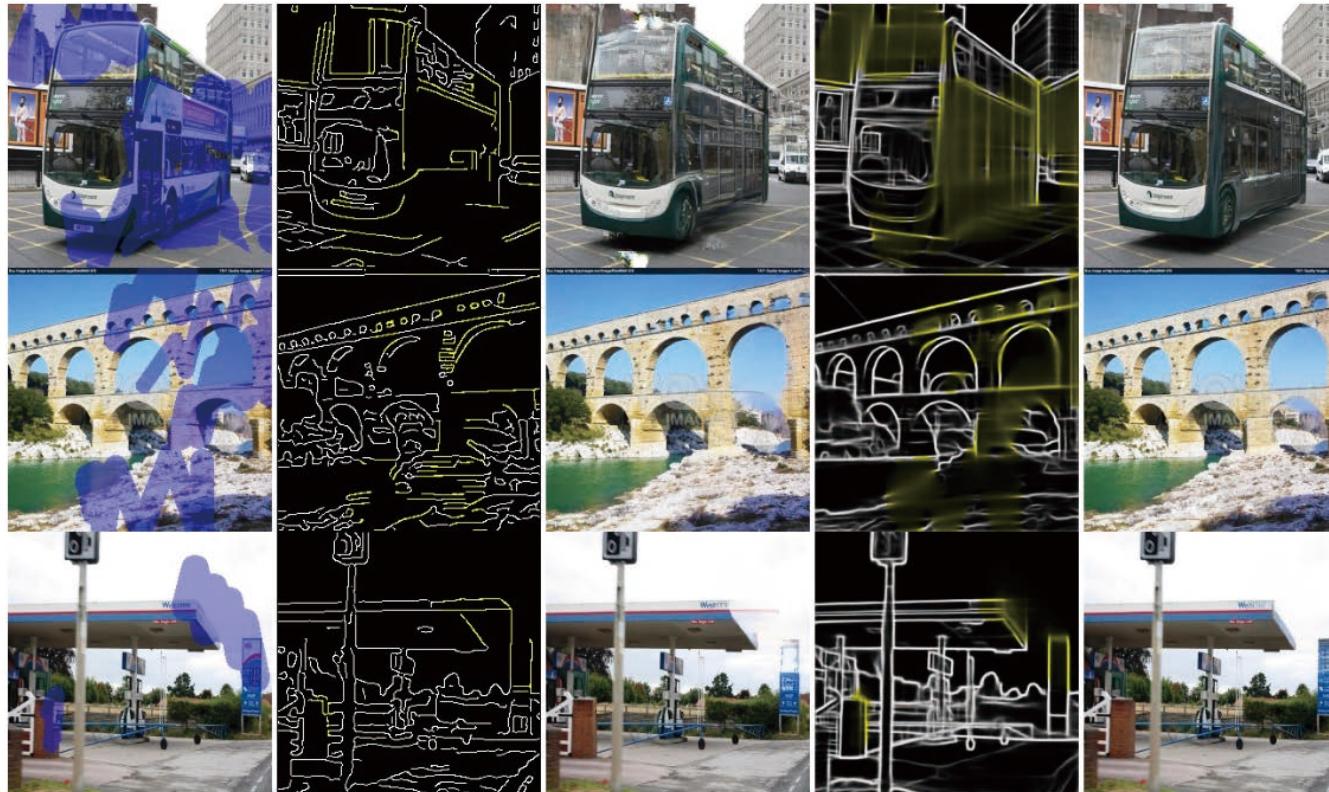
- Incremental Transformer Structure for filling line priors
- Novel Masking Positional Encoding
- Incrementally training strategy to utilize pretrained LaMa.



[https://github.com/DQiaole/ZITS\\_inpainting](https://github.com/DQiaole/ZITS_inpainting)

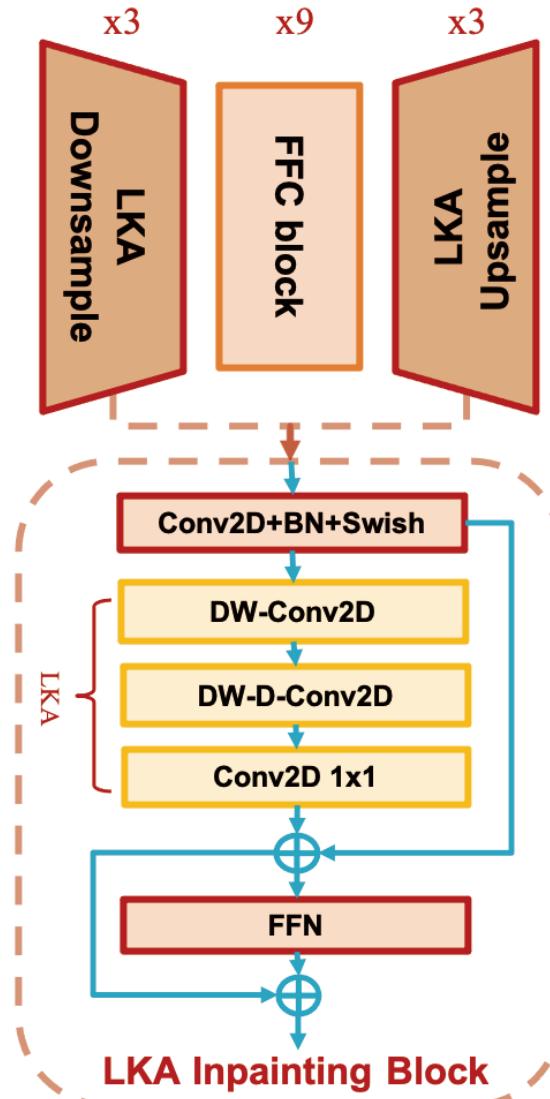
# ZITS++: Improved Incremental Transformer on Structural Priors

- ▶ Using Learning based Edges instead of Canny edge.
- ▶ Further improve the FTR training with large kernel attention (LKA)
- ▶ Discussing more about different priors for image inpainting



ZITS++, in submission to a journal

**Fourier CNN Texture Restoration with LKA (FTR)**



# High resolution results of ZITS++

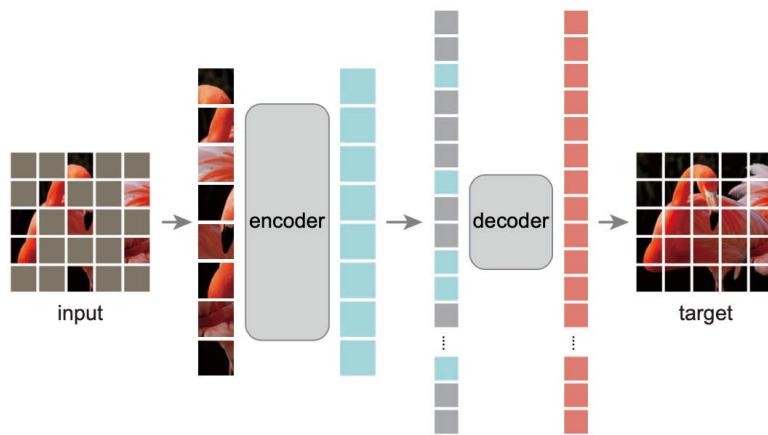


(f) High-resolution inpainting results compared with LaMa (first) and our ZITS++ (second).

1K and 4K resolution

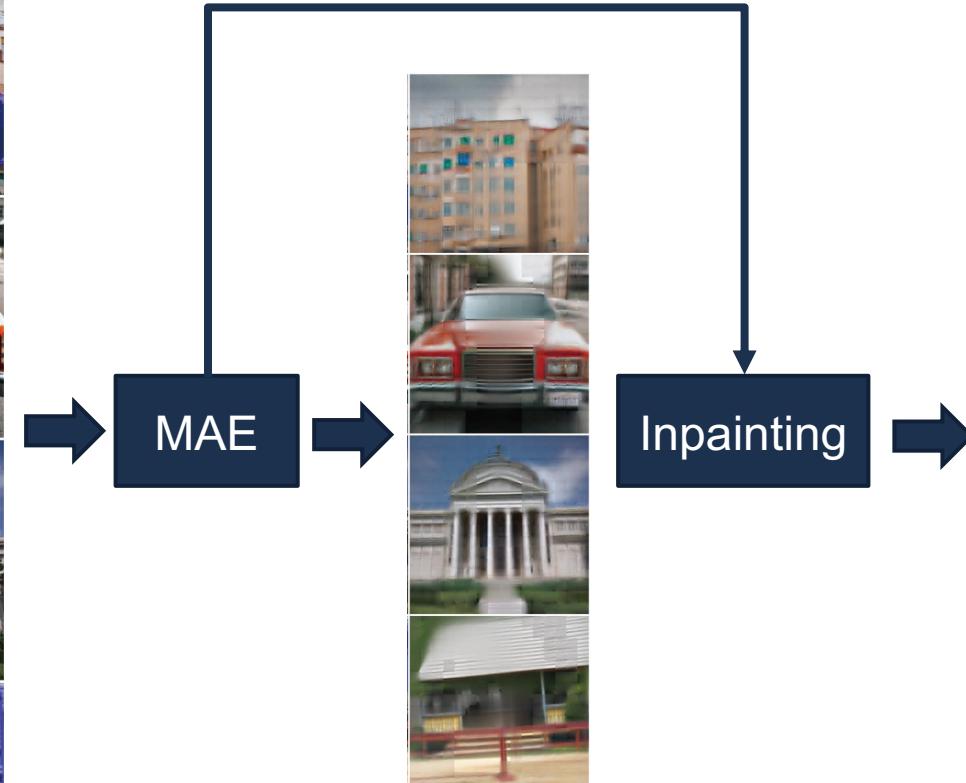
# MAE-FAR: Learning Prior Feature and Attention Enhanced Image Inpainting

Our model provides **proper priors** for  
Image inpainting with pre-trained MAE.



**Masked AutoEncoder (MAE):** A vision transformer that is pre-trained with 75% random masking prediction

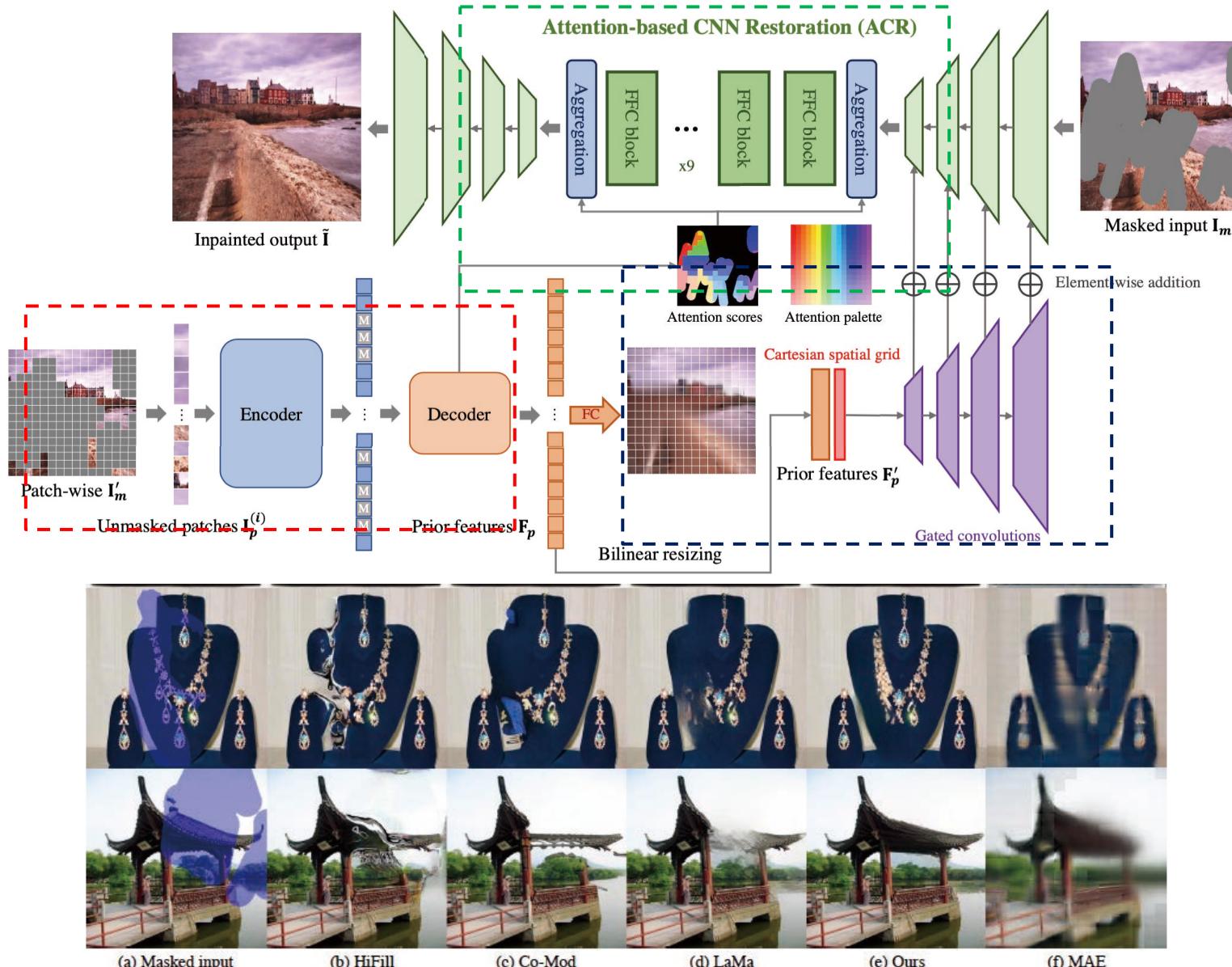
<https://github.com/ewrfcas/MAE-FAR>



MAE results



# Overview





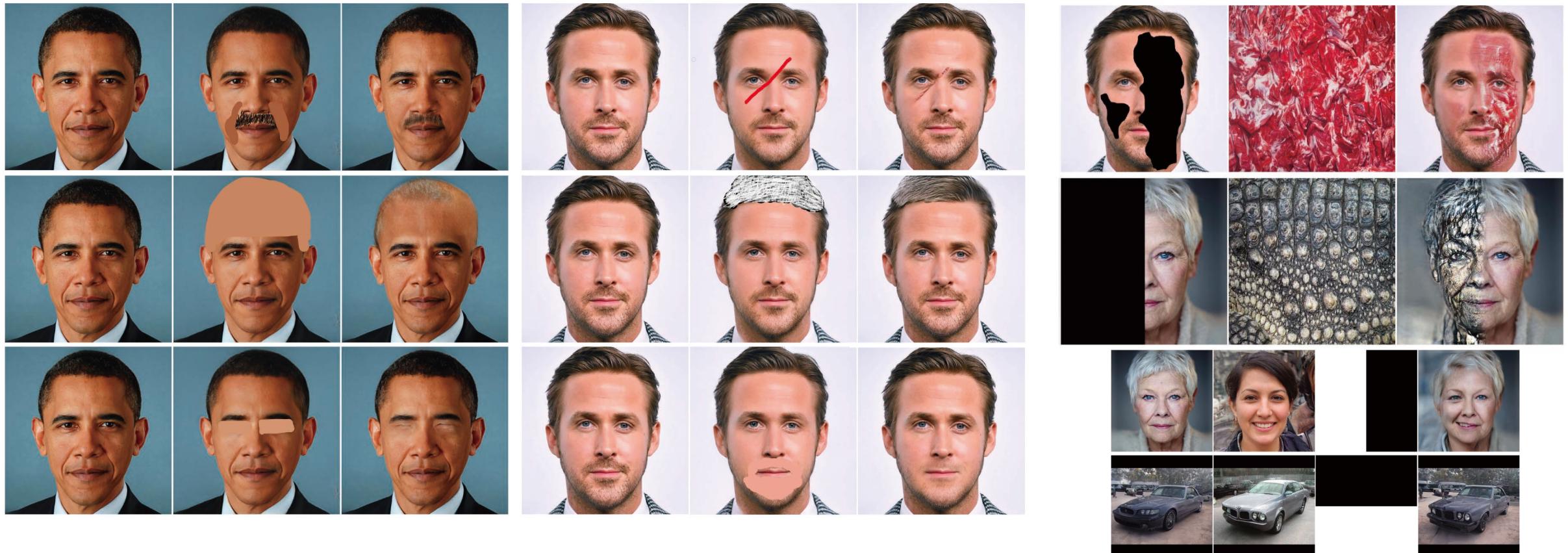
# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion

# Image2StyleGAN

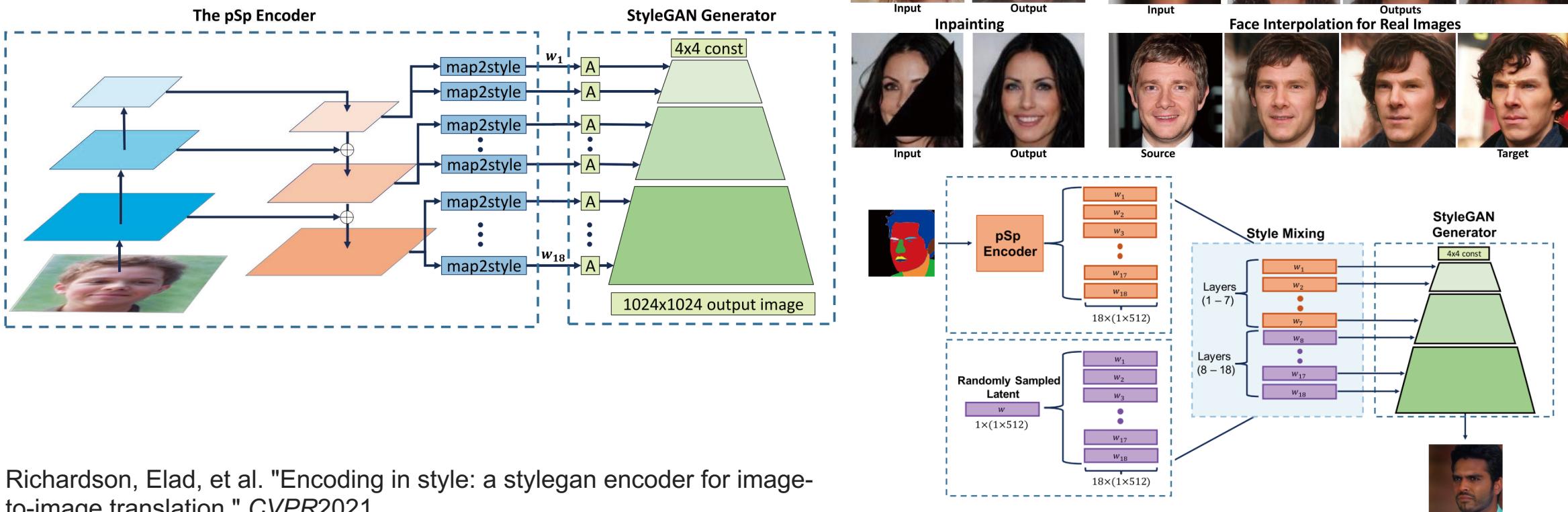
- ▶ Optimizing the latent space for an effective editing
- ▶ Embedding works well into the extended latent space  $W^+$



Rameen Abdal et al. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? ICCV2019  
Rameen Abdal et al. Image2StyleGAN++: How to Edit the Embedded Images? CVPR2020

# Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation

- ▶ Using a pSp Encoder to learn for a latent space  $W^+$
- ▶ Combining with random latent codes and fuse for the mixed styles for stylegan



# Portrait Editing by Differentiable Guided Sketches from Latent Space



(a) Input



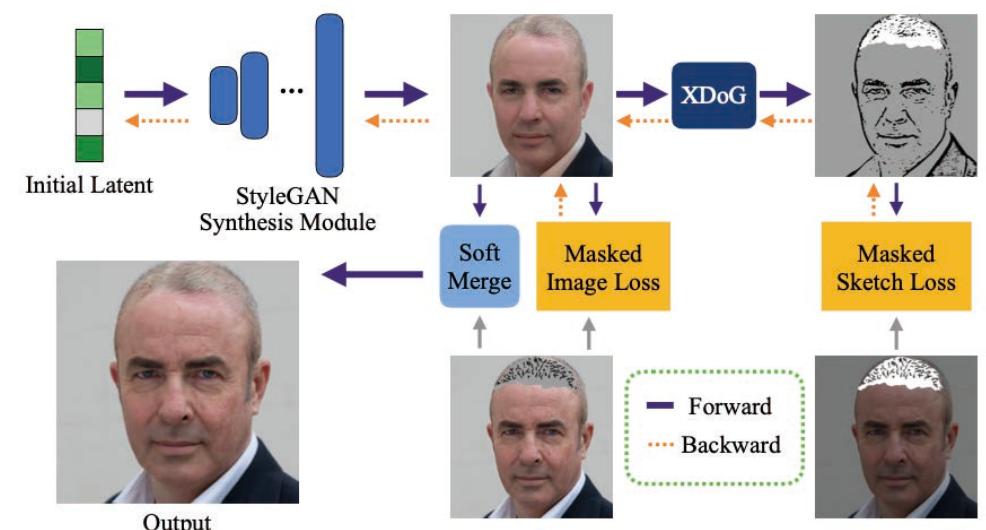
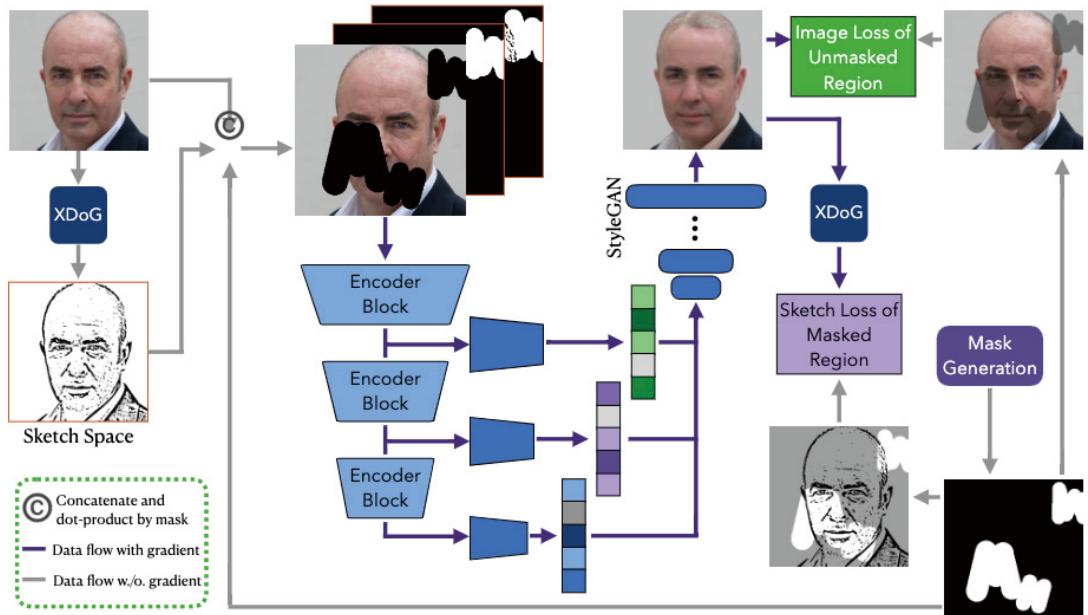
(b) Result by SC-FEGAN



(c) Result by DeepPS



(d) Result by our method



Unify both latent encoding and optimizing with differentiable sketch (XDoG operator)



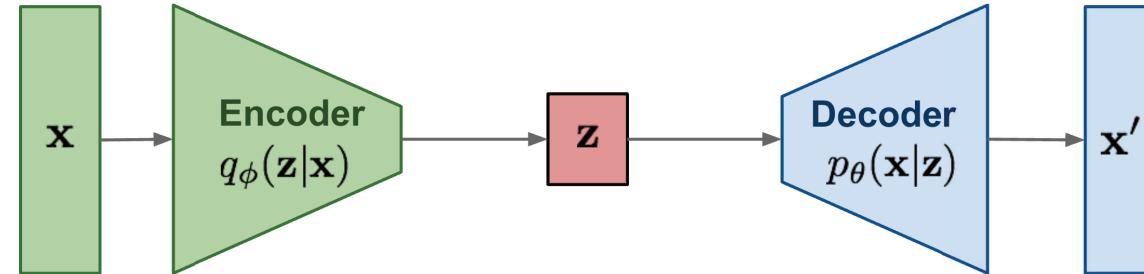
# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion

# Variational Auto-Encoder (VAE)

**VAE:** maximize variational lower bound



Generative model:  $p_{\theta}(\mathbf{z})p_{\theta}(\mathbf{x}|\mathbf{z})$

Variational approximation  $q_{\phi}(\mathbf{z}|\mathbf{x})$  to the posterior  $p_{\theta}(\mathbf{z}|\mathbf{x})$

$$p_{\theta}(z|x) \cong q_{\phi}(z|x)$$

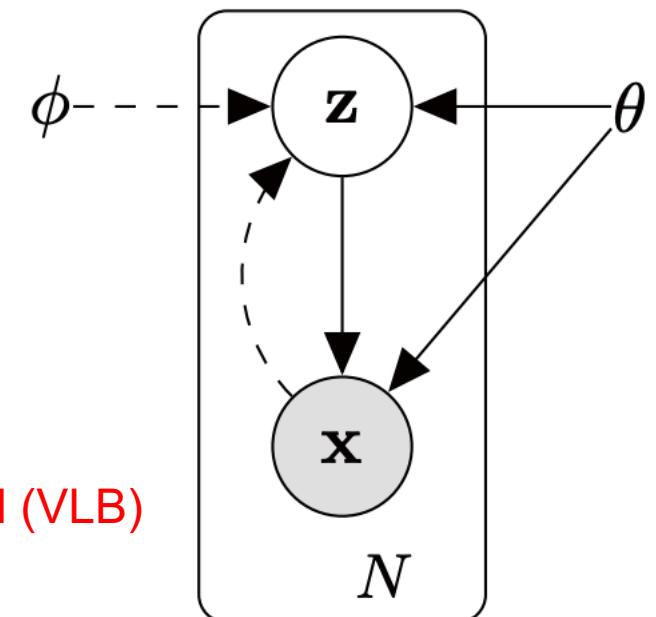
$$\log p_{\theta}(\mathbf{x}^{(i)}) = D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)}) || p_{\theta}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$$

Const

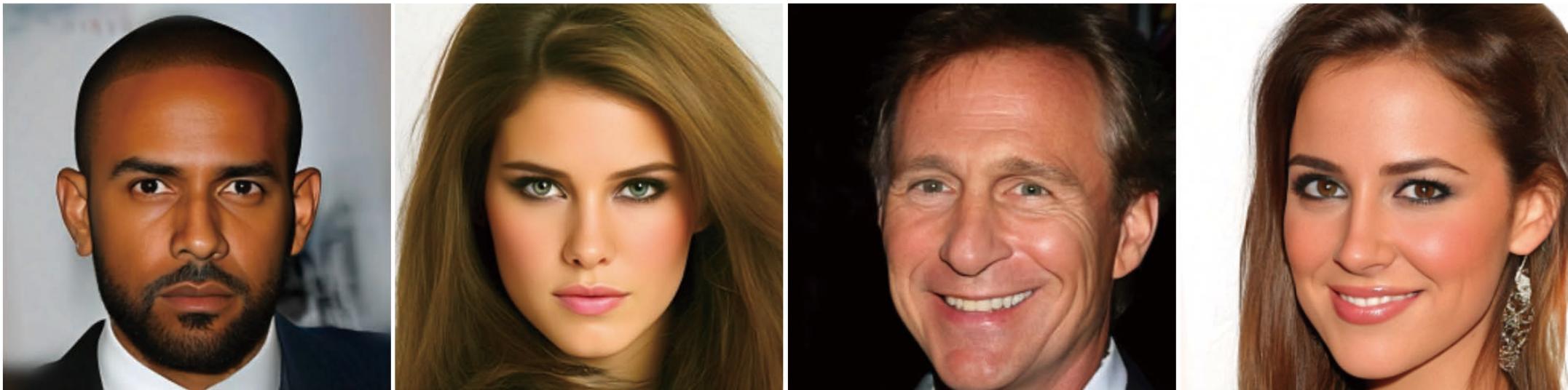
Non-negative

Variational Lower Bound (VLB)

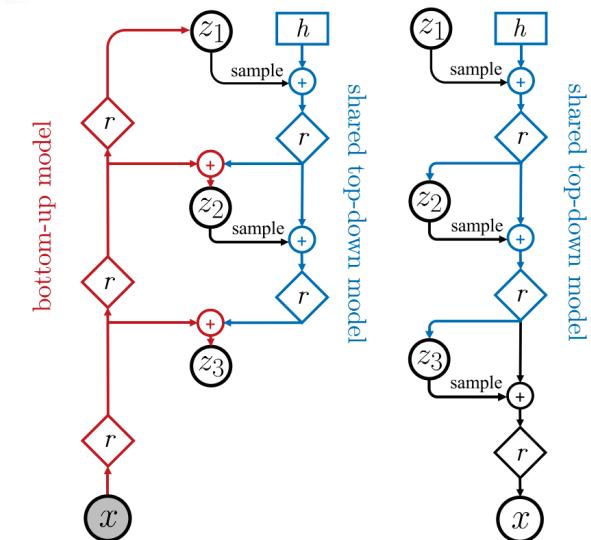
So when VLB is larger, KL is lower



# NVAE: A Deep Hierarchical Variational Autoencoder

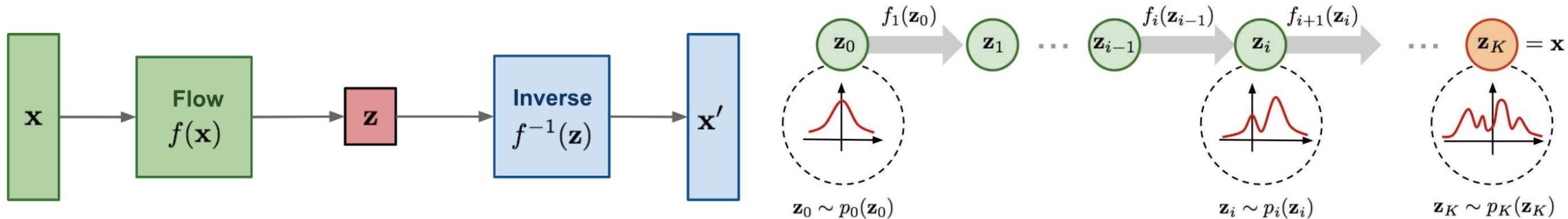


- ▶ VAE: Difficult to approximate  $p(z|x), q(z|x)$ . Unstable to train very deep hierarchical VAE.
- ▶ NVAE:
  - ▶ Autoregressive and multi-scale residual normal distribution
  - ▶ Many architecture improvements
  - ▶ Scaling up the model design



(a) Bidirectional Encoder (b) Generative Model

# Flow-based Generative Model



**Normalizing Flows:** A normalizing flow transforms a simple distribution into a complex one by applying a sequence of **invertible** transformation functions

$$p_i(\mathbf{z}_i) = p_{i-1}(f_i^{-1}(\mathbf{z}_i)) \left| \det \frac{df_i^{-1}}{d\mathbf{z}_i} \right|$$

- ▶ NICE: Affine coupling layer without the scale term, known as additive coupling layer
- ▶ RealNVP: Stacking a sequence of invertible bijective transformation functions
- ▶ Glow: Replacing the reverse operation on the channel ordering with invertible 1x1 convolutions

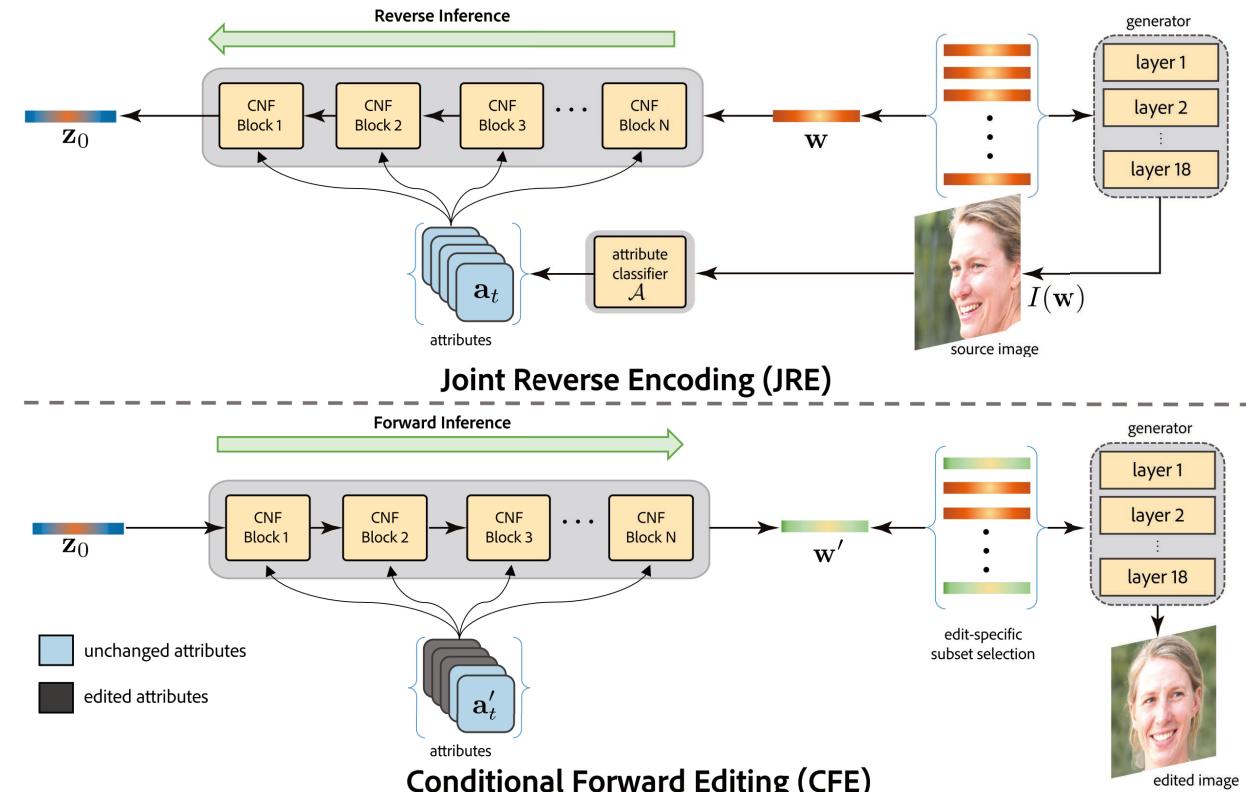
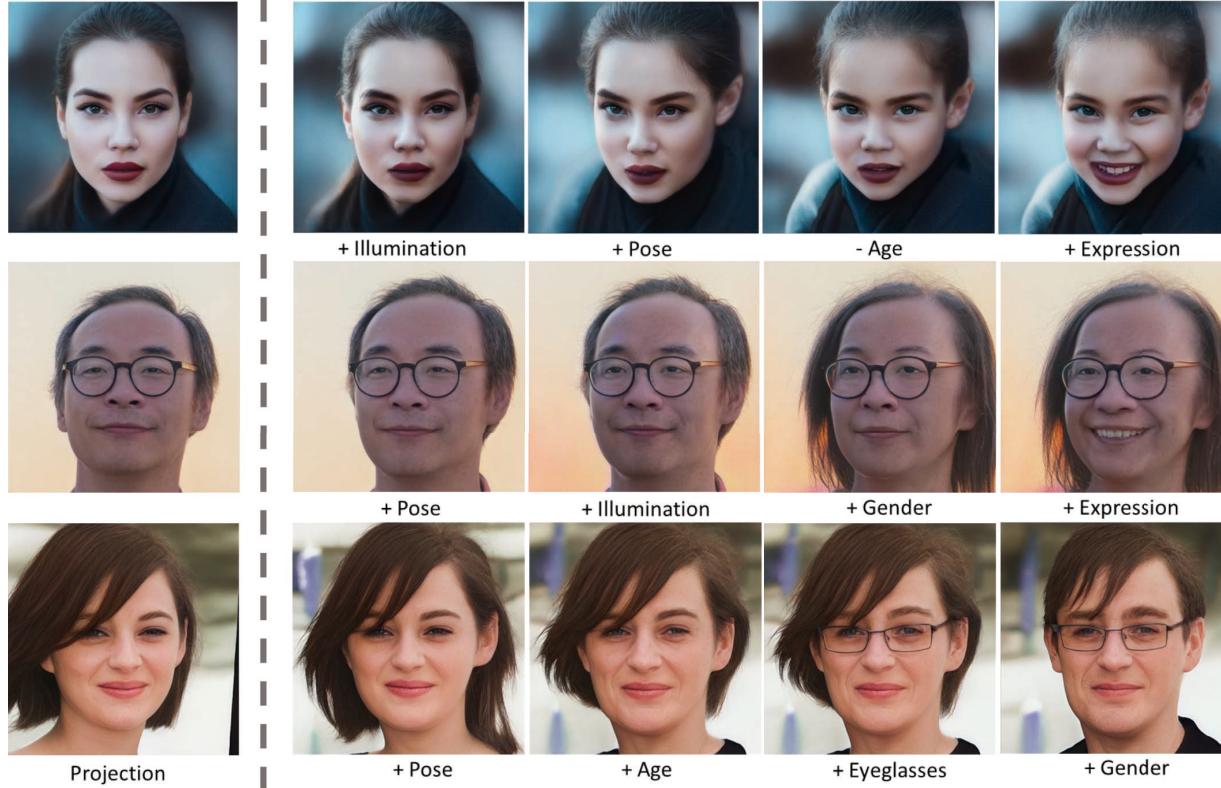
<https://lilianweng.github.io/posts/2018-10-13-flow-models/>

Dinh L, Krueger D, Bengio Y. Nice: Non-linear independent components estimation. ICLR2015.

Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using real nvp. ICLR2017.

Kingma D P, Dhariwal P. Glow: Generative flow with invertible 1x1 convolutions. NeurIPS2018.

# StyleFlow: Normalizing Flow and StyleGAN



- ▶ Flow enjoys good performance when combined with the stylegan through the conditional continuous normalizing flow



# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion

# SOTA Synthesis Results from Transformers with VQGAN

Parti



*A warrior wombat holding a sword and shield in a fighting stance. The wombat stands in front of the Arc de Triomphe on a day shrouded mist with the sun high in the sky. realistic anime illustration.*



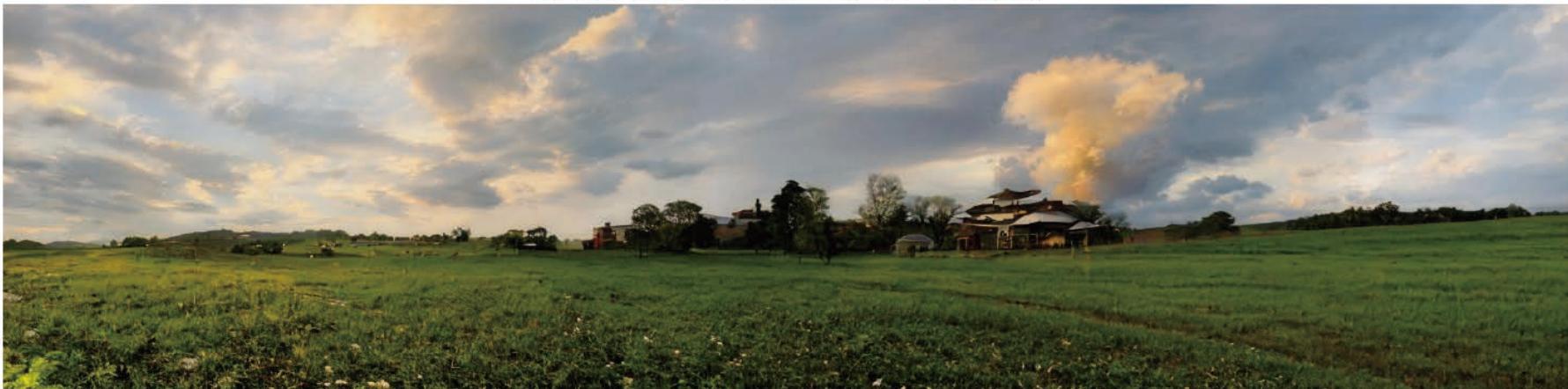
*A sloth in a go kart on a race track. The sloth is holding a banana in one hand. There is a banana peel on the track in the background. DSLR photograph.*



*A robot with a black visor and the number 42 on its chest. It stands proudly in front of an F1 race car. The sun is setting on a cityscape in the background. wide-angle view. comic book illustration.*

Input: a field with a house and a cloudy sky

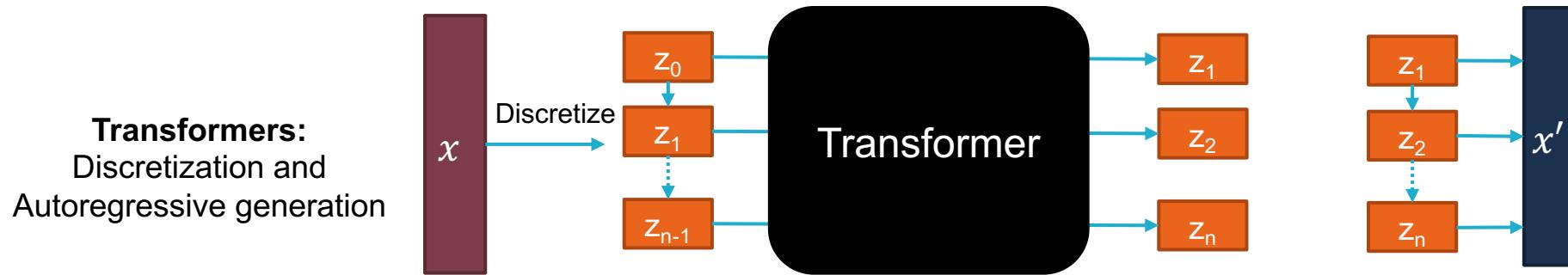
NUWA-infinity



[1] Yu, Jiahui, et al. "Scaling autoregressive models for content-rich text-to-image generation." arXiv preprint arXiv:2206.10789 (2022).

[2] Wu, Chenfei, et al. "NUWA-Infinity: Autoregressive over Autoregressive Generation for Infinite Visual Synthesis." arXiv preprint arXiv:2207.09814 (2022).

# Transformer for Image Generation



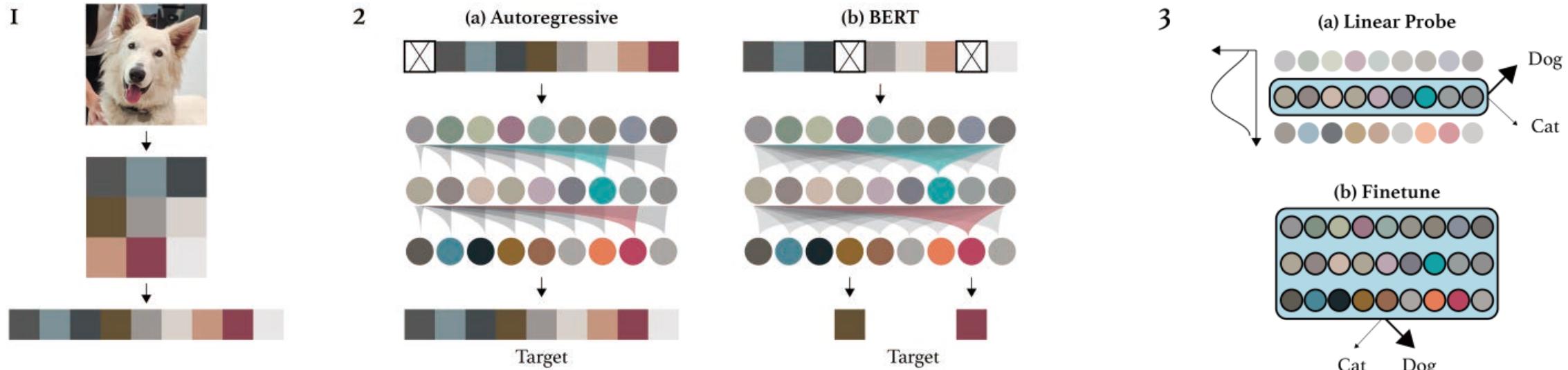
- ▶ Some classical works without discretization (**inefficient!**) :
- ▶ PixelCNN [1]
- ▶ Image Transformer[2]

[1] Van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders." *NIPS* 2016.

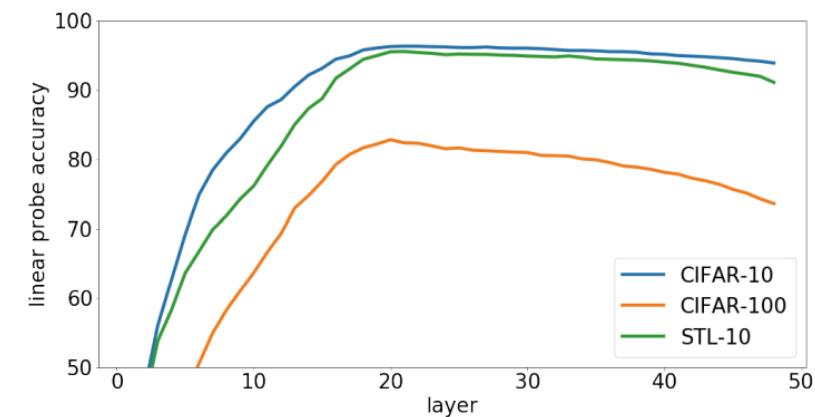
[2] Parmar, Niki, et al. "Image transformer." International conference on machine learning. PMLR, 2018

# iGPT: Generation Task for Model Pretraining

Discretizing RGB to 512 color index with K-means



- Autoregressive:  $p(x) = \prod_{i=1}^n p(x_{\pi_i} | x_{\pi_1}, \dots, x_{\pi_{i-1}}, \theta)$   
 $L_{AR} = \mathbb{E}_{x \sim X} [-\log p(x)]$
- BERT:  $L_{BERT} = \mathbb{E}_{x \sim X} \mathbb{E}_M \sum_{i \in M} [-\log p(x_i | x_{[1,n] \setminus M})]$



# DALLE: Zero-Shot Text-to-Image Generation



(a) a tapir made of accordion.  
a tapir with the texture of an  
accordion.

(b) an illustration of a baby  
hedgehog in a christmas  
sweater walking a dog

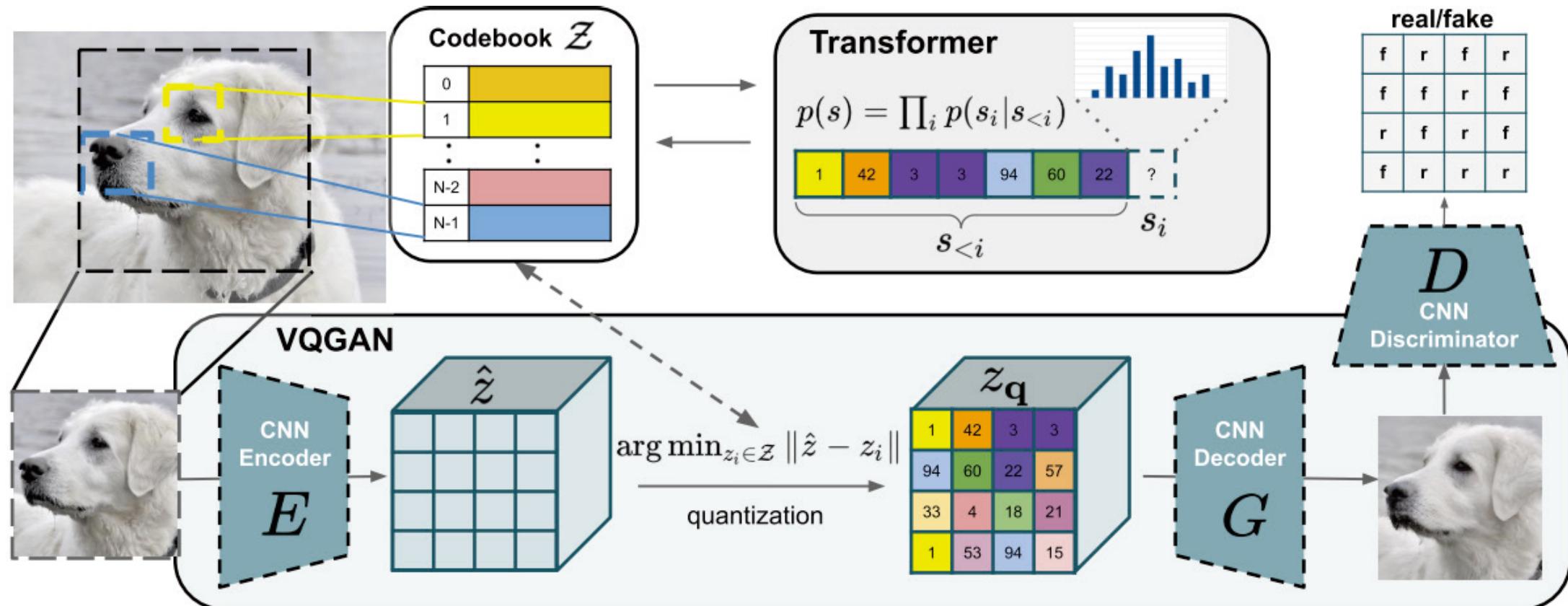
(c) a neon sign that reads  
“backprop”. a neon sign that  
reads “backprop”. backprop  
neon sign

(d) the exact same cat on the  
top as a sketch on the bottom

- Stage 1. Training a **discrete Variational AutoEncoder (dVAE)**
  - compress each  $256 \times 256$  image into a  $32 \times 32$  grid of image tokens, assuming 8192 possible values
- Stage 2. Concatenating 256 BPE-encoded text tokens with the  $32 \times 32 = 1024$  image tokens, and train an autoregressive transformer to model the joint distribution over text and image tokens.

# Taming Transformers for High-Resolution Image Synthesis

- Learning an Effective Codebook of Image Constituents for Use in Transformers  
**(Discretization:VQVAE)**
- Using an Autoregressive Transformer to learn/reconstruct sequential codebook tokens
- Improve VQVAE to VQGAN with adversarial training



# VQVAE VS VQGAN with large downsample scale(f=16)

input



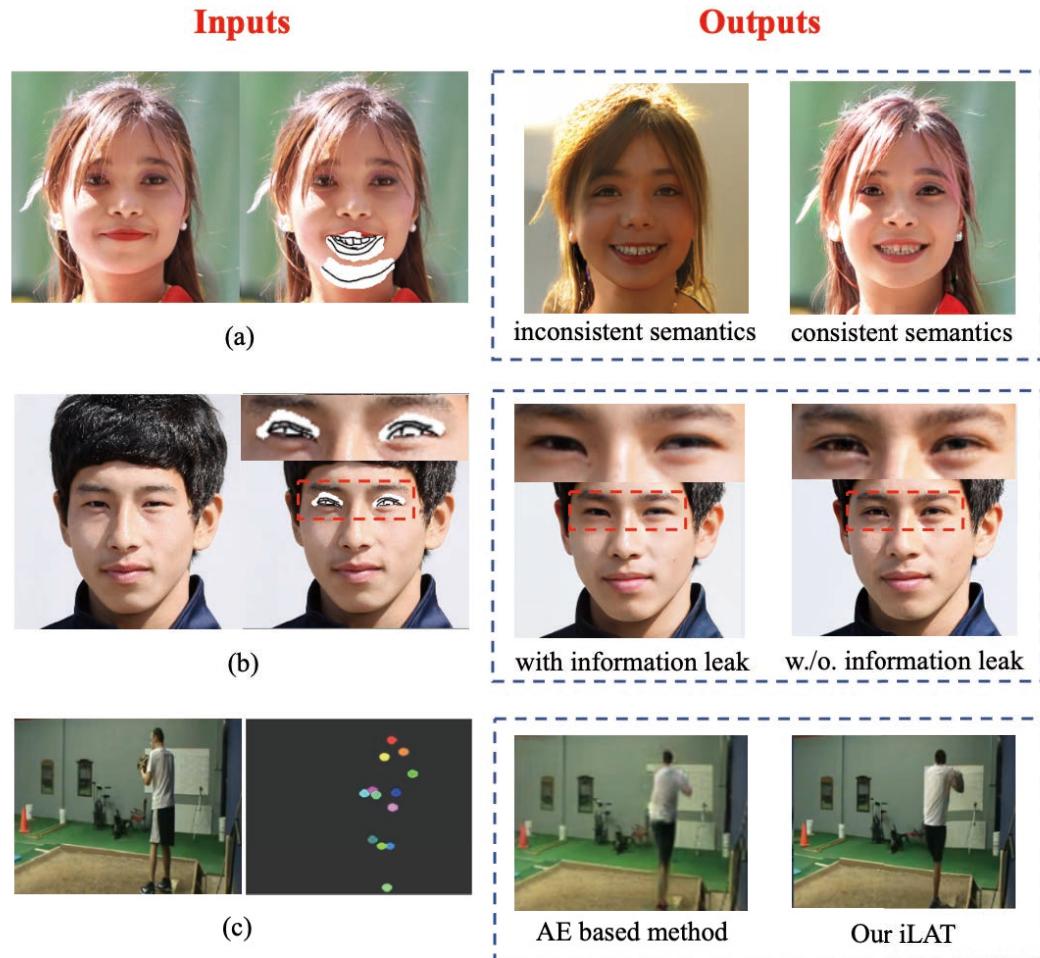
VQVAE [45]



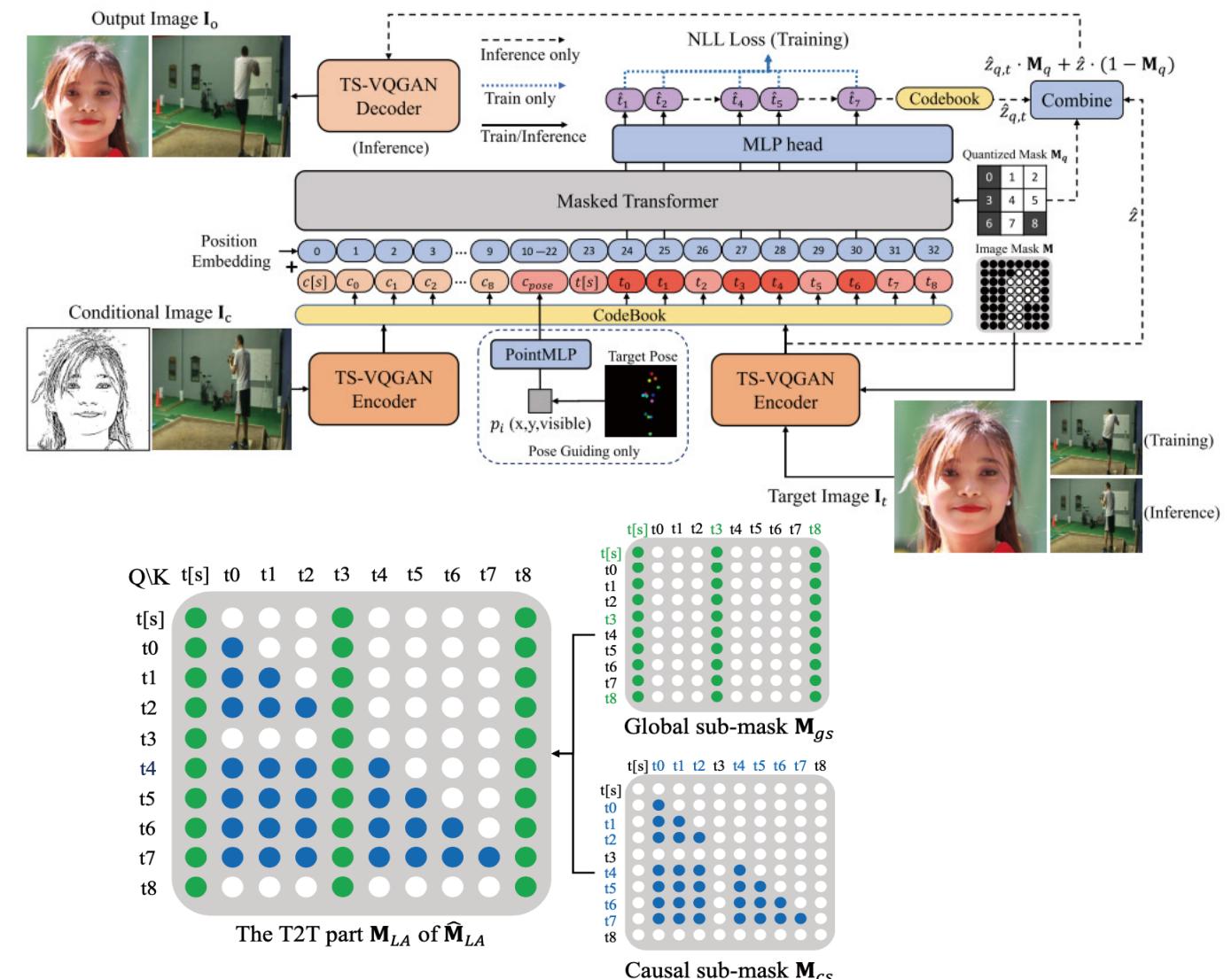
VQGAN (ours)



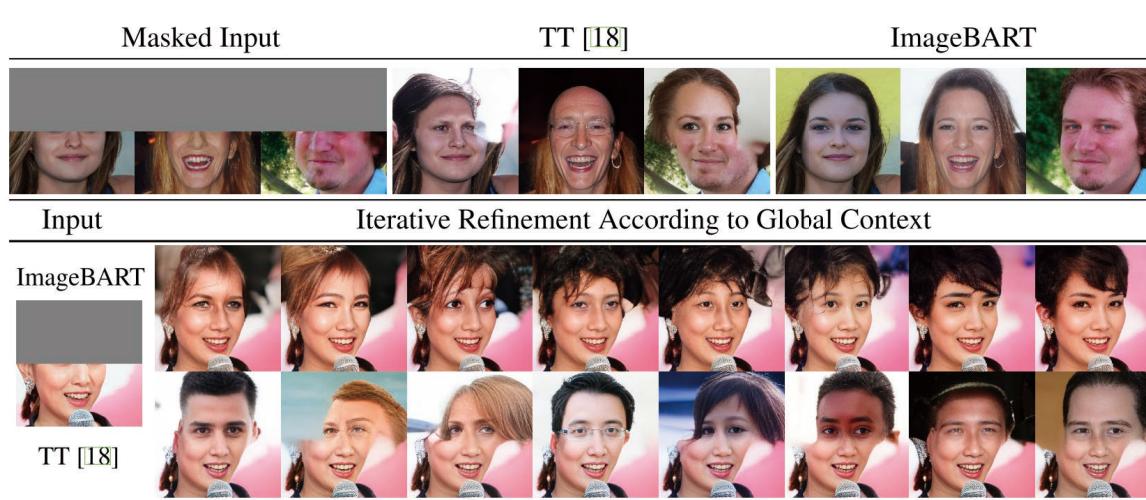
# Image Local Autoregressive Transformer



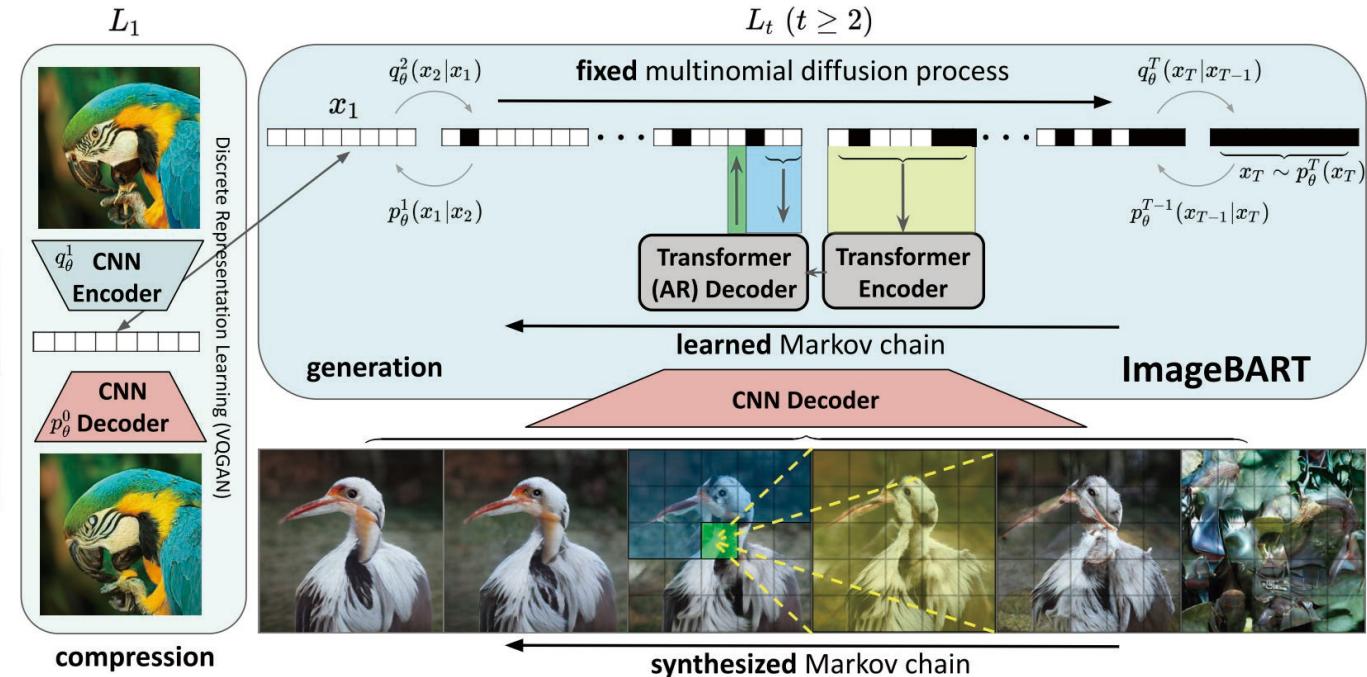
(A) Inputs and outputs of local generation compared with previous works



# ImageBART



Taming Transformer (TT) Esser, Patrick et al. CVPR2021



- ▶ Learning a compact, discrete representation for images (VQGAN)
- ▶ Multinomial diffusion for discrete sequence (extended from binomial diffusion)
- ▶ Diffusion encoder: bidirectional transformer (more layers)
- ▶ Diffusion decoder: autoregressive transformer (less layers)

# ManiTrans

Entity-level Text-guided Image Manipulation

**Text**

Horse. → Zebra.

Shirt. → Trees.

Cat on the plate.  
→  
Sandwiches on the plate.

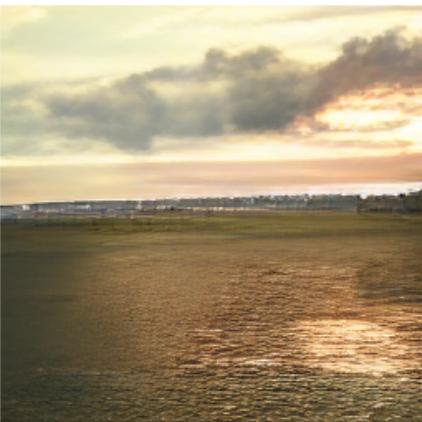
Grass. → River.

Street.  
→  
Snowy Street.

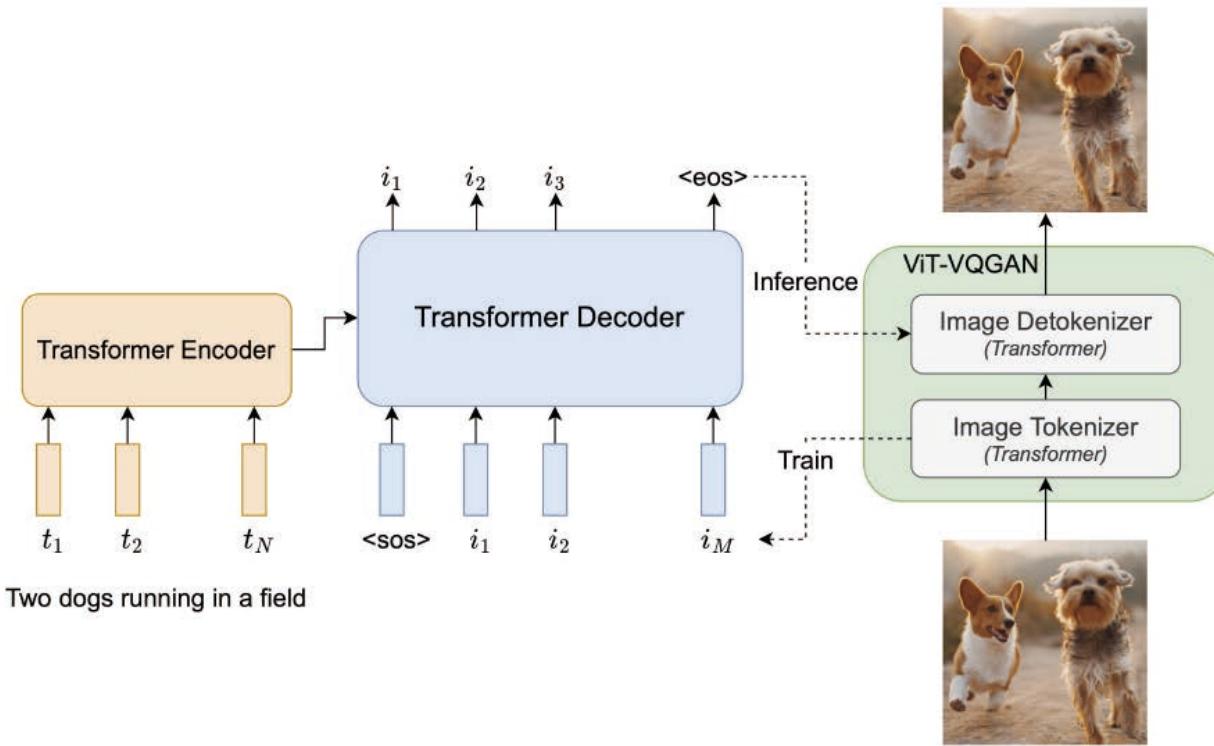
**Original Image**



**ManiTrans**



# Parti: Pathways Autoregressive Text-to-Image Model



- Use ViT-VQGAN with L2-normalization codes and factorized codes
- At the model scale of 350-million to 750-million parameters, the encoder-decoder variants of Parti outperformed decoder-only ones
- Pretrain the text encoder on large datasets



**If model capacities are large enough, autoregressive models are still competitive compared with diffusion models!**

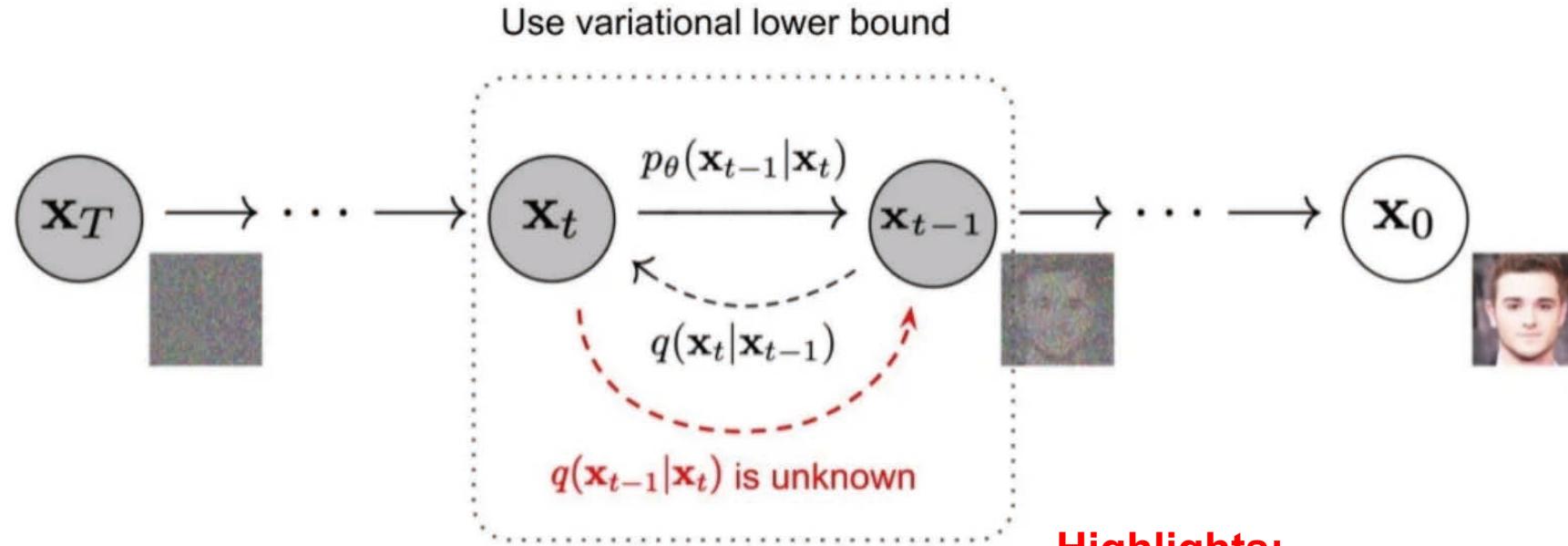


# Contents

---

- ▶ Tasks and Motivation
- ▶ Image Synthesis/Generation Methods
- ▶ GAN
  - ▶ Inpainting with GAN
  - ▶ GAN inversion
- ▶ VAE and Flow
- ▶ Transformer
- ▶ Diffusion

# Denoising Diffusion Probabilistic Models



Denoising diffusion models consist of two processes:

- **Forward diffusion process:** gradually adding noise to input
- **Reverse denoising process:** learning to generate data by denoising

## Highlights:

- **Typically using U-net for implementation**
- **Conditional Diffusion Models**

# Denoising Diffusion Models

Emerging as powerful generative models, outperforming GANs

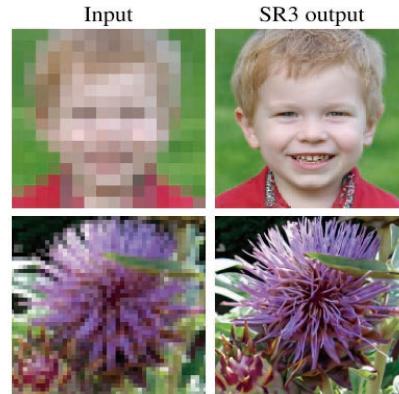


"Diffusion Models Beat GANs on Image Synthesis" Dhariwal & Nichol,  
OpenAI, 2021



"Cascaded Diffusion Models for High Fidelity Image Generation" Ho  
et al., Google, 2021

# Various Usages of Diffusion Models



## Super-resolution

Saharia, Chitwan, et al. "Image super-resolution via iterative refinement." (2021).



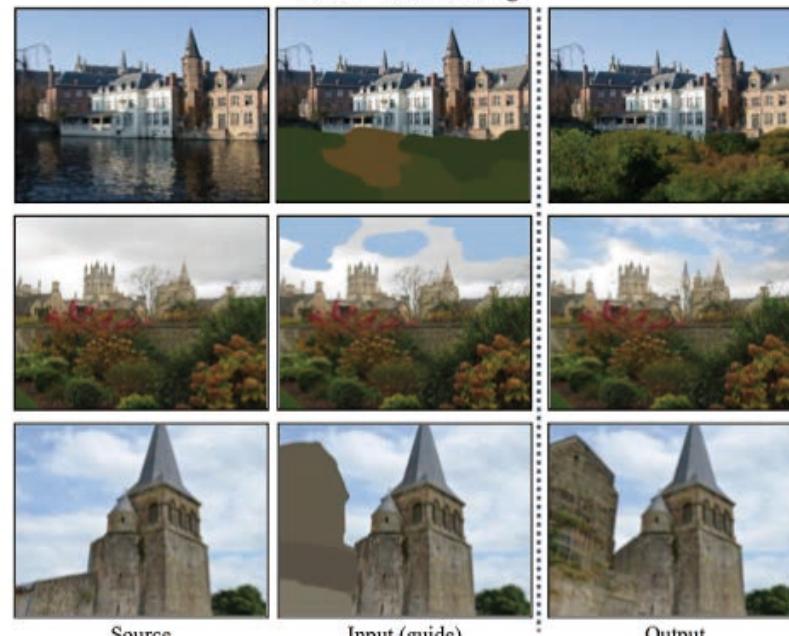
Portrait

Realistic Image

## Domain Transfer

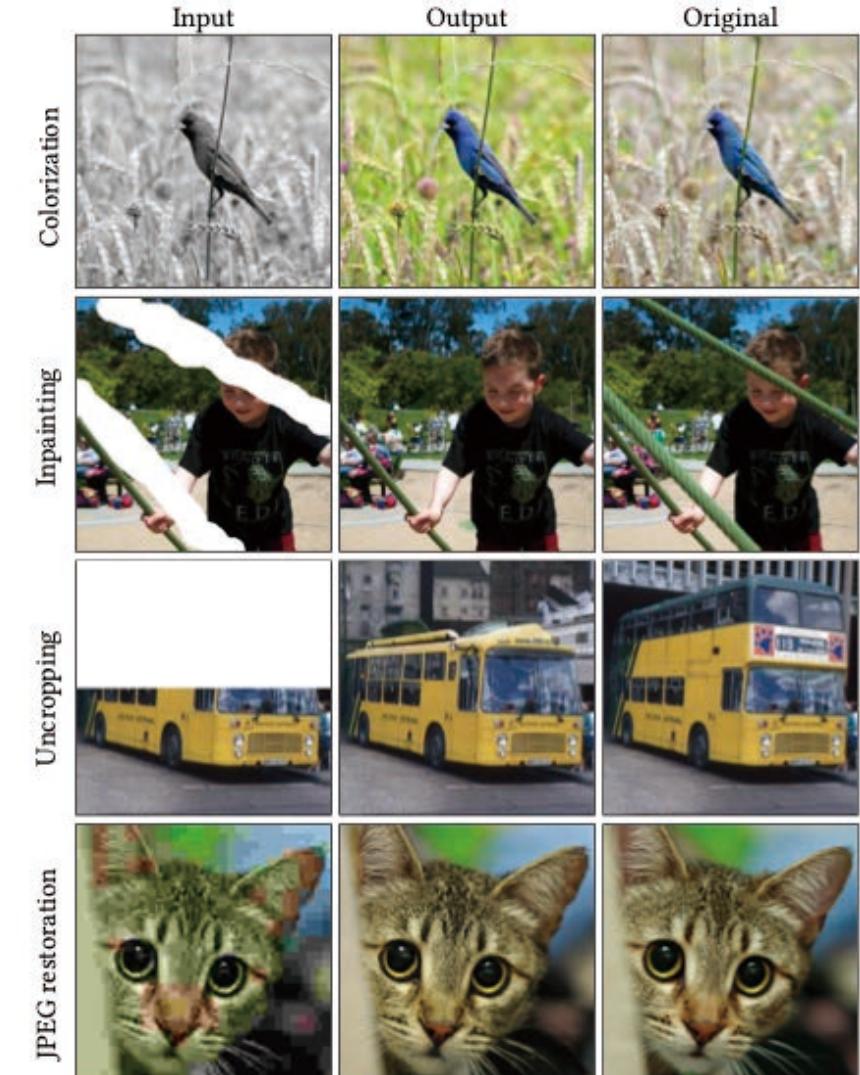
Choi, Jooyoung, et al. "Ilvr: Conditioning method for denoising diffusion probabilistic models." (2021).

### Stroke-based Editing



## Editing

Meng, Chenlin, et al. "Sdedit: Guided image synthesis and editing with stochastic differential equations." 2021.



## Multi-task (colorization, inpainting, restoration)

Saharia, Chitwan, et al. "Palette: Image-to-image diffusion models.". 2022.

# Text-to-Image Models

## DALL·E 2

"a teddy bear on a skateboard in times square"



"Hierarchical Text-Conditional Image Generation with CLIP Latents" Ramesh et al., 2022

## Imagen

A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

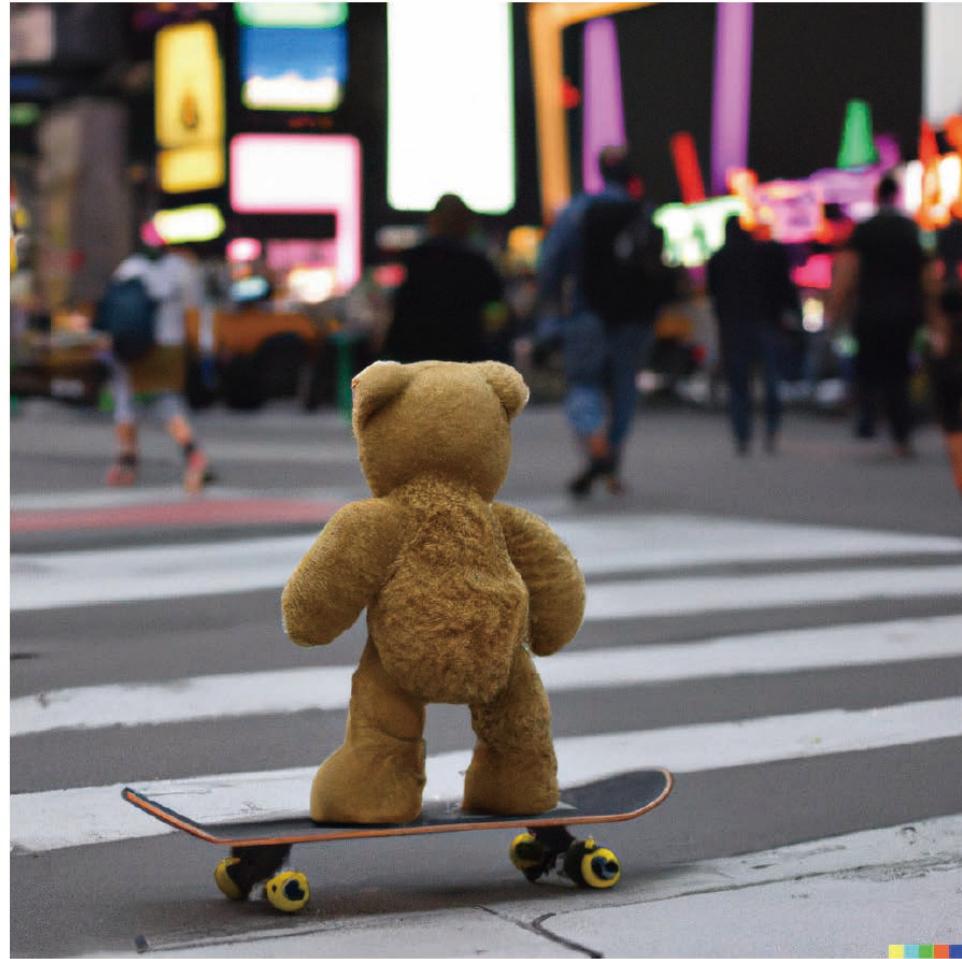


"Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", Saharia et al., 2022

# DALLE 2

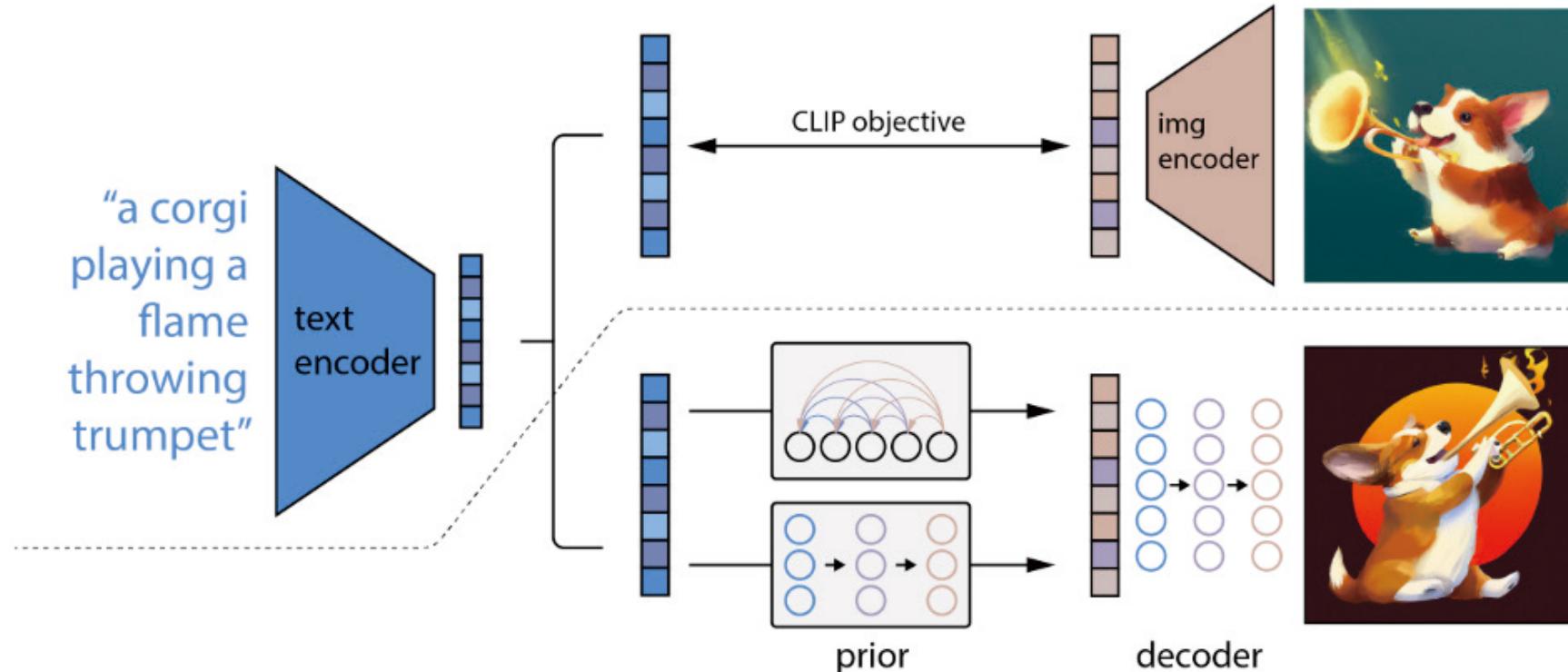


a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese



a teddy bear on a skateboard in times square

# DALLE 2 (unCLIP)



encode any given image  $x$  into a bipartite latent representation  $(z_i, x_T)$

- A *prior*  $P(z_i|y)$  that produces CLIP image embeddings  $z_i$  conditioned on captions  $y$ .
- A *decoder*  $P(x|z_i, y)$  that produces images  $x$  conditioned on CLIP image embeddings  $z_i$  (and optionally text captions  $y$ ).

# Imagen

**Pretrained text encoders are important for Text-to-Image!**

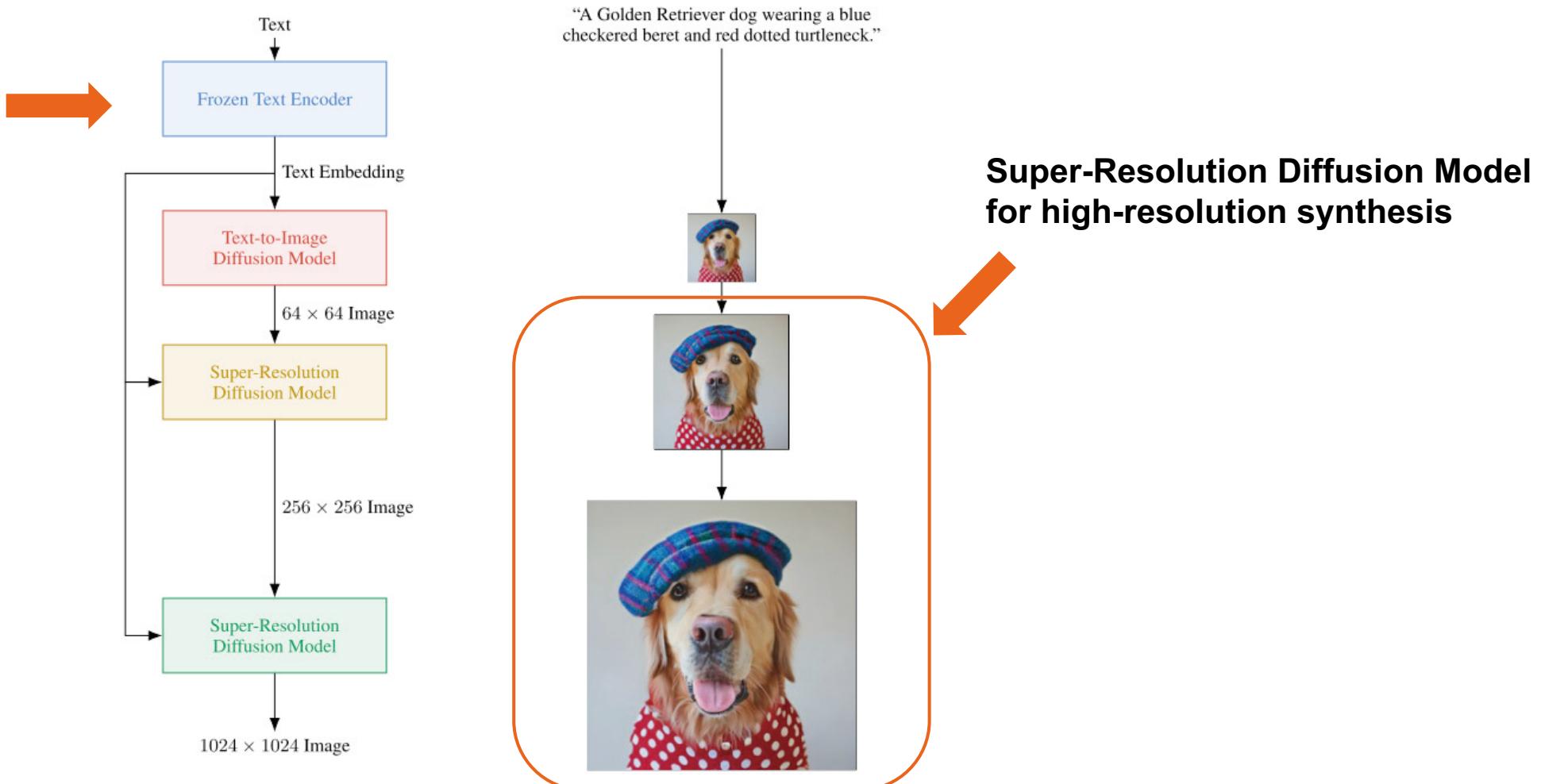
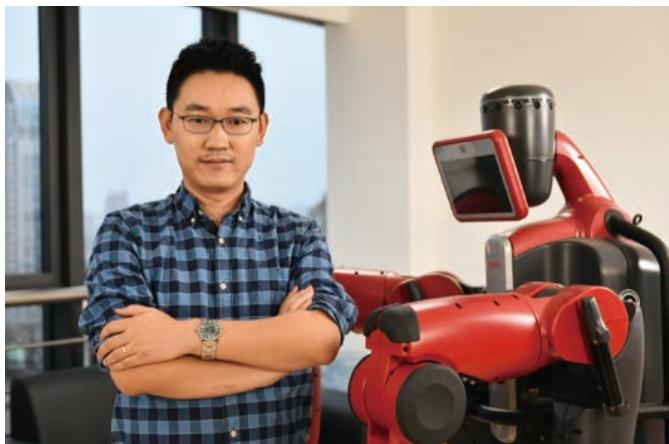


Figure A.4: Visualization of Imagen. Imagen uses a frozen text encoder to encode the input text into text embeddings. A conditional diffusion model maps the text embedding into a  $64 \times 64$  image. Imagen further utilizes text-conditional super-resolution diffusion models to upsample the image, first  $64 \times 64 \rightarrow 256 \times 256$ , and then  $256 \times 256 \rightarrow 1024 \times 1024$ .

Thanks!  
[yanweifu@fudan.edu.cn](mailto:yanweifu@fudan.edu.cn)

Dr. Yanwei Fu  
the School of Data Science,  
Fudan University



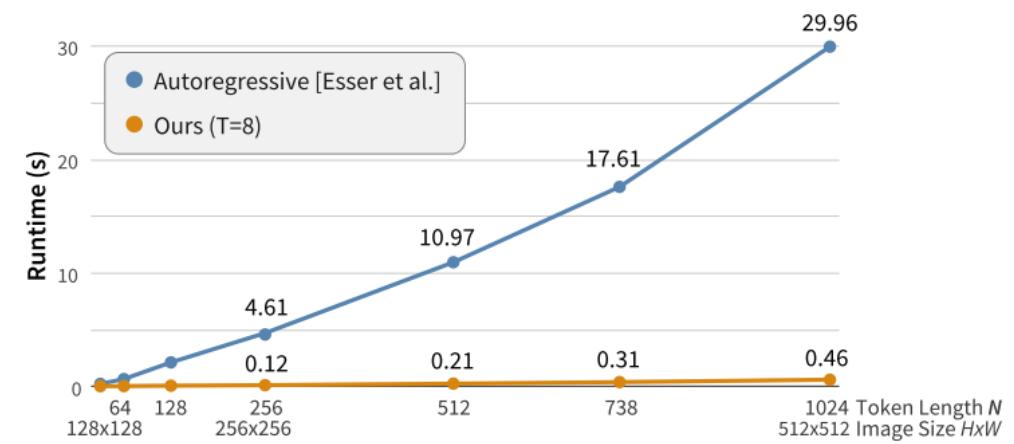
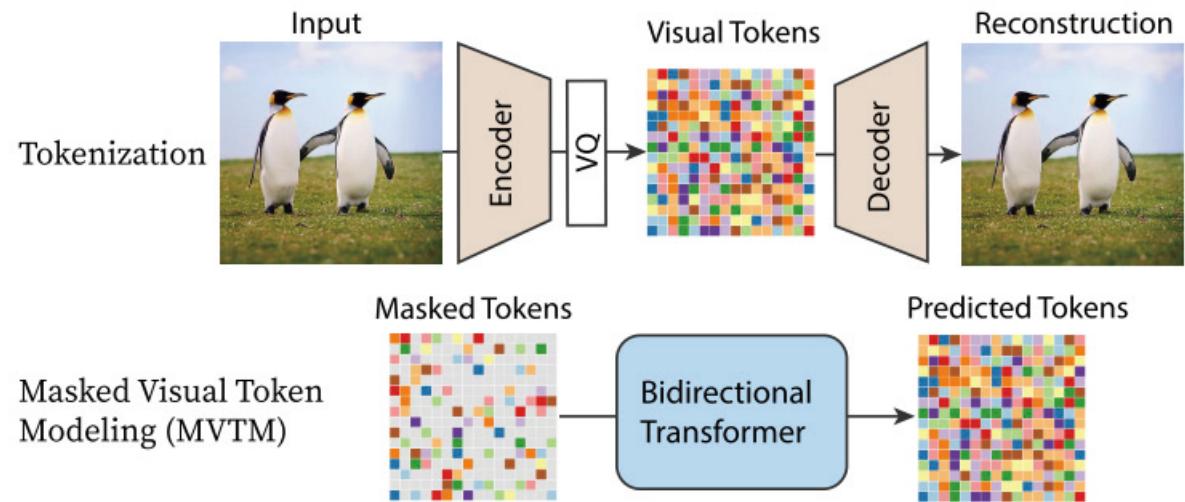
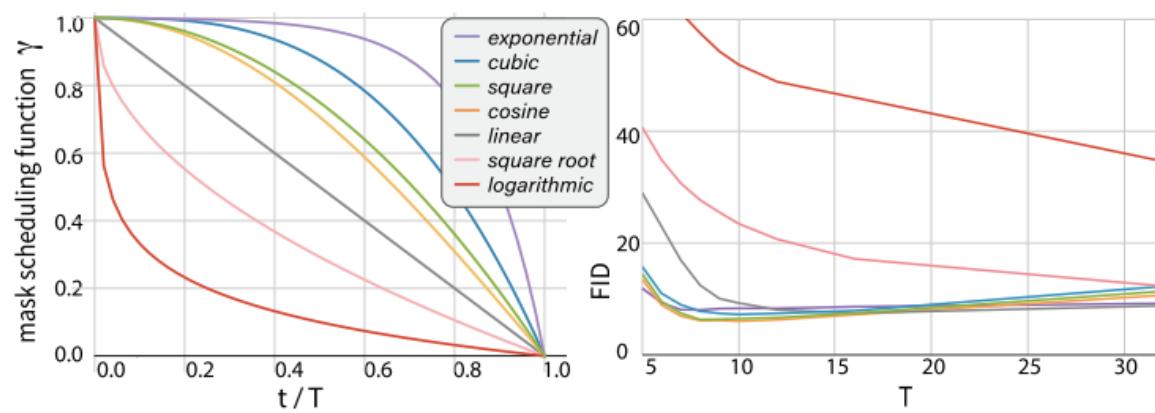
# MaskGIT: Masked Generative Image Transformer

- Bidirectional masked image prediction:

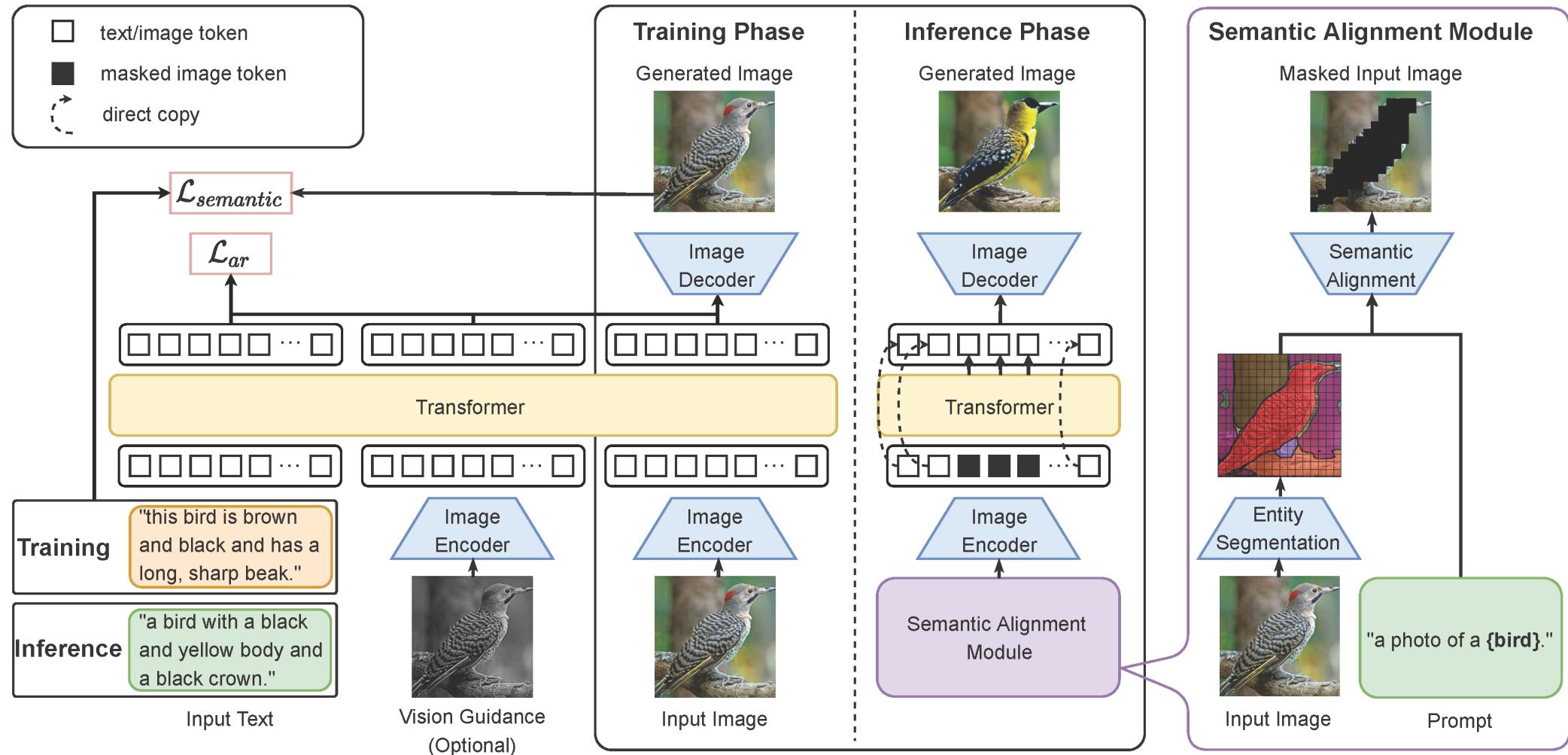
$$\mathcal{L}_{\text{mask}} = - \mathbb{E}_{Y \in \mathcal{D}} \left[ \sum_{\forall i \in [1, N], m_i=1} \log p(y_i | Y_{\overline{\mathbf{M}}}) \right]$$

- Iterative Decoding

- Predict** probabilities of all un-masked tokens.
- Sample** tokens based on probabilities
- Compute the number of tokens to mask according to the **mask scheduling**
- Re-mask** least-confidence tokens

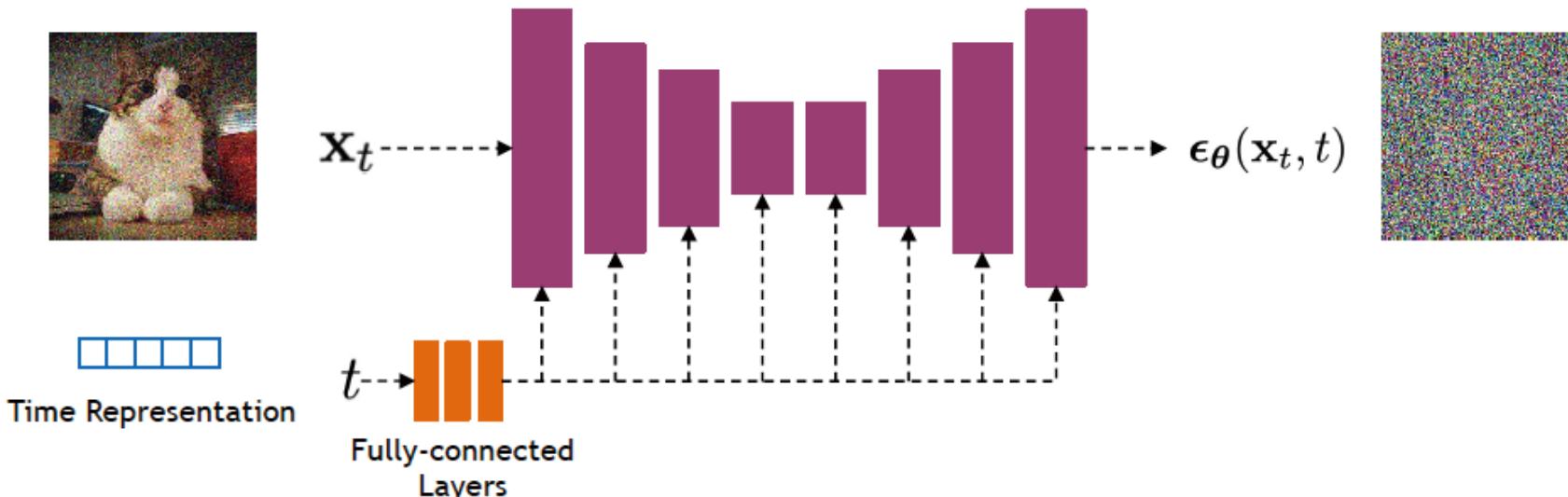


# ManiTrans



# Implementation Considerations

Diffusion models often use U-Net architectures with ResNet blocks and self-attention layers to represent  $\epsilon_\theta(\mathbf{x}_t, t)$



Time representation: sinusoidal positional embeddings or random Fourier features.

Time features are fed to the residual blocks using either simple spatial addition or using adaptive group normalization layers. (see [Dhariwal and Nichol NeurIPS 2021](#))

[Kingma et al. NeurIPS 2022](#) introduce a new parameterization of diffusion models using signal-to-noise ratio (SNR), and show how to learn the noise schedule by minimizing the variance of the training objective.

We can also train  $\sigma_t^2$  while training the diffusion model by minimizing the variational bound ([Improved DPM by Nichol and Dhariwal ICML 2021](#)) or after training the diffusion model ([Analytic-DPM by Bao et al. ICLR 2022](#)).

# Conditional Diffusion Models

- Classifier guidance

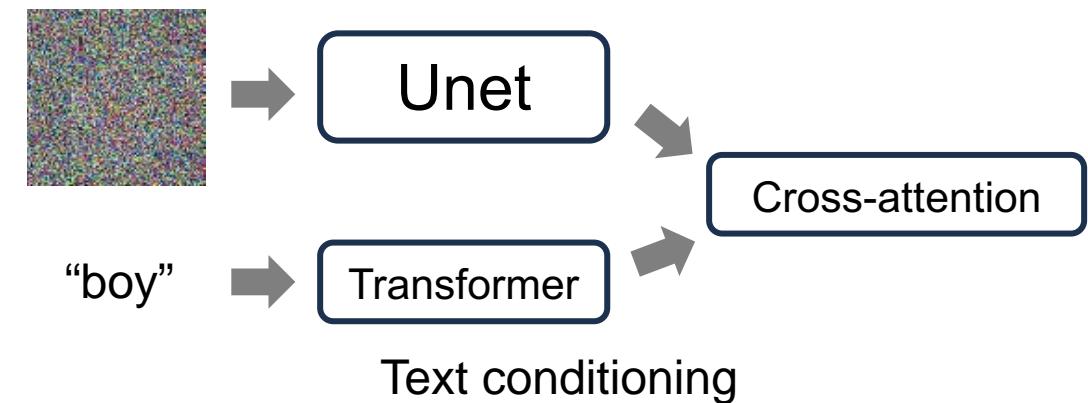
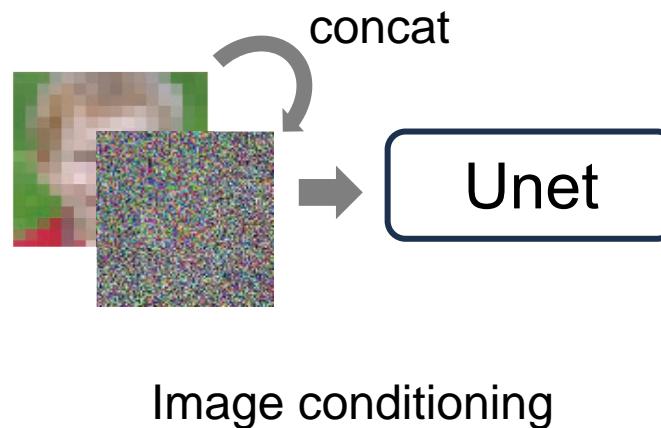
$$\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)$$

- Classifier-free guidance

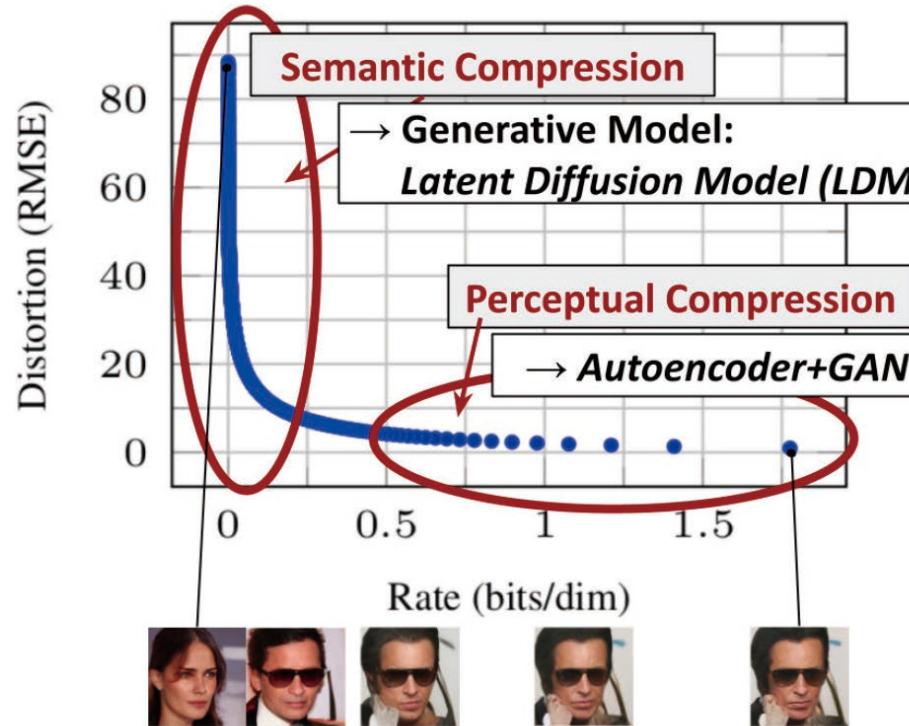
$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\epsilon_\theta(\mathbf{z}_\lambda)$$

- Incorporate conditions into U-Net diffusion model

- Image conditioning: channel-wise concatenation of the conditional image.
- Text conditioning: spatial addition or adaptive normalization / cross-attention.



# Latent diffusion

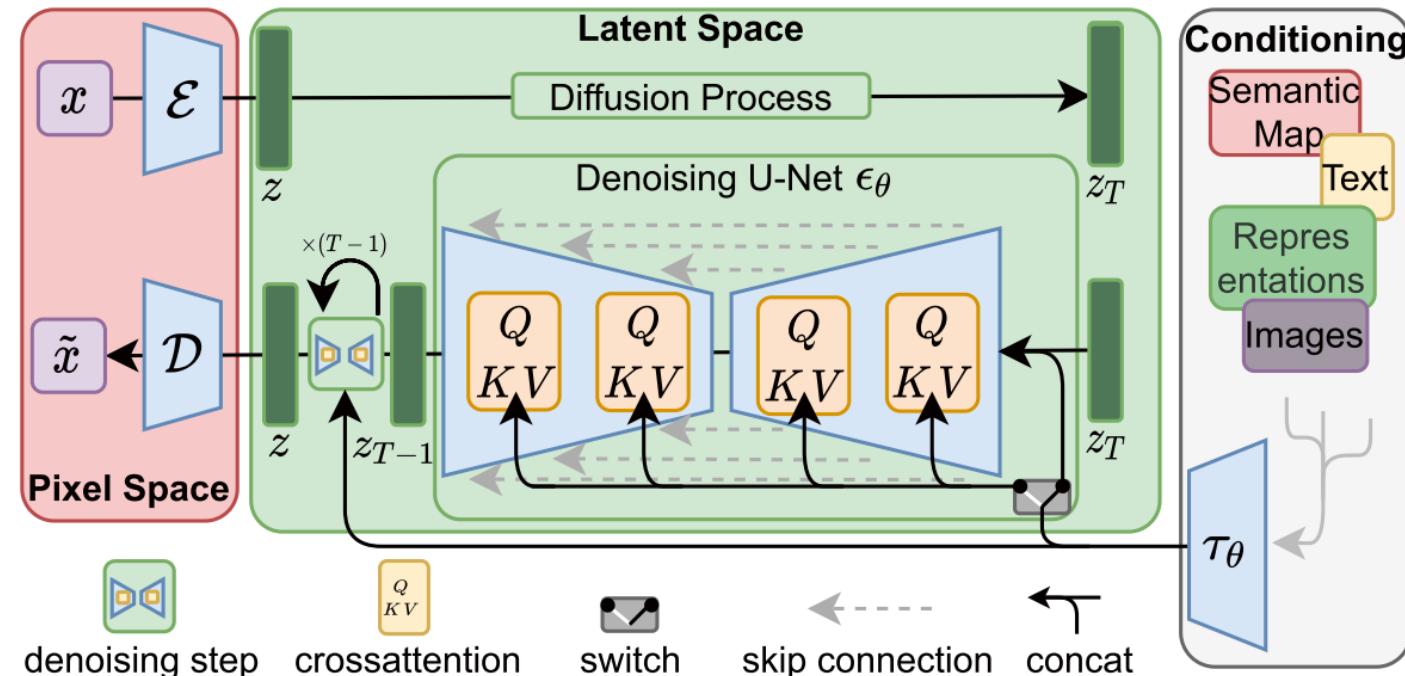


## Perceptual Image Compression:

Downsample input pixels  $x$  to latent  $z$ . Through VQGAN or KL-regularized (similar to VAE).

## Latent Diffusion Models:

Then optimize  $z$  based on the diffusion model.



# Imagen

Tricks:

- Using dynamic thresholding instead of static thresholding

Static thresholding: clip prediction to  $[-1, 1]$ .

Dynamic thresholding:

- at each sampling step, clip to  $[-s, s]$  and divide by  $s$ .



(a) No thresholding.

(b) Static thresholding.

(c) Dynamic thresholding.