CS 2770: Conditional Generative Models

PhD. Nils Murrugarra-Llerena
nem177@pitt.edu

Conditional Generative Models: Applications

[Class Conditional Generation]

- **Task**: Given a class label indicating the image type, sample a new image from the model with that type
- Image classification is the problem of taking in an image and predicting its label p(y|x)
- Class conditional generation is doing this in reverse p(x|y)

https://www.cs.cmu.edu/~mgormley/courses/10423/

Figure from Razavi et al. (2019)

Conditional Generative Models: Applications

[Super Resolution]

- Given a low resolution image, generate a high resolution reconstruction of the image.
- Compelling on low resolution inputs (see example to the left) but also effective on high resolution inputs.

Figure from Li et al. (2021)

Conditional Generative Models: Applications

[Image Editing]

A variety of tasks involve automatic editing of an image:

- **Inpainting** fills in the (prespecified) missing pixels.
- **Colorization** restores color to a greyscale image
- **Uncropping** creates a photo-realistic reconstruction of a missing side of an image

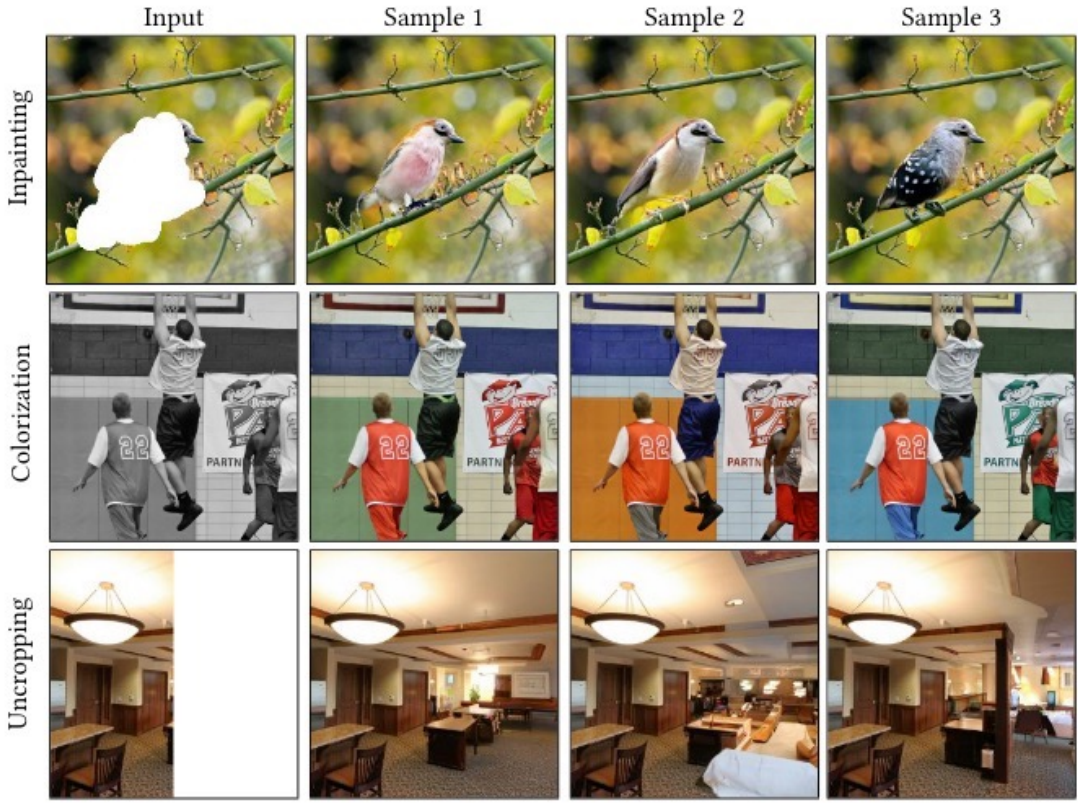


Figure from Saharia et al. (2022)

Conditional Generative Models: Applications

[Style Transfer]

- The goal of style transfer is to blend two images
- Yet, the blend should retain the semantic content of the source image presented in the style of another image

https://www.cs.cmu.edu/~mgormley/courses/10423/

Figure from Gatys et al. (2016)

Conditional Generative Models: Applications

[Text-to-Image Generation]

- Given a text description, sample an image that depicts the prompt
- The following images are samples from SDXL with refinement

Prompt: A propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese.

Figure from Podell et al. (2023)

Conditional Generative Models: Problem Statement

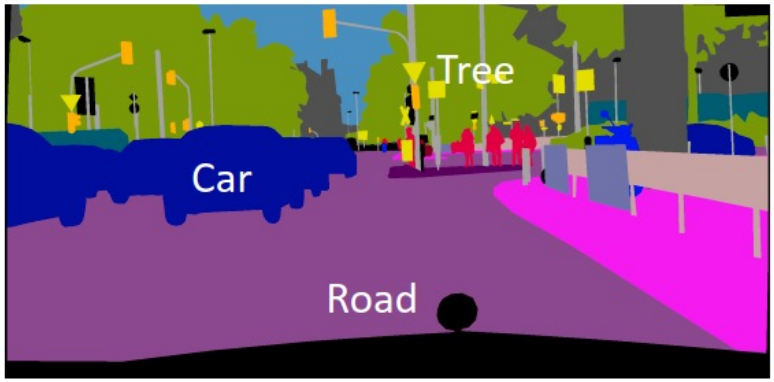
Input

Output

Goal: synthesize a photograph given an input image

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Conditional Generative Models: Problem Statement



Input

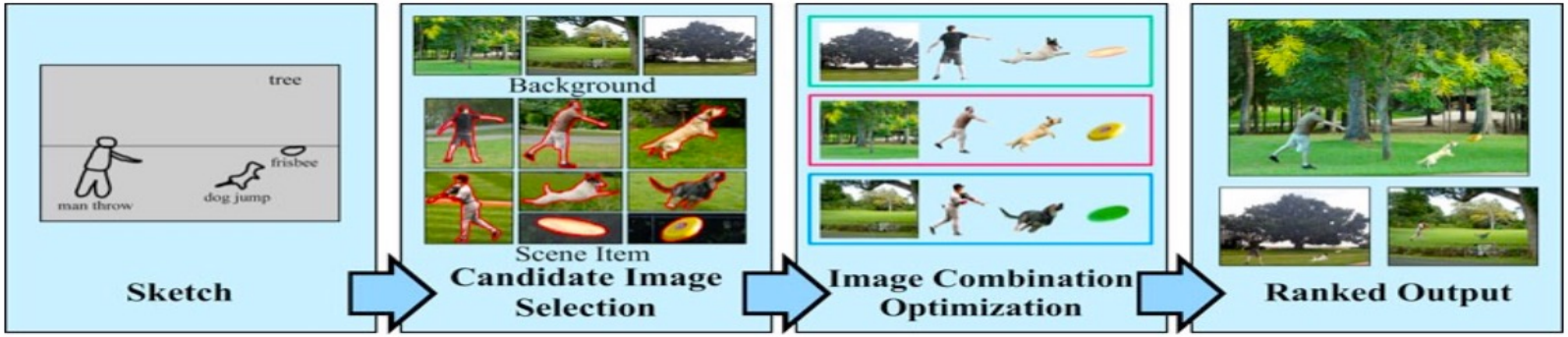
Output

Goal: synthesize a photograph given an input image

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Conditional Generative Models: Early Work

Semantic Photo Synthesis [Johnson et al., Eurographics 2006]

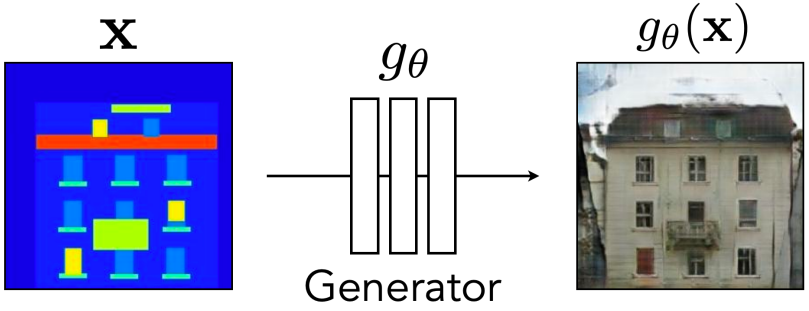


Sketch2Photo [Tao et al., SIGGRAPH Asia 2009]

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

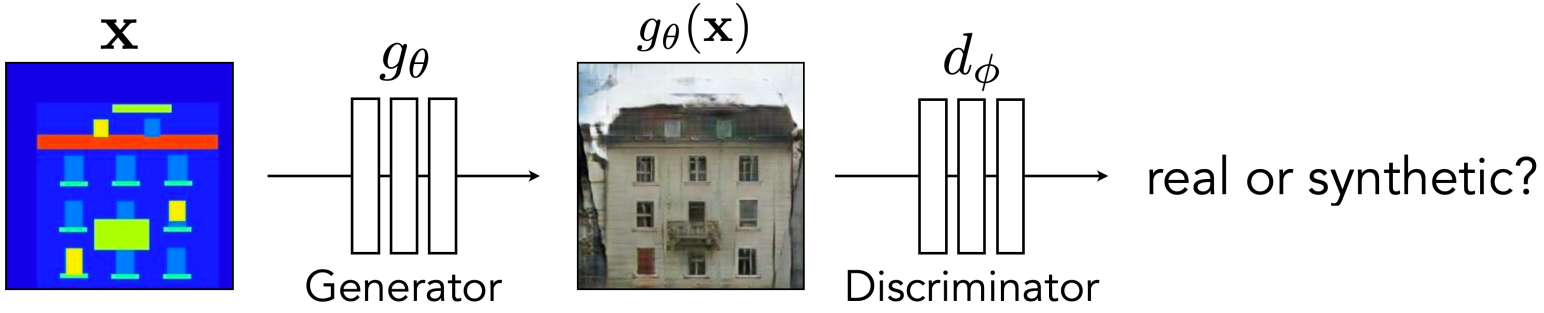
Conditional GANs

Conditional GANs



For example: pix2pix [Isola et al. 2017]

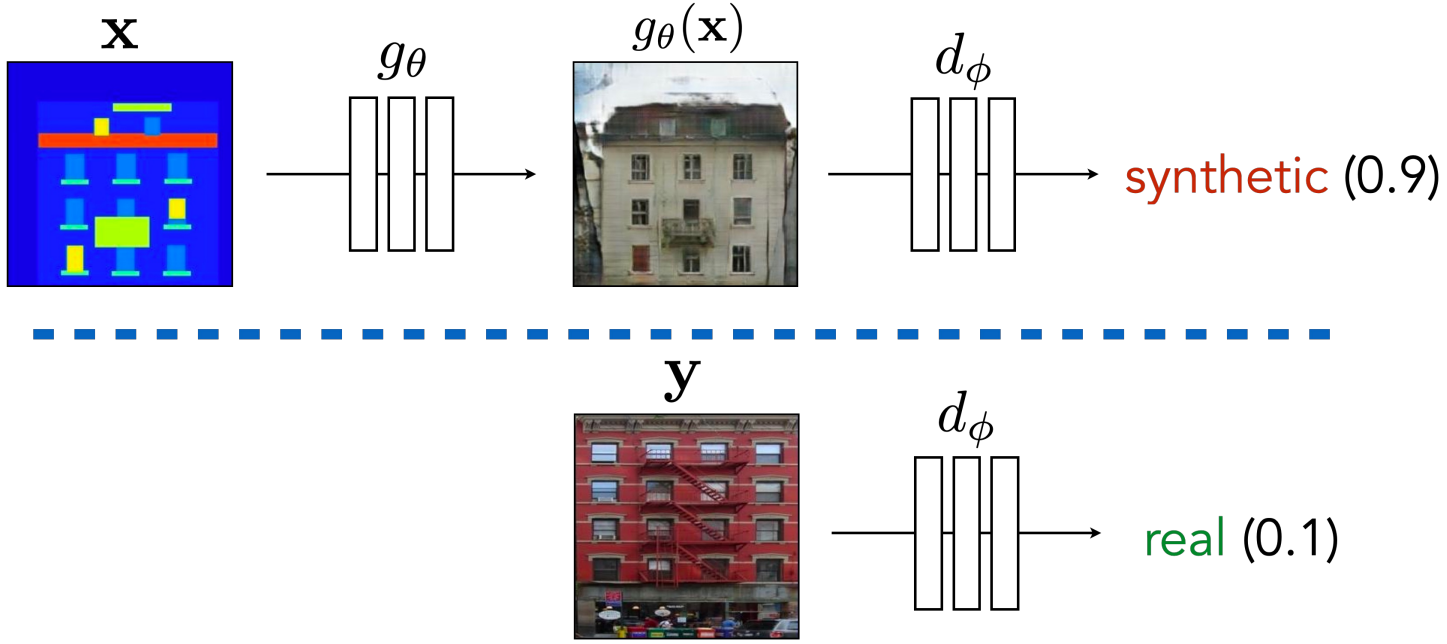
Conditional GANs



g tries to synthesize fake images that fool d

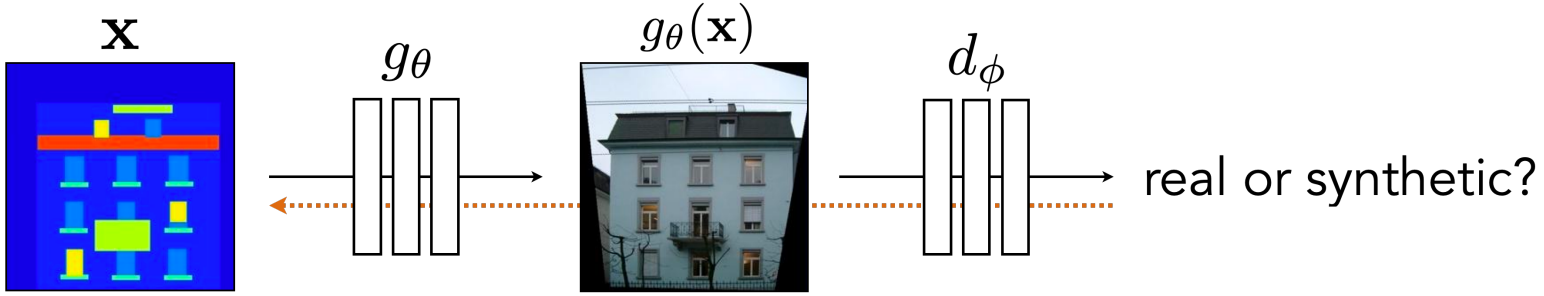
d tries to identify the fakes

Conditional GANs



$$d_\phi^* = \arg\max_\phi \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\log d_\phi(g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{y})) \right]$$

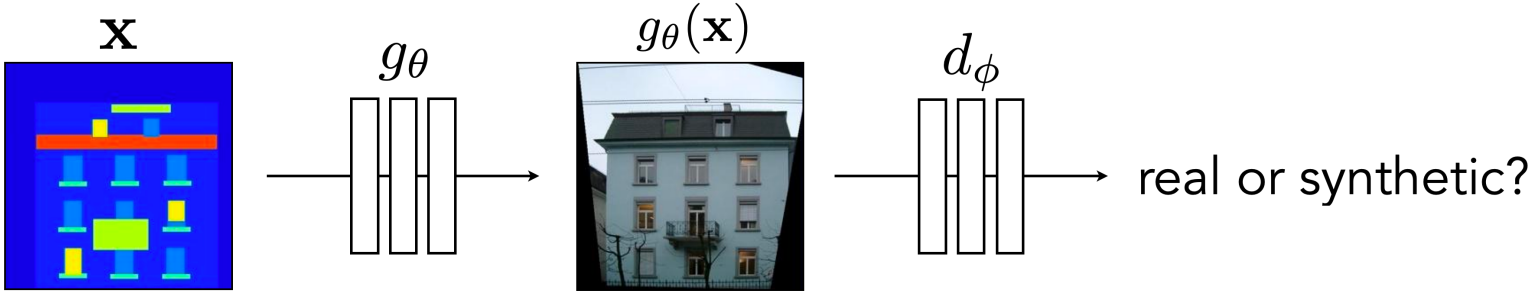
Conditional GANs



g tries to synthesize fake images that *fool* d:

$$g_\theta^* = \boxed{\arg\min_\theta} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(g_\theta(\mathbf{x}))]$$

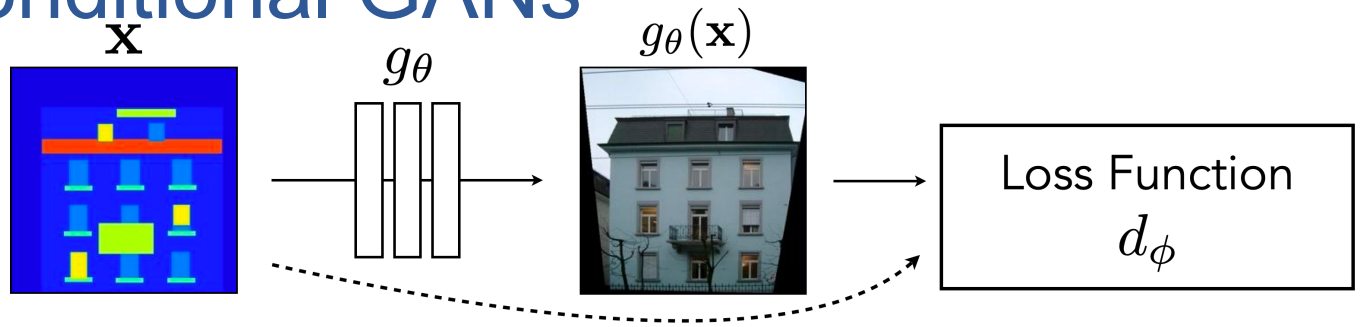
Conditional GANs



g tries to synthesize fake images that *fool* the *best* d:

$$\arg \min_{\theta} \max_{\phi} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{y}))]$$

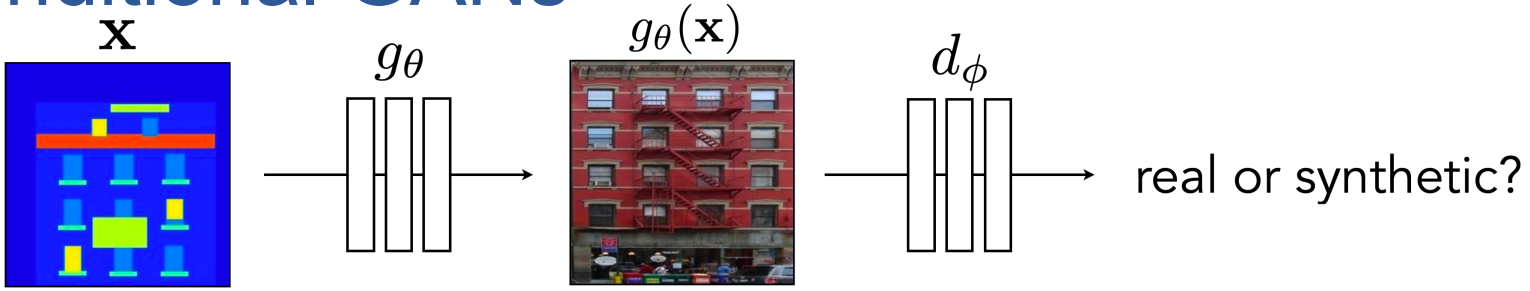
Conditional GANs



g's perspective: d is a loss function.

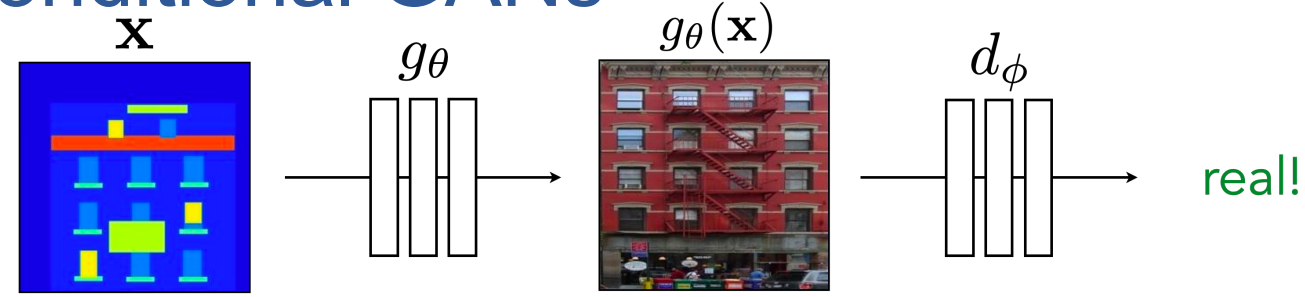
Rather than being hand-designed, it is *learned* and *highly structured*.

Conditional GANs



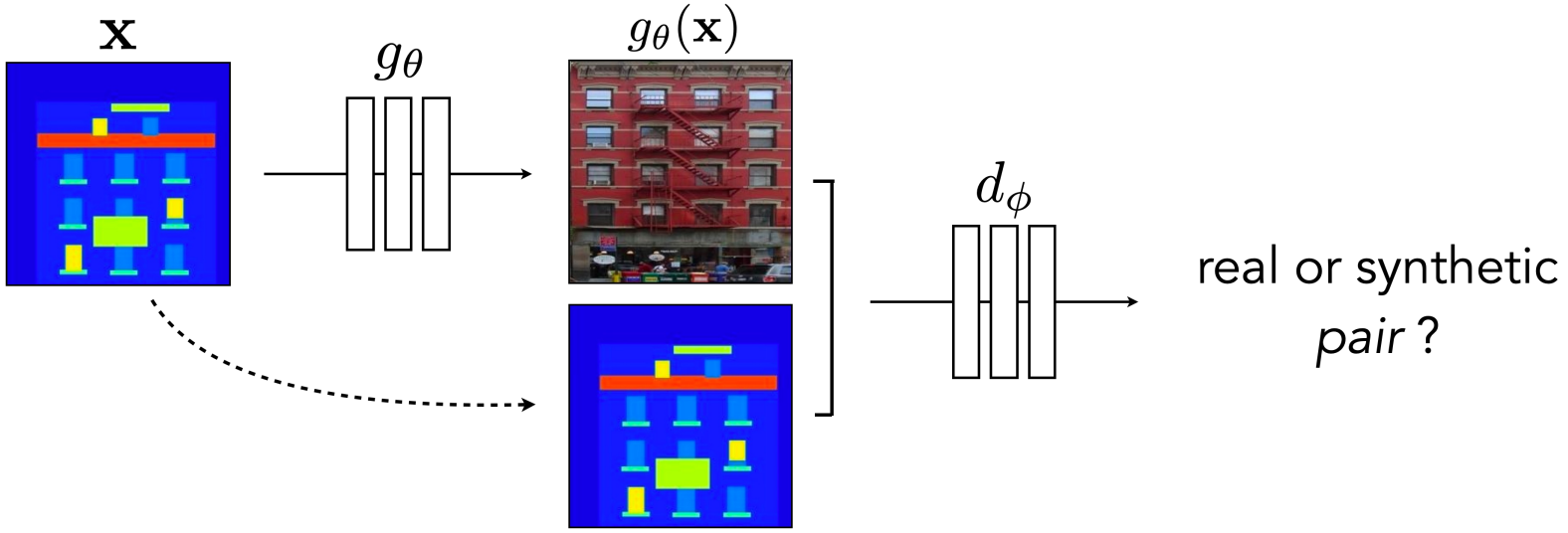
$$\arg\min_{\theta}\max_{\phi}\mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(g_\theta(\mathbf{x}))+\log(1-d_\phi(\mathbf{y}))]$$

Conditional GANs



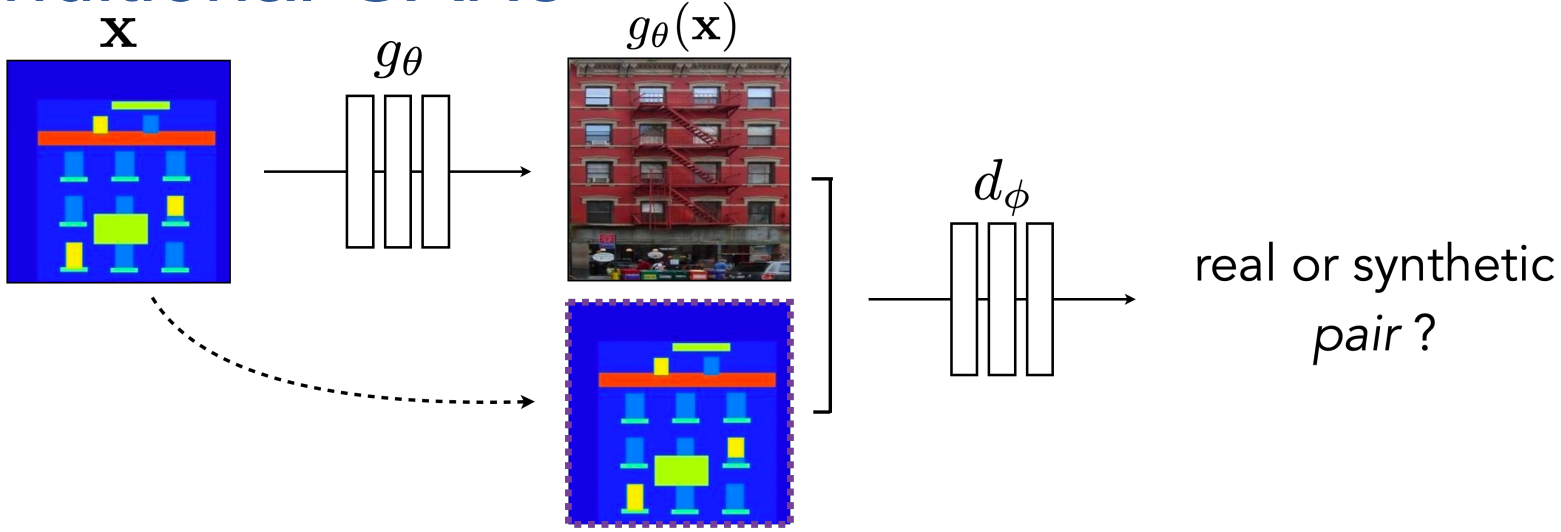
$$\arg\min_{\theta}\max_{\phi}\mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(g_\theta(\mathbf{x}))+\log(1-d_\phi(\mathbf{y}))]$$

Conditional GANs



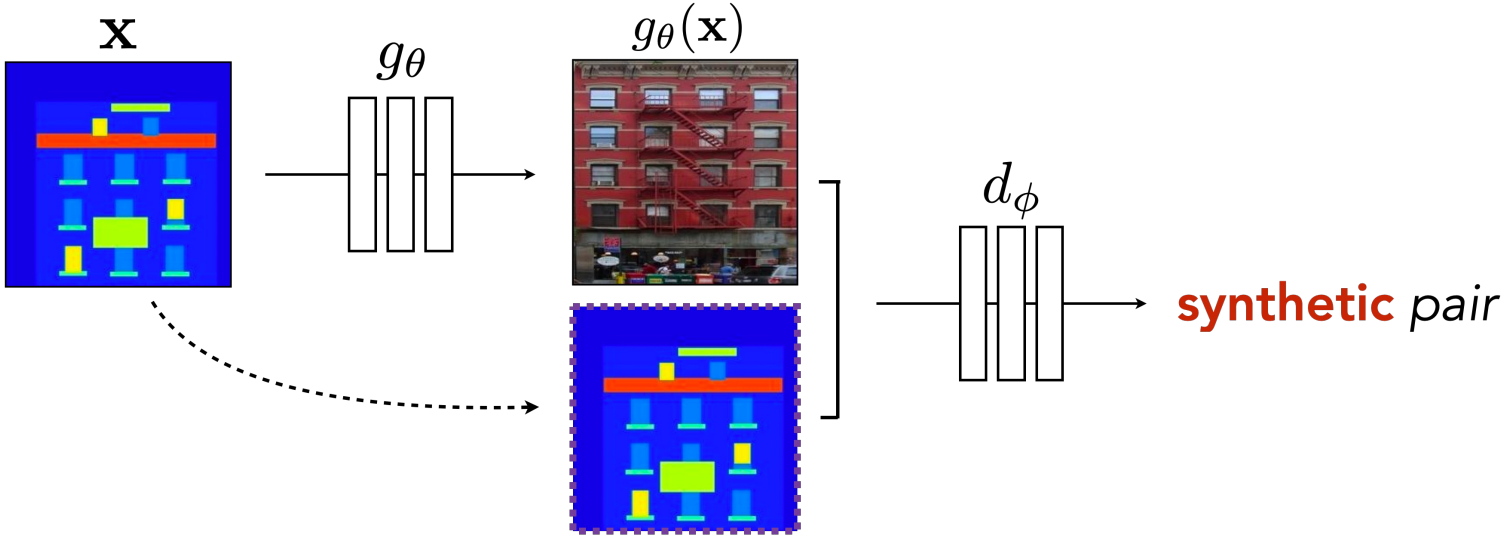
$$\arg\min_{\theta}\max_{\phi}\mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{y}))]$$

Conditional GANs



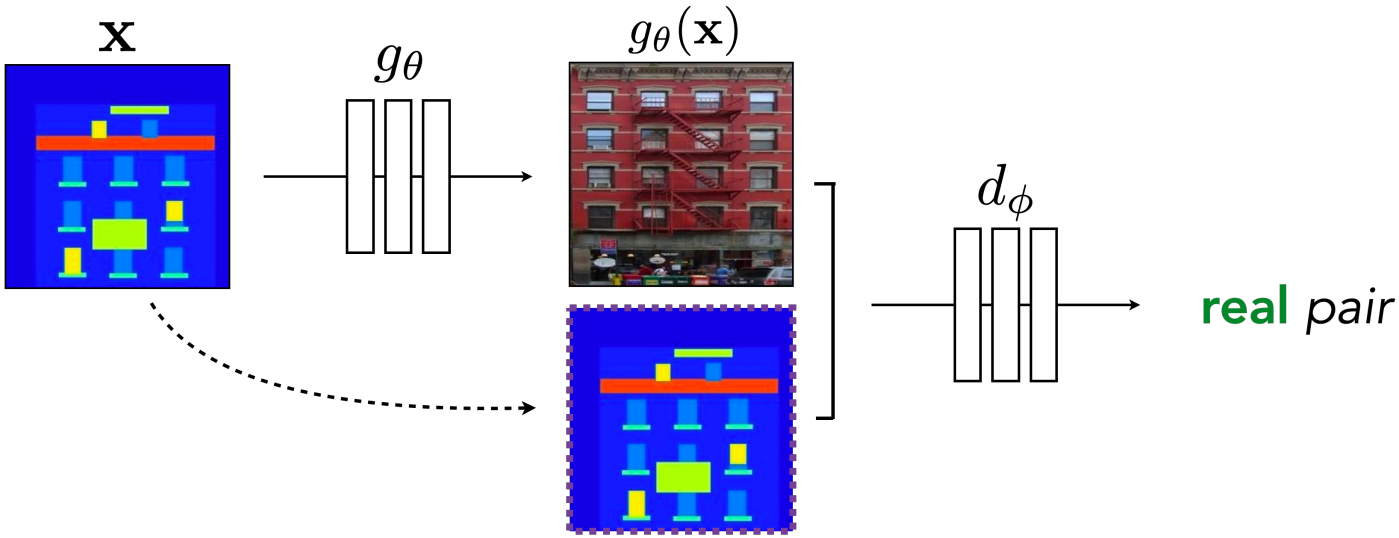
$$\arg\min_\theta \max_\phi \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(\mathbf{x}, g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{x}, \mathbf{y}))]$$

Conditional GANs



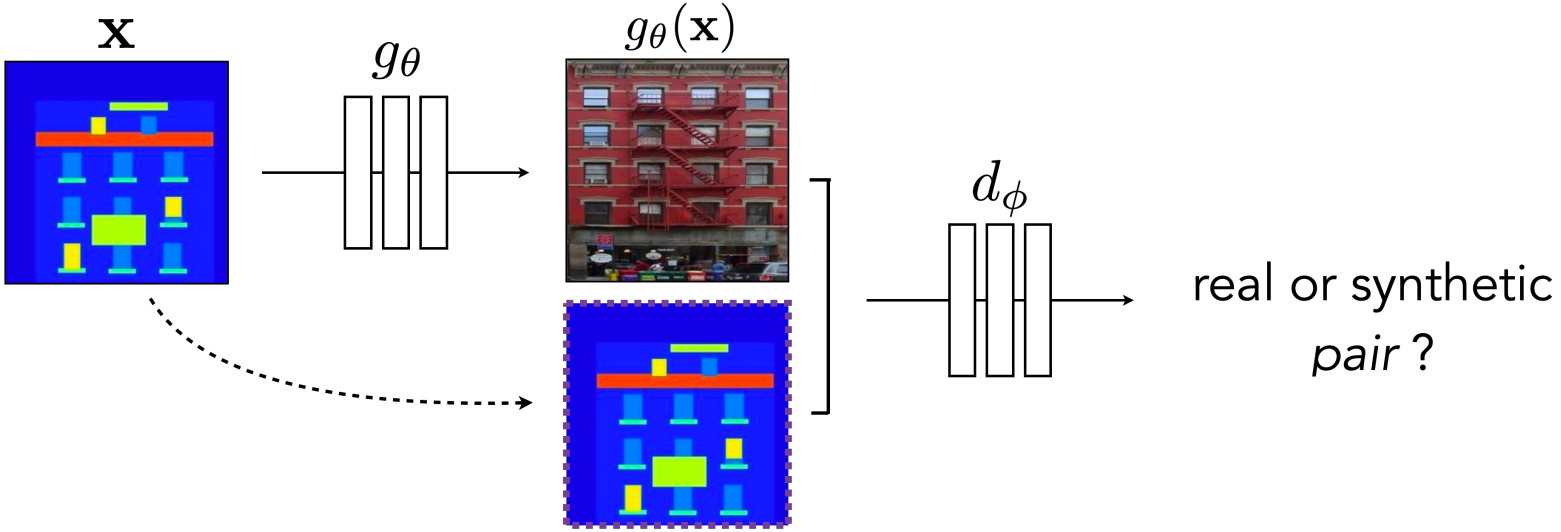
$$\arg\min_\theta \max_\phi \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(\mathbf{x}, g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{x}, \mathbf{y}))]$$

Conditional GANs



$$\arg\min_\theta \max_\phi \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(\mathbf{x}, g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{x}, \mathbf{y}))]$$

Conditional GANs

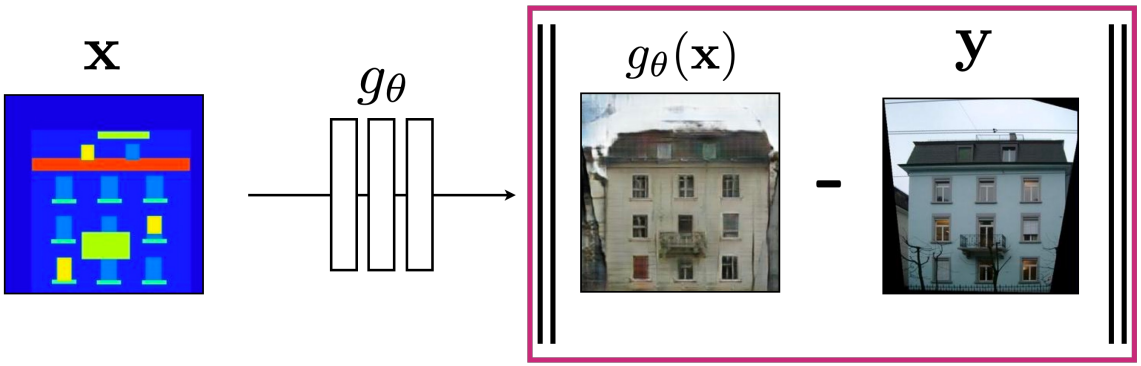


$$\arg\min_{\theta}\max_{\phi}\mathbb{E}_{\mathbf{x},\mathbf{y}}[\log d_\phi(\mathbf{x}, g_\theta(\mathbf{x})) + \log(1 - d_\phi(\mathbf{x}, \mathbf{y}))]$$

Conditional GANs

Training Details: Loss function

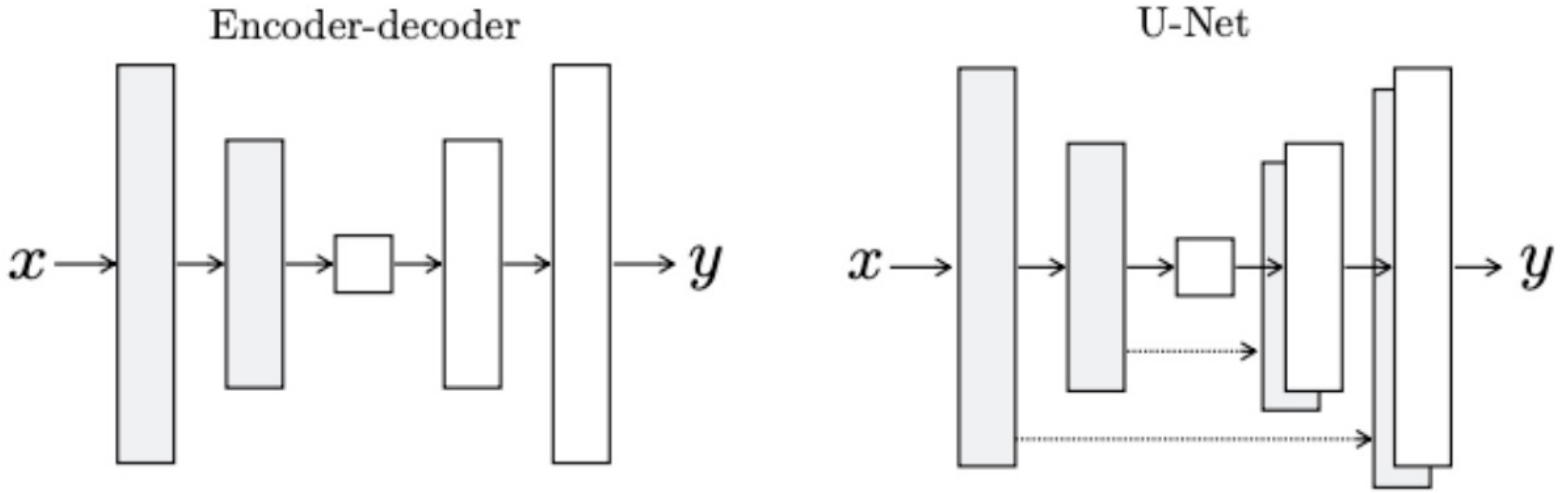
$$g_\theta^* = \arg\min_\theta \max_\phi \mathcal{L}_{\texttt{cGAN}}(\theta, \phi) + \boxed{\lambda \mathcal{L}_{\texttt{L1}}(\theta)}$$



Stable training + fast convergence

[c.f. Pathak et al. CVPR 2016]

Conditional GANs: pix2pix Generator



U-Net [Ronneberger et al.]: popular CNN backbone for biomedical image segmentation
U-Net: preserve high-frequency information (e.g., edge) of the input image.
Encoder-decoder: lose high-frequency details due to the information bottleneck

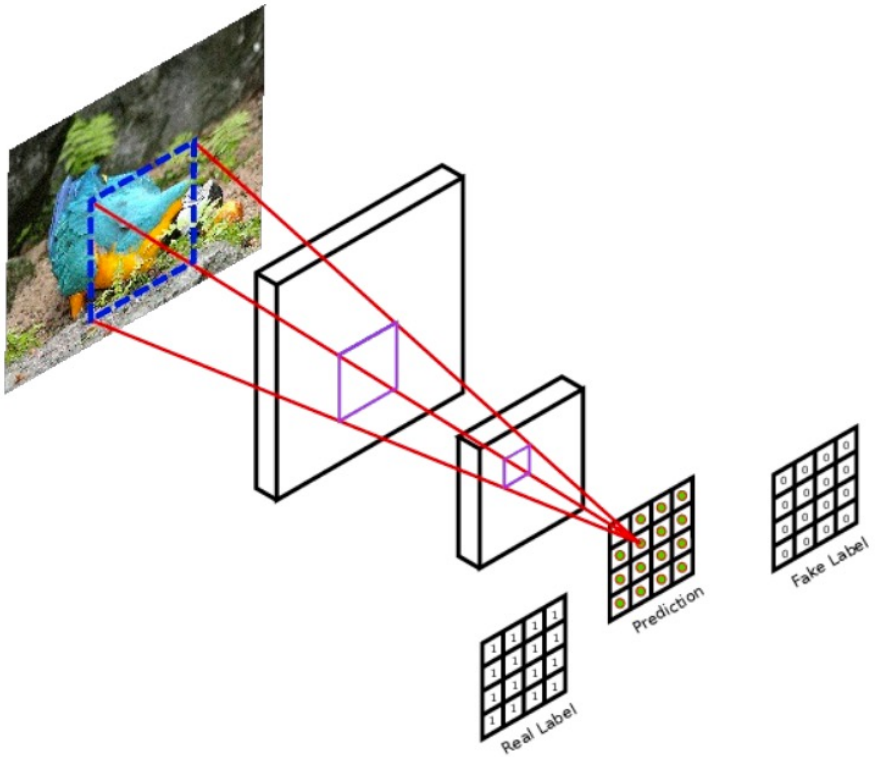
Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Conditional GANs: pix2pix Discriminator

- Rather than penalizing if output image looks fake, penalize if each overlapping patches looks fake
- Focus on local visual cues (color, textures).
- Global structure: the input image has already encoded global structure. L1 loss can help as well.

Advantages:

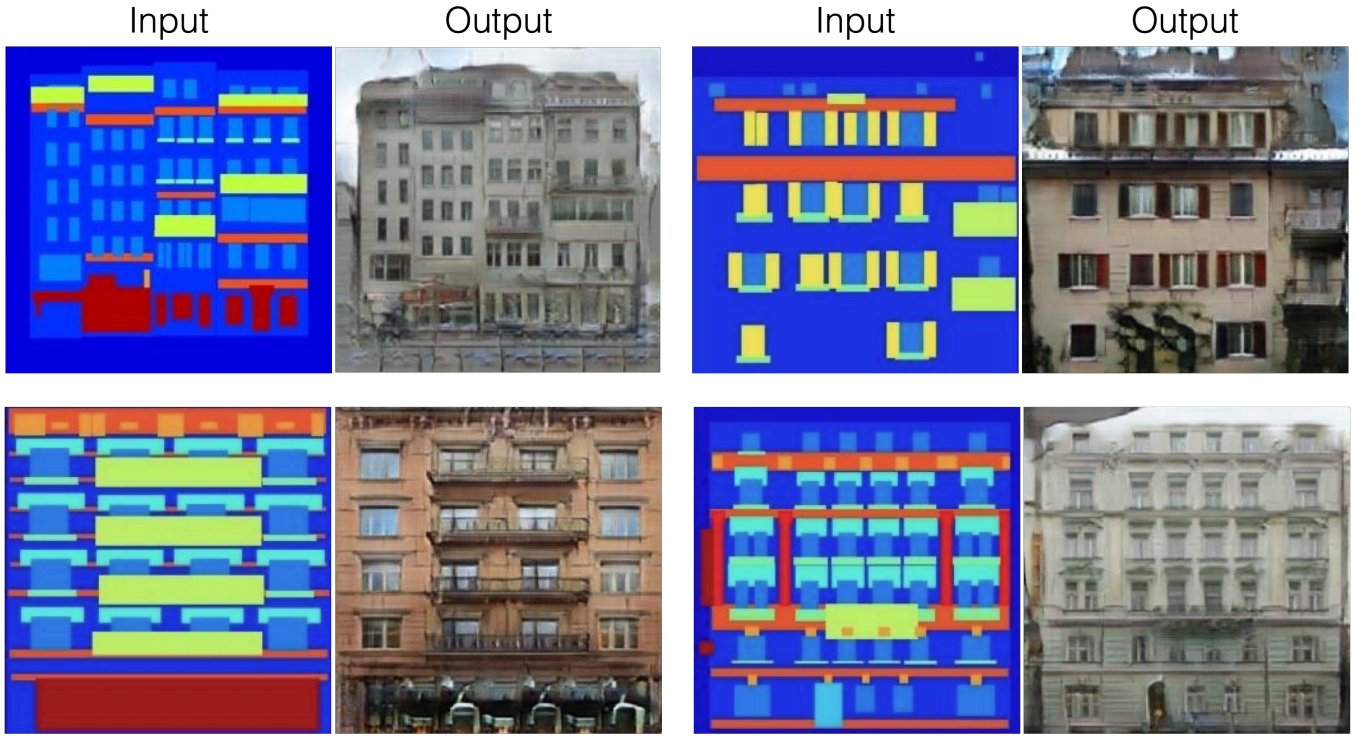
- Faster, fewer parameters
- More supervised observations
- Applies to arbitrarily large images



Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Conditional GANs

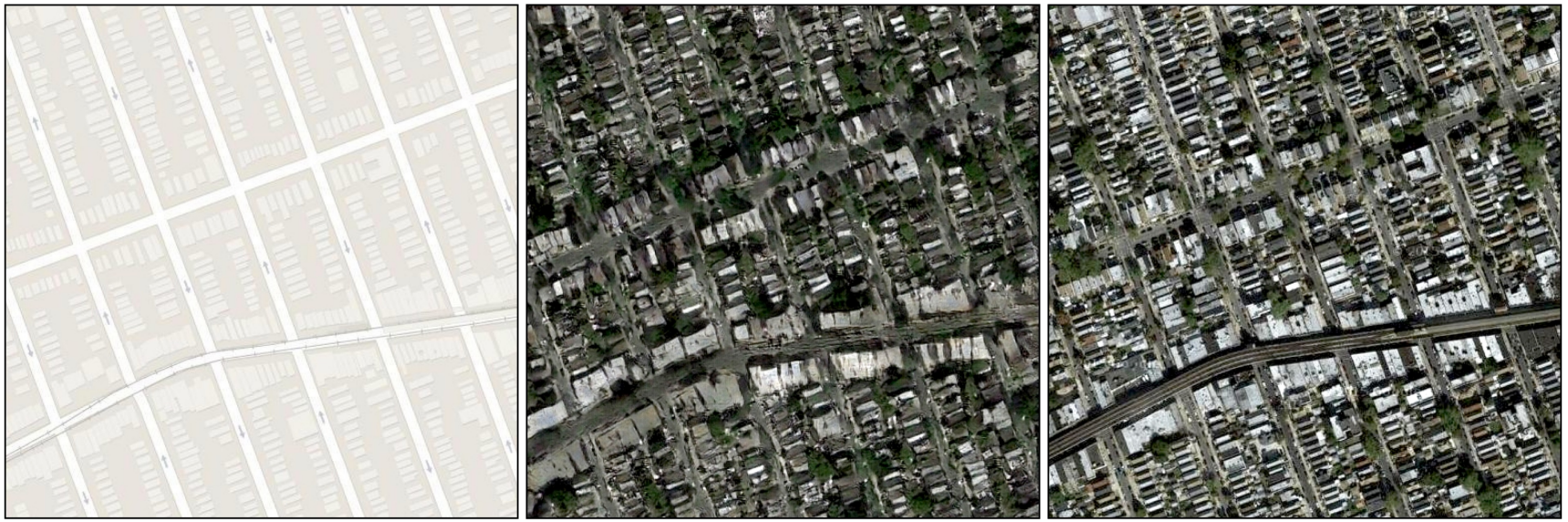
Labels → Facades



Data from [Tylecek, 2013]

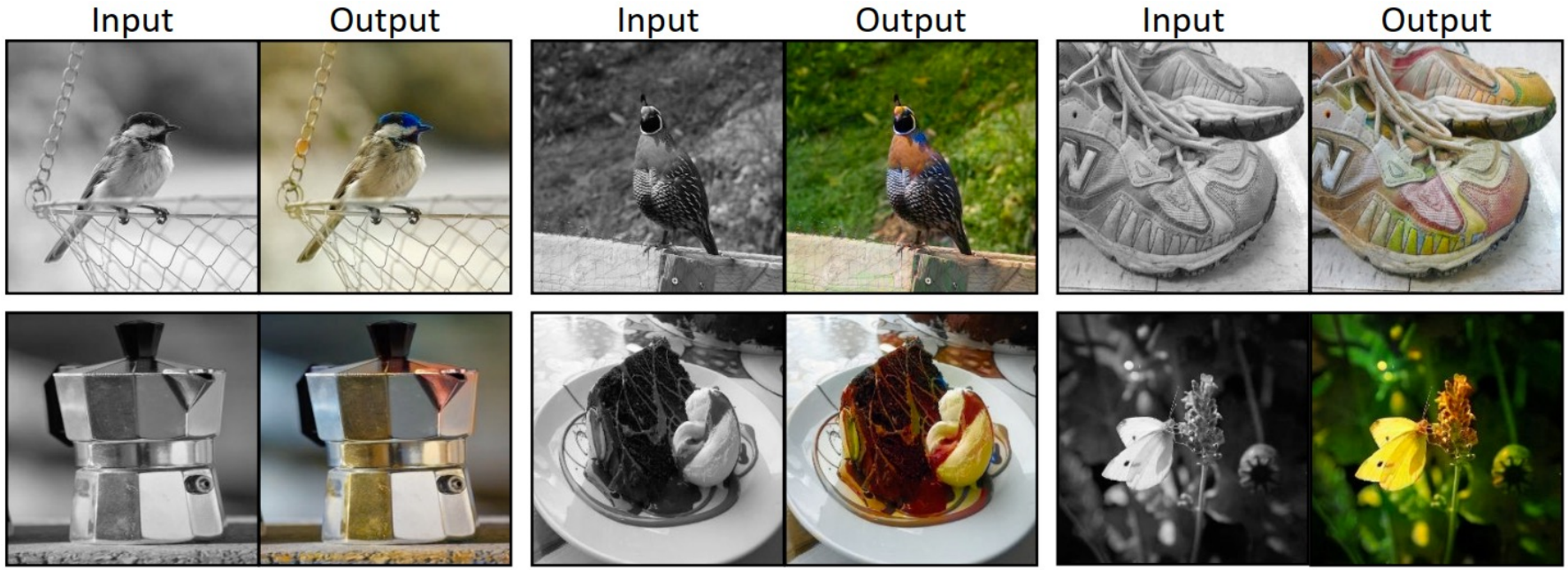
Conditional GANs

Input | Output | Groundtruth



Data from [maps.google.com]

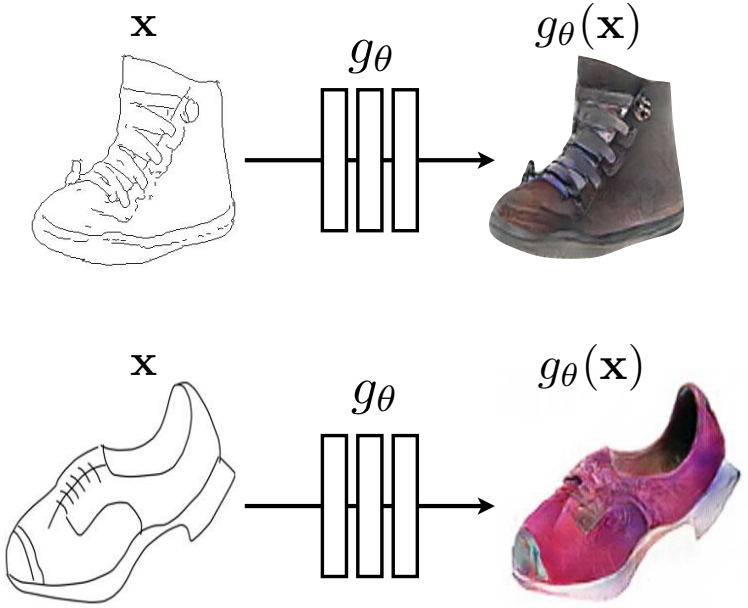
Conditional GANs: Automatic Colorization



Data from [Russakovsky et al. 2015]

Conditional GANs

[HED, Xie & Tu, 2015]

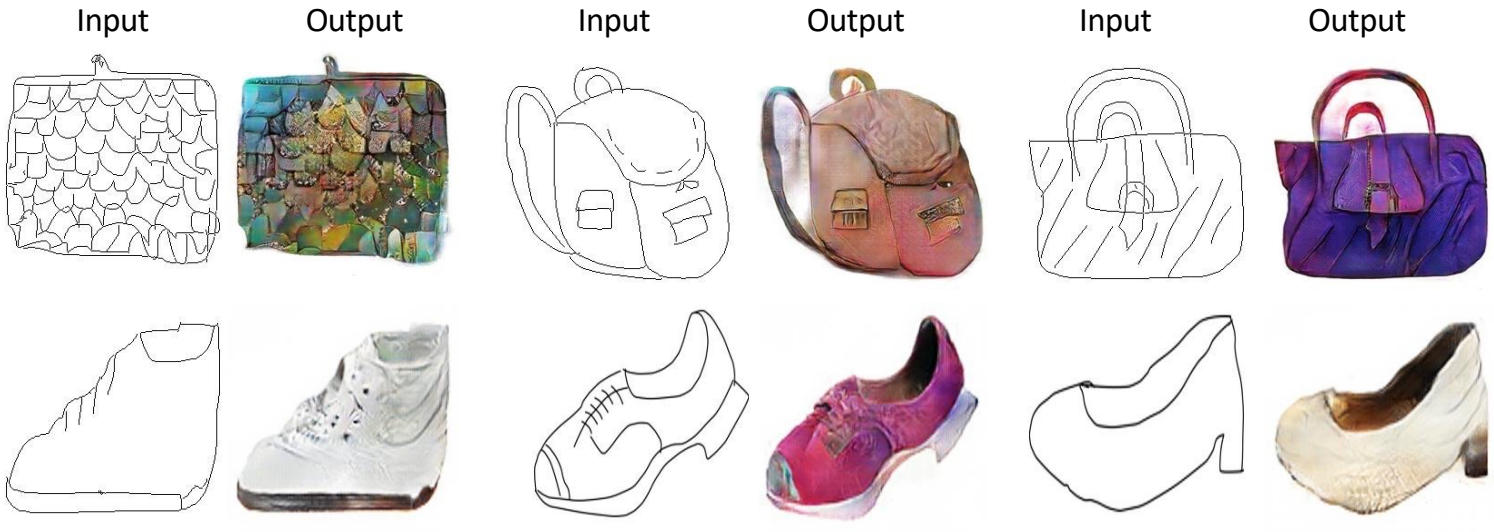


Conditional GANs: Edges to Images

Edges from [Xie & Tu, 2015]

Pix2pix / CycleGAN

Conditional GANs: Sketches to Images



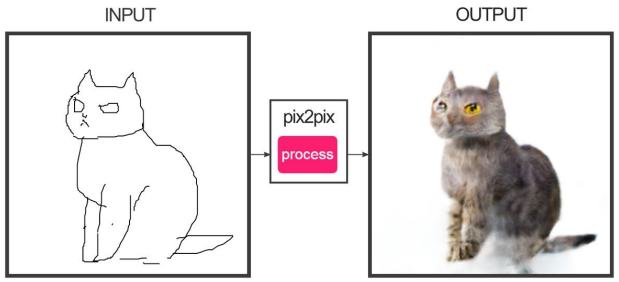
Trained on Edges → Images

Data from [Eitz, Hays, Alexa, 2012]

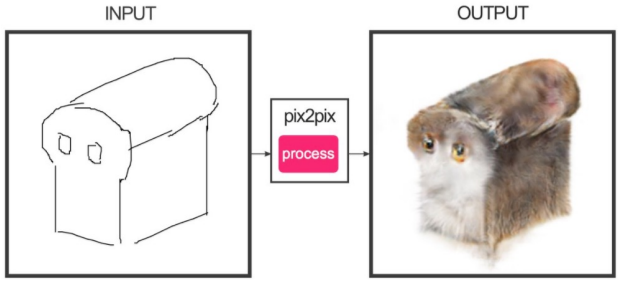
Pix2pix / CycleGAN

Conditional GANs: Edges to Images

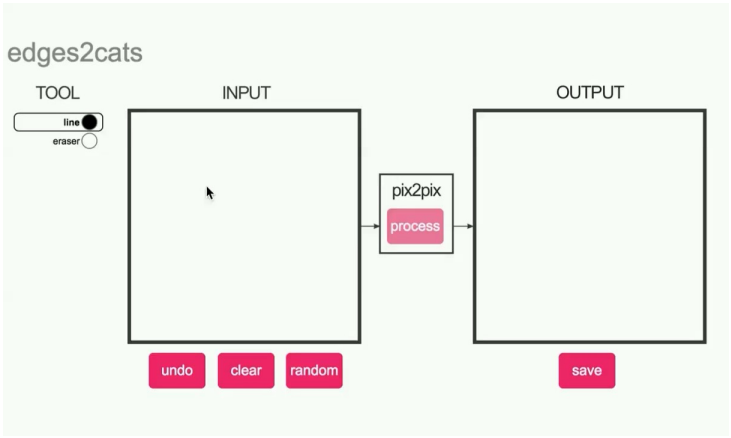
#edges2cats [Christopher Hesse]



@gods_tail



Ivy Tasi @ivymyt



@matthematician

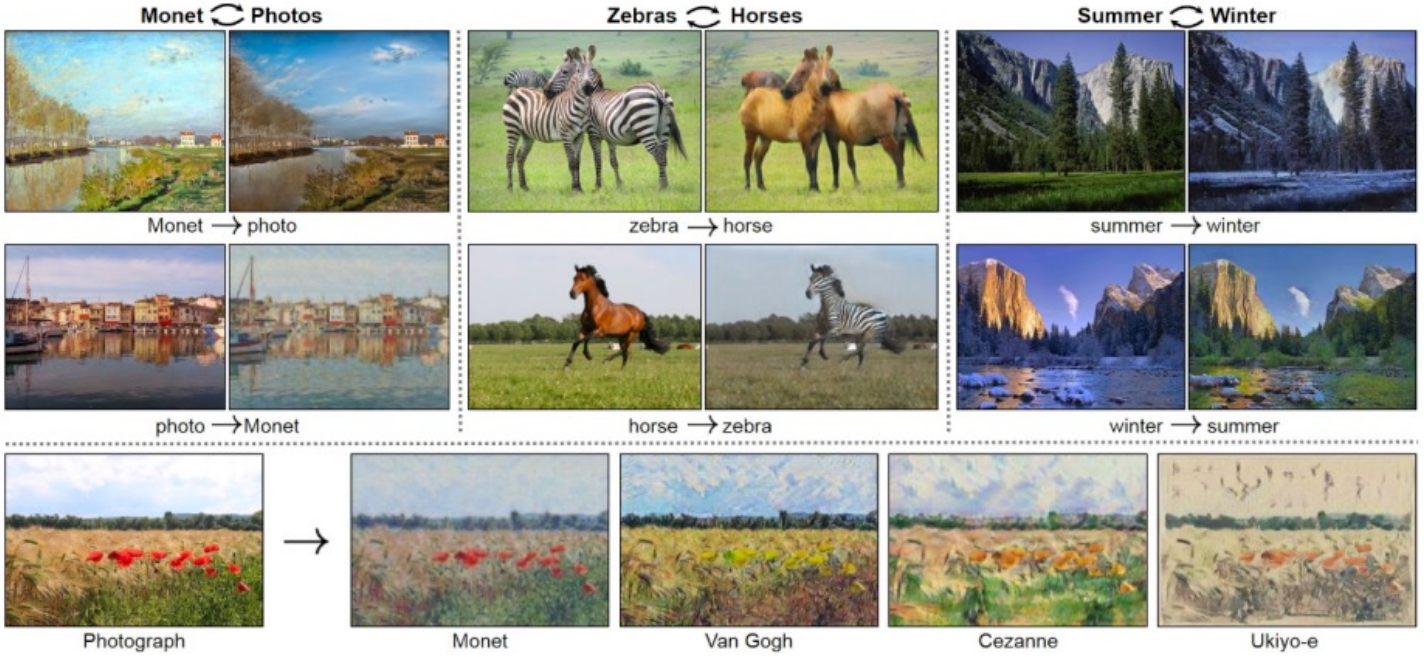
Vitaly Vidmirov @vvid

https://affinelayer.com/pix2pix/

Pix2pix / CycleGAN

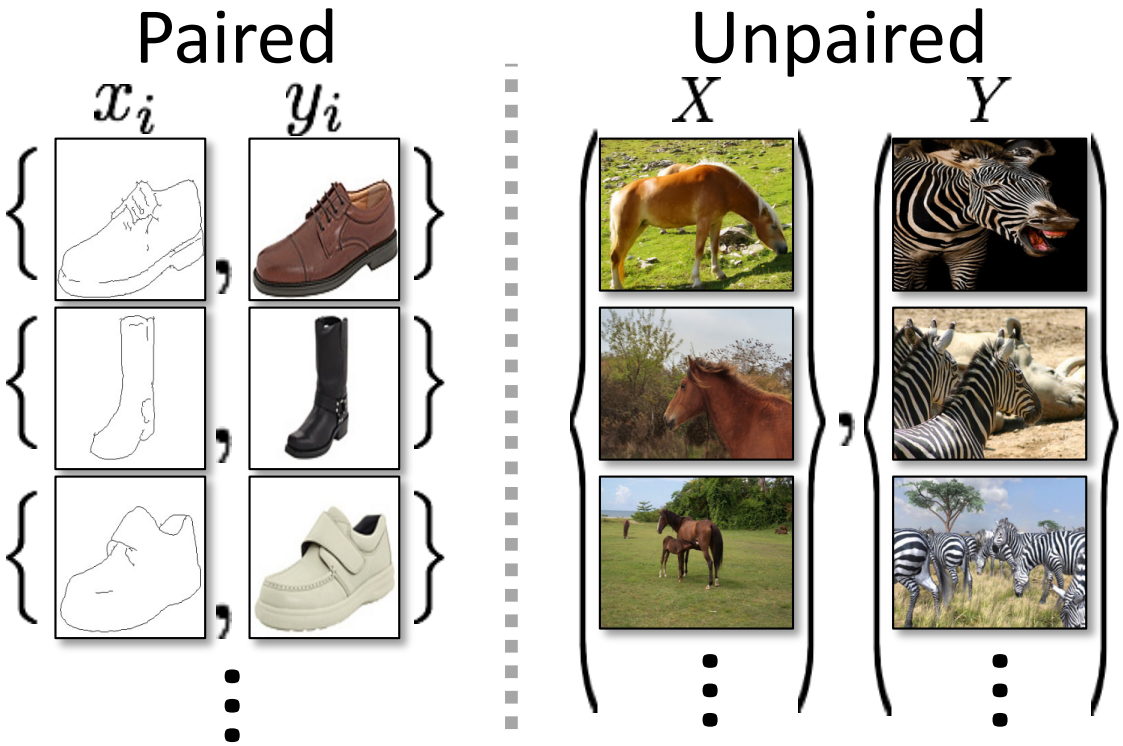
Conditional GANs: Unpaired Translations

Style transfer problem: change the style of an image while preserving the content.



Data: Two unrelated collections of images, one for each style

Conditional GANs: Unpaired Translations - CycleGAN



Conditional GANs: CycleGAN

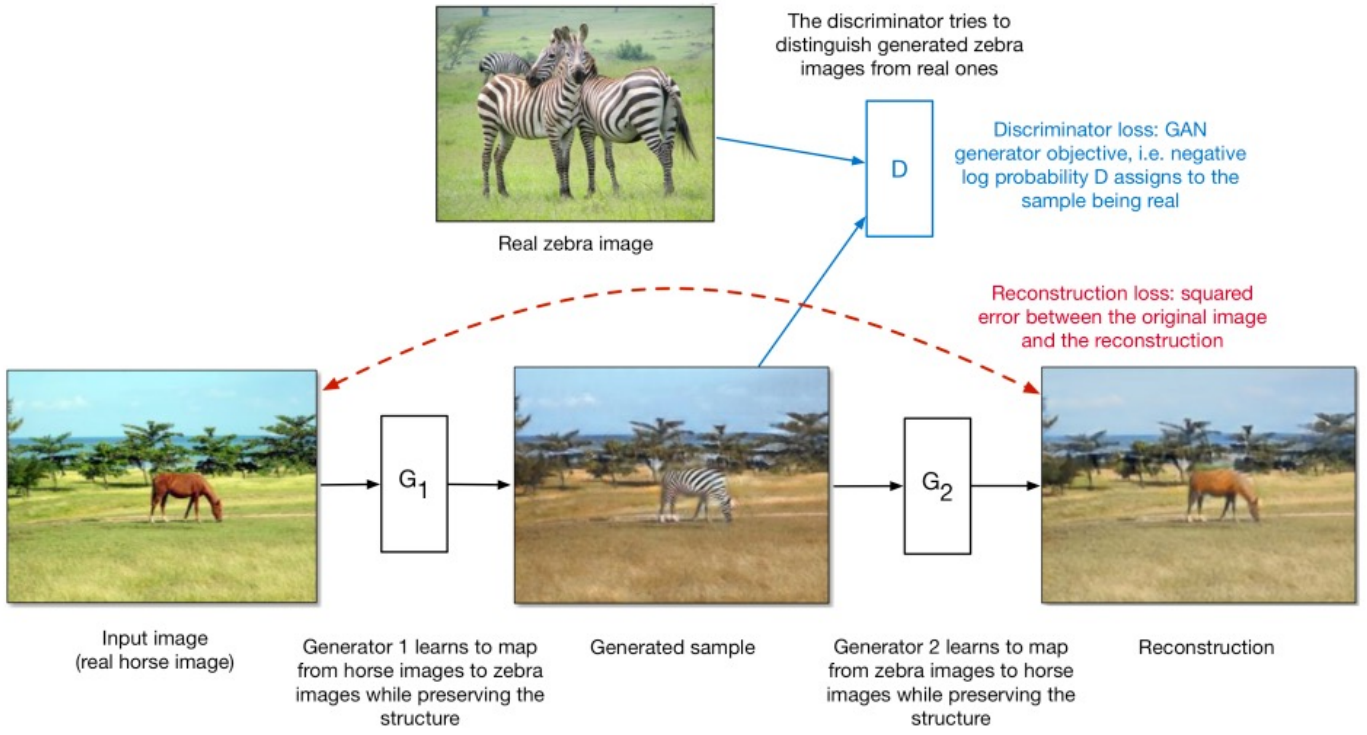
If we had paired data (same content in both styles), this would be a supervised learning problem. But this is hard to find!

The CycleGAN architecture learns to do it from unpaired data.

- Train two different generator nets to go from style 1 to style 2, and vice versa.
- Make sure the generated samples of style 2 are indistinguishable from real images by a discriminator net.
- Make sure the generators are cycle-consistent:

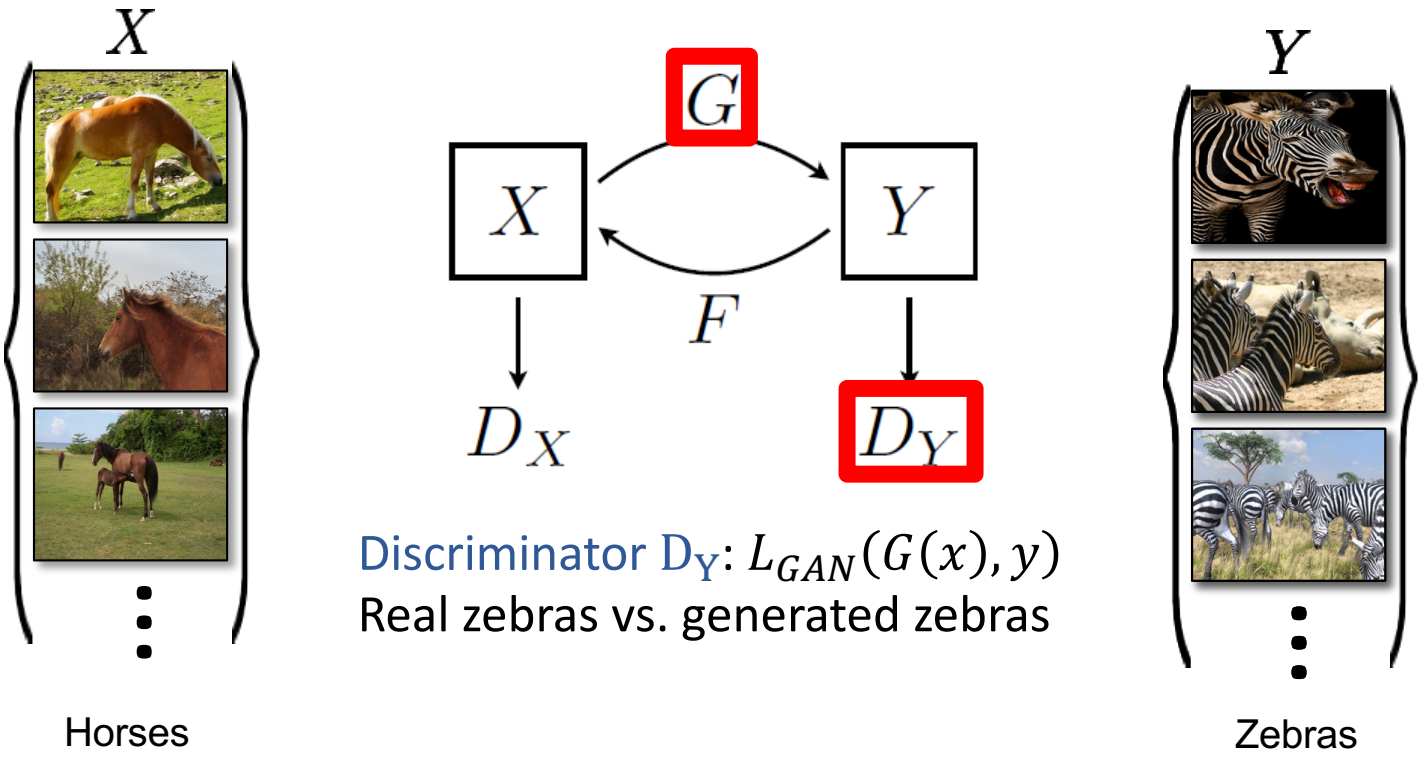
  Mapping from style 1 to style 2 and back again should give you almost the original image.

Conditional GANs: CycleGAN



Total loss = discriminator loss + reconstruction loss

Cycle GAN: Cycle Consistency

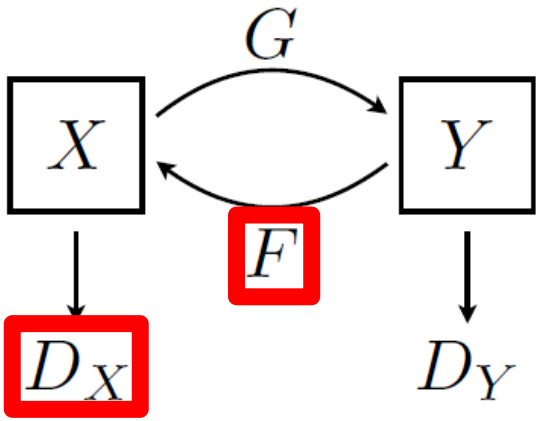


Discriminator D_Y: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle GAN: Cycle Consistency

Horses



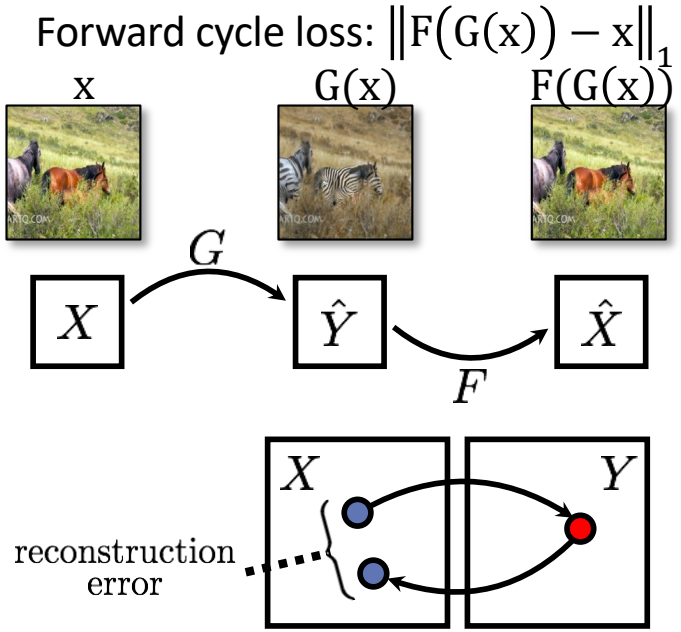
Discriminator D_Y: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras

Discriminator D_X: $L_{GAN}(F(y), x)$
Real horses vs. generated horses

Zebras

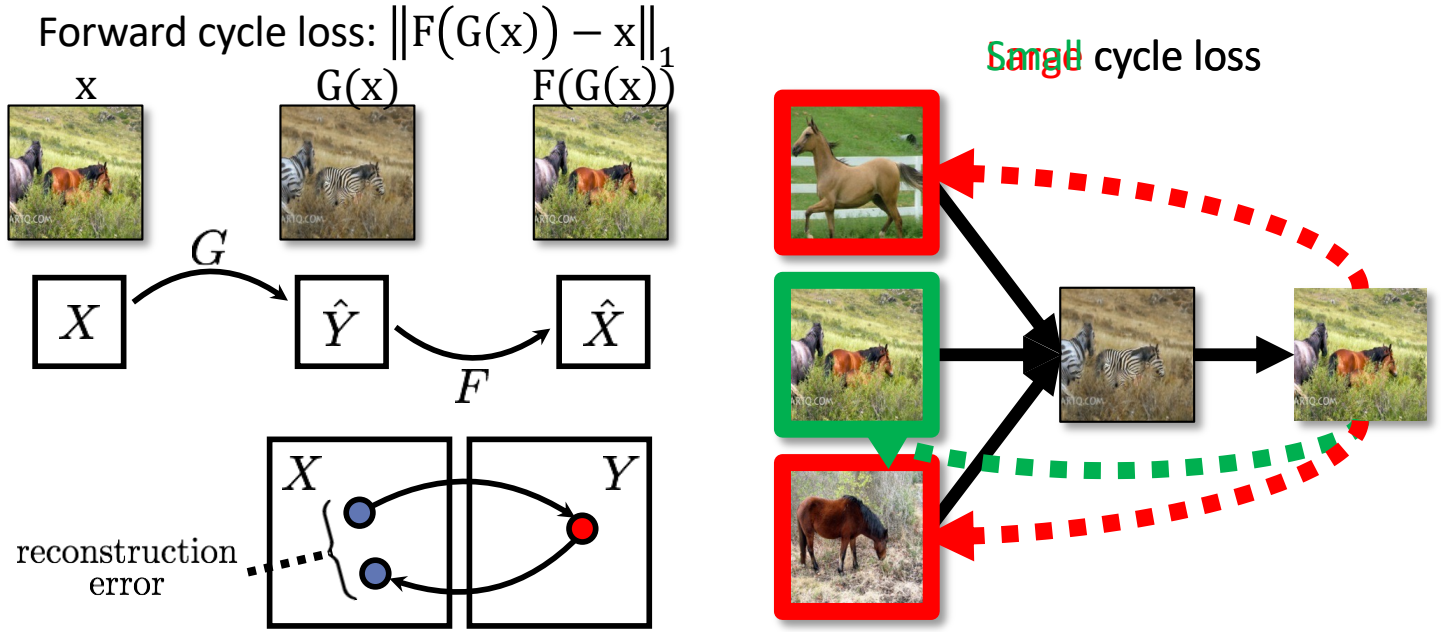
Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle GAN: Cycle Consistency



Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle GAN: Cycle Consistency



Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle GAN: Training

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) =& \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],\end{aligned}$$

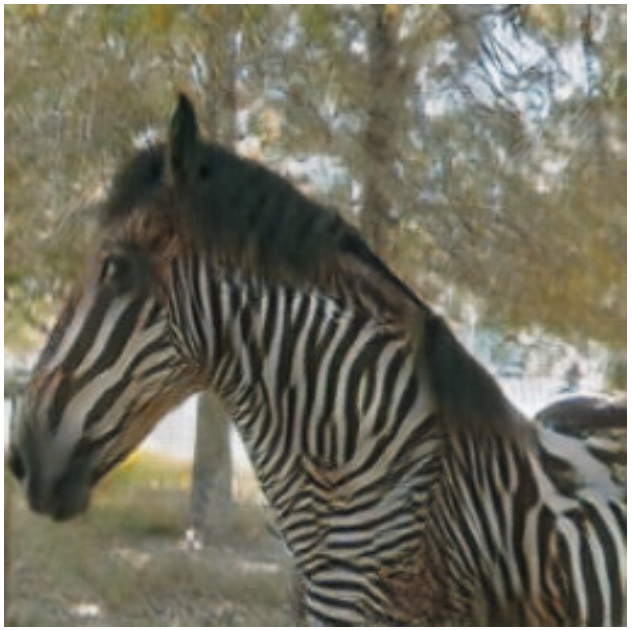
$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) =& \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned}$$

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) =& \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned}$$

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

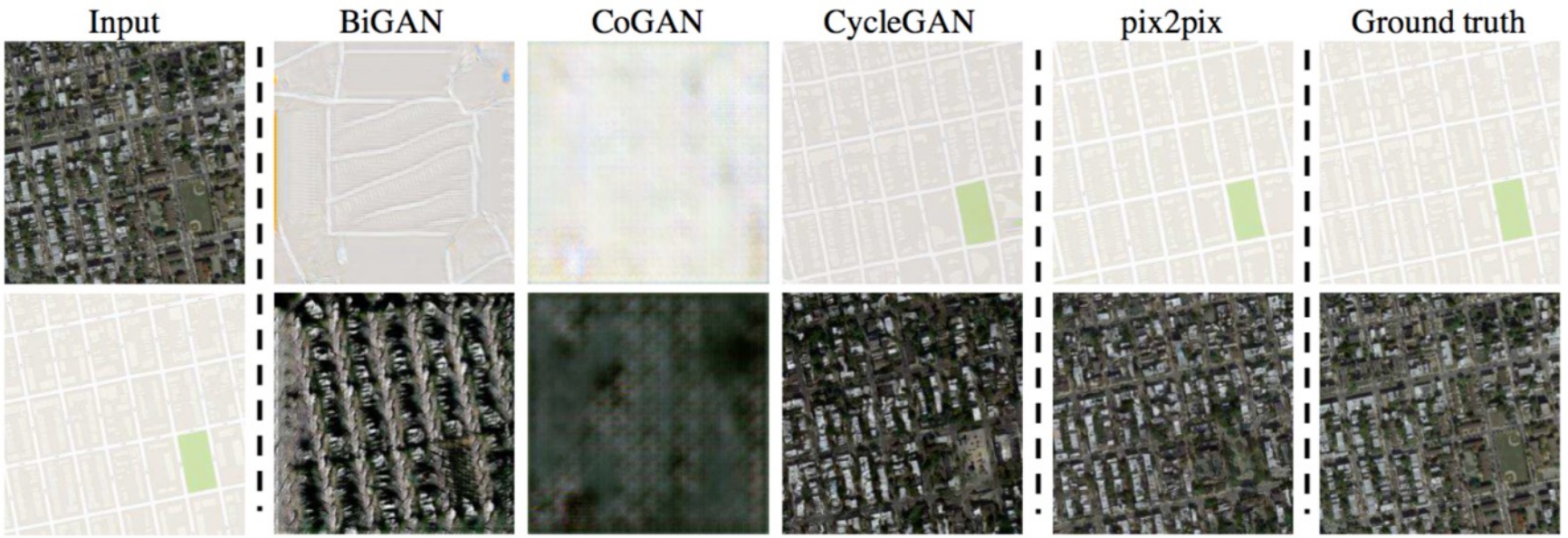
Cycle GAN: Examples



Pix2pix / CycleGAN

Cycle GAN: Application

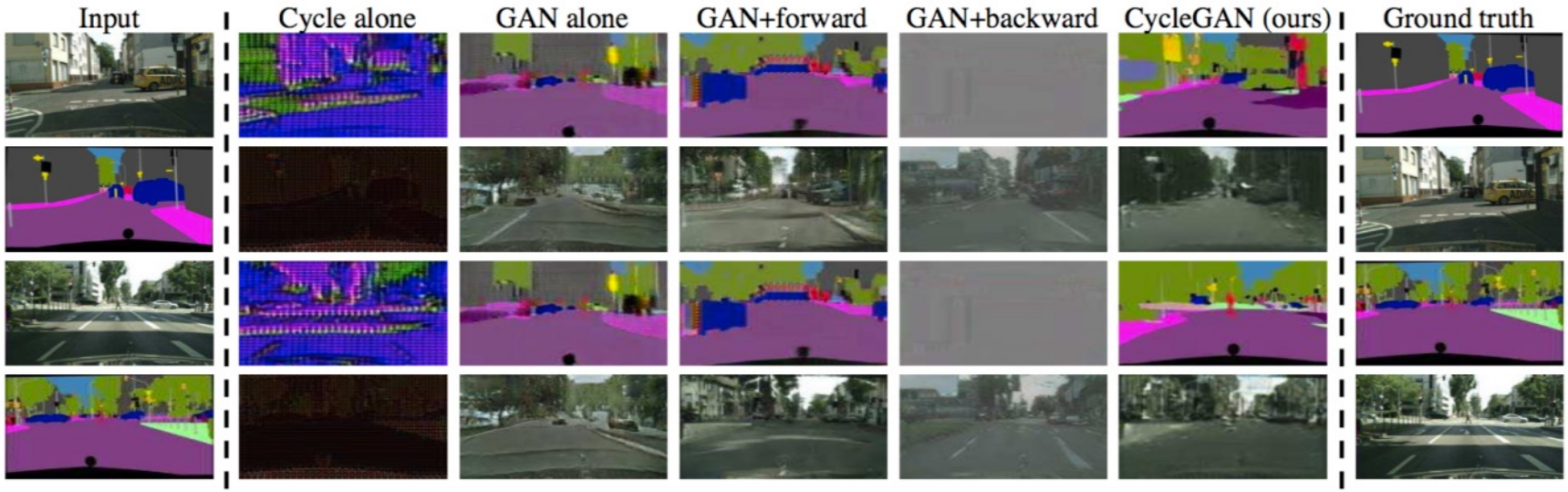
Style transfer between aerial photos and maps:



Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle GAN: Application

Style transfer between road scenes and semantic segmentations (labels of every pixel in an image by object category):

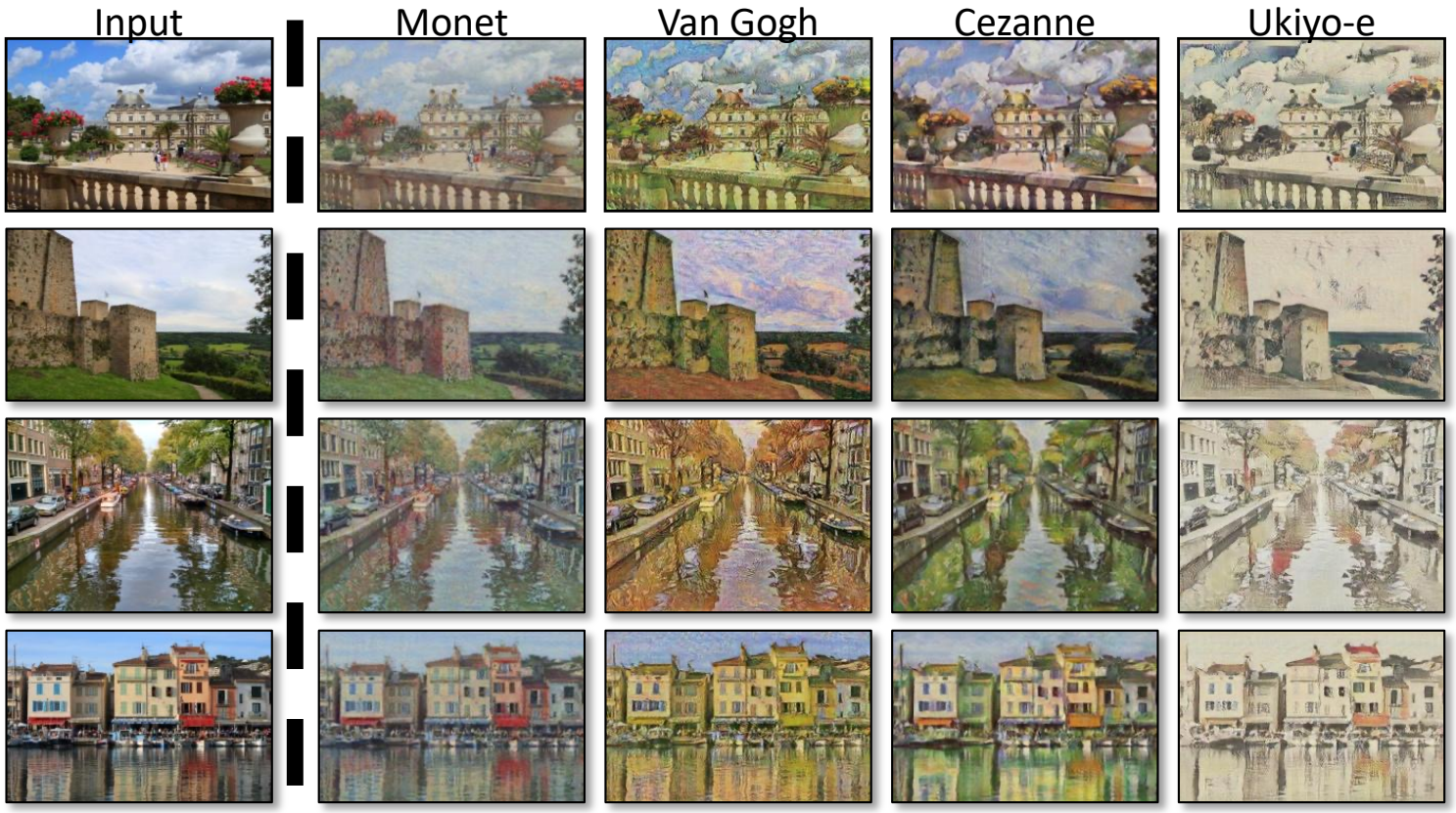


Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle GAN: Application

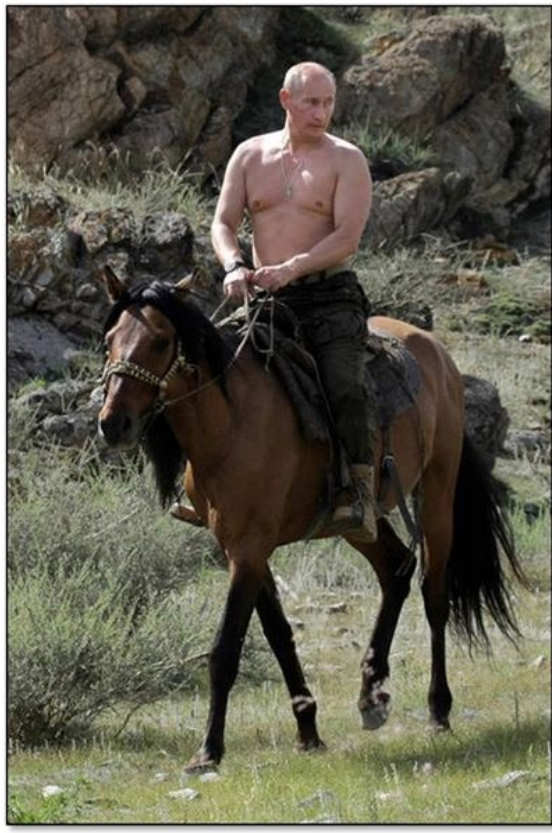
Pix2pix / CycleGAN

Cycle GAN: Application



Pix2pix / CycleGAN

Cycle GAN: Failure Example



Pix2pix / CycleGAN

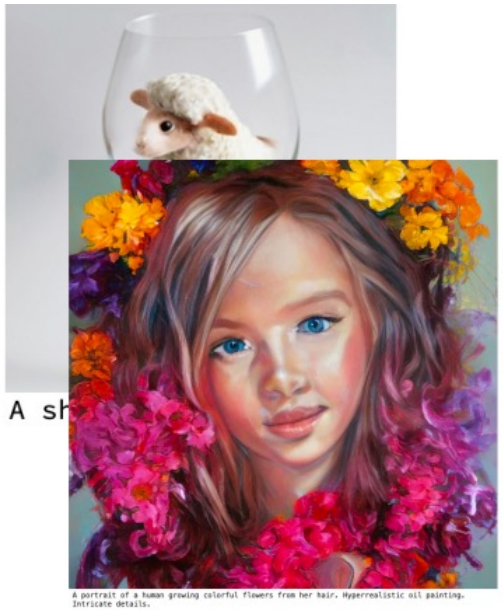
Cycle GAN: Failure Example

Pix2pix / CycleGAN

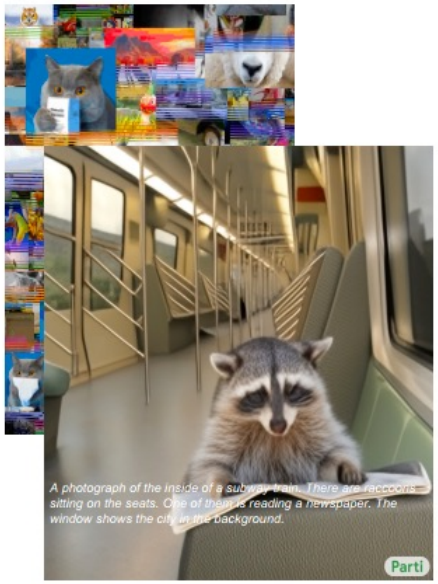
Conditional GANs

What other modalities?

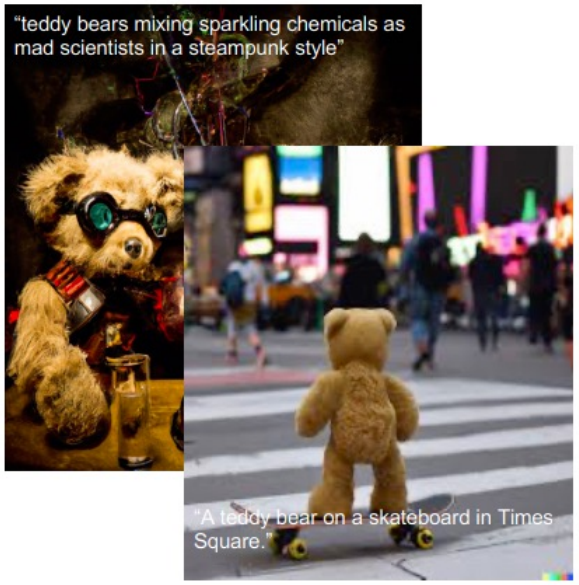
Text-conditional GANs: Motivation



GANs, Masked GIT
(GigaGAN, MUSE)



Autoregressive models
(Image GPT, Parti)



Diffusion models
(DALL-E 2, Imagen)

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Timeline: Text2Image Generation

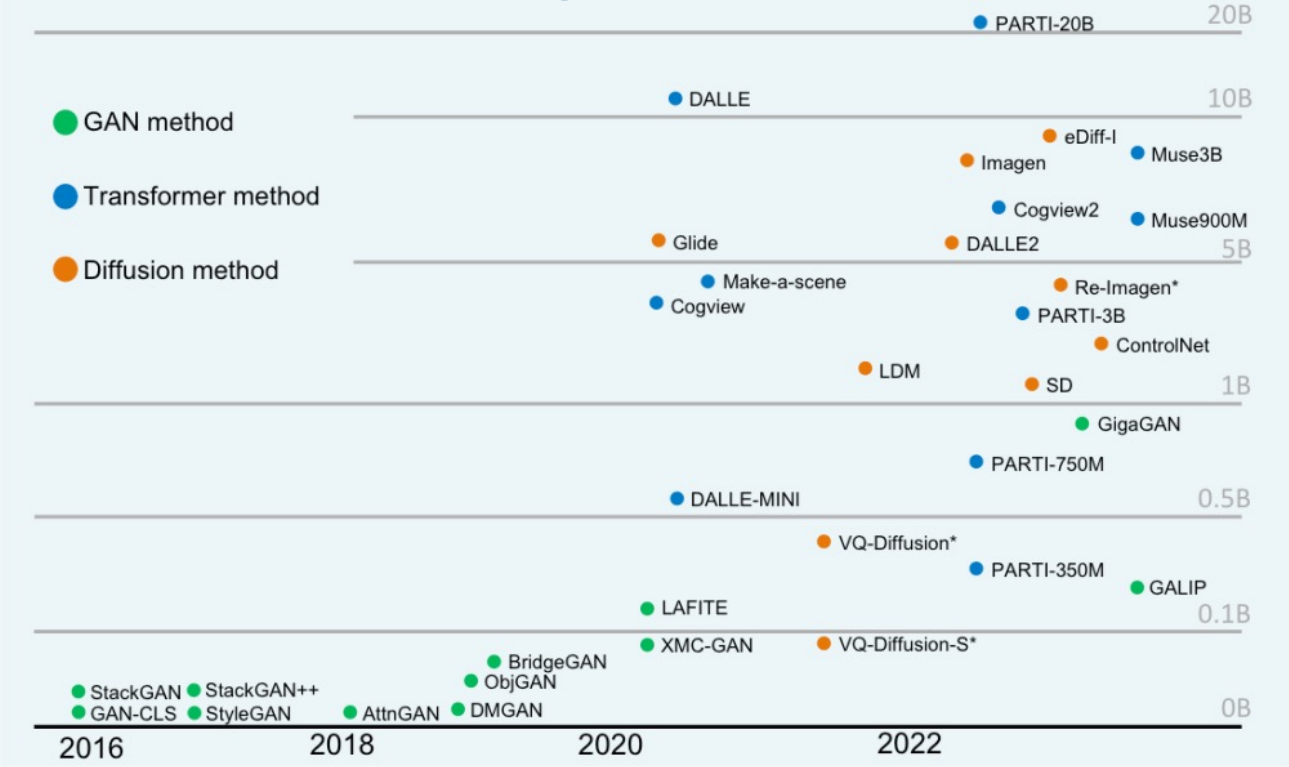


Fig. 5. Timeline of TTI model development, where green dots are GAN TTI models, blue dots are autoregressive Transformers and orange dots are Diffusion TTI models. Models are separated by their parameter, which are in general counted for all their components. Models with asterisk are calculated without the involvement of their text encoders.

https://www.cs.cmu.edu/~mgormley/courses/10423/schedule.html

Figure from Bie et al. (2023)

Timeline: Text2Image Generation

A comparison of the text to image methods discussed highlighting their date published, model configuration and evaluation results. For the model type, green dot refers to the GAN model TTI, blue dot refers to the autoregressive TTI and orange dot indicates the Diffusion TTI. For evaluation metrics, IS and FID score are provided under the evaluation of MSCOCO dataset in a zero-shot fashion. The last column provides the specific model size in scale of Million(M) or Billion(B); ⋆ : no zero-shot results found, use standard results instead.

Method	Date	Model Type	Data Size	Open Source	IS evaluation	FID evaluation	Model size
AttnGAN [33]	11/2017	🟢	120K	✗	20.80	35.49 ⋆	13M
StyleGAN [34]	11/2017	🟢	120K	✗	20.80	35.49 ⋆	-
Obj-GAN [220]	09/2019	🟢	120K	✓	24.09	36.52 ⋆	34M
Control-GAN [221]	09/2019	🟢	120K	✓	23.61	33.10 ⋆	-
DM-GAN [35]	04/2019	🟢	120K	✓	32.32	27.34 ⋆	21M
XMC-GAN [165]	01/2021	🟢	120K	✗	30.45	9.33 ⋆	90M
LAFITE [44]	11/2021	🟢	-	✓	26.02	26.94	150M
Retreival-GAN [208]	08/2022	🟢	120K	✗	29.33	9.13 ⋆	25M
GigaGAN [46]	01/2023	🟢	-	✗	-	10.24	650M
GALIP [45]	03/2023	🟢	3M-12M	✓	-	12.54	240M
DALLE [39]	02/2021	🔵	250M	✗	-	27.5	12B
Cogview [189]	06/2021	🔵	300M	✓	-	27.1	4B
Make-A-Scene	03/2022	🔵	35M	✗	-	11.84	4B
Cogview2 [43]	05/2022	🔵	300M	✓	-	24.0	6B
PARTI-350M [5]	06/2022	🔵	∼1000M	✗	-	14.10	350M
PARTI-20B [5]	06/2022	🔵	∼1000M	✗	-	7.23	20B
DALLE-mini [187]	07/2021	🔵	250M	✗	-	-	∼500M
MUSE-3B [31]	03/2023	🔵	∼1000M	✗	-	7.88	7.6B
GLIDE [40]	12/2021	🟠	250M	✓	-	12.24	5B
VQ-diffusion-F [68]	11/2021	🟠	>7M	✓	-	13.86 ⋆	370M
DALLE-2 [4]	04/2022	🟠	250M	✗	-	10.39	5.2B
Imagen [30]	05/2022	🟠	∼860M	✗	-	7.27	7.6B
LDM [3]	08/2022	🟠	400M	✓	30.29	12.63	1.45B
eDiff-I [197]	11/2022	🟠	1000M	✗	-	6.95	9B
Shift Diffusion[158]	08/2022	🟠	900M	✓	-	10.88	-
Re-Imagen[203]	09/2022	🟠	50M	✗	-	6.88	∼8B
ControlNet [159]	03/2023	🟠	-	✓	-	-	∼2.2B

https://www.cs.cmu.edu/~mgormley/courses/10423/schedule.html

Figure from Bie et al. (2023) http://arxiv.org/abs/2309.00810

Text-conditional GANs: Start

First the farmer gives hay to the goat. Then the farmer gets milk from the cow.

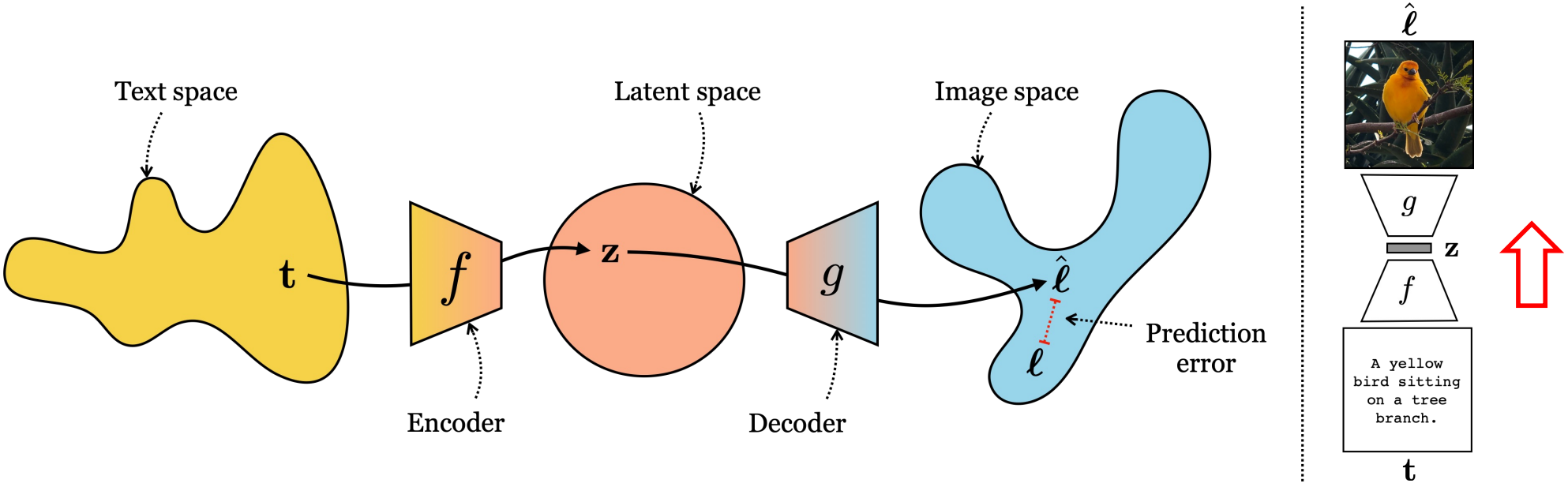
Step 1: Image Selection.
Step 2: Layout Optimization (Minimum overlap, Centrality, Closeness)

A Text-to-Picture Synthesis System for Augmenting Communication
Xiaojin Zhu, Andrew Goldberg, Mohamed Eldawy, Charles Dyer, and Bradley Strock. AAAI 2007

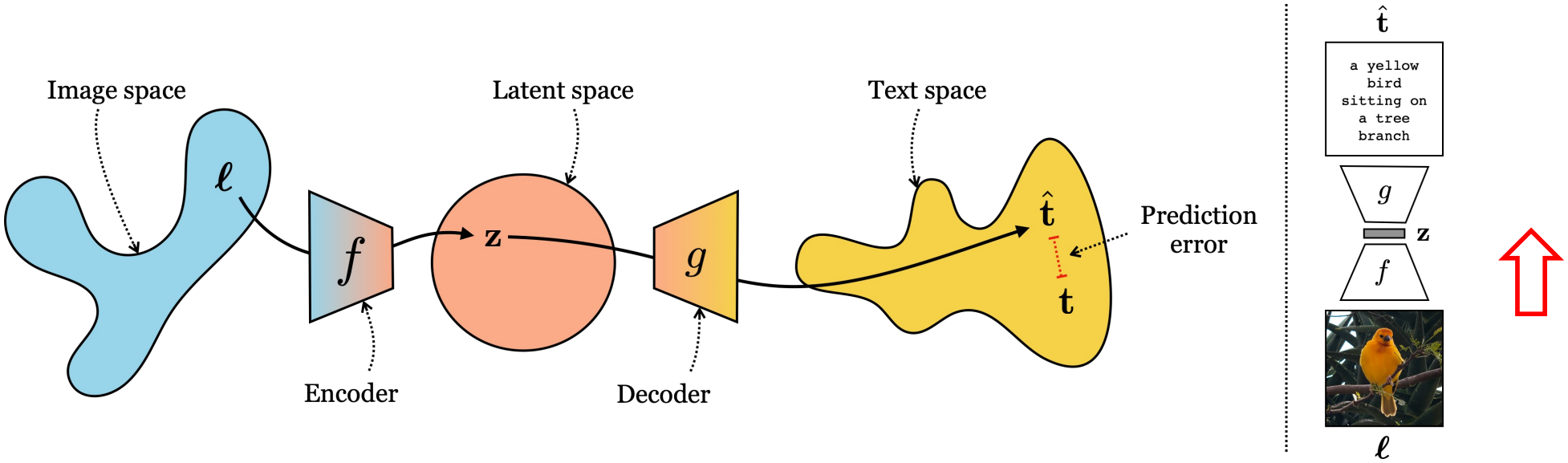
Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional AE Models

Text-conditional Auto Encoders



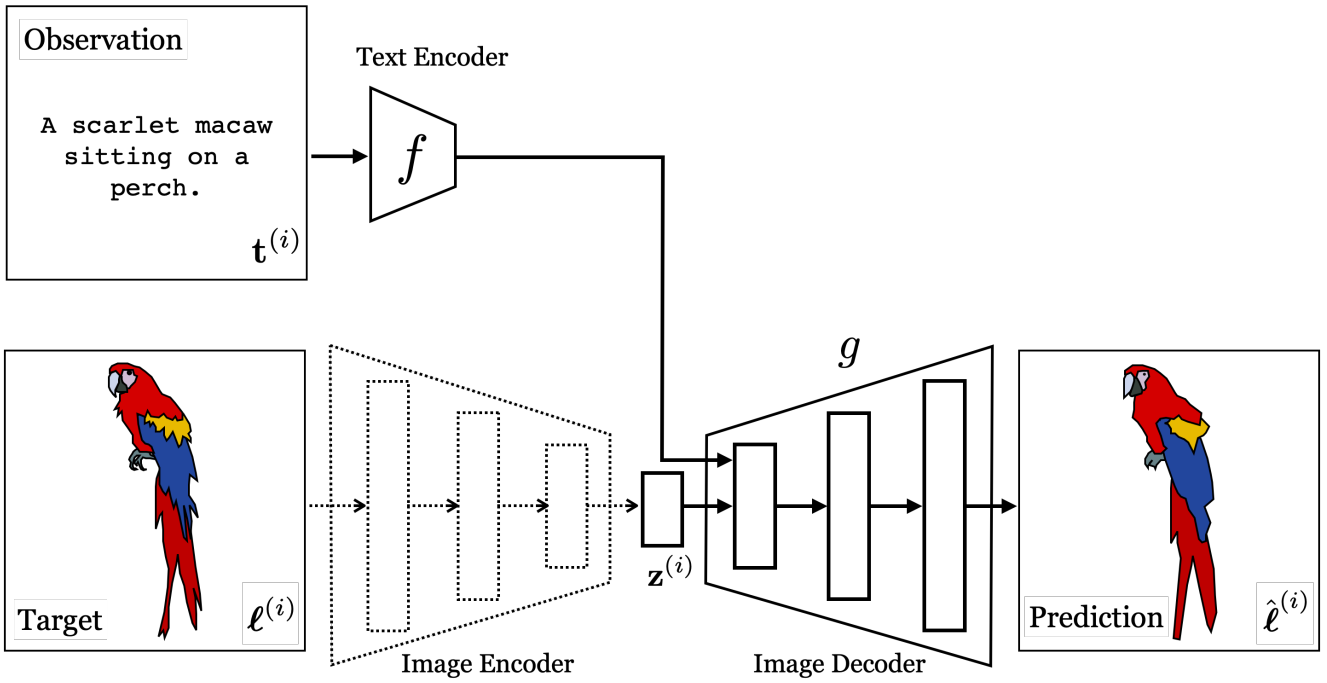
Text-conditional Auto Encoders



Text-conditional VAE Models

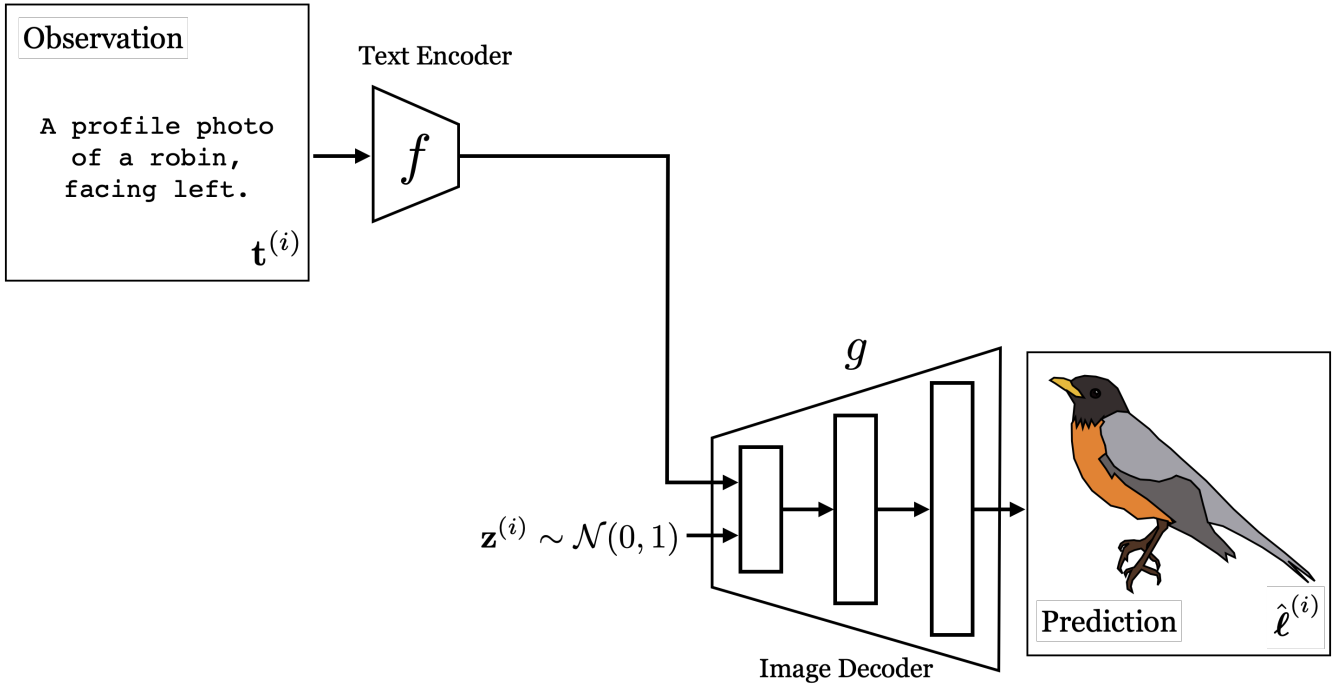
Text-conditional: Learning

Text-to-image architecture (cVAE) — training



Text-conditional: Learning

Text-to-image architecture (cVAE) — inference



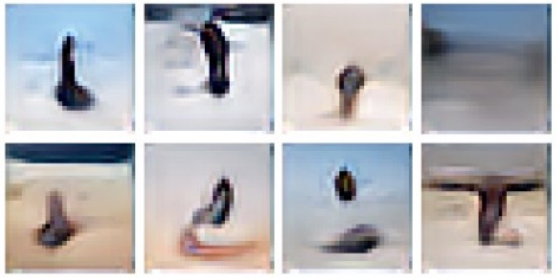
Text-conditional GANs Models

Text-conditional GANs: Start DL

A stop sign is flying in blue skies.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

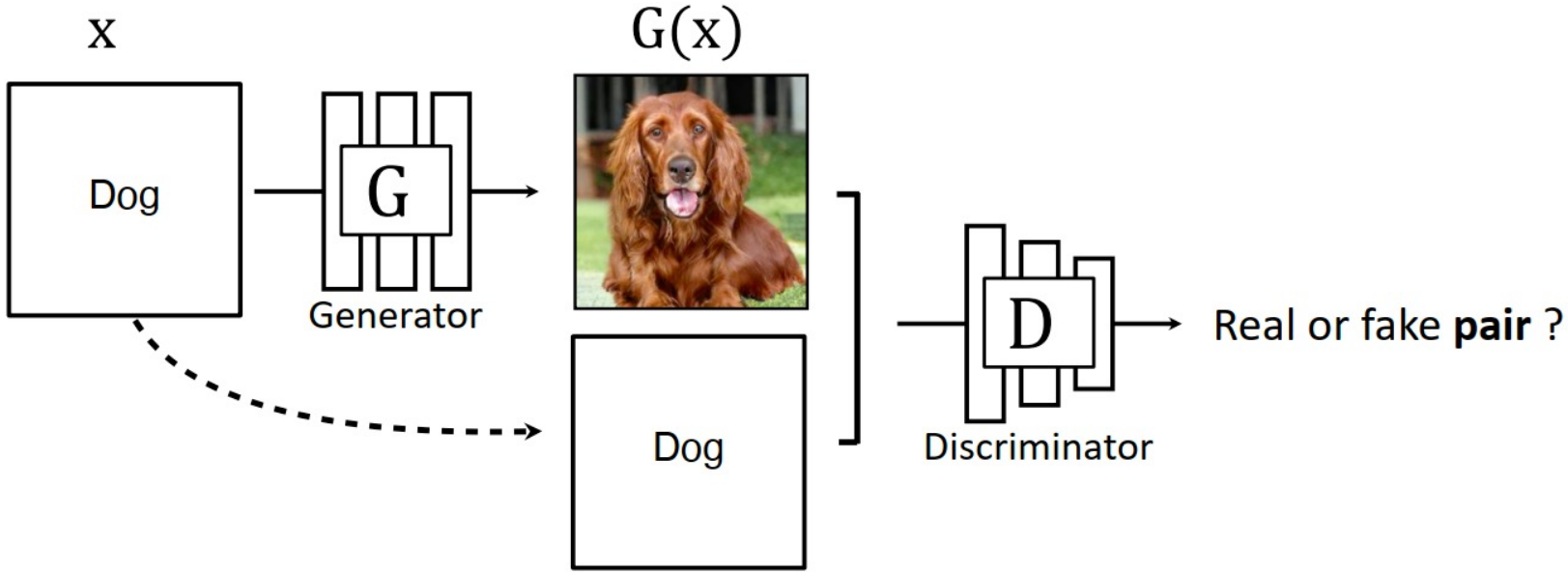


A person skiing on sand clad vast desert.

Generating Images from Captions with Attention.
Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, Ruslan Salakhutdinov. ICLR 2016

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-Conditional GANs: Class / Category

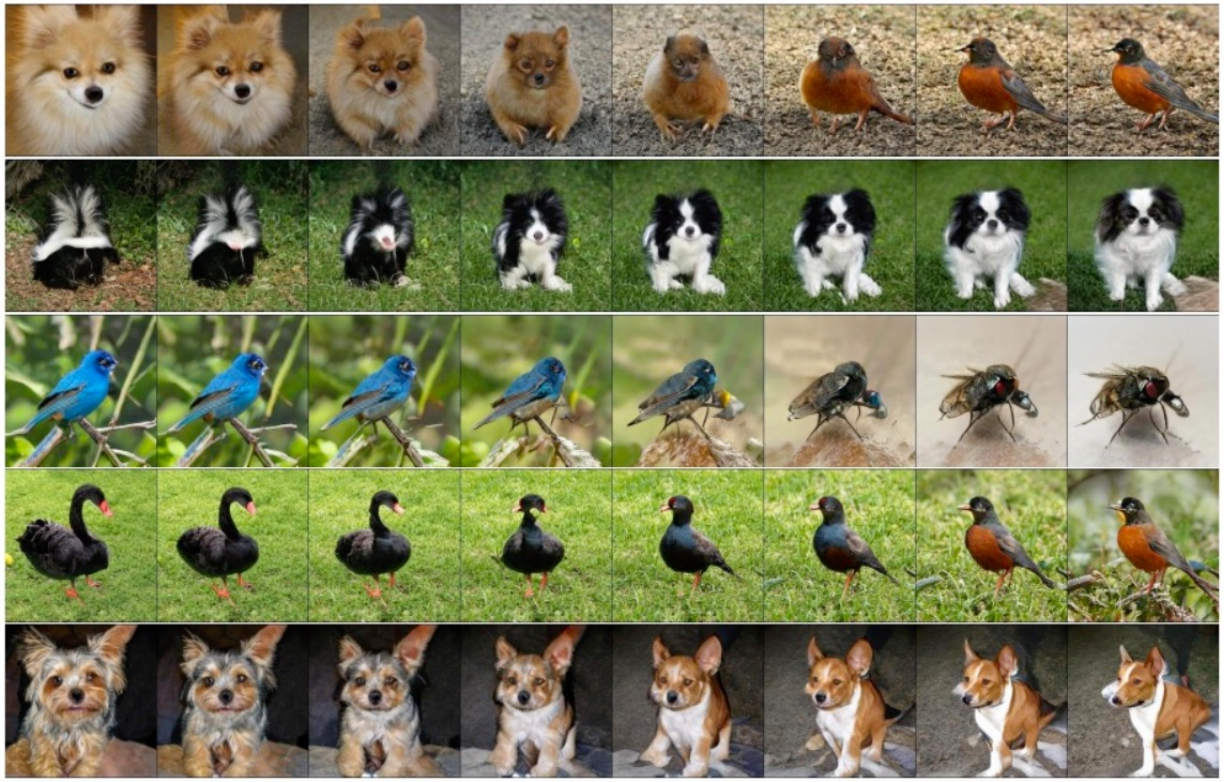


Input: **Class** → Output: **Photo**

Class-conditional GANs

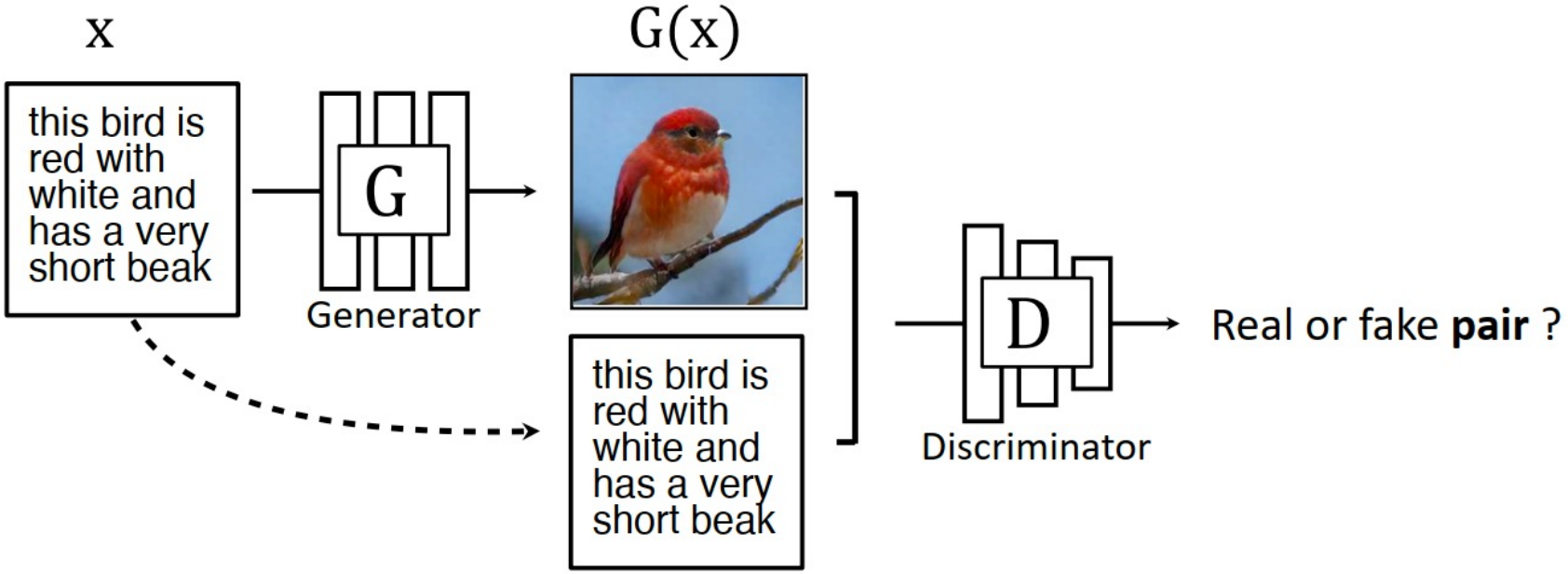
cGANs [Mirza and Osindero. 2014], SAGAN [Zhang et al., 2018], BigGAN [Brock et al., 2019] StyleGAN-XL [Sauer et al., 2022]

Text-conditional GANs: Class Conditioned – BigGAN - Feature Space Interpolation



cGANs [Mirza and Osindero. 2014], SAGAN [Zhang et al., 2018], BigGAN [Brock et al., 2019] StyleGAN-XL [Sauer et al., 2022]

Text-conditional GAN



Input: **Text** → Output: **Photo**

Text-to-Image Synthesis

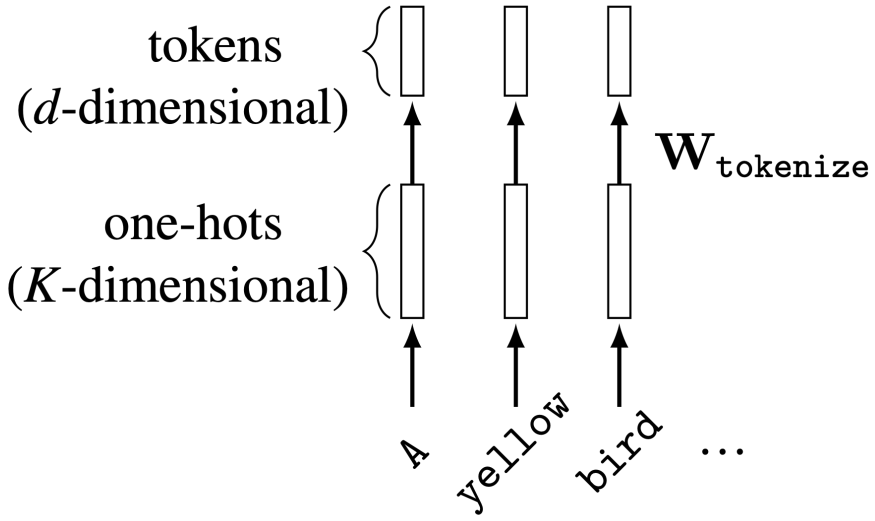
StackGAN, StackGAN++ [Zhang et al., 2016 and 2017], AttnGAN [Xu et al., 2018]

Text-conditional GANs: Representation

How to model Text?

Text-conditional GANs: Representation

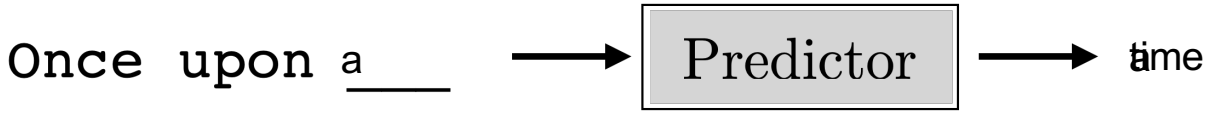
How to represent text as tokens?

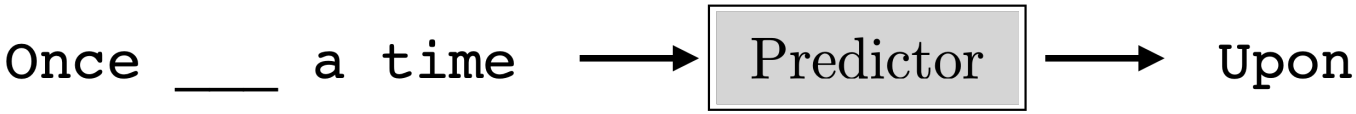


Note: sometimes the word "token" is used to refer to a unit of the discrete vocabulary we will model (the one-hots here). We use a more general definition, where a token can be discrete or continuous — each layer of a transformer consists of a set of tokens.

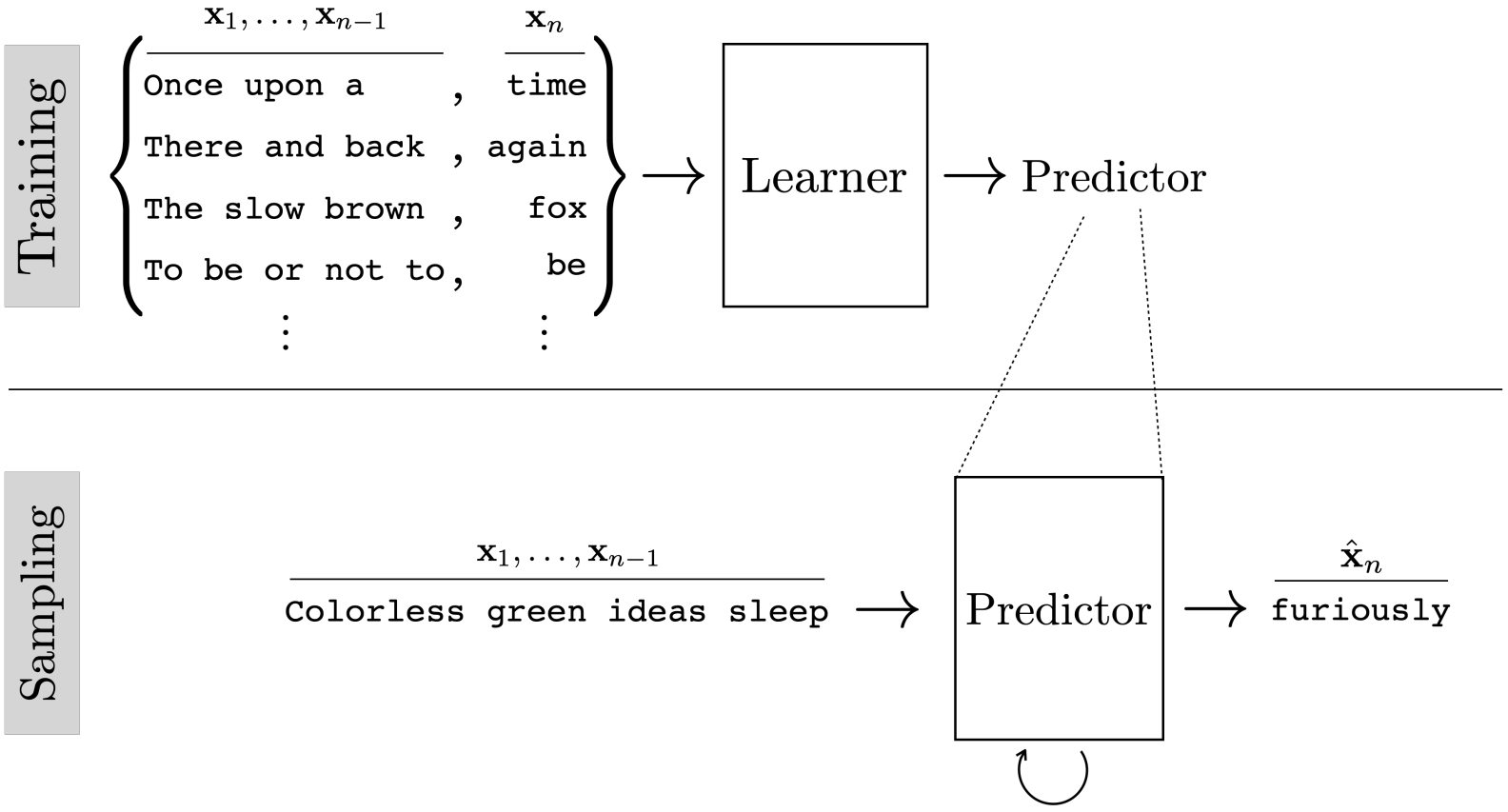
Text-conditional GANs: Learning

Language Models — Autoregressive



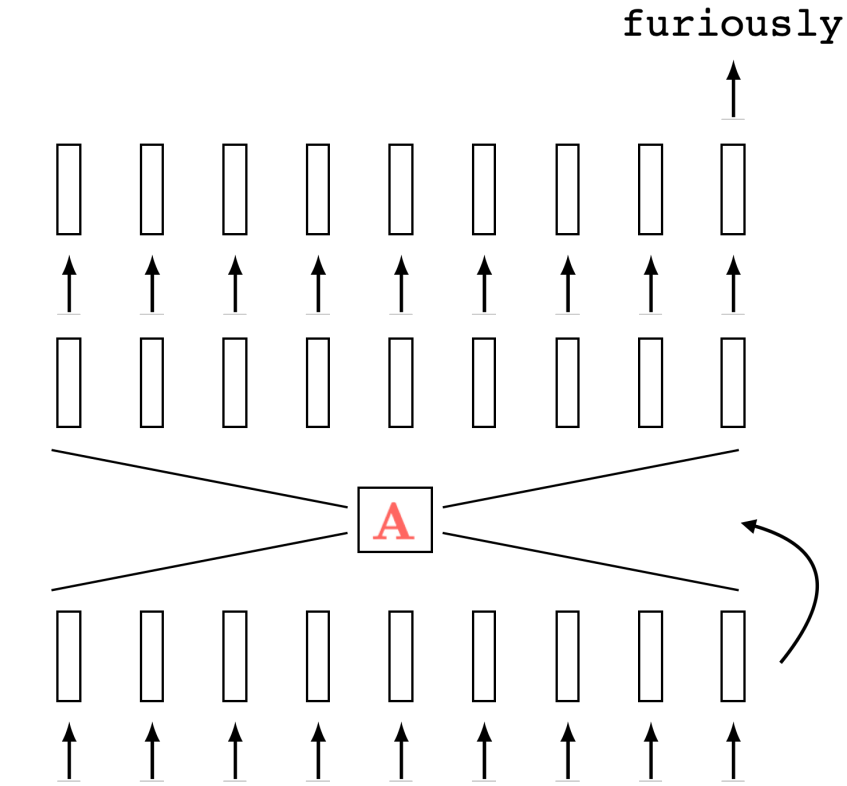


Text-conditional GANs: Learning



Text-conditional GANs: Learning

Language Models — Autoregressive



Text-conditional GANs: Details

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

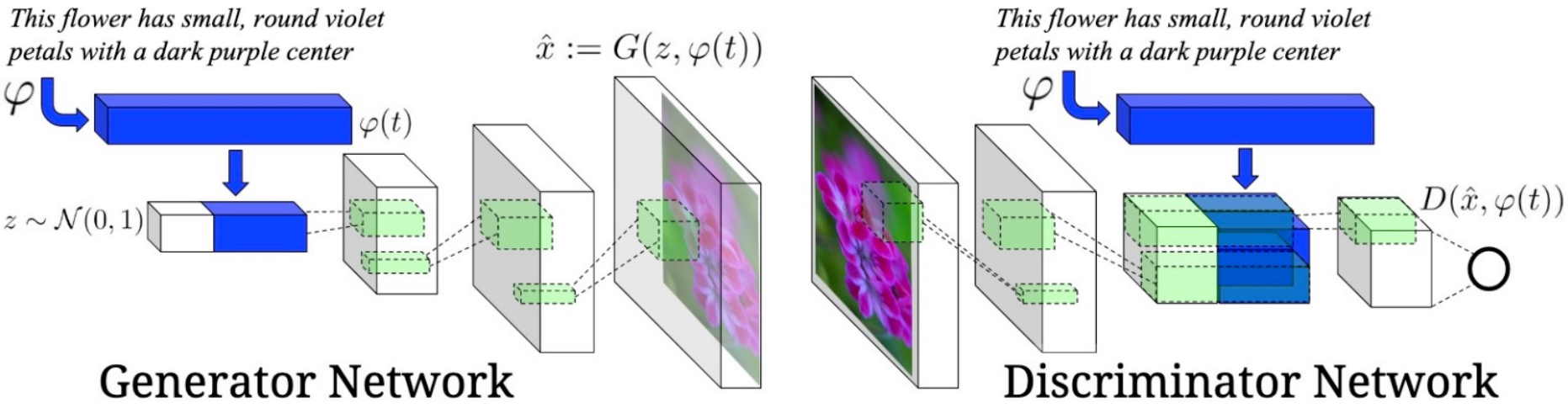
the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

Generative Adversarial Text to Image Synthesis
Scott Reed et al., ICML 2016

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs: Details



Conditional GAN + CNN + concatenation

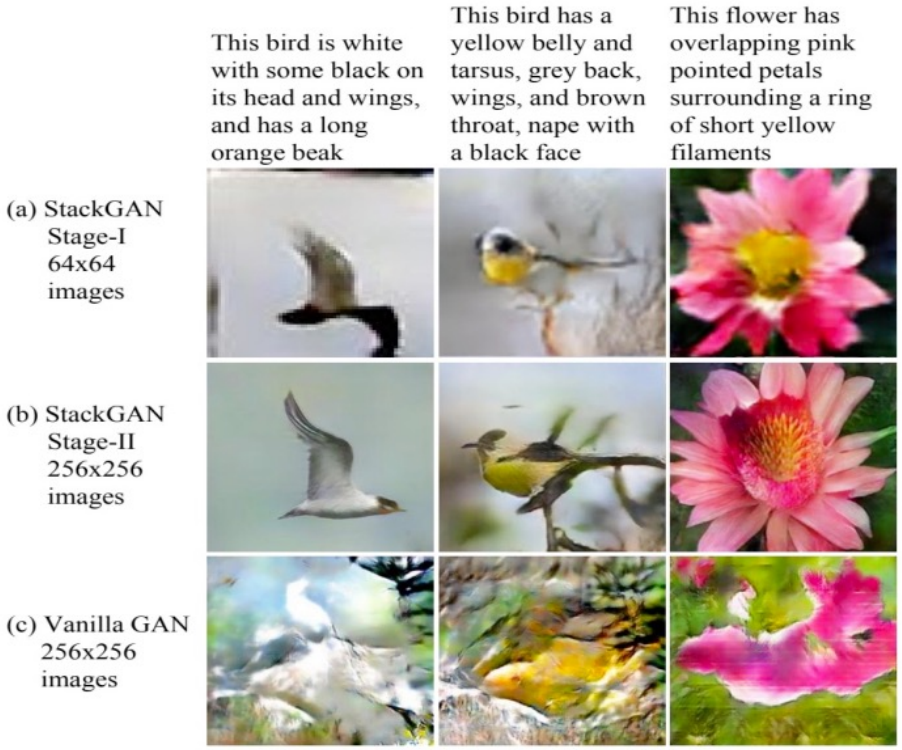
Generative Adversarial Text to Image Synthesis
Scott Reed et al., ICML 2016

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs

How to improve resolution?

Text-conditional GANs: Two-stage Model

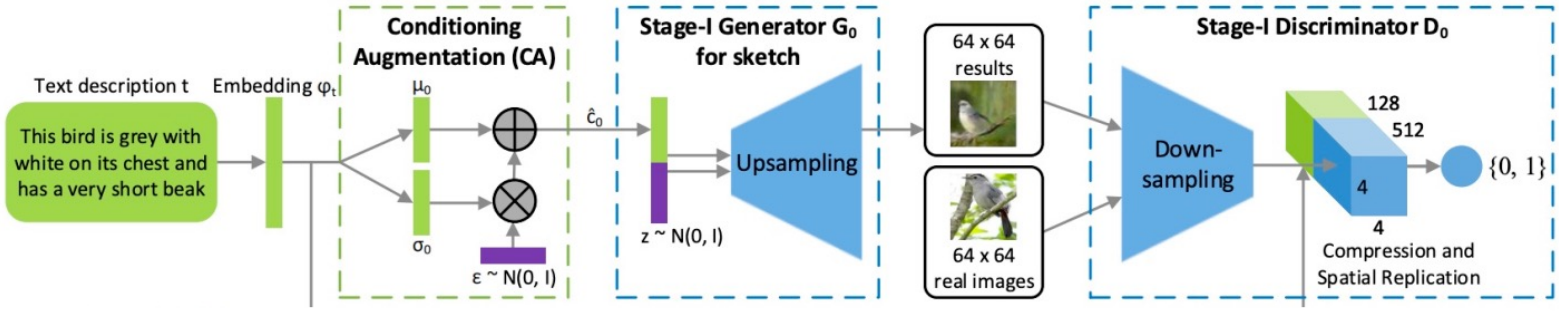


Two-stage Conditional GAN + CNN + concatenation

StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
Han Zhang et al., ICCV 2017

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs: Two-stage Model



Two-stage Conditional GAN + CNN + concatenation

StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
Han Zhang et al., ICCV 2017

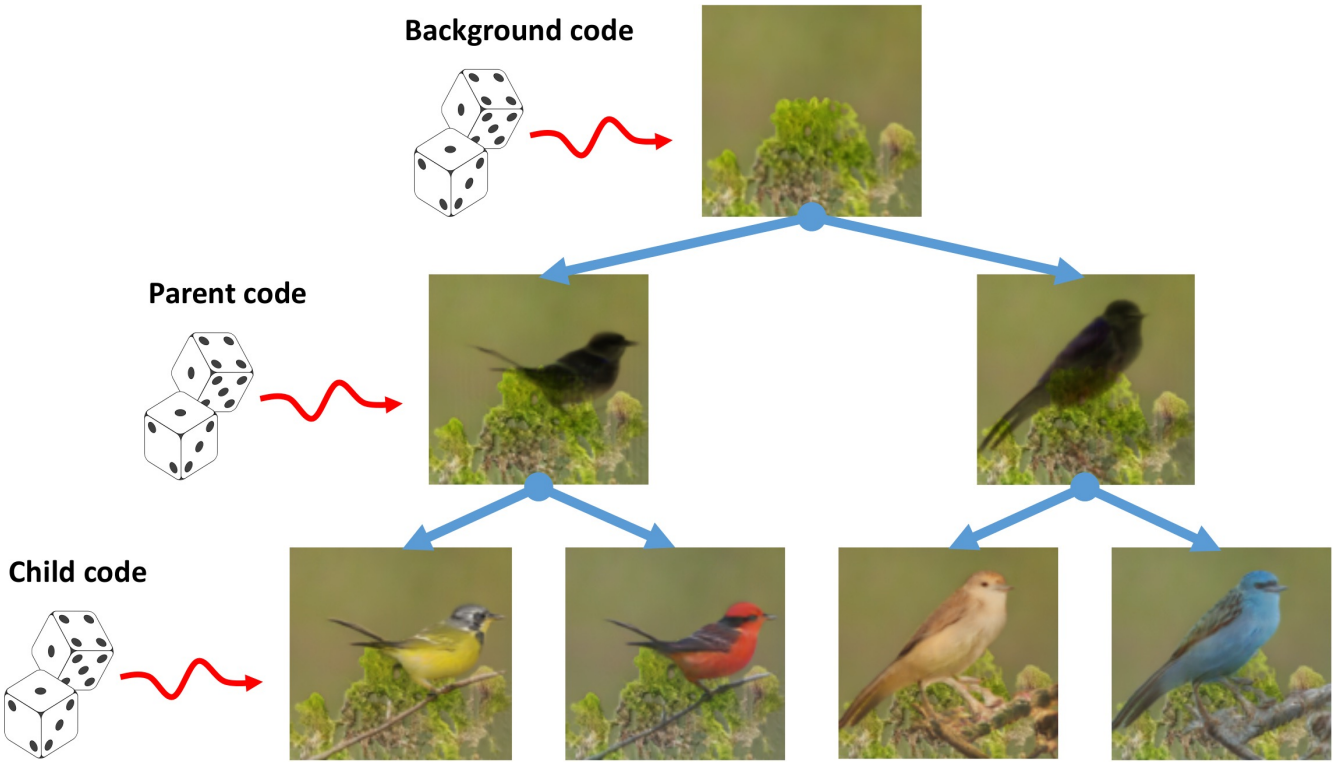
Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs: Two-stage Model

StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
Han Zhang et al., ICCV 2017

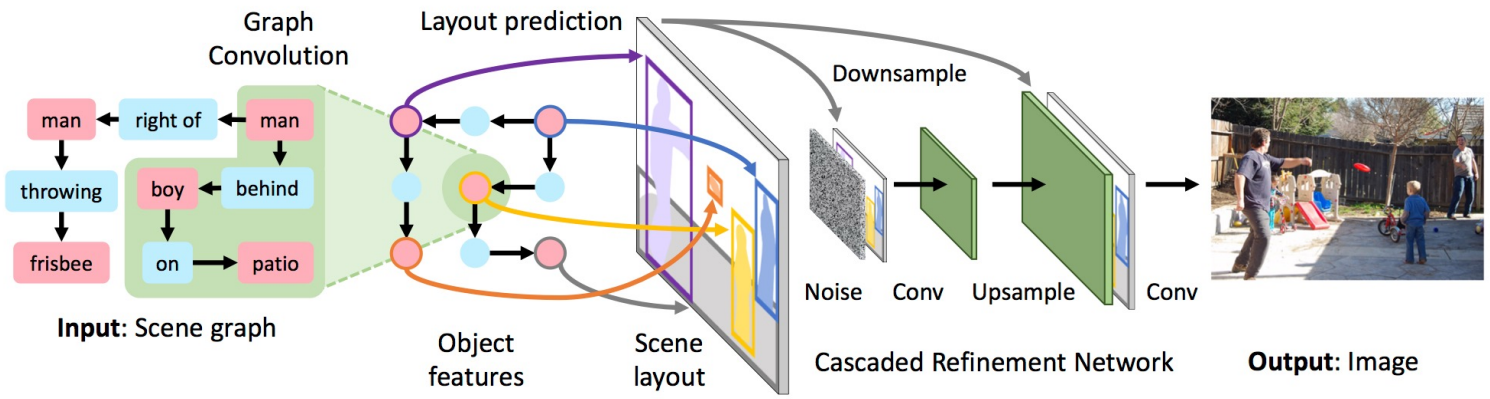
Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Conditional GANs: Multi-stage Model



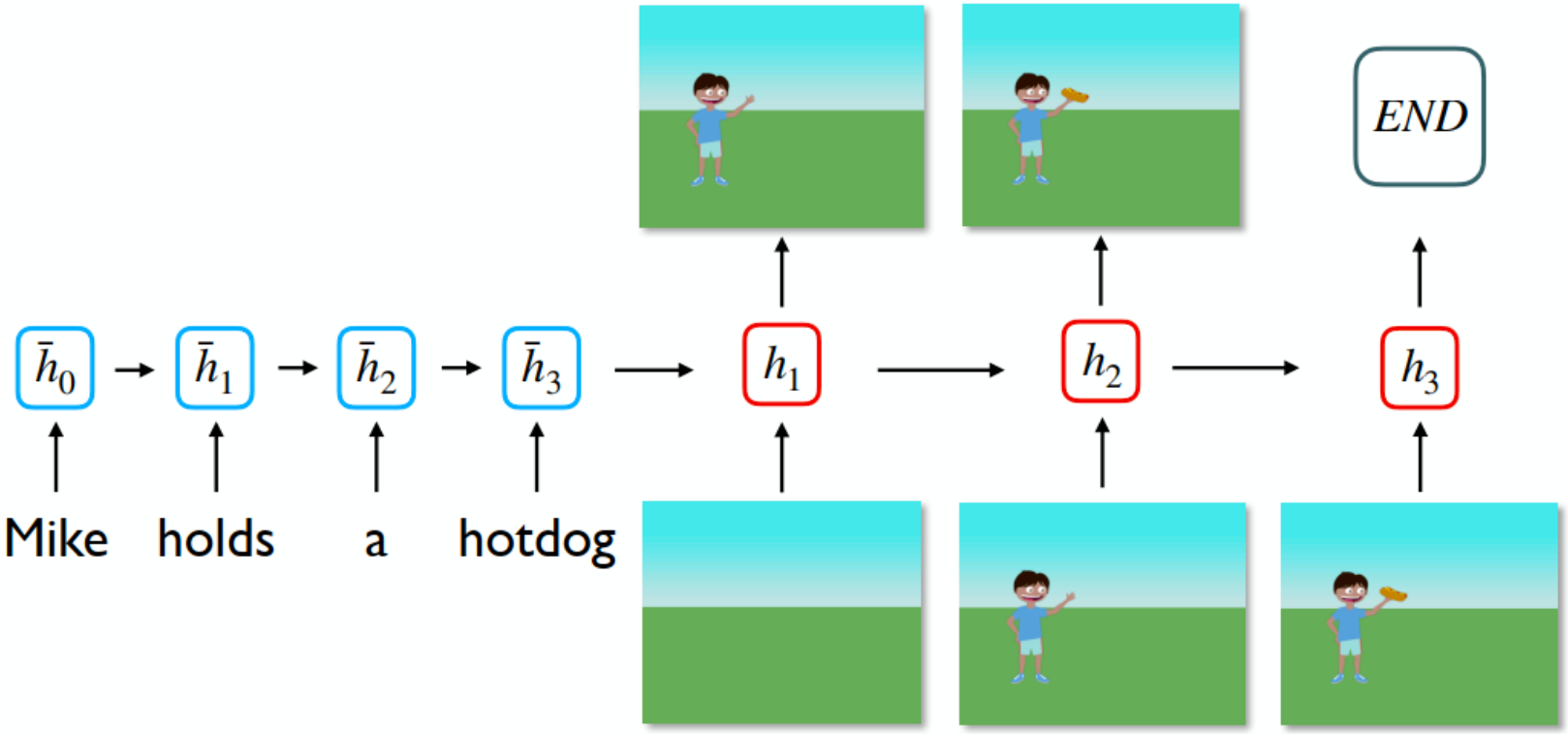
Singh et al., "FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery", CVPR 2019

Conditional GANs: Multi-stage Model



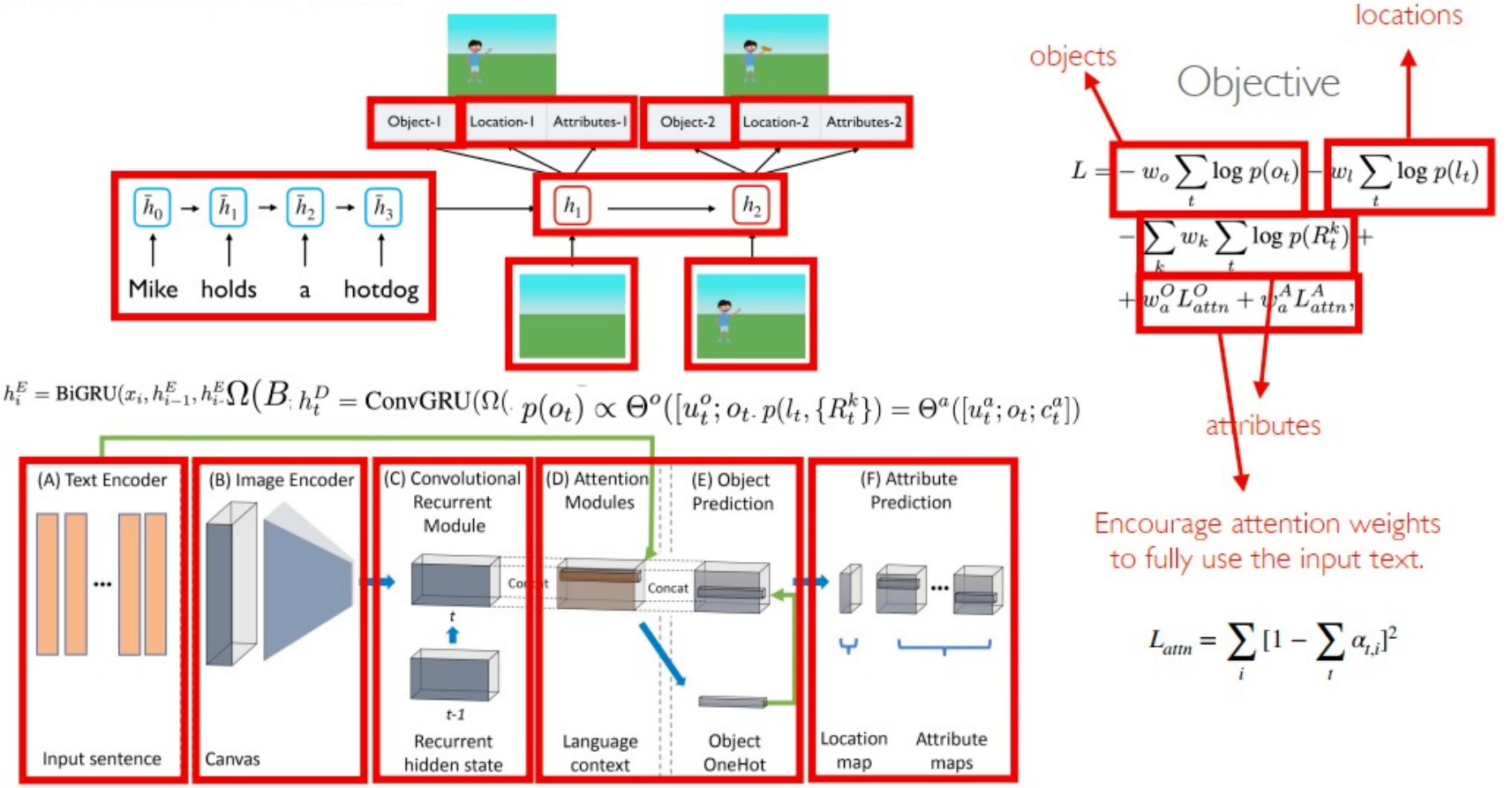
Johnson et al., "Image Generation from Scene Graphs", CVPR 2018

Text to Scene as Machine Translation!



Text2Scene: Generating Compositional Scenes from Textual Descriptions Fuwen Tan, Song Feng, Vicente Ordonez. Intl. Conference on Computer Vision and Pattern Recognition. CVPR 2019

Text to Scene as Machine Translation!



$$h_i^E = \text{BiGRU}(x_i, h_{i-1}^E, h_{i-1}^E) \quad \Omega(B, h_t^D = \text{ConvGRU}(\Omega(\quad p(o_t) \propto \Theta^o([u_t^o; o_t \quad p(l_t, \{R_t^k\}) = \Theta^a([u_t^a; o_t; c_t^a])$$

$$L = -w_o \sum_t \log p(o_t) - w_l \sum_t \log p(l_t) - \sum_k w_k \sum_t \log p(R_t^k) + w_a^O L_{attn}^O + w_a^A L_{attn}^A,$$

$$L_{attn} = \sum_i [1 - \sum_t \alpha_{t,i}]^2$$

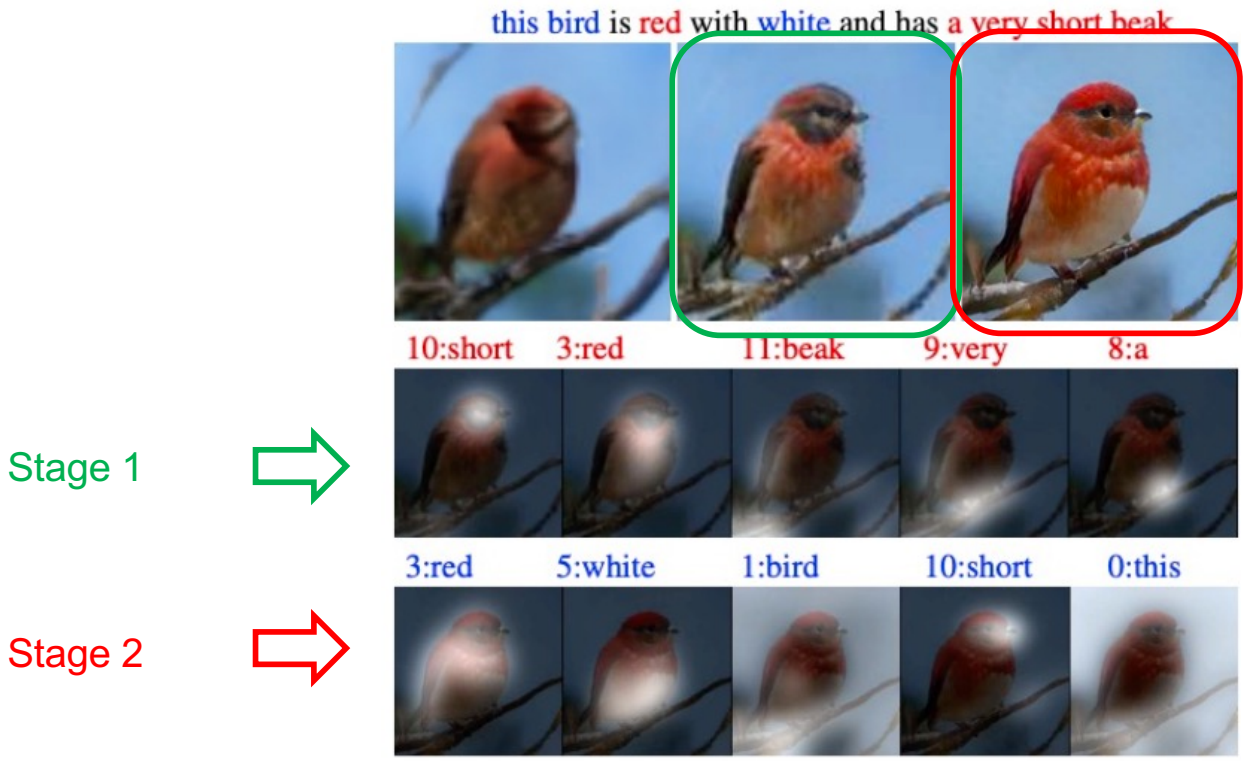
Text2Scene: Generating Compositional Scenes from Textual Descriptions Fuwen Tan, Song Feng, Vicente Ordonez. Intl. Conference on Computer Vision and Pattern Recognition. CVPR 2019

Text to Scene as Machine Translation!

https://vislang.ai/text2scene

Text2Scene: Generating Compositional Scenes from Textual Descriptions Fuwen Tan, Song Feng, Vicente Ordonez. Intl. Conference on Computer Vision and Pattern Recognition. CVPR 2019

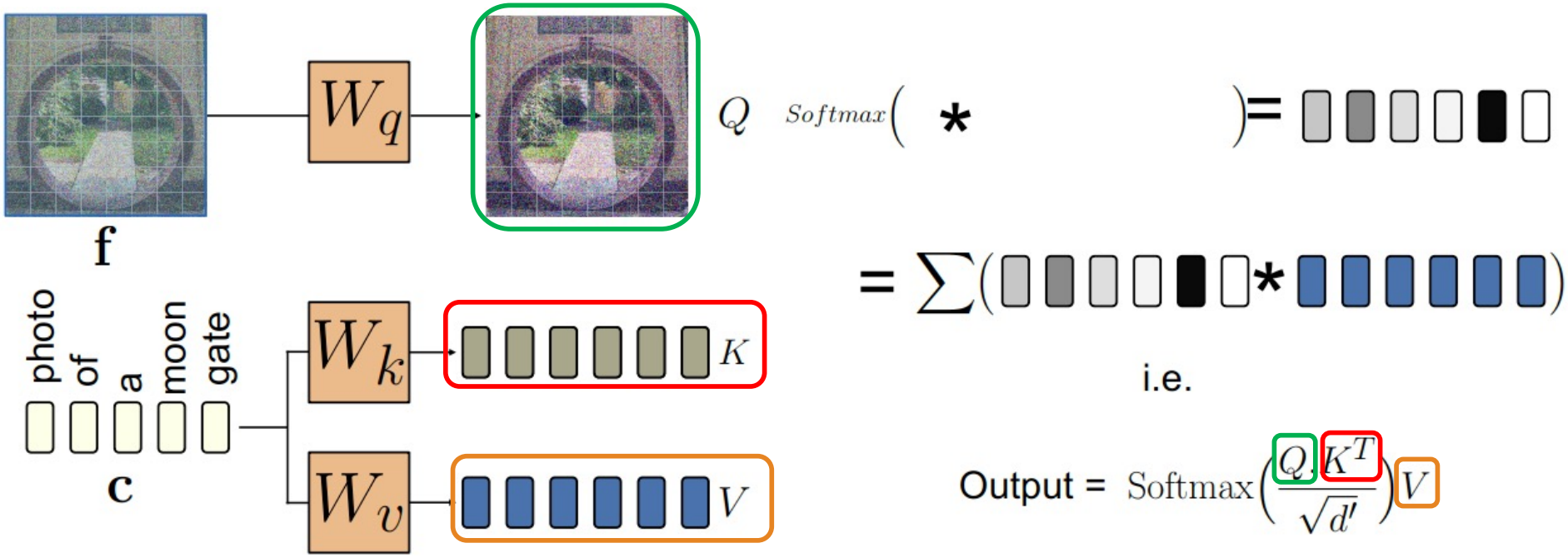
Text-conditional GANs: + Attention



AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks
Tao Xu et al., CVPR 2018

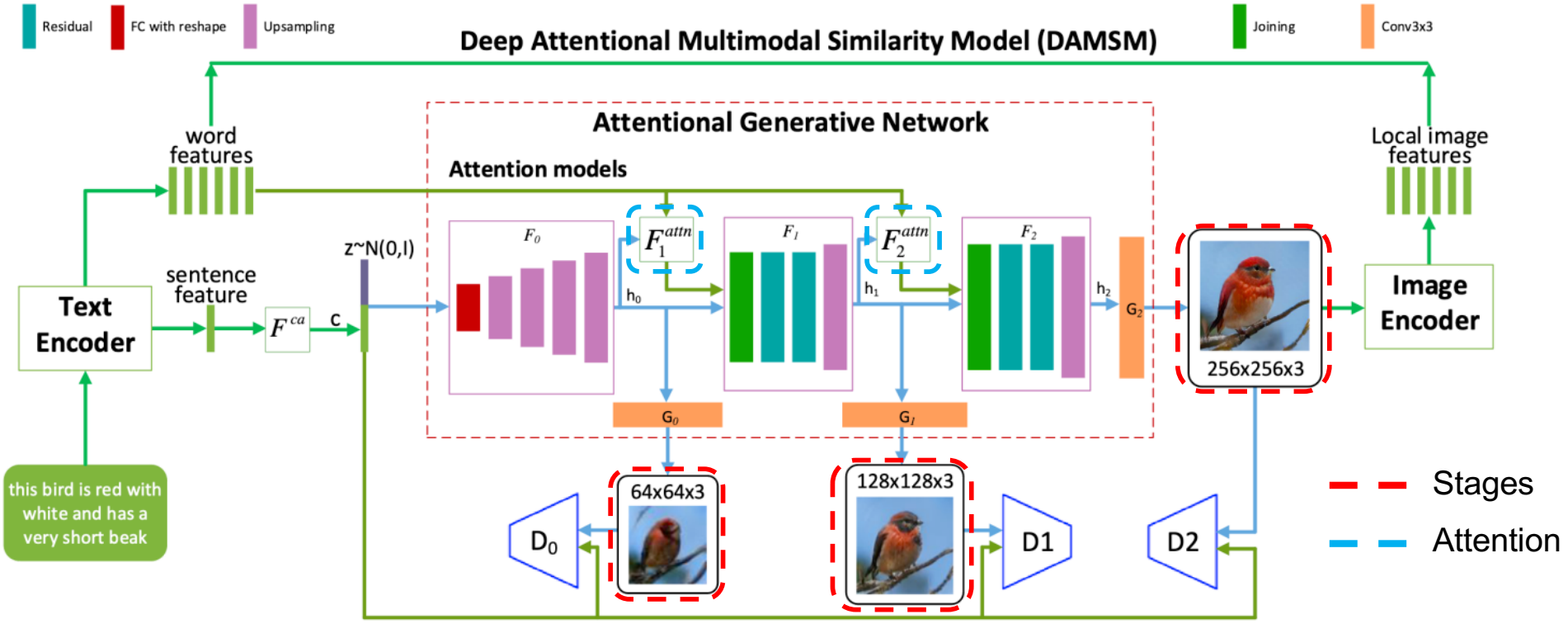
Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs: Cross-attention



Slides from [Kumari et al., CVPR 2023]

Text-conditional GANs: Cross-attention



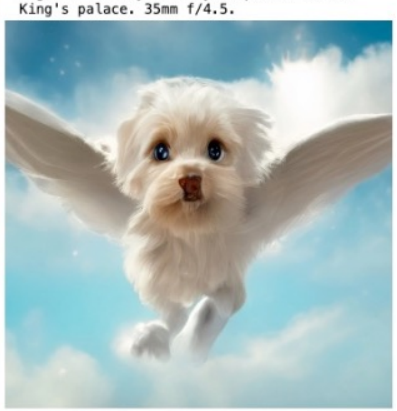
AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks
Tao Xu et al., CVPR 2018

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs: GigaGAN – Scaling up GAN

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

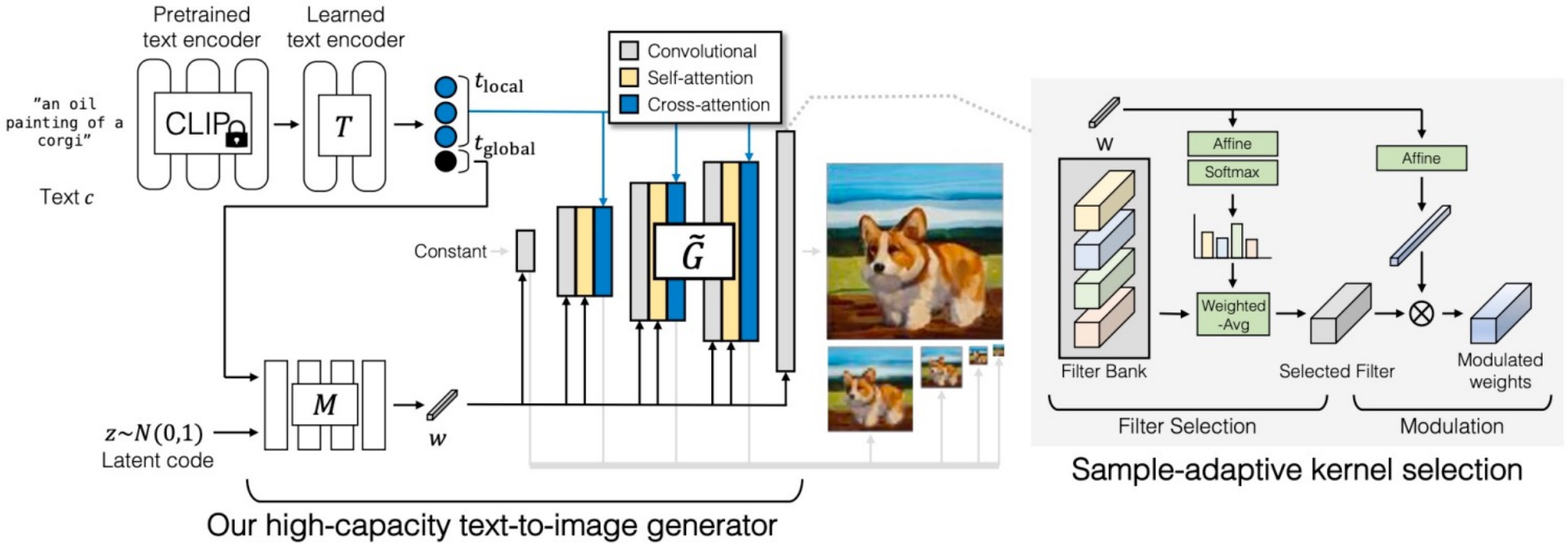
A golden luxury motorcycle parked at the King's palace. 35mm f/4.5.



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

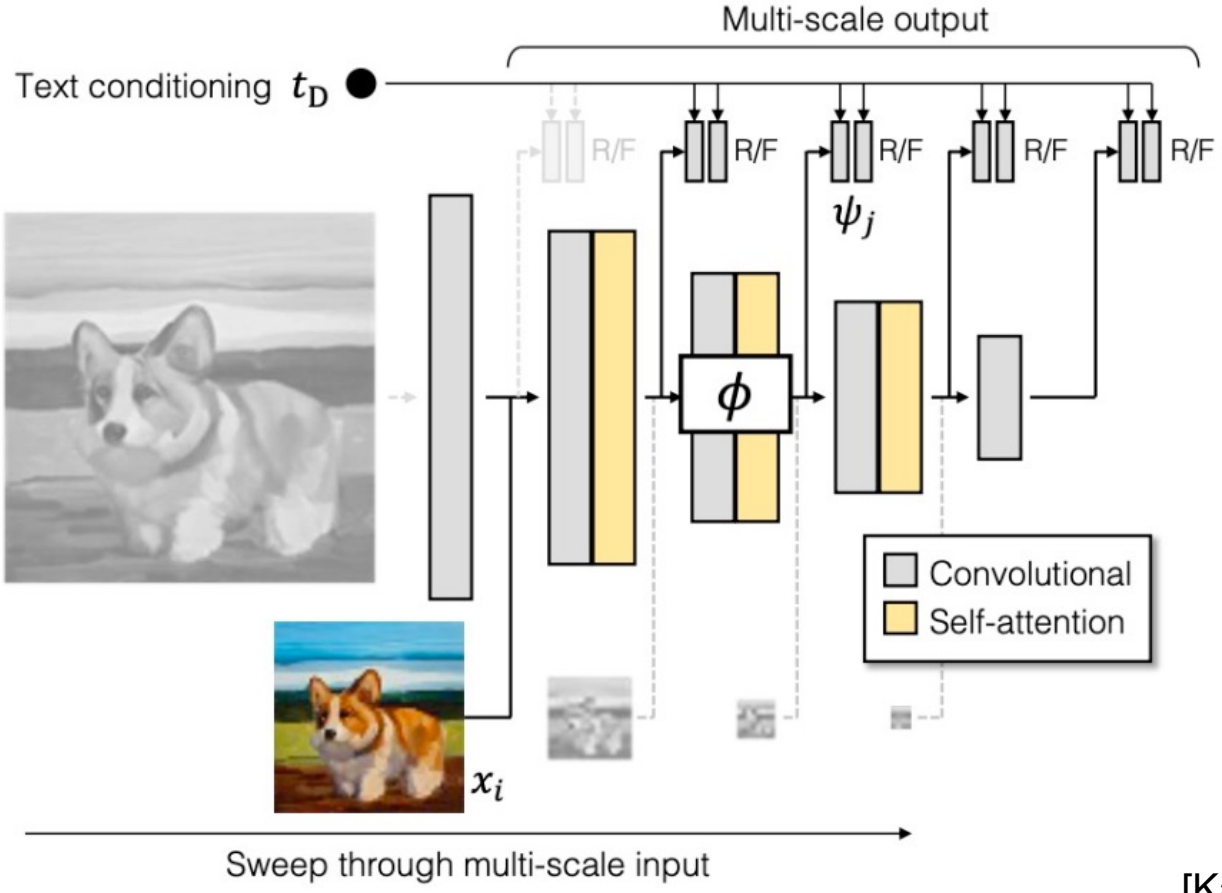
[Kang et al., CVPR 2023]

Text-conditional GANs: GigaGAN Generator



[Kang et al., CVPR 2023]

Text-conditional GANs: GigaGAN Discriminator

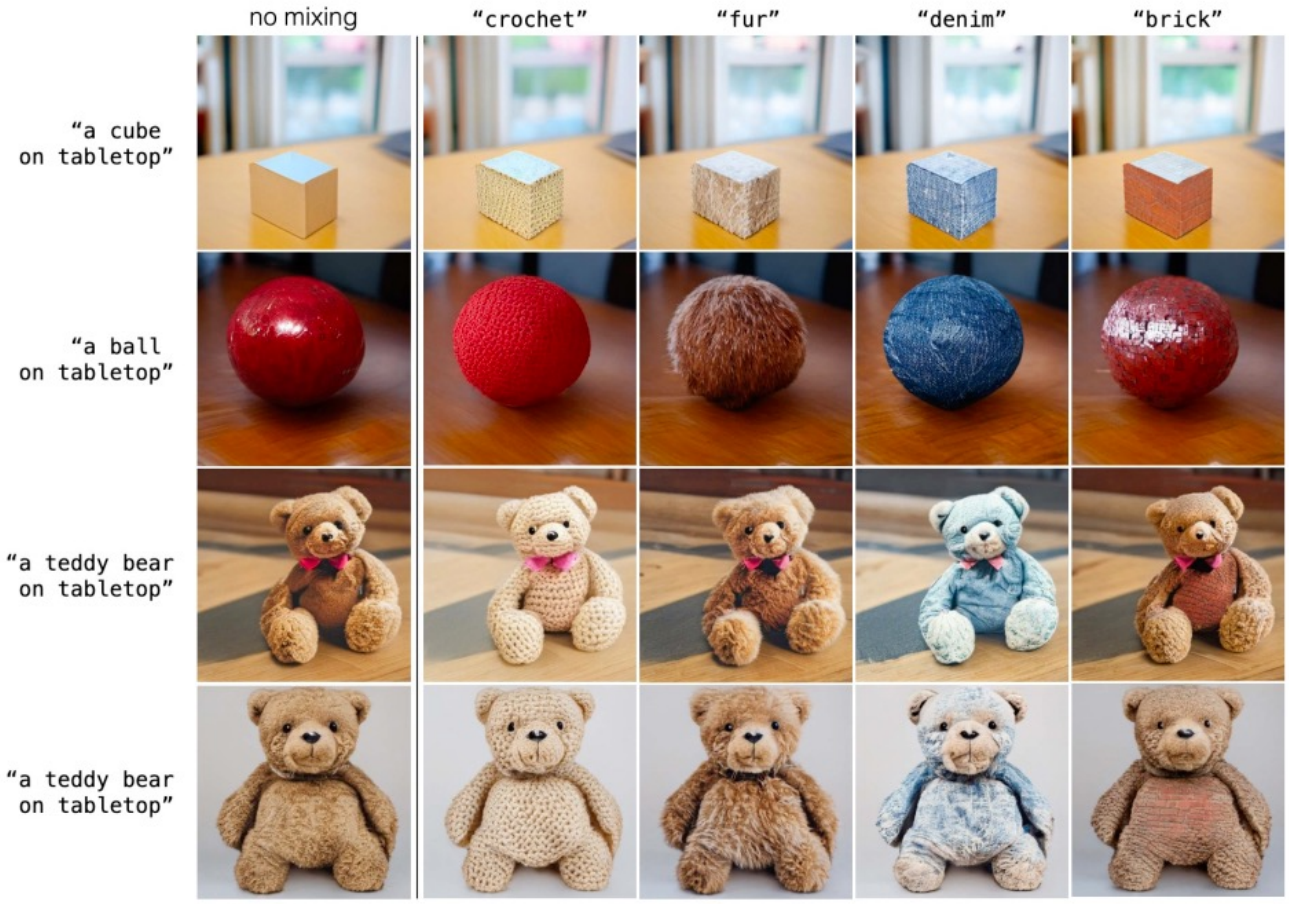


[Kang et al., CVPR 2023]

Text-conditional GANs: GigaGAN Style Mixing

[Kang et al., CVPR 2023]

Text-conditional GANs: GigaGAN Prompt Mixing



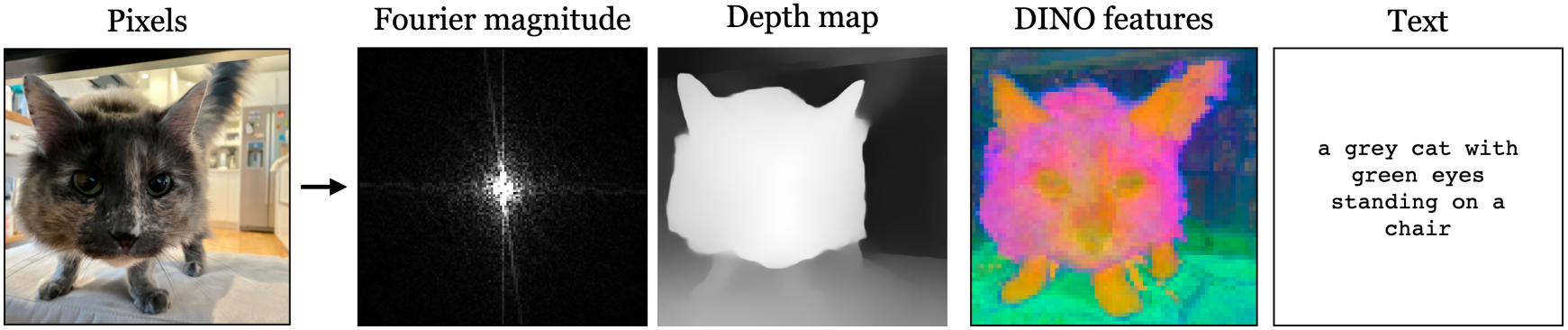
[Kang et al., CVPR 2023]

Text-conditional GANs: GigaGAN Results

[Kang et al., CVPR 2023]

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional GANs: Applications

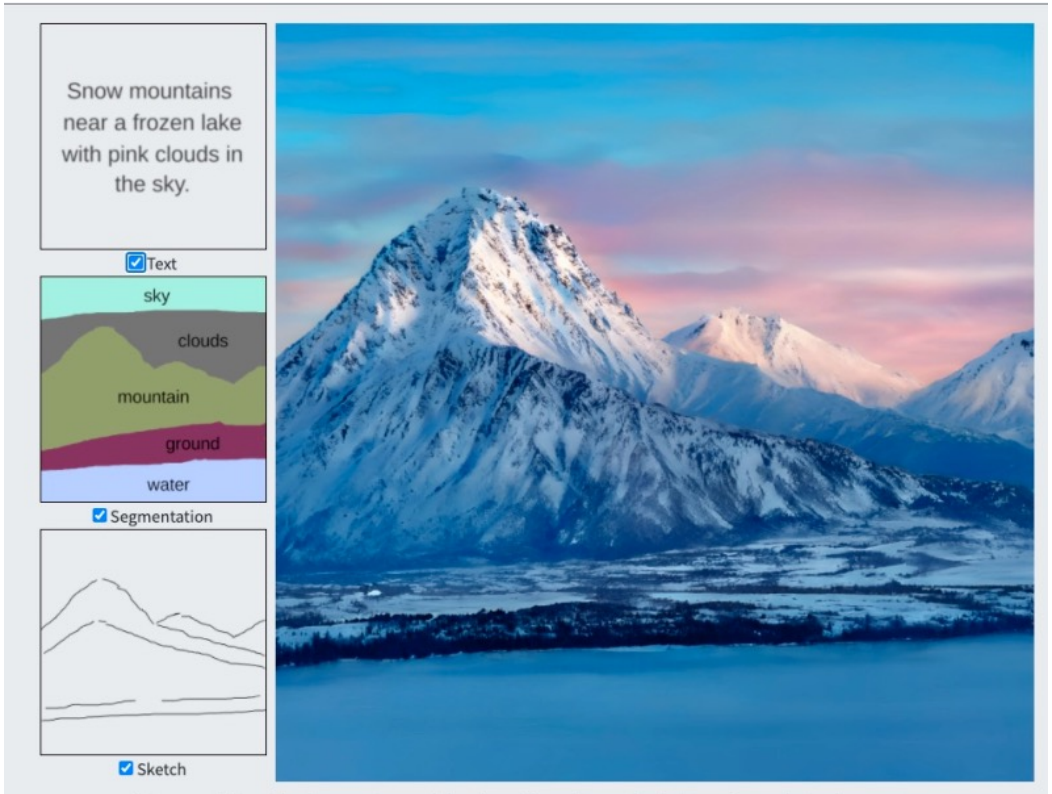


Different kinds of visual representations

Conditional GANs: Applications [Example-Guided Translation]

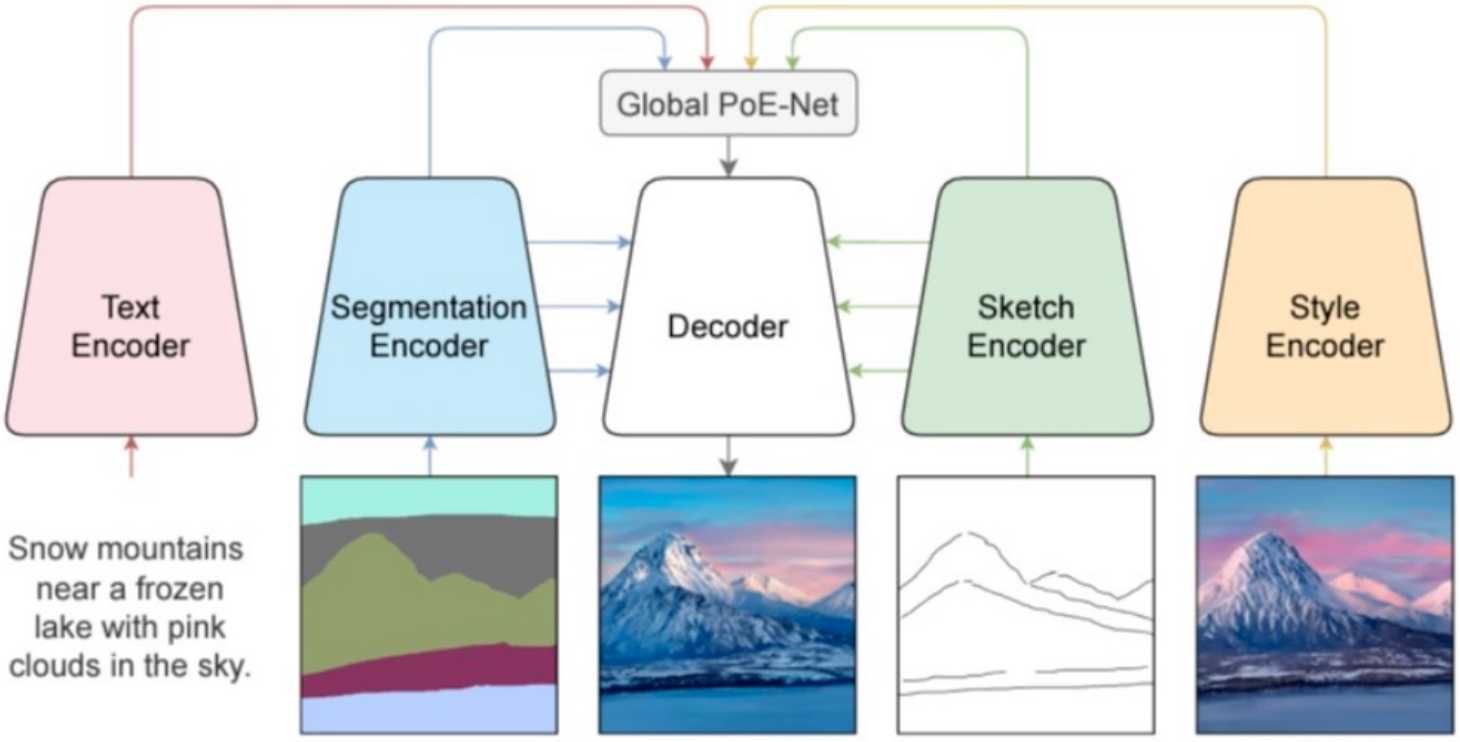
Multimodal Unsupervised Image-to-Image Translation - Xun Huang et al. ECCV 2018

Conditional GANs: Applications



Multimodal Conditional Image Synthesis with Product-of-Experts GANs [Huang et al., 2021]

Conditional GANs: Applications

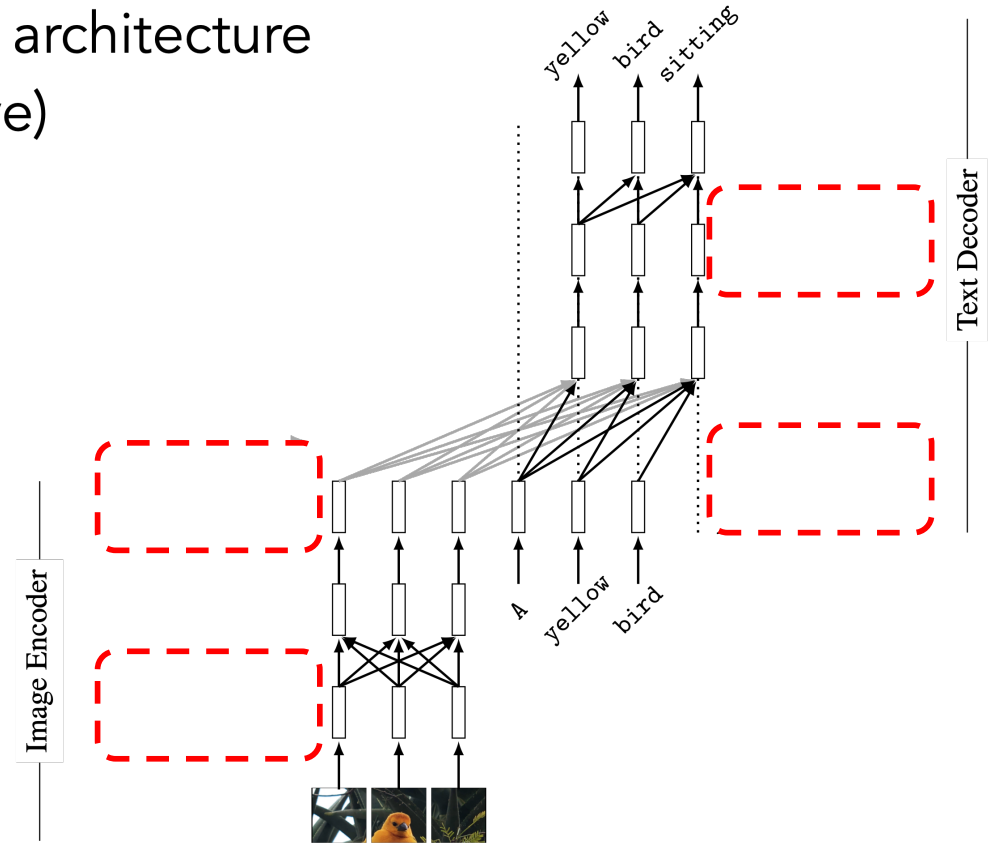


Multimodal Conditional Image Synthesis with Product-of-Experts GANs [Huang et al., 2021]

Text-conditional Autoregressive Models

Text-conditional: Transformers

Image-to-text architecture
(autoregressive)



Text-conditional Autoregressive - Parti: Scaling VQGAN

See released “Parti” paper by Google (text-to-image model)

- https://parti.research.google/

Slide credit: Robin Rombach

Text-conditional Autoregressive - Parti: Scaling VQGAN

[Step 1] Image tokenization (ViT-VQGAN)

- Pre-train a model to convert images into image tokens (discrete set of embeddings)

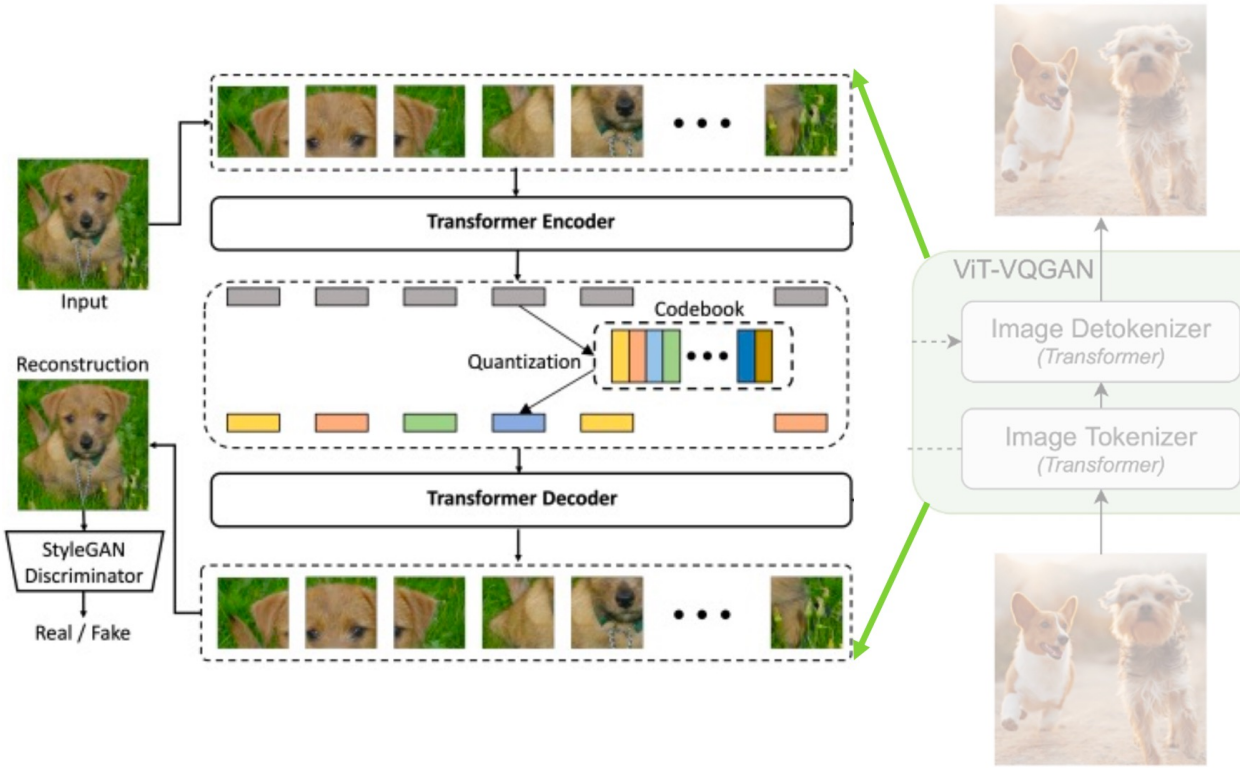


Figure adapted from https://arxiv.org/pdf/2206.10789 and https://arxiv.org/pdf/2110.04627

Text-conditional Autoregressive - Parti: Scaling VQGAN

[Step 2] Training

- treat image generation as a ***sequence-to-sequence problem***
- text prompt is input to encoder (pretrained BERT)
- sequence of image tokens is output of decoder

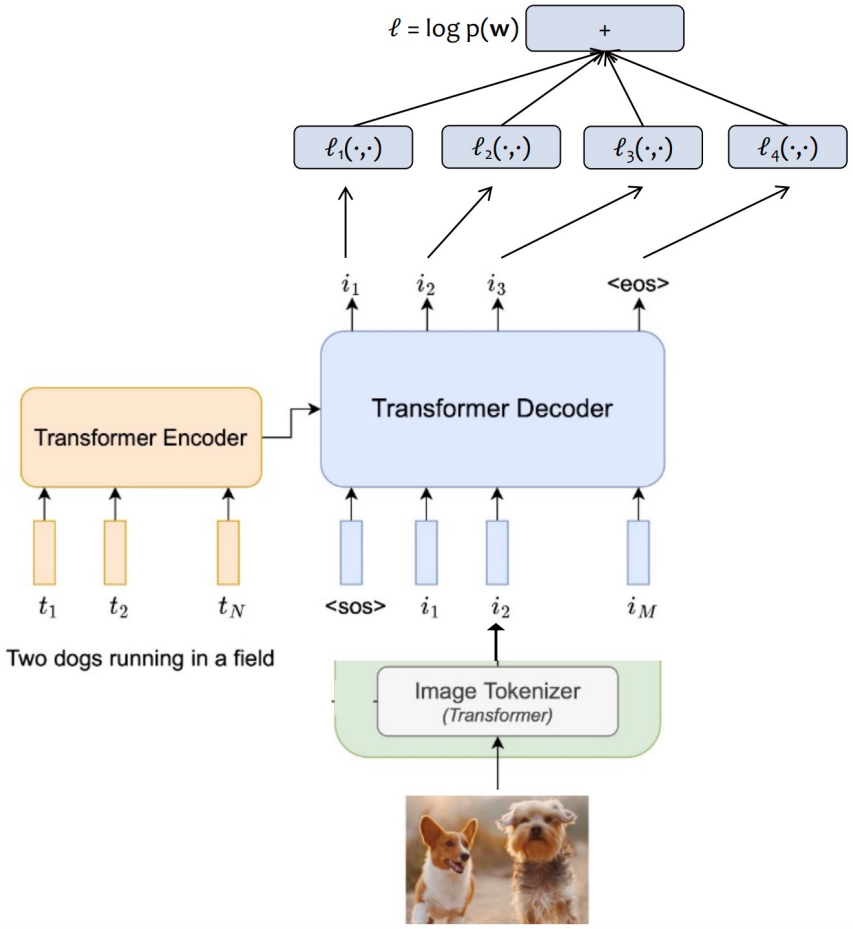


Figure adapted from https://arxiv.org/pdf/2206.10789

Text-conditional Autoregressive - Parti: Scaling VQGAN

[Step 3] Generation

- ViT-VQGAN takes in the image tokens and generates a high quality image

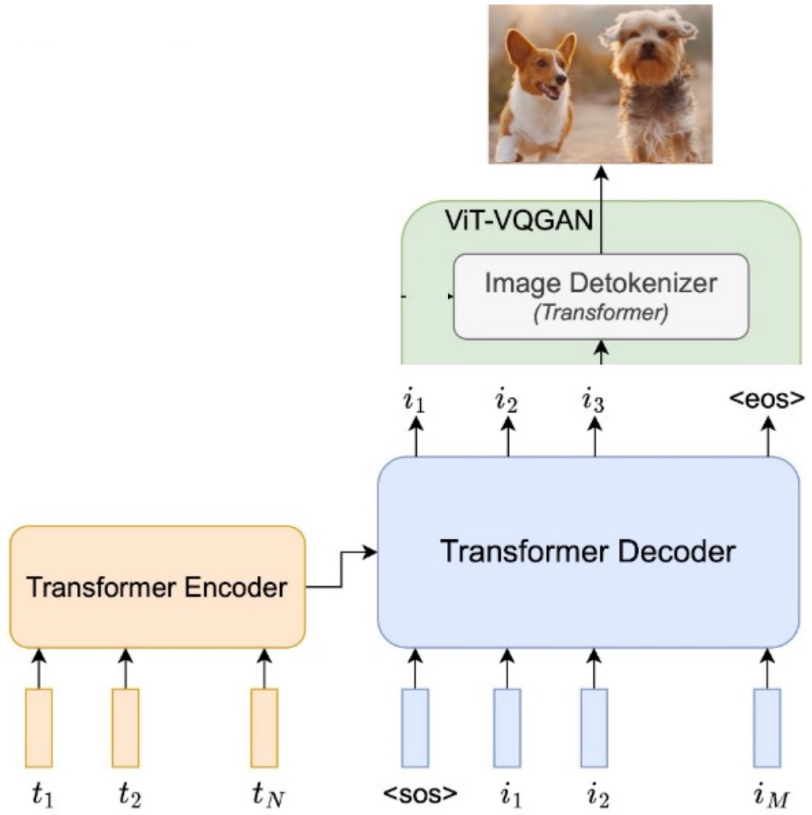
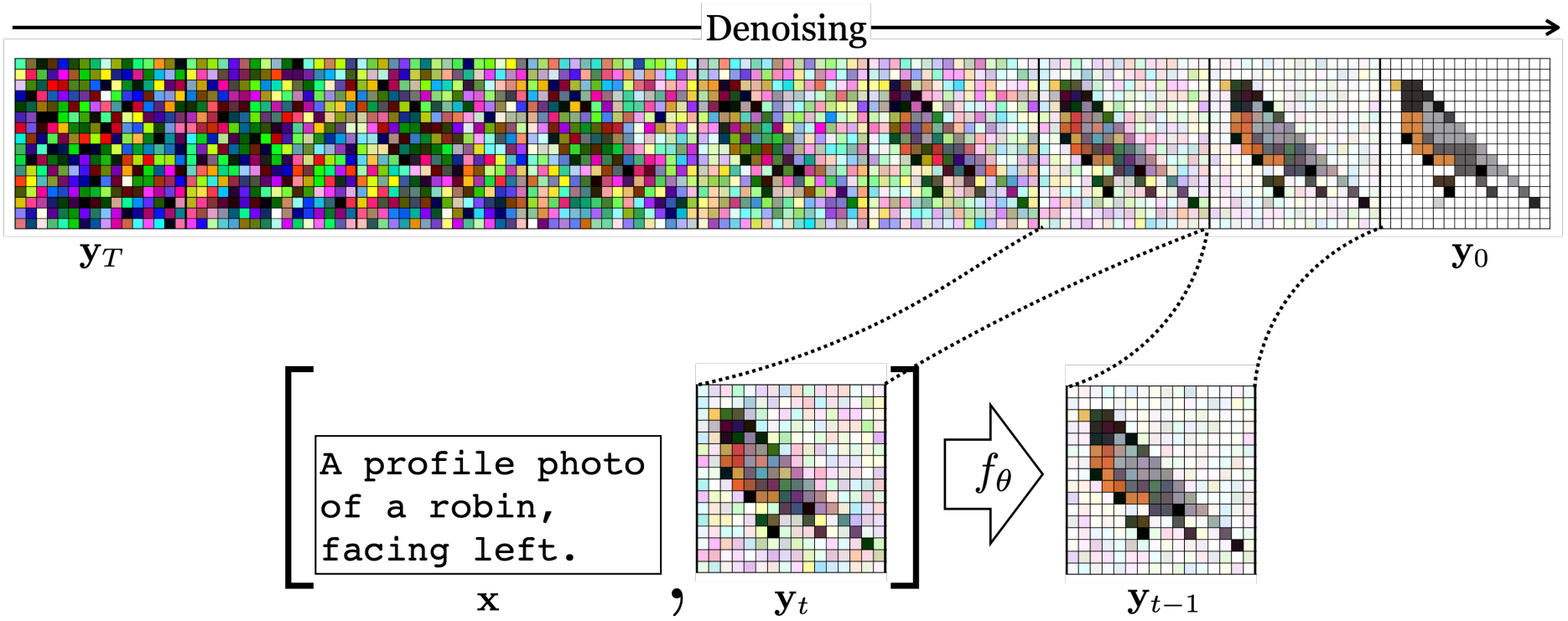


Figure adapted from https://arxiv.org/pdf/2206.10789

Text-conditional Diffusion Models

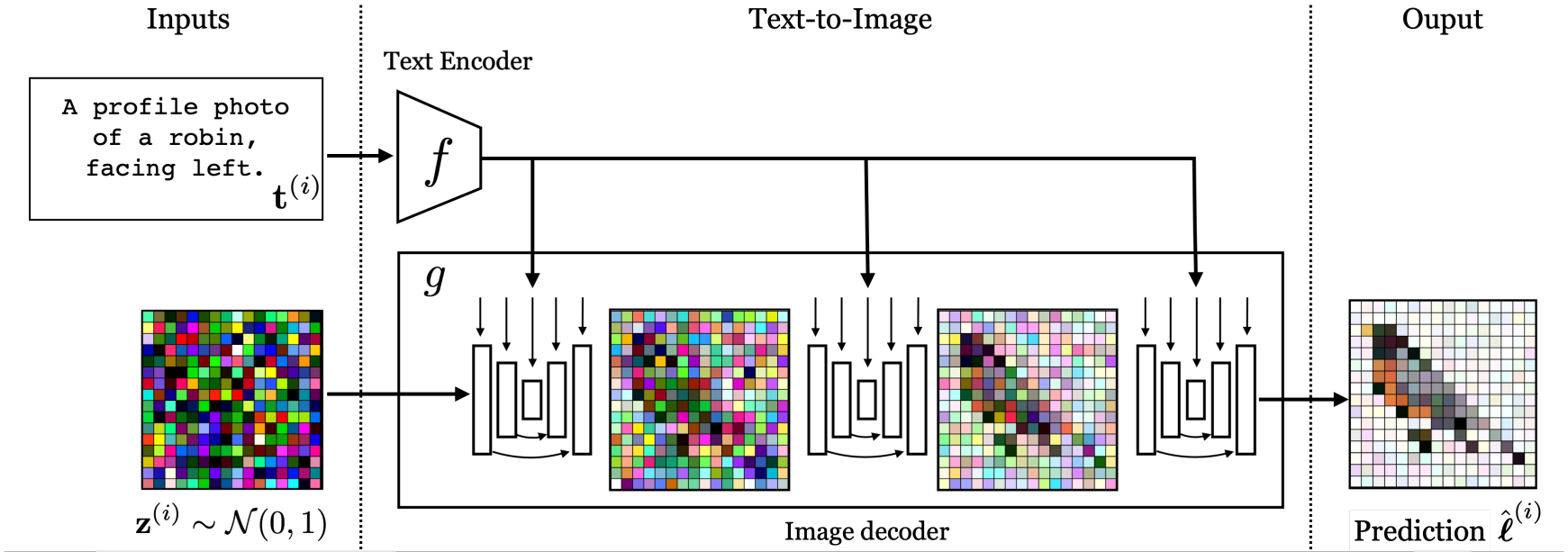
Text-conditional: Diffusion - Learning



For example: DALL-E 2 [Ramesh et al. 2022], Stable Diffusion [Rombach*, Blattman* et al. 2022]

Text-conditional: Diffusion - Learning

Text-to-image architecture (diffusion)



Text-conditional: Diffusion

What if you have 1,000+ GPUs/TPUs

DALL-E (v1)

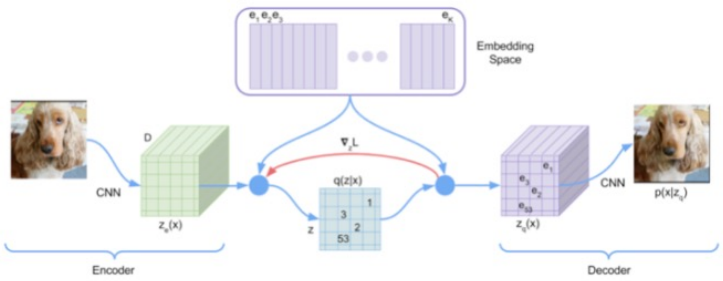
Zero-Shot Text-to-Image Generation

OpenAI Feb 2021

Aditya Ramesh [1] Mikhail Pavlov [1] Gabriel Goh [1] Scott Gray [1]
Chelsea Voss [1] Alec Radford [1] Mark Chen [1] Ilya Sutskever [1]

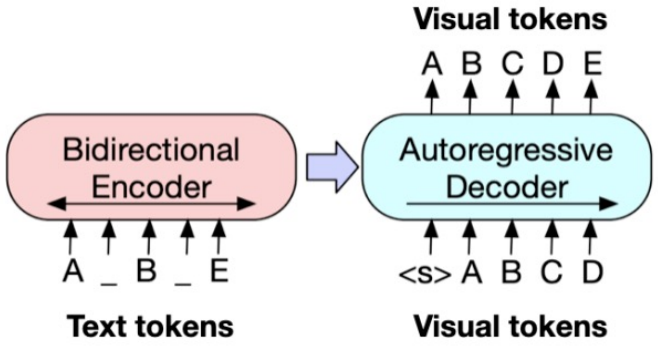
Step 1:

Learn Discrete Dictionary of Visual Tokens



Step 2:

Build a scene as a composition of discrete visual tokens



VQVAE — Oord, Vinyals, Kavukcuoglu, 2017
VQGAN — Esser, Rombach, Ommer, 2021
dVAE - DALL-E — Ramesh et al 2021

BART, GPT-3, etc

DALL-E (v1): Example

an armchair in the shape of an avocado. . . .

Text-conditional: DALL·E 2, Imagen

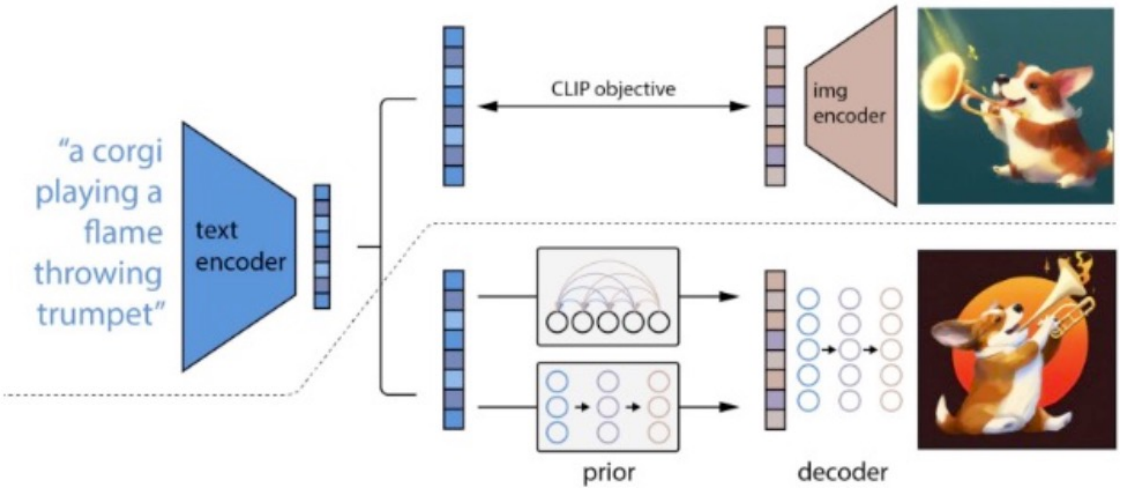
- Pixel-based Diffusion (No encoder-decoder)
- Pre-trained text encoder (CLIP, t5)
- Diffusion model + classifier-free guidance
- Cascaded models: 64->128->512

https://cdn.openai.com/papers/dall-e-2.pdf
https://arxiv.org/abs/2205.11487

DALL.E 2 | OpenAI

Conditioning on CLIP embeddings

- Helps capture multimodal representations
- The bi-partite latent enables several text-controlled image manipulation tasks



Hierarchical Text-Conditional Image Generation with CLIP Latents. Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, Mark Chen - https://arxiv.org/abs/2204.06125

DALL.E 2 | OpenAI

- 1k x 1k text-conditioned image generation
- Uses a **prior** to produce CLIP embeddings conditioned on the text-caption
- Uses a **decoder** to produce images conditioned on the CLIP embeddings

Hierarchical Text-Conditional Image Generation with CLIP Latents. Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, Mark Chen - https://arxiv.org/abs/2204.06125

Text-conditional: Imagen

- Imagen uses a **text to-image diffusion** model coupled with a **super-resolution** diffusion model
- All the models operate in pixel space
- While effective, the compute requirements are very high

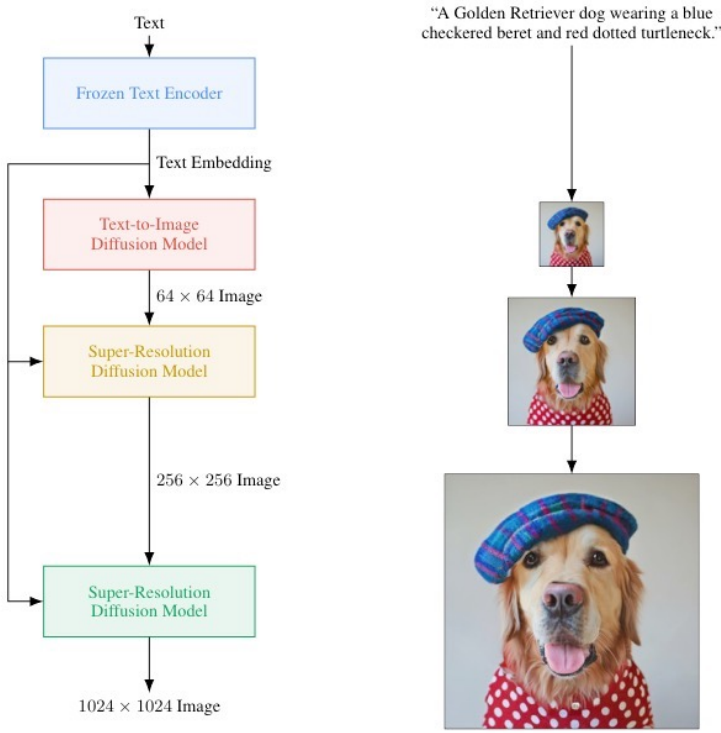
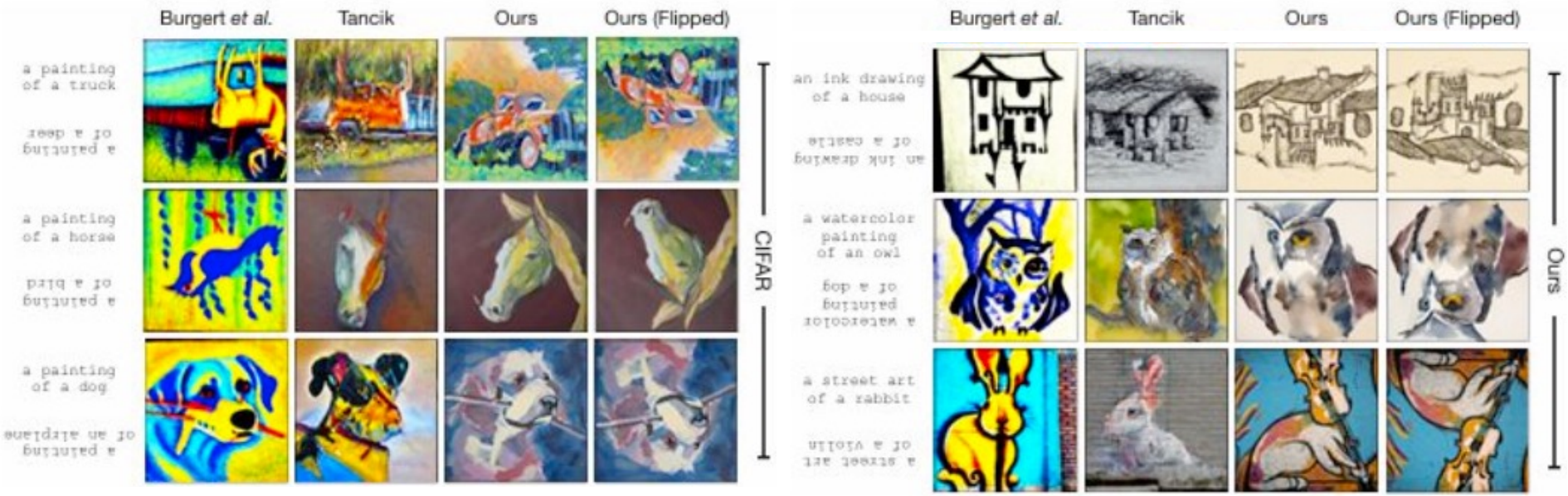


Figure A.4: Visualization of Imagen. Imagen uses a frozen text encoder to encode the input text into text embeddings. A conditional diffusion model maps the text embedding into a 64×64 image. Imagen further utilizes text-conditional super-resolution diffusion models to upsample the image, first $64 \times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

https://www.cs.cmu.edu/~mgormley/courses/10423/

Text-conditional: Diffusion - Applications



Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models, CVPR 2024

CFG: Classifier-free Guidance

- Diffusion models (unlike GANs) are great at generating diverse samples

- But when **diffusion** is conditioned on some input (text, label, etc.) that diversity may cause it to stray away from the prompt

- **Classifier-free guidance** helps diffusion to adhere to the prompt, yielding higher quality images

https://diffusion.csail.mit.edu/2026/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

CFG – Vanilla Guided Sampling

Algorithm 7 Guided Sampling Procedure

Require: A trained guided vector field $u_t^\theta(x|y)$.

1: Select a prompt $y \in \mathcal{Y}$, such as "a cat baking a cake"
2: Initialize $X_0 \sim p_{\text{init}}$.
3: Simulate $\mathrm{d}X_t = u_t^\theta(X_t|y)\mathrm{d}t$ from $t = 0$ to $t = 1$.

Prompt: "Corgi dog"

These images do not fit well to the prompt and they have errors!

https://diffusion.csail.mit.edu/2026/

Image source: Classifier-free diffusion guidance [5].

CFG: Classifier-free Guidance

A swamp ogre with a pearl earring by Johannes Vermeer

A car made out of vegetables.

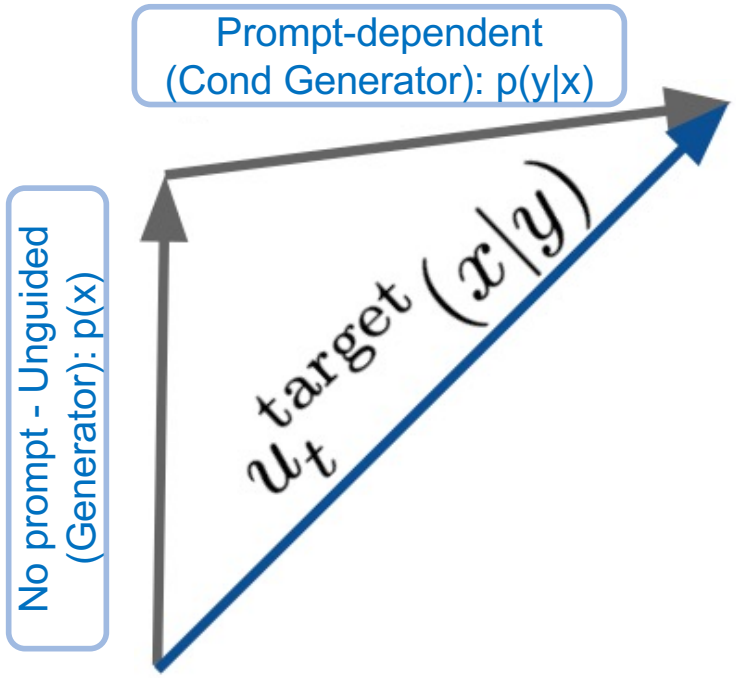
heat death of the universe, line art

Unguided / Unconditional: “Generate an image.”

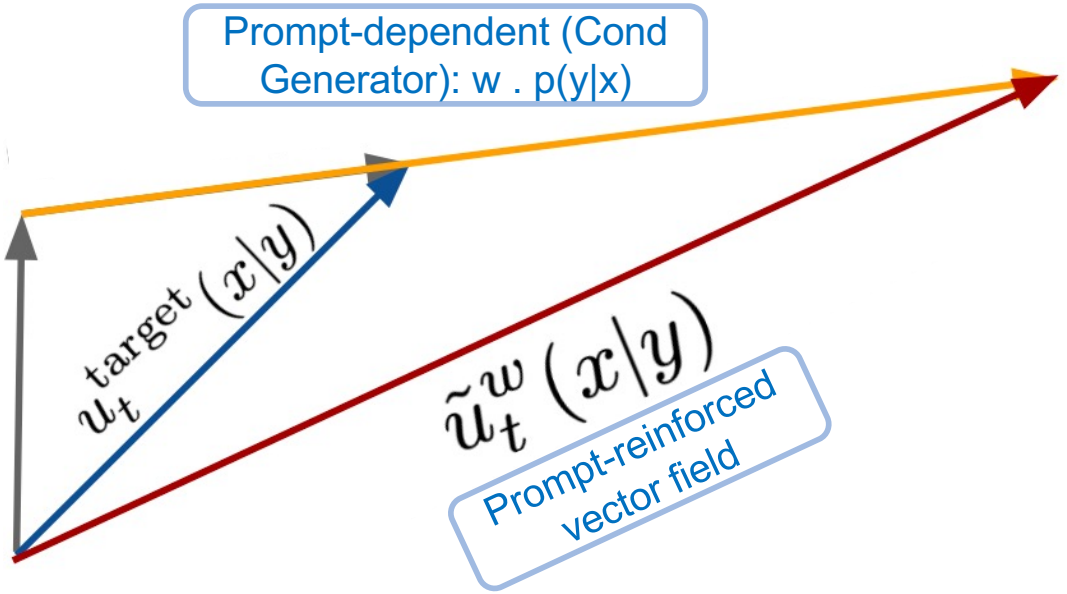
Guided / Conditional: “Generate an image of a cat baking a cake.”

Image source: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis [1]

CFG: Classifier-free Guidance - Intuition



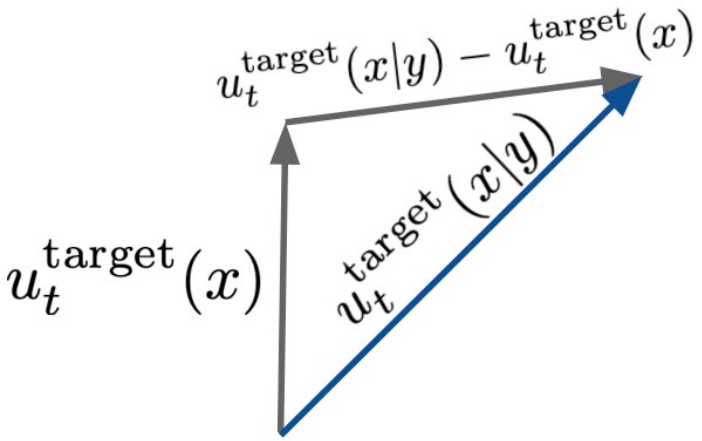
y: Prompt
x: Image

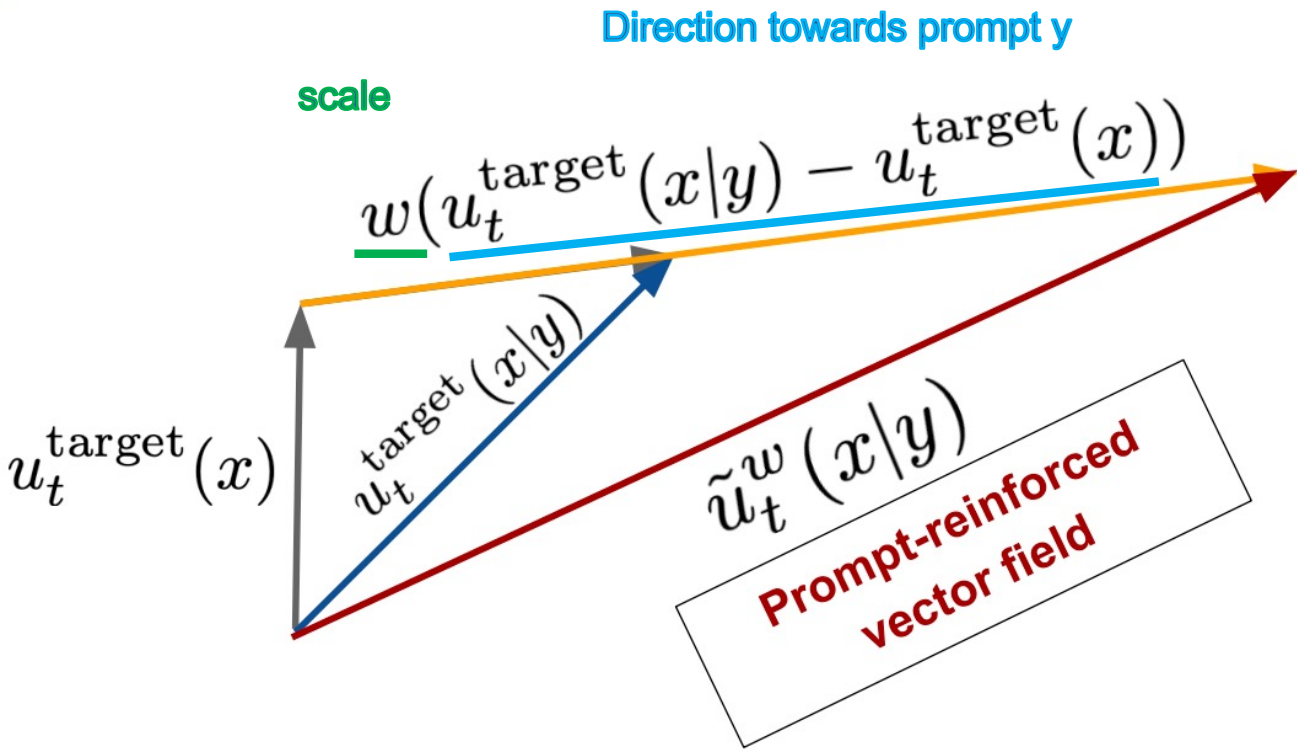


Empty Prompt: "Just draw... something."
Generate an image that represents the "center" of model training data

https://diffusion.csail.mit.edu/2026/

CFG: Classifier-free Guidance - Intuition





https://diffusion.csail.mit.edu/2026/

CFG: Classifier-free Guidance - Training

Algorithm 5 Classifier-free guidance training

Require: Paired dataset $(z, y) \sim p_{\text{data}}$, neural network u_t^θ

1: **for** each mini-batch of data **do**
2: Sample a data example (z, y) from the dataset.
3: Sample a random time $t \sim \text{Unif}_{[0,1]}$.
4: Sample noise $\epsilon \sim \mathcal{N}(0, I_d)$
5: Set $x = \alpha_t z + \beta_t \epsilon$
6: With probability p drop label: $y \leftarrow \varnothing$
7: Compute loss

$$\mathcal{L}(\theta) = \|u_t^\theta(x|y) - u_t^{\text{target}}(x|z)\|^2$$

8: Update the model parameters θ via gradient descent on $\mathcal{L}(\theta)$.
9: **end for**

Drop label with a certain probability!

y: prompt/caption
z: image

x: noisy image

Noise Predictor
Learnt Model

https://diffusion.csail.mit.edu/2026/

CFG: Classifier-free Guidance - Sampling

$$u_t^{\theta,w}(x) = (1-w)u_t^{\theta}(x|\varnothing) + wu_t^{\theta}(x|y)$$

Algorithm 8 Classifier-Free Guidance Sampling Procedure

Require: A trained guided vector field $u_t^{\theta}(x|y)$.

1: Select a prompt $y \in \mathcal{Y}$, or take $y = \varnothing$ for unguided sampling.
2: Select a **guidance scale** $w > 1$.
3: Initialize $X_0 \sim p_{\text{init}}$.
4: Simulate $\mathrm{d}X_t = \left[(1-w)u_t^{\theta}(X_t|\varnothing) + wu_t^{\theta}(X_t|y)\right] \mathrm{d}t$ from $t = 0$ to $t = 1$.

https://diffusion.csail.mit.edu/2026/

CFG – Experimental Results

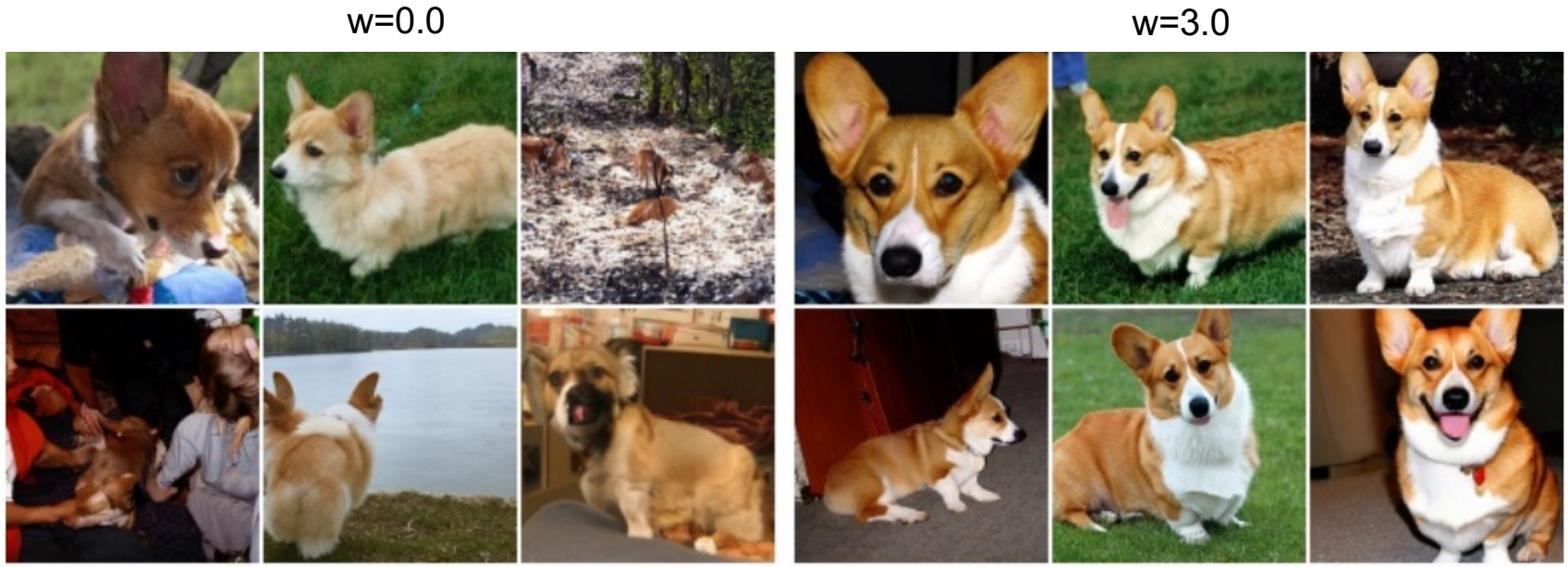
- Increasing guidance scale yields samples that more closely adhere to the class label
- Guidance scale w increases from the left block of samples to the right block of samples

https://diffusion.csail.mit.edu/2026/

Figure from https://arxiv.org/pdf/2207.12598

CFG – Experimental Results

- Increasing guidance scale yields samples that more closely adhere to the class label
- Guidance scale w increases from the left block of samples to the right block of samples

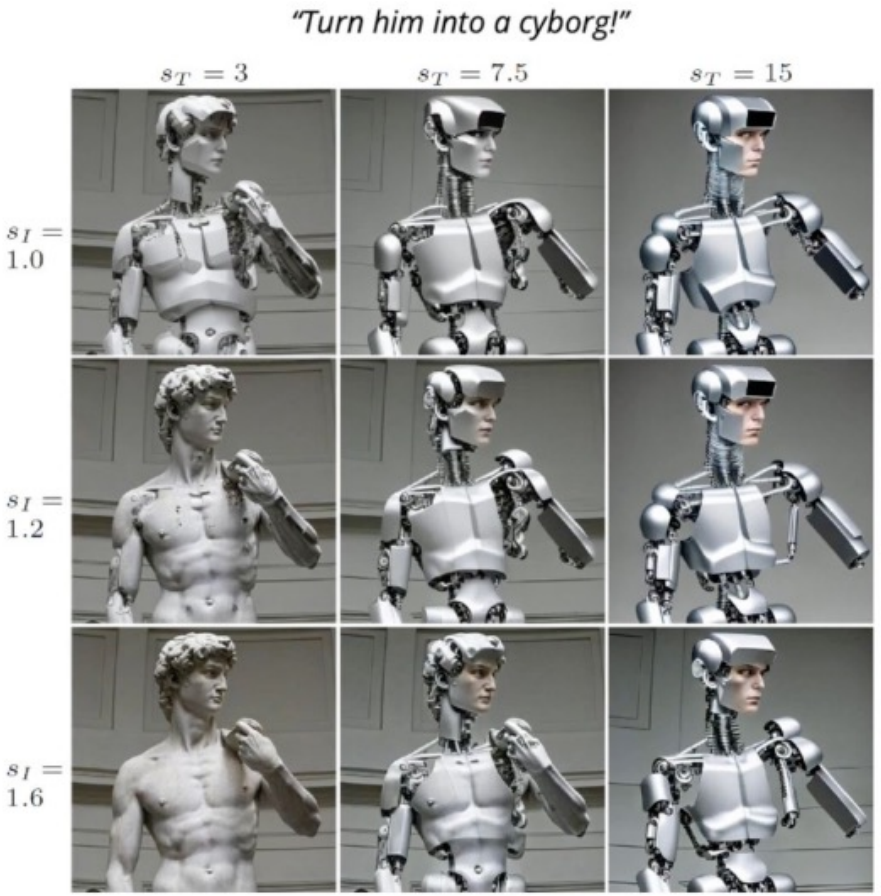


https://diffusion.csail.mit.edu/2026/

Figure from https://arxiv.org/pdf/2207.12598

CFG – Applications

Authors apply CFG with separate scales for image and text conditioning



Slide Credit: Tim Brooks

InstructPix2Pix: Learning to Follow Image Editing Instructions, CVPR 2023

Text-conditional Diffusion Models!

great results for image synthesis

Denoising Diffusion Probabilistic Models

Jonathan Ho, Ajay Jain, et al

https://arxiv.org/abs/2006.11239

Diffusion Models beat GANs on Image Synthesis

Prafulla Dhariwal, Alex Nichol

https://arxiv.org/abs/2105.05233

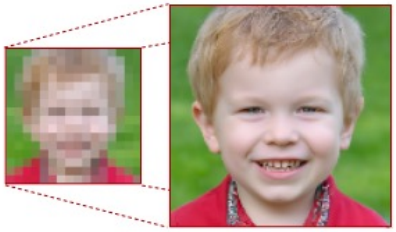


Image Super-Resolution via Iterative Refinement

Chitwan Saharia, et al

https://arxiv.org/abs/2104.07636

… but very expensive :(

Slide credit: Robin Rombach

Text-conditional Latent Diffusion Models

Text-conditional: Latent Diffusion Modeling (LDM)

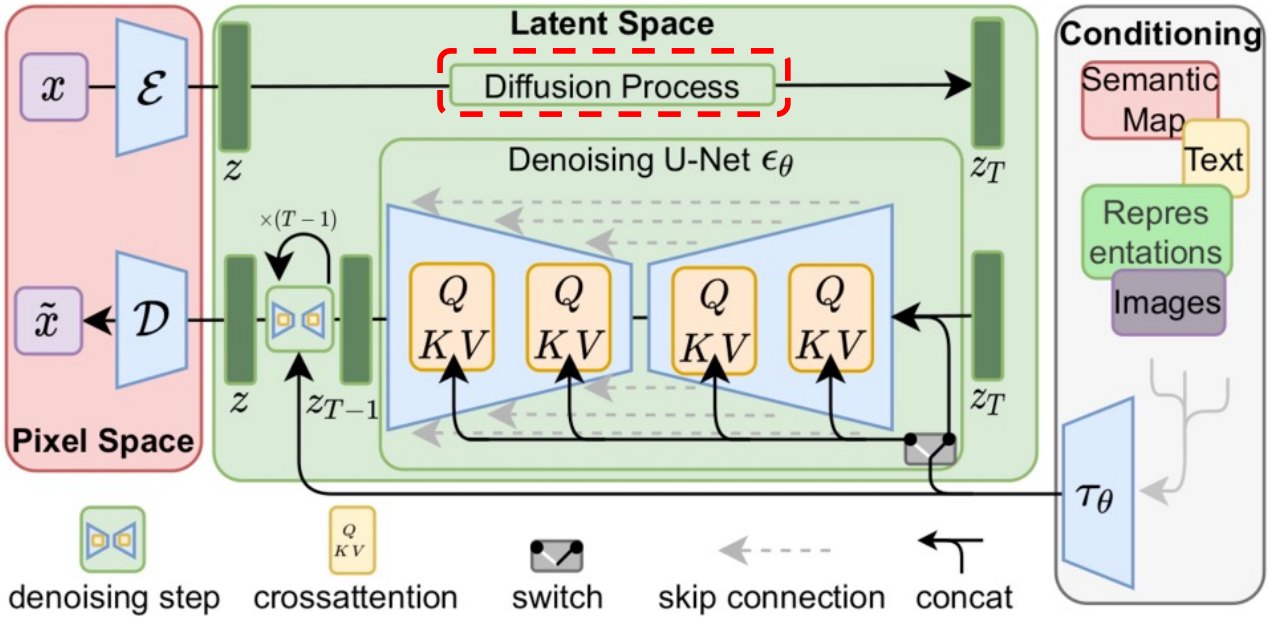
https://www.cs.cmu.edu/~mgormley/courses/10423/

Text-conditional: Latent Diffusion Modeling (LDM)

Autoencoder with KL or VQ regularization.

VQ-reg.: $\mathcal{L}_{\text{total}} = \mathcal{L}_{rec} + \mathcal{L}_{VQ} + \lambda \mathcal{L}_{GAN}$ where $\lambda = \frac{\nabla_{G_L}[\mathcal{L}_{rec}]}{\nabla_{G_L}[\mathcal{L}_{GAN}]+\delta}$

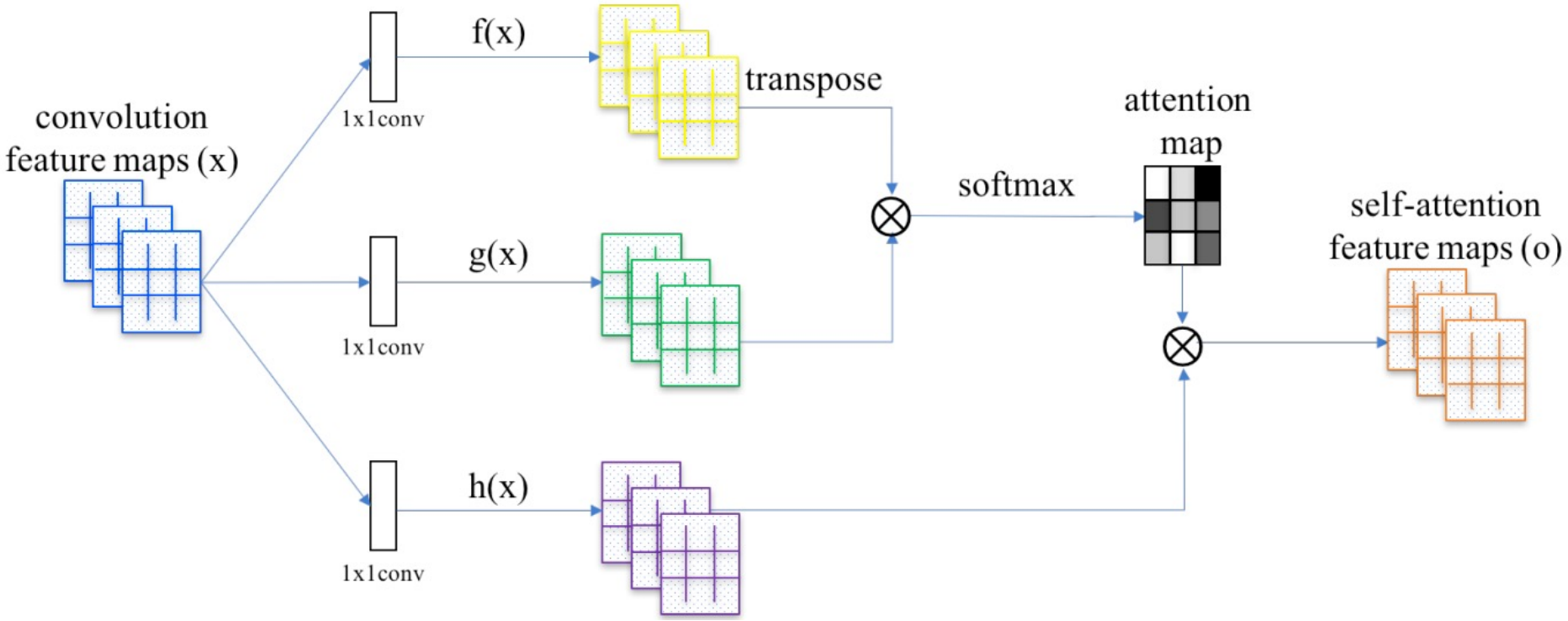
KL-reg.: $\mathcal{L}_{\text{total}} = \mathcal{L}_{rec} + \beta \mathcal{L}_{KL} + \lambda \mathcal{L}_{GAN}$



Slide credit: Robin Rombach

Text-conditional: Latent Diffusion Modeling (LDM)

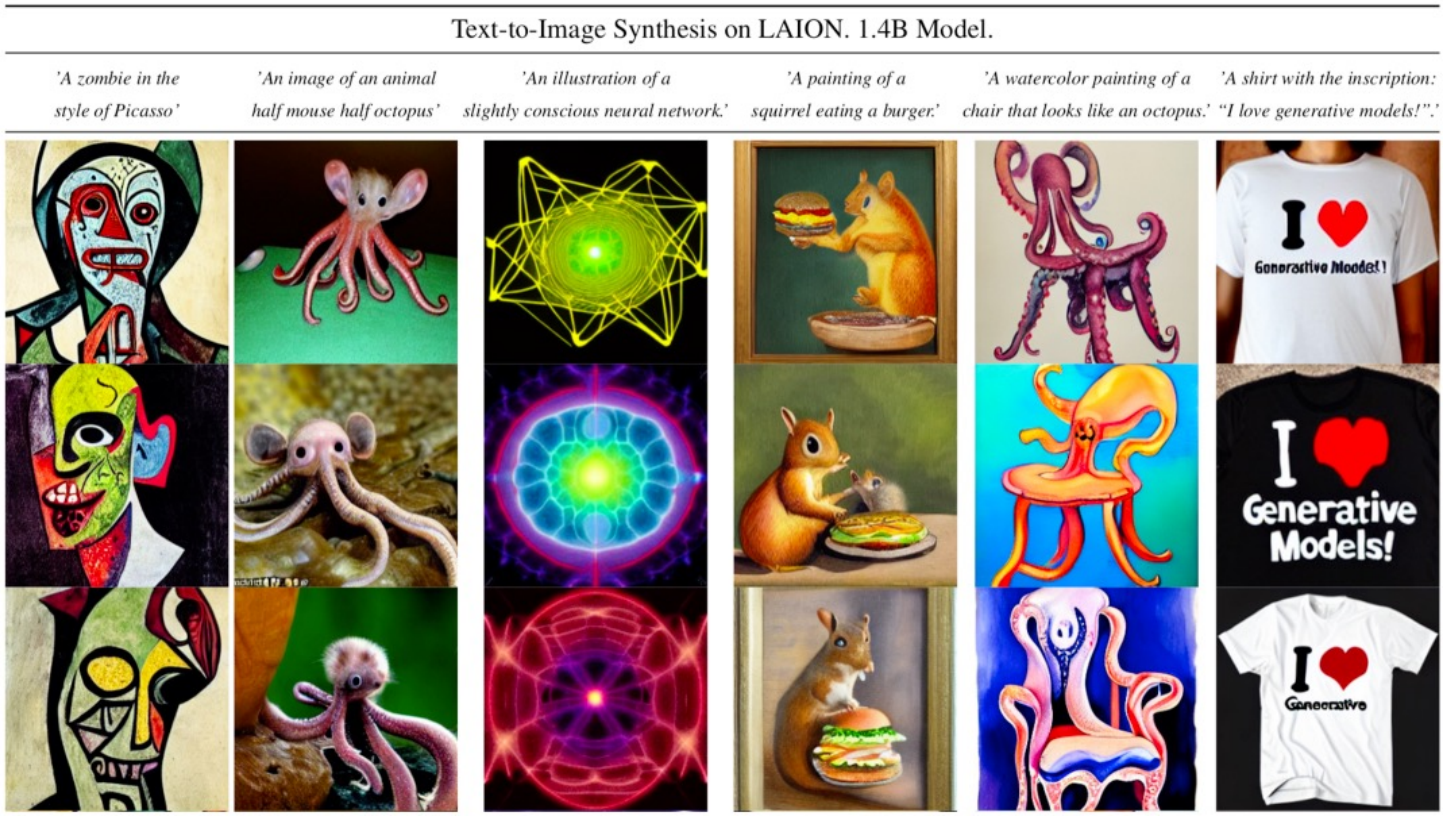
(Spatial) Self-attention Layer



Han Zhang et al, “Self-Attention Generative Adversarial Networks”, ICML 2018

Text-conditional: LDM results

- 32x32 cont. space
- 600M Transformer
- 800M UNet
- 400M Image/Text Pairs

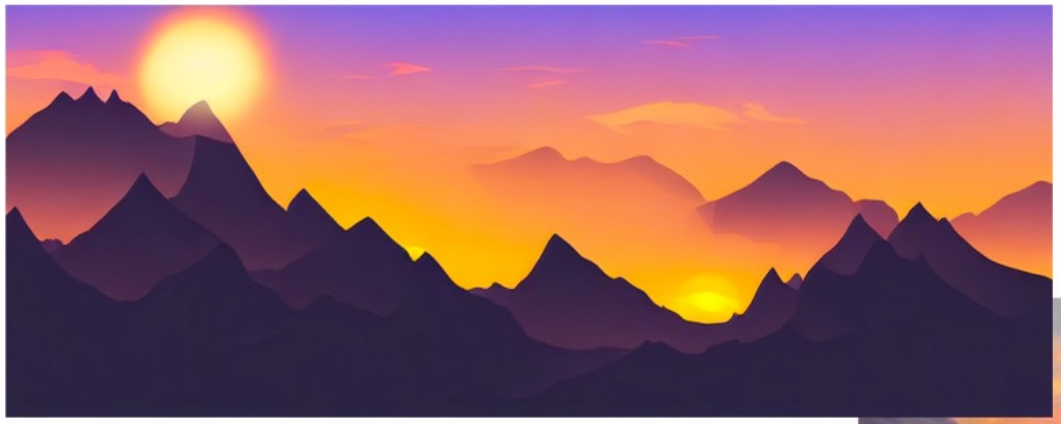


Slide credit: Robin Rombach

Text-conditional: LDM results

convolutional sampling (train on 256^2, generate on $>256^2$)

"A sunset over a mountain range, vector image"

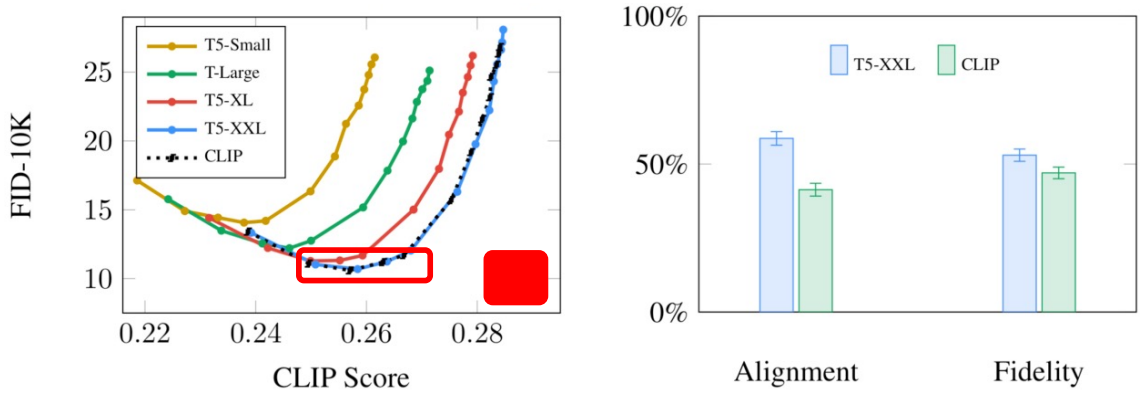


"A sunset over a mountain range, oil on canvas"

Slide credit: Robin Rombach

Text-conditional: Stable Diffusion

- Goal: achieve a small model that people can actually run locally on "small" GPUs (~10GB VRAM)
- Progressive training: pretrain on 256x256, then continue on 512x512
- Fix text encoder (as in Imagen)
- → choose CLIP (ViT-L/14) since performance/size tradeoff seems significant



(a) Pareto curves comparing various text encoders. (b) Comparing T5-XXL and CLIP on DrawBench.

Figure from Imagen, https://arxiv.org/abs/2205.11487

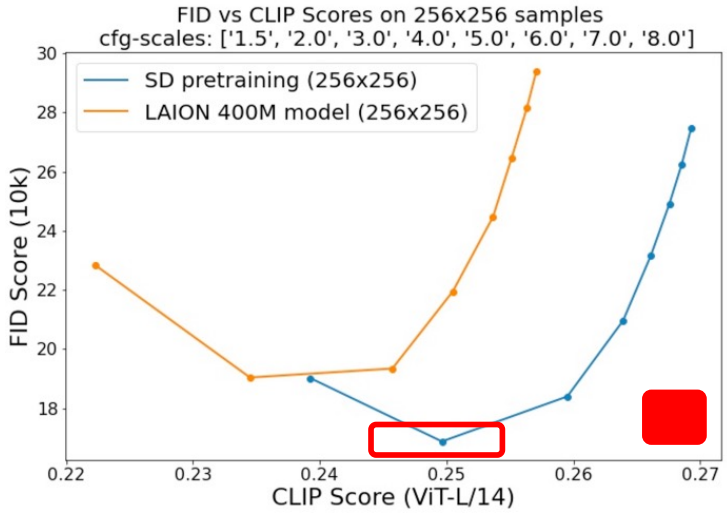
Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional: Stable Diffusion

Stage 1: Pretraining @256x256

- 237k steps at resolution 256x256 on LAION 2B(en)
- batch-size = 2048
- ~ 64 A100 GPUs

* FID score is a metric used to evaluate the quality of images generated by generative models, with lower scores indicating a better match between generated and real images.



10k random COCO val captions / 50 decoding steps

FID Score: https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance

Figure from Imagen, https://arxiv.org/abs/2205.11487

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional: Stable Diffusion

Stage 2: Training @512x512. batch-size=2048, #gpus=256

[Part 1 (v1.1)]

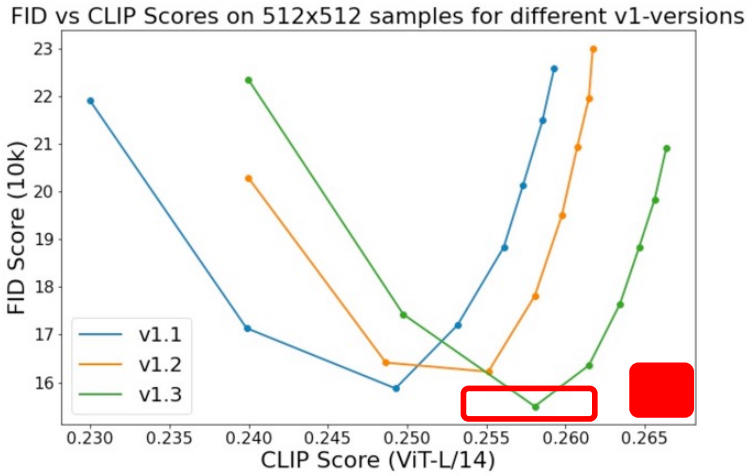
- 194k steps at resolution 512x512 on laion-high-resolution (170M examples from LAION-5B with resolution >= 1024x1024).

[Part 2 (v1.2)]

- 515k steps at resolution 512x512 on "laion-improved-aesthetics" (a subset of laion2B-en, filtered to images with an original size >= 512x512, estimated aesthetics score > 5.0, and an estimated watermark probability < 0.5

[Part 3/4 (v1.3/v1.4)]

- 195k/225k steps at resolution 512x512 on "laion-improved aesthetics" and 10% dropping of the text-conditioning



10k random COCO val captions / 50 decoding steps

FID Score: https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance

Figure from Imagen, https://arxiv.org/abs/2205.11487

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional: Stable Diffusion

Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

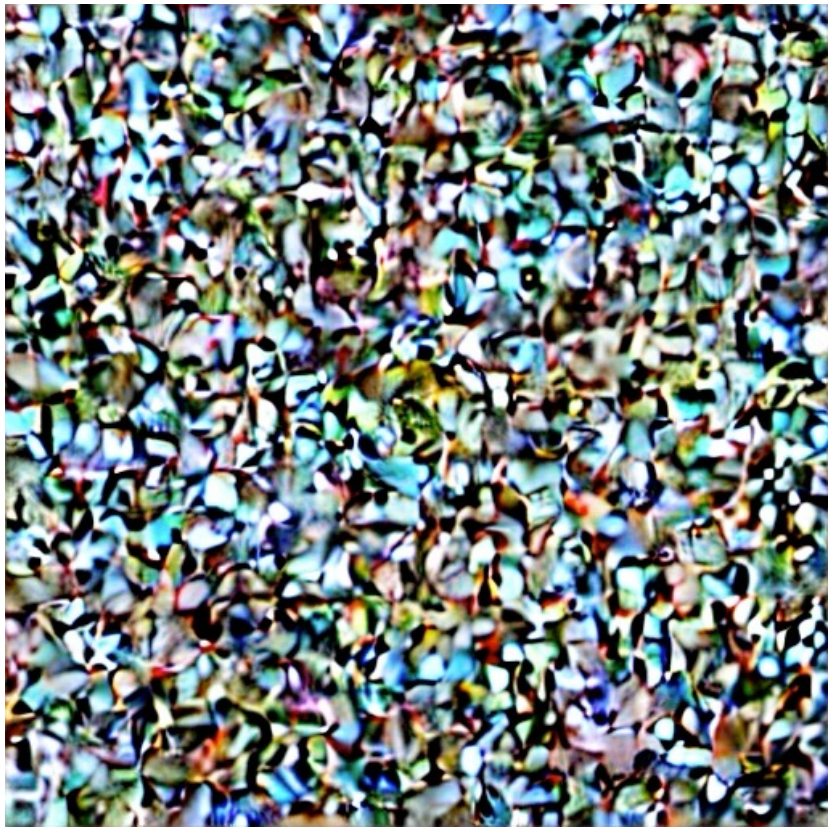
Text-conditional: Stable Diffusion - Demo

Stable Diffusion Demo: link

Slide credit: Robin Rombach

Text-conditional: Stable Diffusion - Demo

prompt = "a lovely cat running in the desert in Van Gogh style, trending art."



link

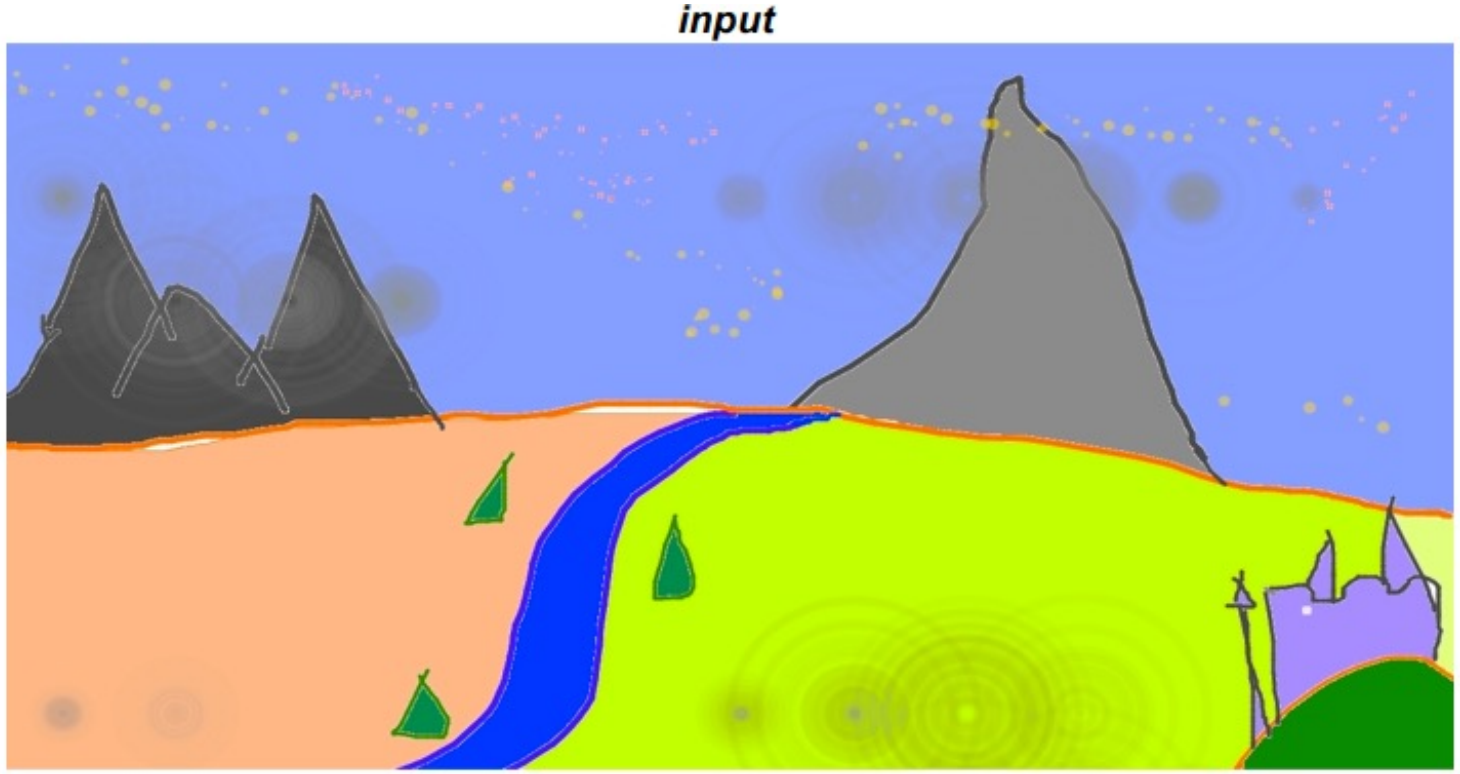
Text-conditional: Stable Diffusion

Diffusion Explainer: link

https://arxiv.org/pdf/2305.03509

Video: https://www.youtube.com/watch?v=Zg4gxdIWDds

Text-conditional: Text-guided Image-to-image



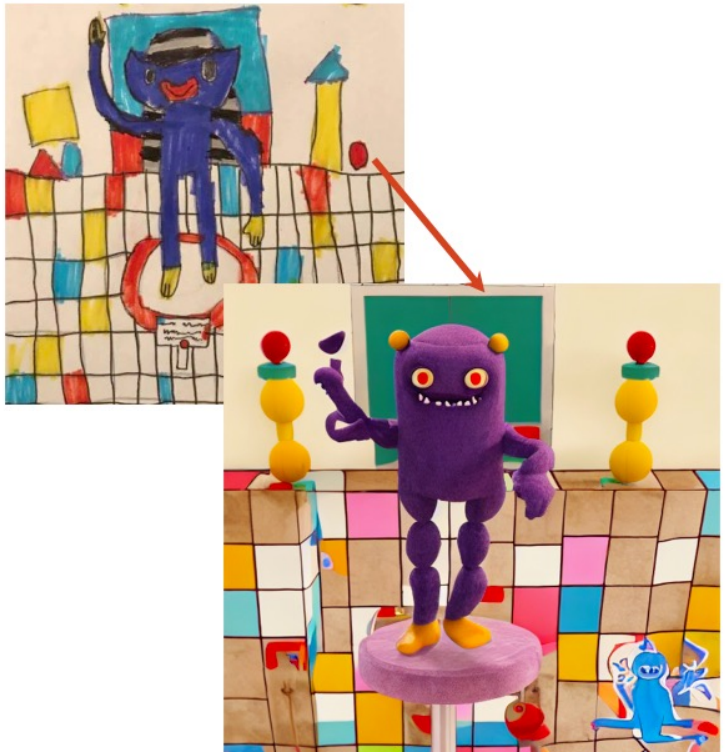
Slide credit: Robin Rombach

Text-conditional: Text-guided Image-to-image

Slide credit: Robin Rombach

Text-conditional: Text-guided Image-to-image

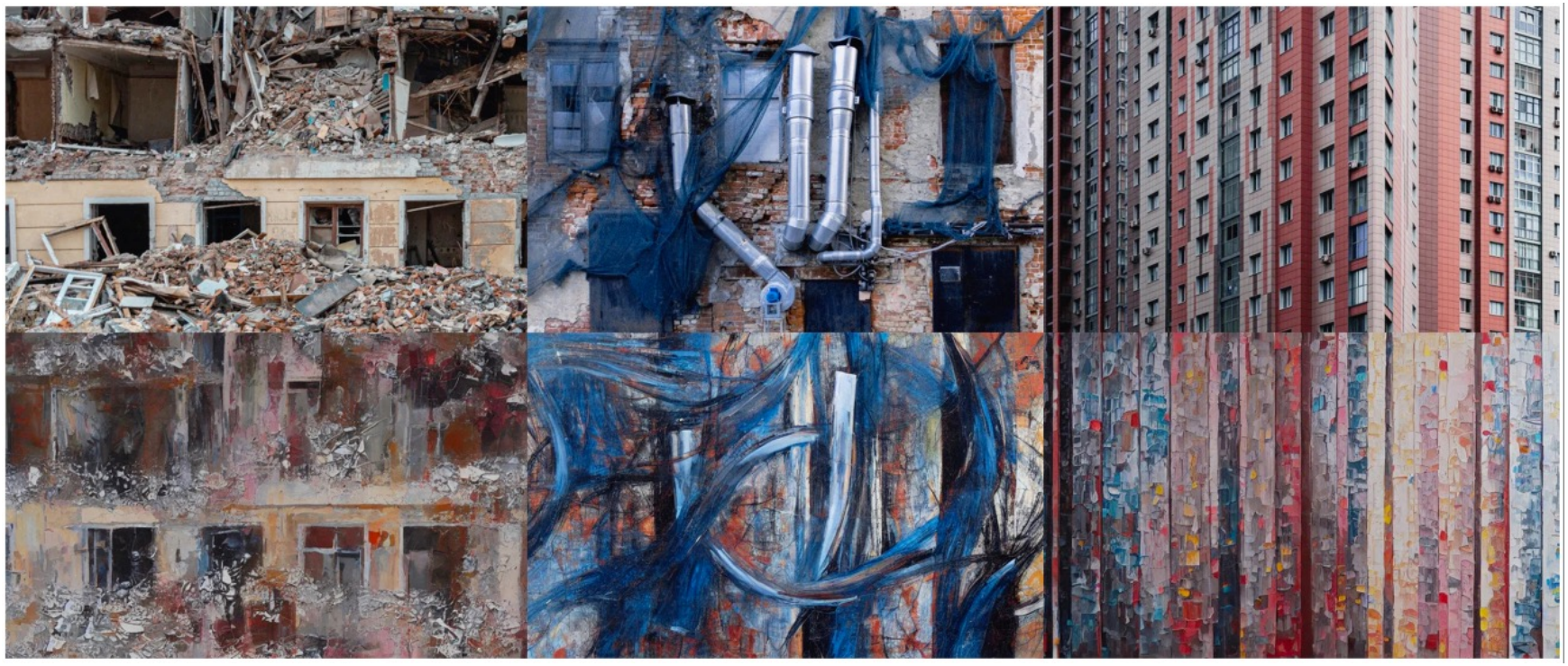
“Upgrade” your child’s artwork
original post: https://www.reddit.com/r/StableDiffusion/comments/wyq04v/using_img2img_to_upgrade_my_sons_artwork/



Slide credit: Robin Rombach

Text-conditional: Text-guided Image-to-image

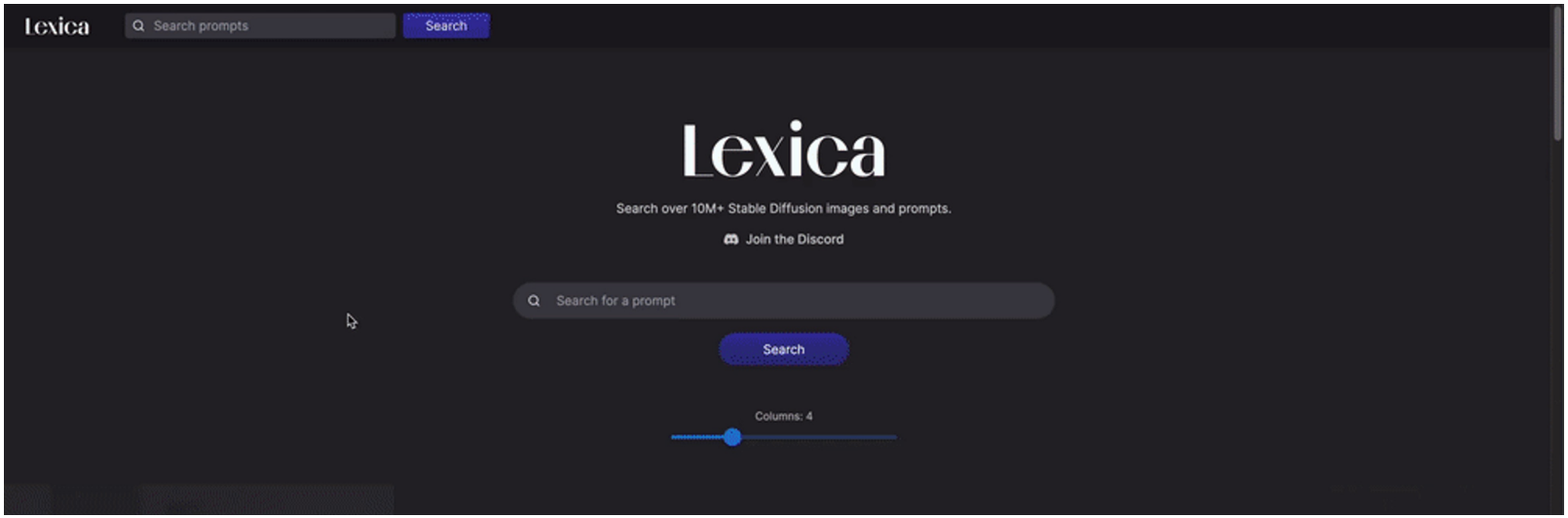
original post by u/Pereulkov:
https://www.reddit.com/r/StableDiffusion/comments/xhhyad/i_made_abstract_art_from_my_photos/



Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional: Prompting

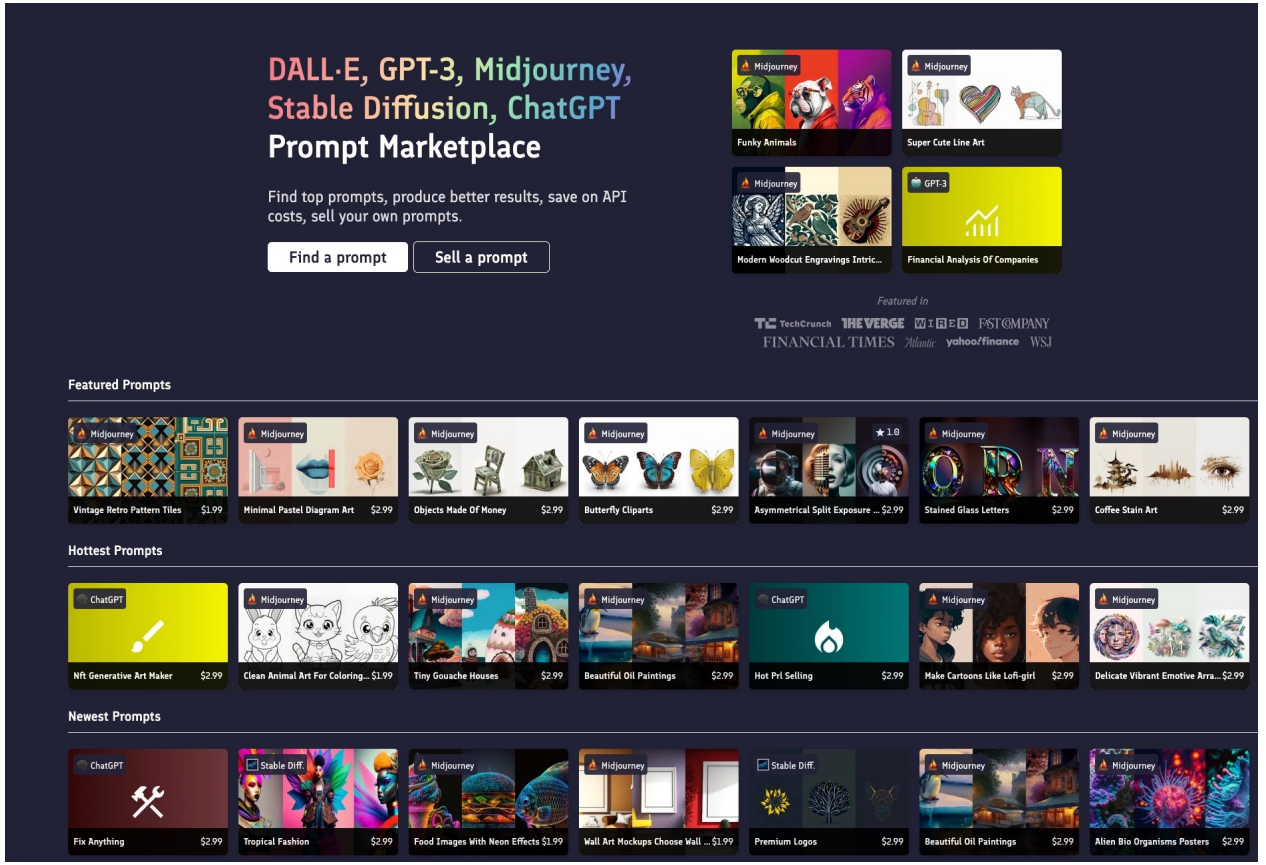
Prompt Search Engine (lexica.art)



Slide credit: Robin Rombach

Text-conditional: Prompting

Prompt Marketplace
(promptbase.com)



Slide credit: Jun-Yan Zhu - Learning-Based Image Synthesis

Text-conditional Diffusion Transformer Models

Text-conditional: Diffusion Transformer (DiT)

2015
- UNet introduced for medical image segmentation

2016-2018
- Numerous UNet variants used to good effect
- pixel-level autoregressive models
- conditional GANs
- Pixel-CNN++ set the standard variant

2020
- UNet used by Ho et al. for DDPM to good effect

2021
- Dhariwal & Nichol include conditional information via Transformer block

2022-2023
- Everyone continues using UNet for diffusion models because it seems to work well

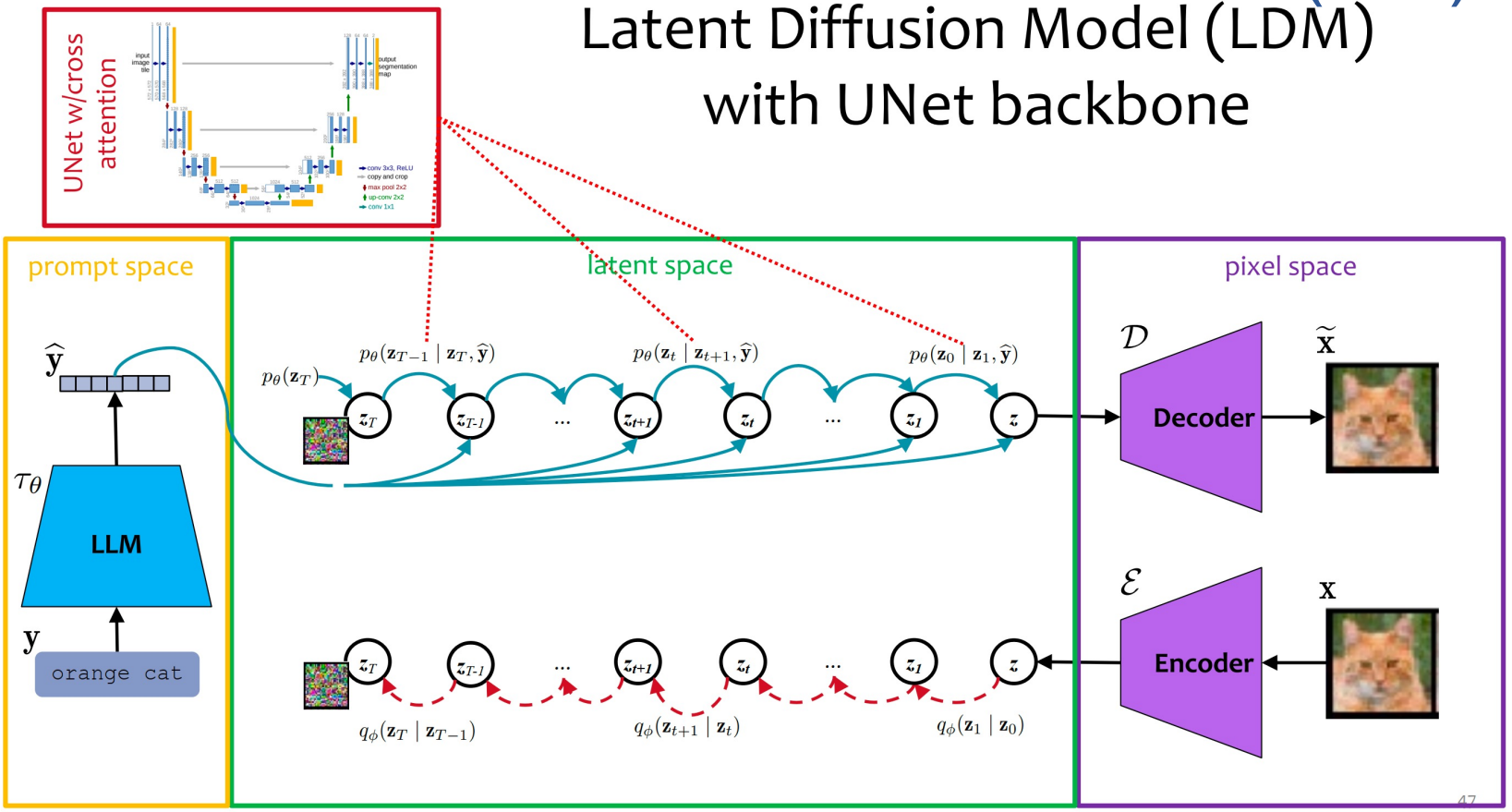
May 2023
- Diffusion Transformer shows we don't actually need UNet

https://www.cs.cmu.edu/~mgormley/courses/10423/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

Text-conditional: Diffusion Transformer (DiT)

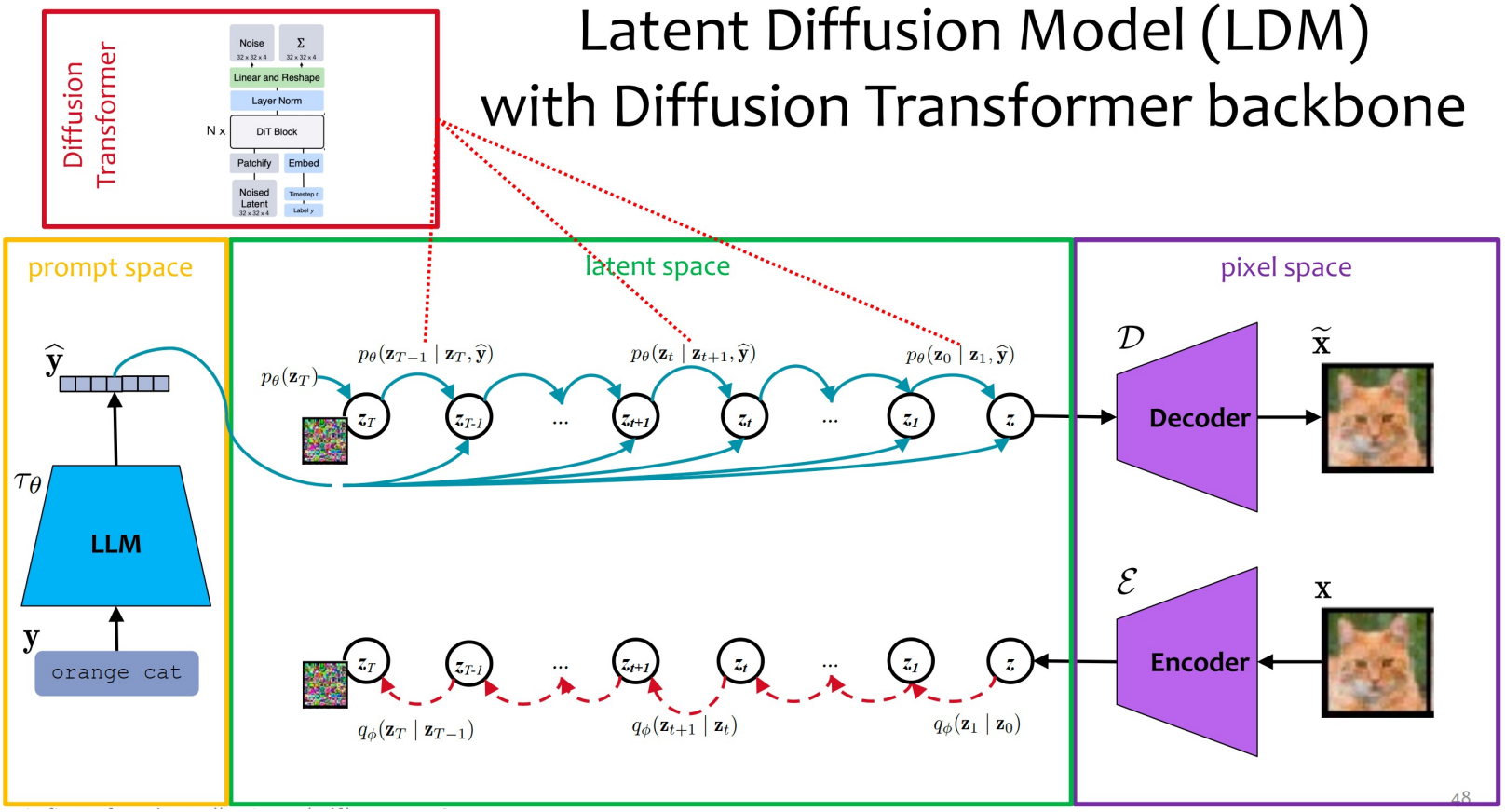
Latent Diffusion Model (LDM) with UNet backbone



https://www.cs.cmu.edu/~mgormley/courses/10423/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

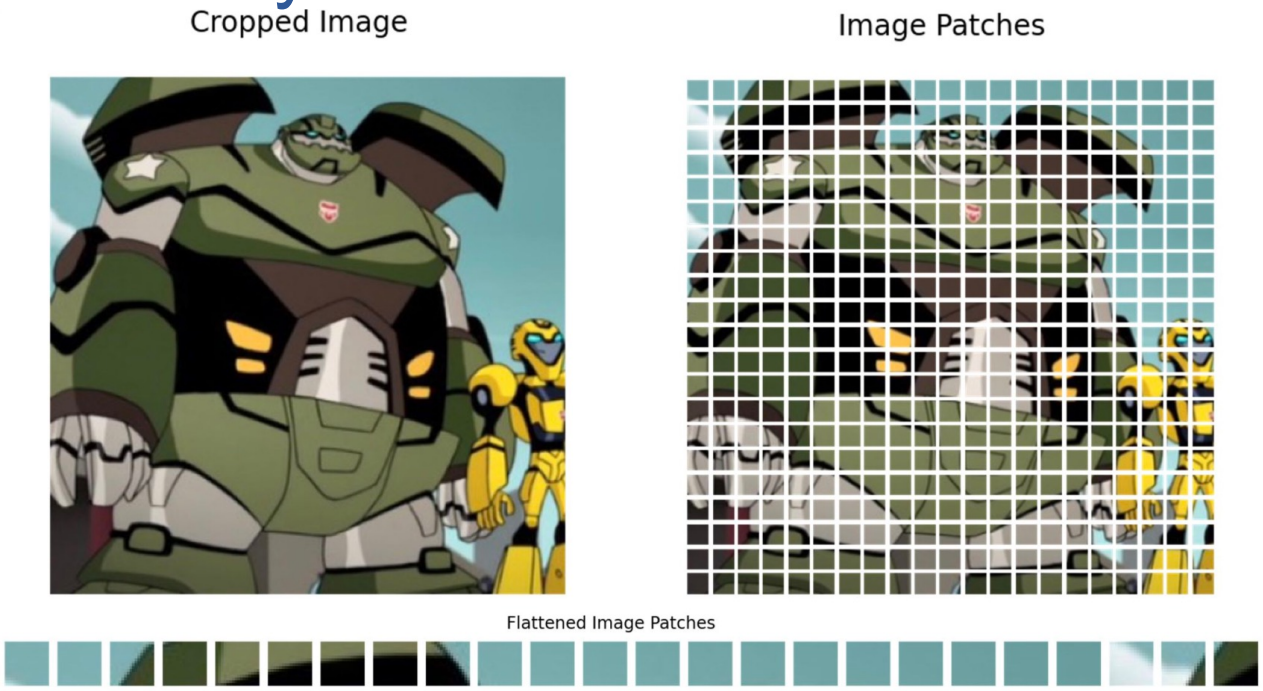
Text-conditional: Diffusion Transformer (DiT)



https://www.cs.cmu.edu/~mgormley/courses/10423/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

Text-conditional: Diffusion Transformer (DiT) - Patchify



$$x \in \mathbb{R}^{3 \times H \times W}$$

$$\tilde{x} = \text{Patchify}(x)$$

$$\tilde{x} \in \mathbb{R}^{L \times k}$$

L is the sequence length

https://diffusion.csail.mit.edu/2026/

https://gowrishankar.info/blog/transformers-everywhere-patch-encoding-technique-for-vision-transformersvit-explained/

Text-conditional: Diffusion Transformer (DiT) – Encoding Language Prompt

"A dog running on grass in a park at sunshine in an Italian city."

To embed text, most models rely on pre-trained language embeddings:

- Use CLIP embeddings (Contrastive Language-Image Pre-training)
- T5 embeddings, etc. (other pre-trained models)
- Can also use LLM embeddings

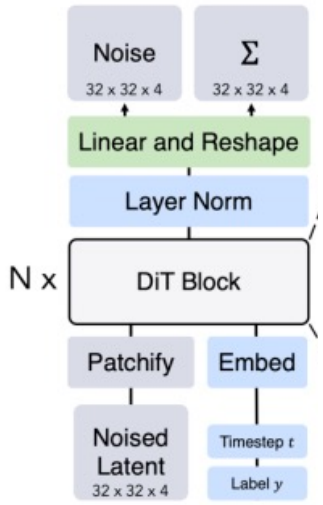
Prompt embedding: The result of these embeddings is that the prompt a sequence of vectors of length S:

$$\text{PromptEmbed}(y_{\text{raw}}) \in \mathbb{R}^{S \times k}$$

https://diffusion.csail.mit.edu/2026/

Text-conditional: Diffusion Transformer (DiT)

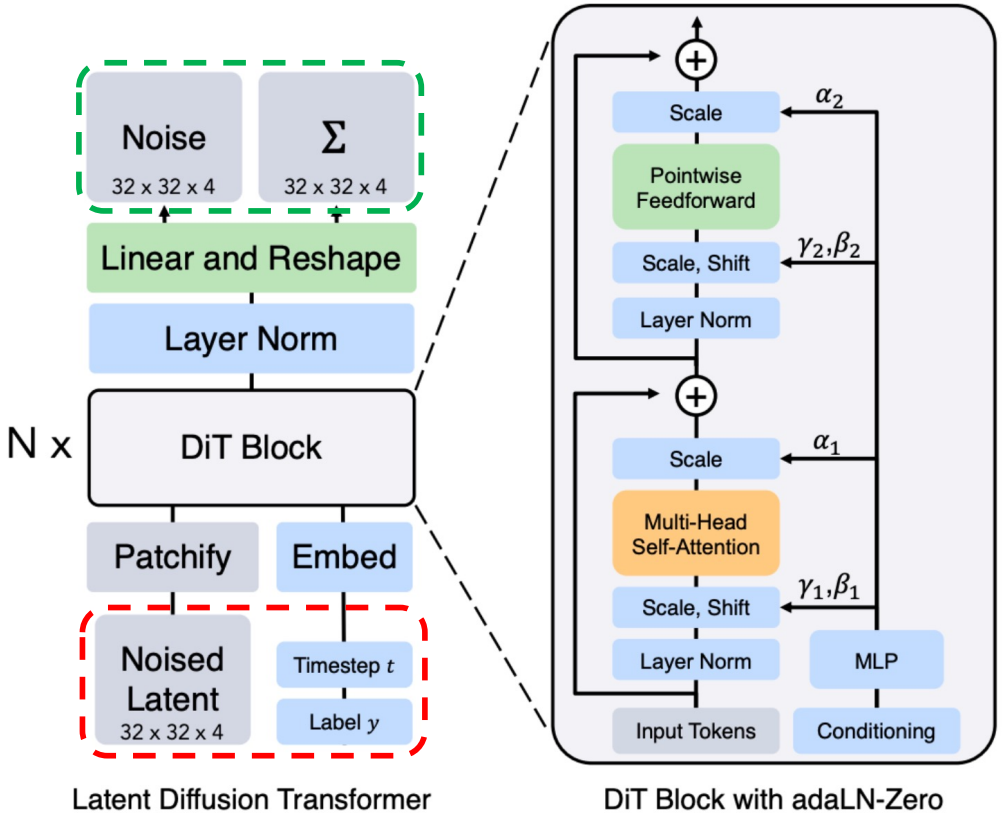
Diffusion uses standard Transformer blocks!
Main question: How to inject conditioning (timestep t, text, …)



Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

Text-conditional: Diffusion Transformer (DiT)

- DiT backbone is essentially a Vision Transformer (ViT) with some tweaks
- Input is a noisy latent, a timestep, and a label (or other conditional information)
- Output is a mean and covariance output, each of fixed size
- After a final layer norm, a linear layer is used to convert from a sequence of T token embeddings to fixed size output



https://www.cs.cmu.edu/~mgormley/courses/10423/

DiT figure from https://arxiv.org/pdf/2212.09748

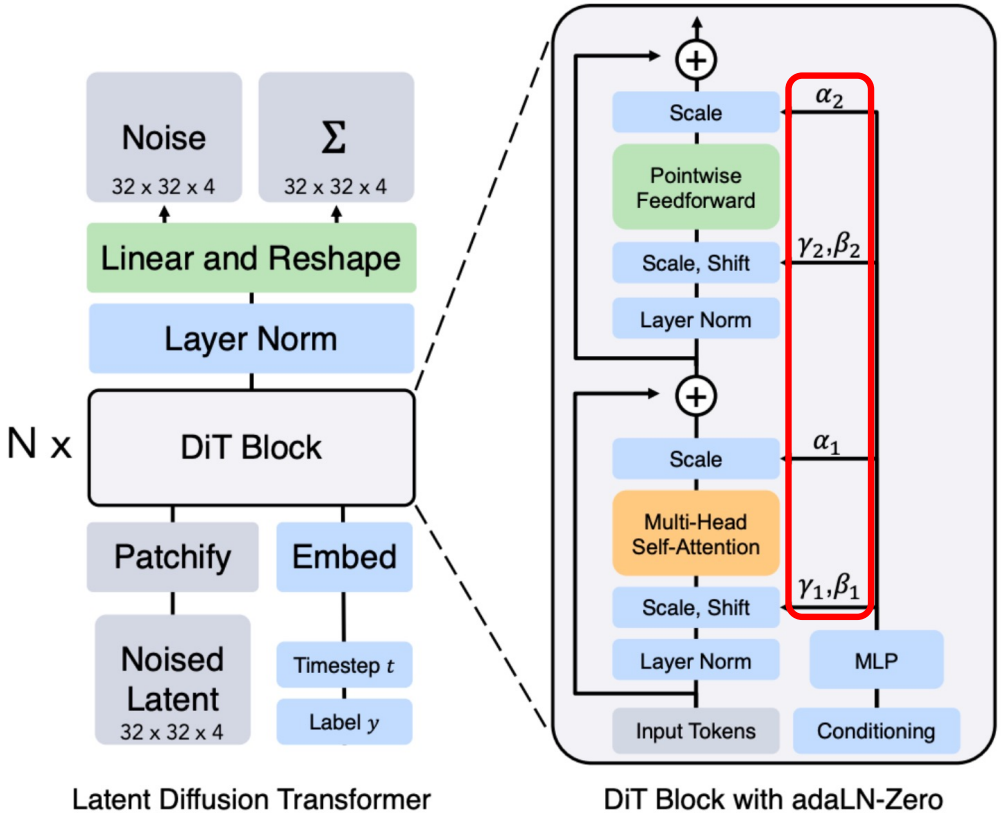
Text-conditional: Diffusion Transformer (DiT) – Adaptive LayerNorm

Original DiT paper tries out various approaches:

- In-context conditioning
- Cross-attention block
- Adaptive layer norm (adaLN) block
- Adaptive layer norm with zero initialization strategy (adaLNZero)

AdaLN-Zero is the best approach empirically

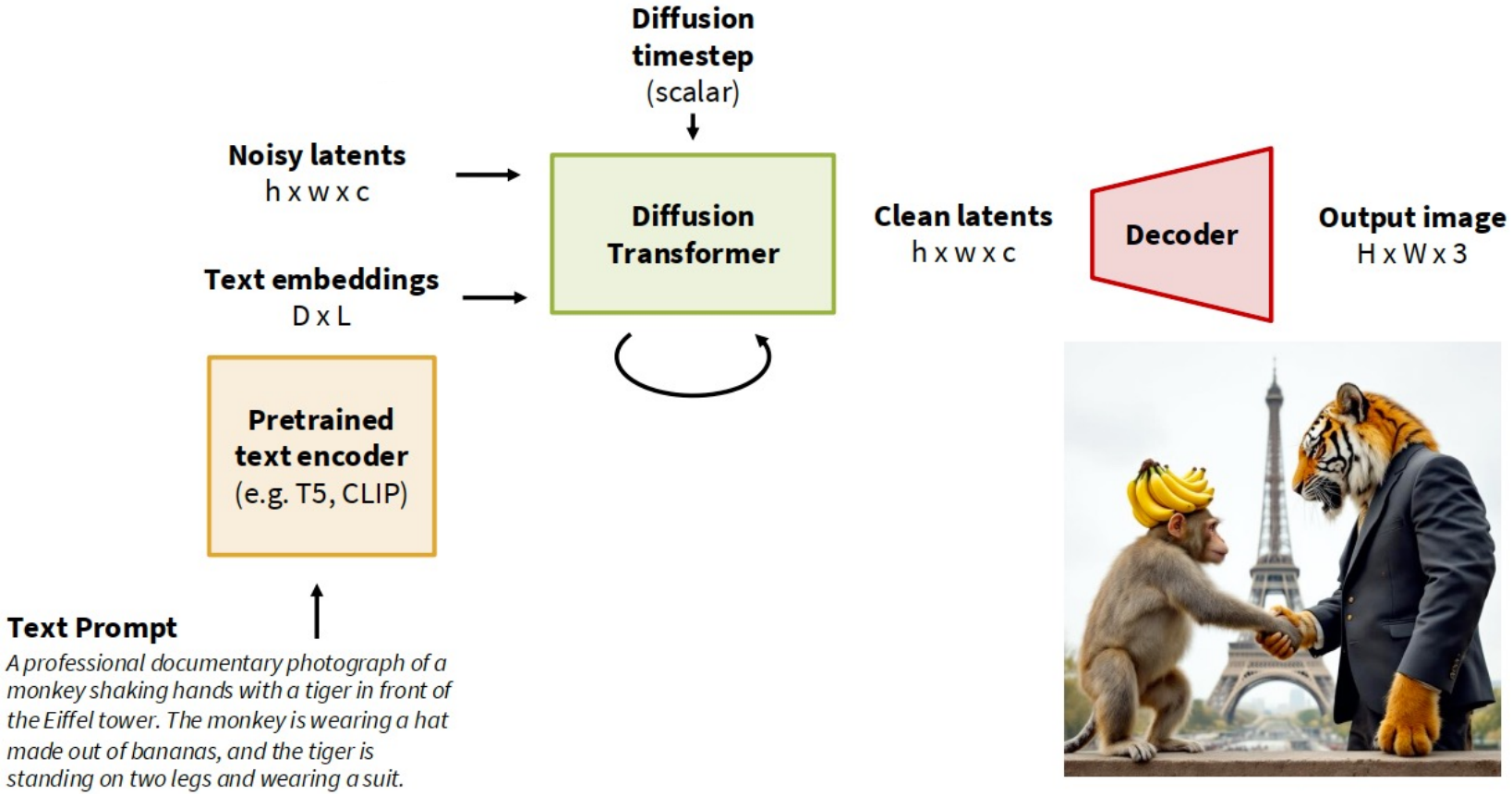
- **key insight**: learn an MLP that outputs the scale and shift parameters for LayerNorm and residual connections



https://www.cs.cmu.edu/~mgormley/courses/10423/

DiT figure from https://arxiv.org/pdf/2212.09748

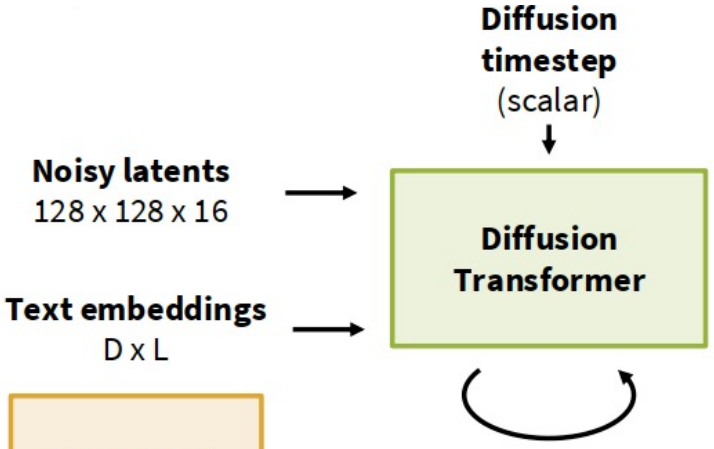
Text-conditional: Diffusion Transformer (DiT)

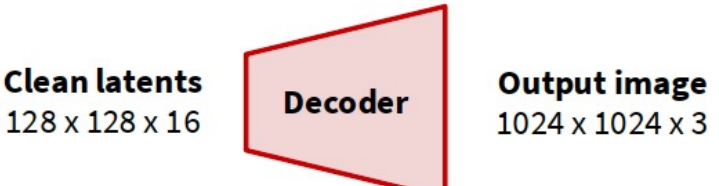


https://cs231n.stanford.edu/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

Text-conditional: Diffusion Transformer (DiT)





Example: FLUX.1 [dev]

Text Encoder: T5 + CLIP
Encoder/Decoder: 8x8 downsampling
Diffusion model: 12B parameter model
2x2 patchify => 64x64 = 1024 image tokens

https://cs231n.stanford.edu/ https://github.com/black-forest-labs/flux Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

Image Customization

Image Customization: GANSpace

GANSpace: Discovering PCA directions

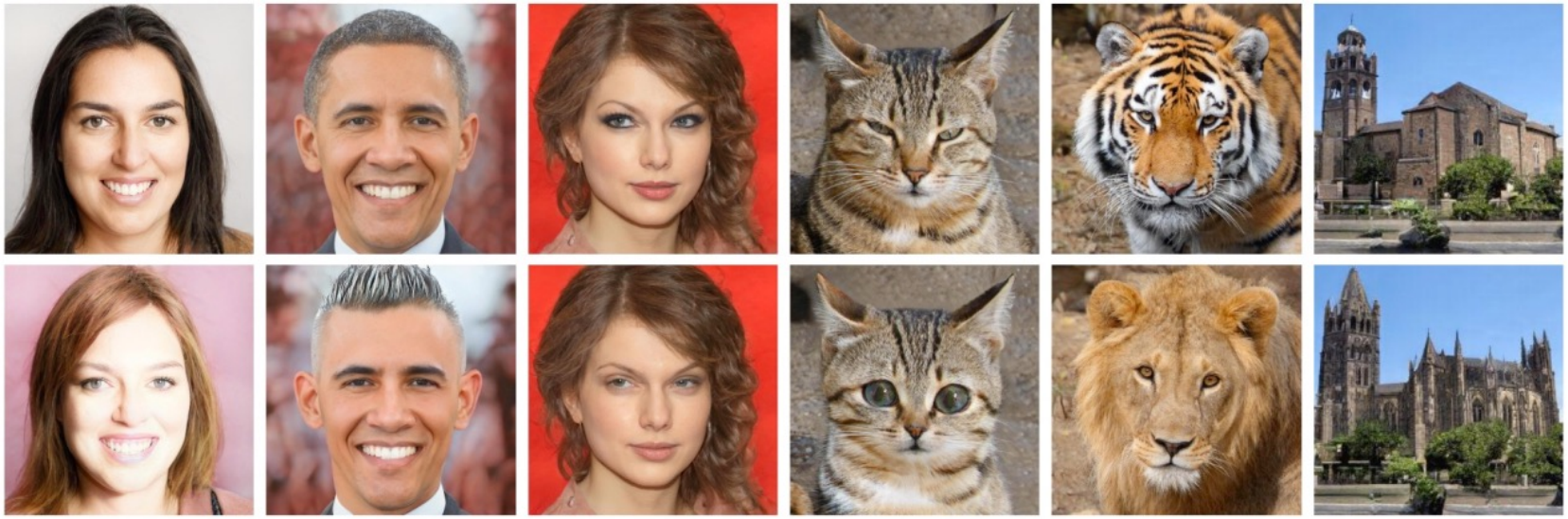
First compute potential directions (PCA), then name them

https://learning-image-synthesis.github.io/sp25/

GANspace [Härkönen et al. 2020]

Image Customization: Manipulate Latent Space

CLIP-guided Directions



$$\arg\min_{w \in \mathcal{W}+} D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \|w - w_s\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w)$$

Output is close to the text — Close to the original latent — Output is close to input

https://learning-image-synthesis.github.io/sp25/

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery [Patashnik et al., ICCV 2021]

Image Customization: Manipulate Latent Space

CLIP-guided Directions

$$\arg\min_{w \in \mathcal{W}+} D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \|w - w_s\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w)$$

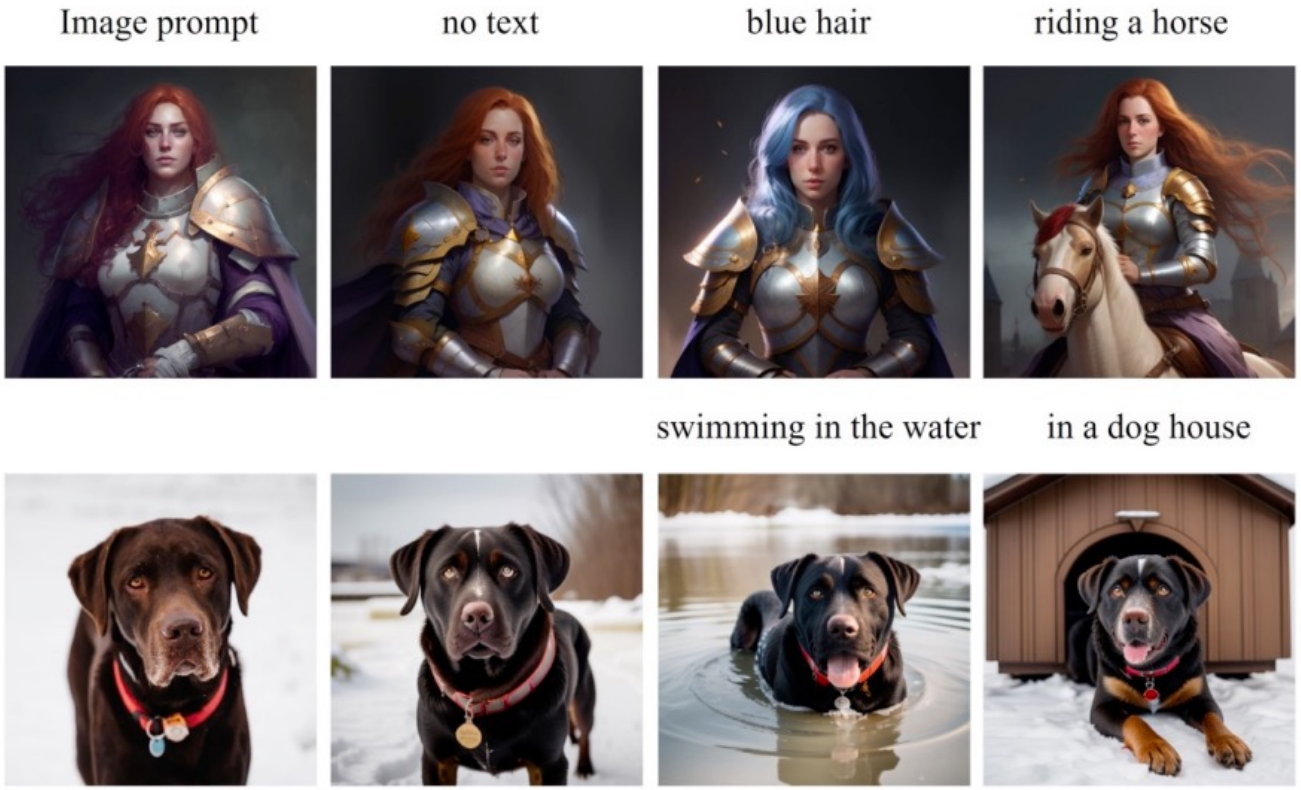
Output is close to the text — Close to the original latent — Output is close to input

https://learning-image-synthesis.github.io/sp25/

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery [Patashnik et al., ICCV 2021]

Image Customization: Encoder Approaches

Image Prompt Adapter (IP-Adapter)

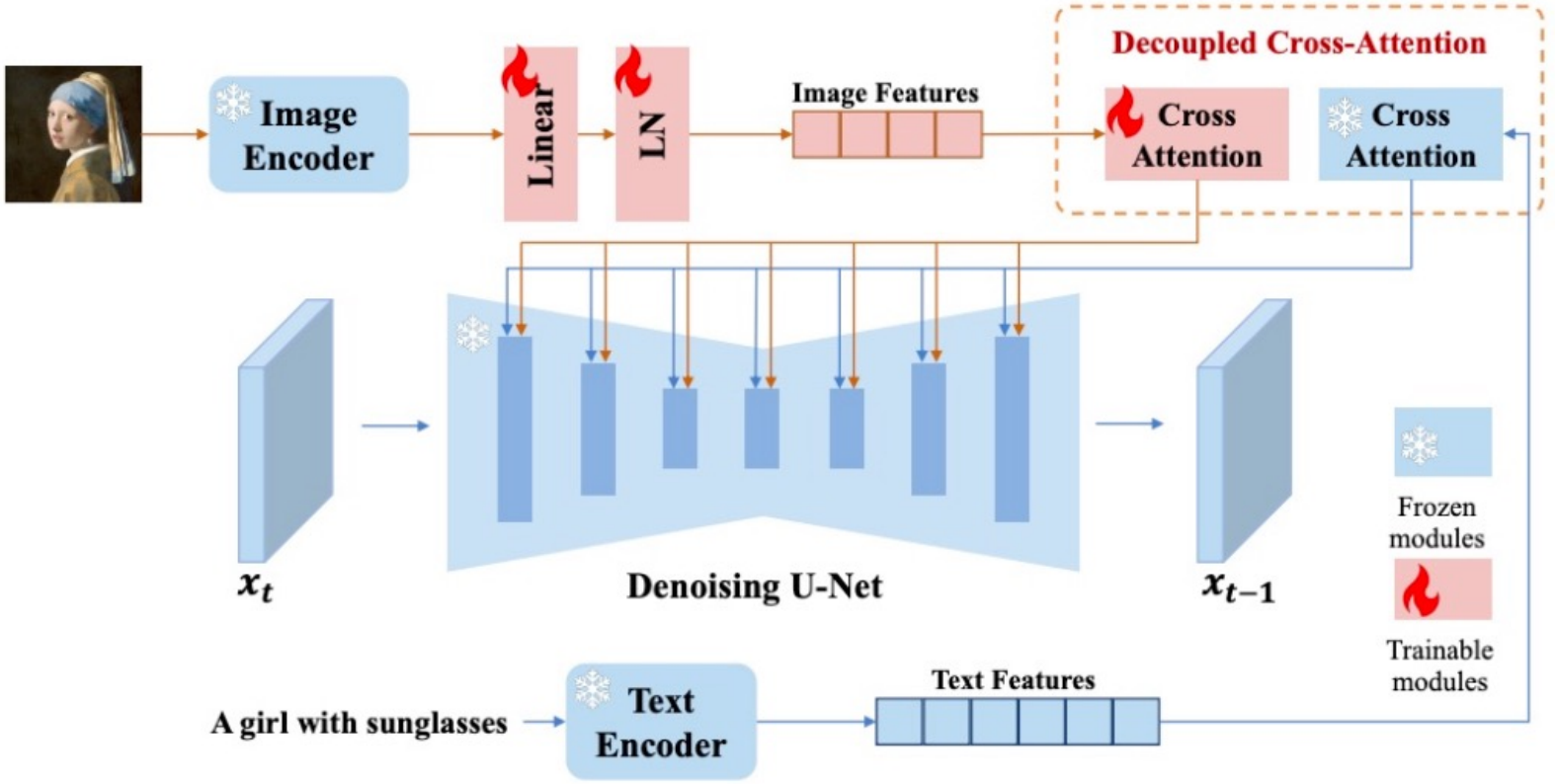


https://learning-image-synthesis.github.io/sp25/

[He Yu et al., CVPR 2024]

Image Customization: Encoder Approaches

Image Prompt Adapter (IP-Adapter)

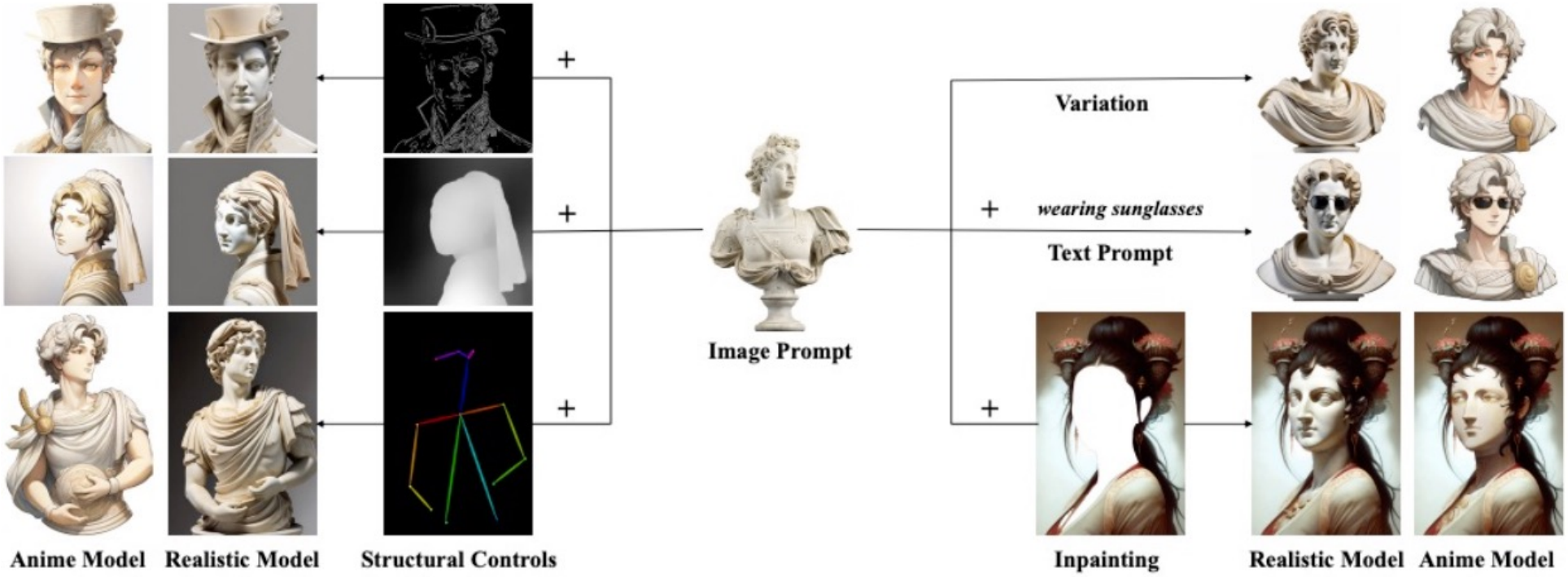


https://learning-image-synthesis.github.io/sp25/

[He Yu et al., CVPR 2024]

Image Customization: Encoder Approaches

Image Prompt Adapter (IP-Adapter)



https://learning-image-synthesis.github.io/sp25/

[He Yu et al., CVPR 2024]

Image Customization

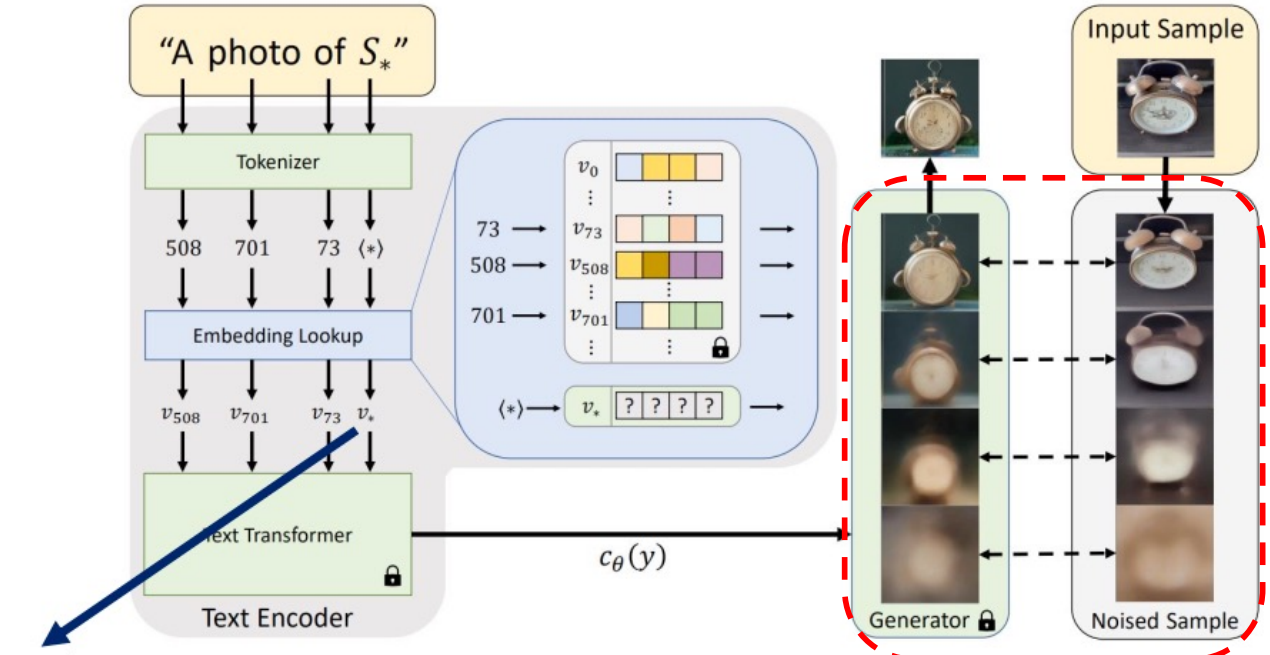
Textual Inversion: Optimizing Text Embedding

https://learning-image-synthesis.github.io/sp25/

[Rinon Gal et al., ICLR 2023]

Image Customization

Textual Inversion: Optimizing Text Embedding



$$v^* = \arg\min_v \mathbb{E}_{\epsilon,\mathbf{x},\mathbf{c},t}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t)||_2]$$

https://learning-image-synthesis.github.io/sp25/

GANs inversion [Zhu et al., 2016] and soft prompting [Lester et al., 2021] - [Rinon Gal et al., ICLR 2023]

Image Customization

Textual Inversion: Optimizing Text Embedding - Results

https://learning-image-synthesis.github.io/sp25/

[Rinon Gal et al., ICLR 2023]

Image Customization

Textual Inversion: Optimizing Text Embedding - Results

https://learning-image-synthesis.github.io/sp25/

[Rinon Gal et al., ICLR 2023]

Image Customization

Textual Inversion: Optimizing Text Embedding – Results – Artistic Style

https://learning-image-synthesis.github.io/sp25/

[Rinon Gal et al., ICLR 2023]

Image Customization

Textual Inversion: Optimizing Text Embedding – Personalized Concepts

How to describe personalized concepts?

V* dog

Where V* is a modifier token in the text embedding space

https://learning-image-synthesis.github.io/sp25/

[Rinon Gal et al., ICLR 2023]

Image Customization: prompt2prompt

Editing Cross Attention

- **Goal**: edit images with text only and do not require the user to provide a mask
- **Key Idea**:
  - given pre-trained latent diffusion model
  - run diffusion model with **original prompt** and store the attention weights and cross-attention weights (from the pixels back to the text)
  - re-run diffusion with **edited prompt**, but (carefully) copy in the cross-attention weights from the previous run
- **Inference only**: no training is involved! we only modify how the samples are drawn from the pre-trained latent diffusion model

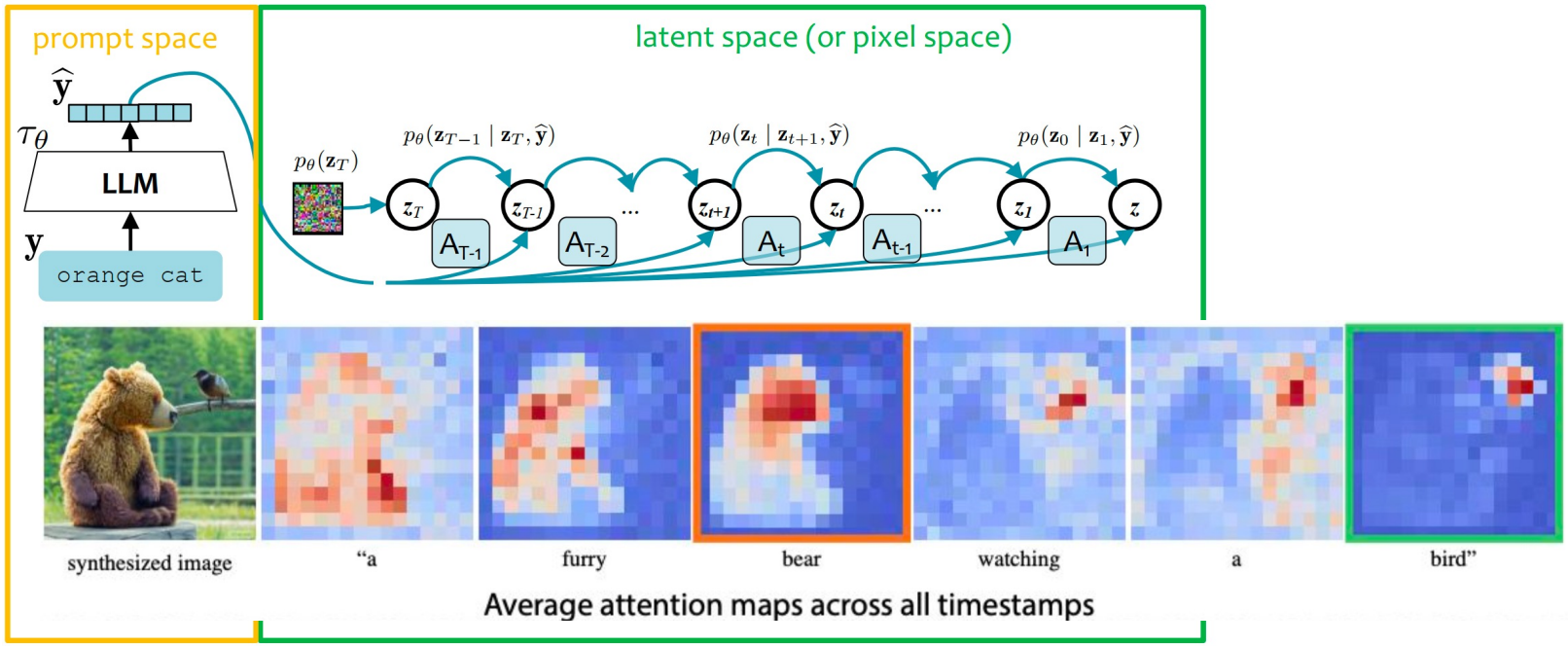
Prompt-to-Prompt Editing (Hertz et al.) Hertz et al.

Prior Editing Methods Hertz et al.

Image Customization: prompt2prompt

Editing Cross Attention

1. encode the original prompt y
2. run diffusion on y and obtain attention weights $A_{T-1},\ldots,A_1$

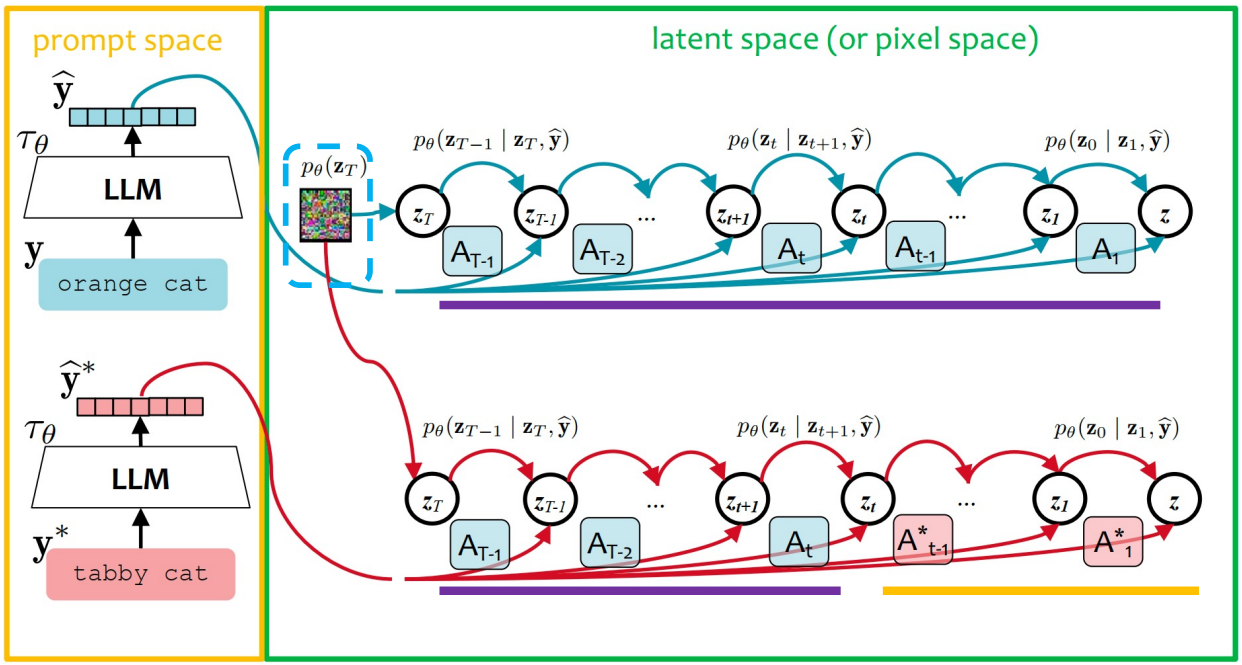


https://www.cs.cmu.edu/~mgormley/courses/10423/

Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023

Image Customization: prompt2prompt

Editing Cross Attention



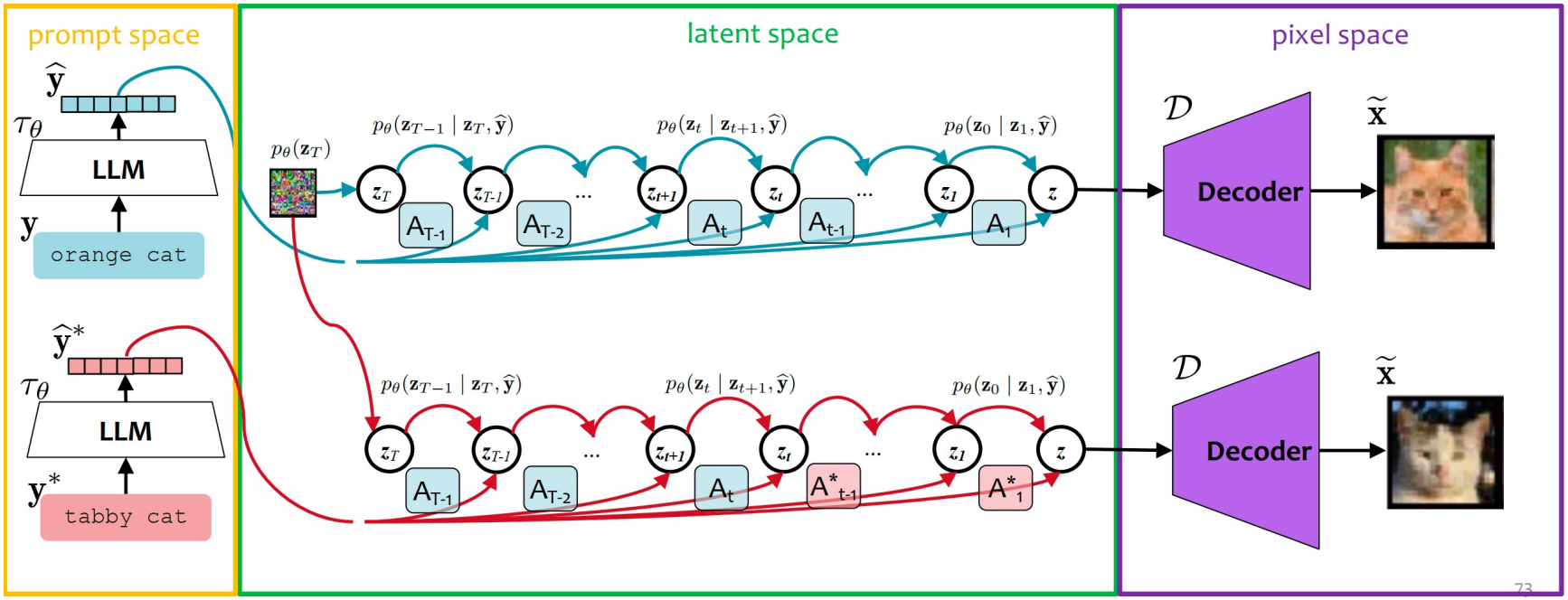
1. encode the original prompt y
2. run diffusion on y and obtain attention weights $A_{T-1},\dots,A_1$
3. encode the modified prompt y*
4. run diffusion again
   a) reuse the noise z_T from the original run
   b) use the attention weights from the original run until timestep τ $A_{T-1},\dots,A_t$
   c) then switch to using attention weights from this current run $A^*_{t-1},\dots,A^*_1$
   d) regardless of which attention weights, you still attend to y*

Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023

Image Customization: prompt2prompt

Editing Cross Attention

5. if running in latent space, then use decoder to recover pixel space representation



https://www.cs.cmu.edu/~mgormley/courses/10423/

Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023

Image Customization: prompt2prompt

Editing Cross Attention

Algorithm 1: Prompt-to-Prompt image editing

Input: A source prompt $\mathcal{P}$, a target prompt $\mathcal{P}^*$, and a random seed s

Output: A source image x_{src} and an edited image x_{dst}.

$z_T \sim N(0, I)$ a unit Gaussian random variable with random seed s;

$z_T^* \leftarrow z_T$;

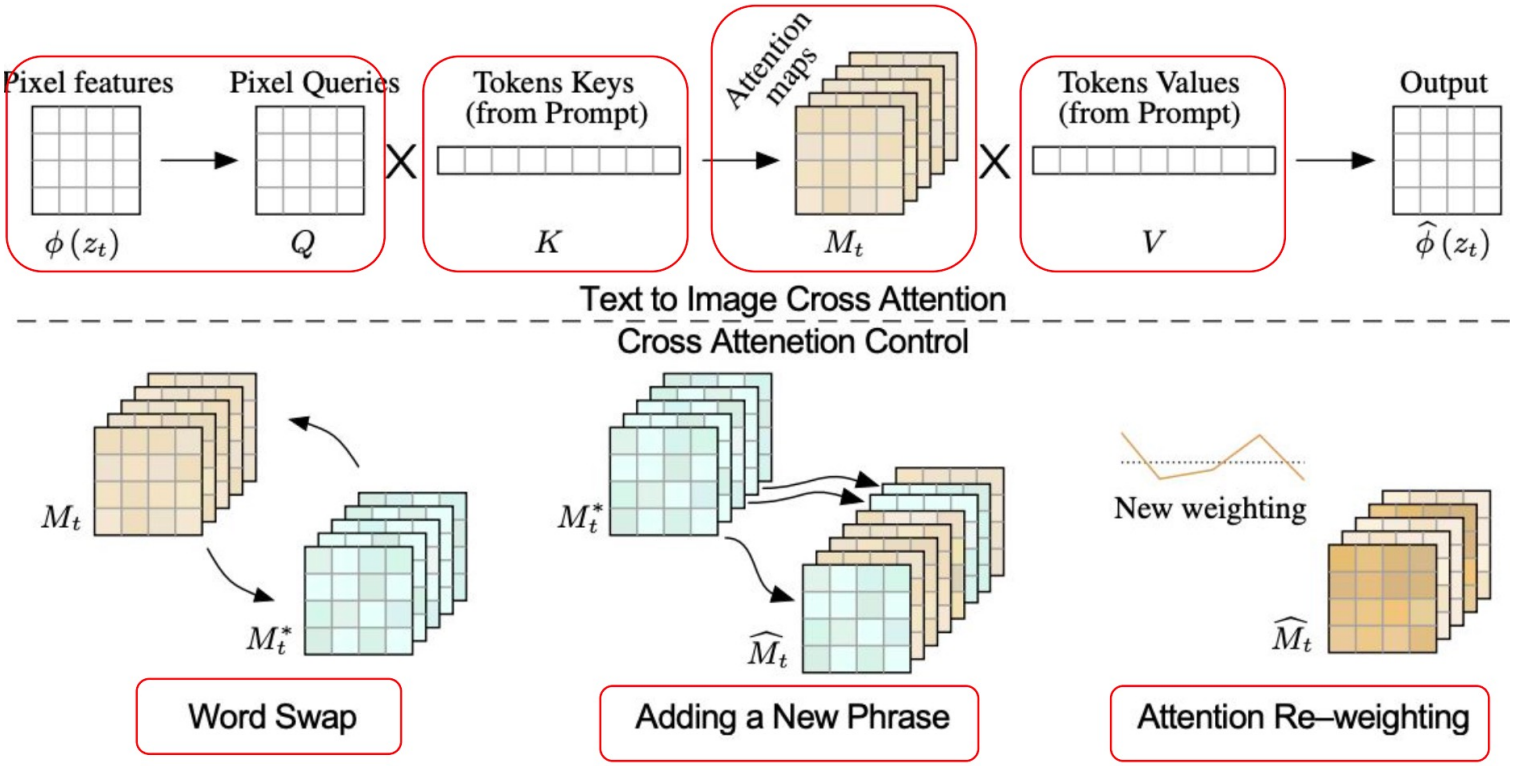
for $t = T, T-1, \ldots, 1$ **do**

- $z_{t-1}, M_t \leftarrow DM(z_t, \mathcal{P}, t, s)$; ← Source Pass
- $M_t^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s)$; ← Target Pass
- $\widehat{M}_t \leftarrow Edit(M_t, M_t^*, t)$; ← Edit Attention
- $z_{t-1}^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s_t)\{M \leftarrow \widehat{M}_t\}$; ← Apply Edited Attention

end

Return (z_0, z_0^*)

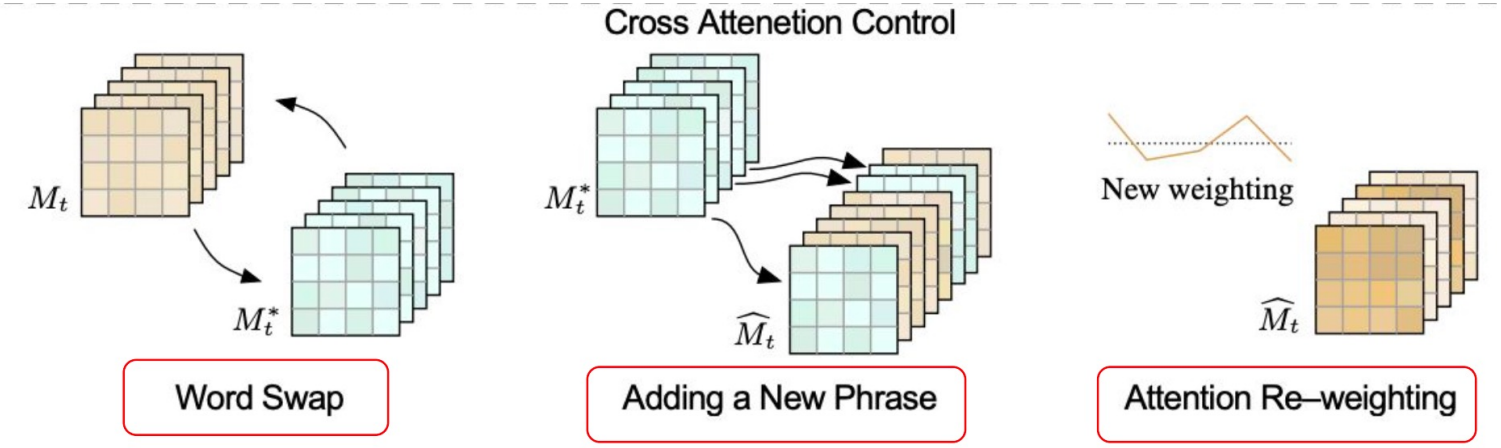
Image Customization: prompt2prompt

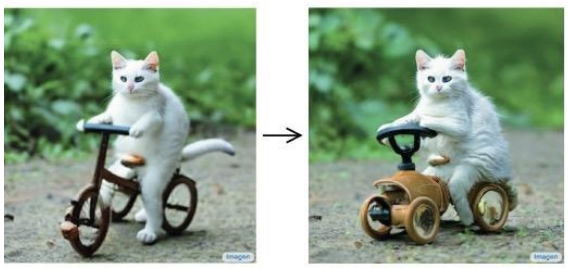


https://www.cs.cmu.edu/~mgormley/courses/10423/

Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023

Image Customization: prompt2prompt





"Photo of a cat riding on a bicycle." car

"Children drawing of a castle next to a river."

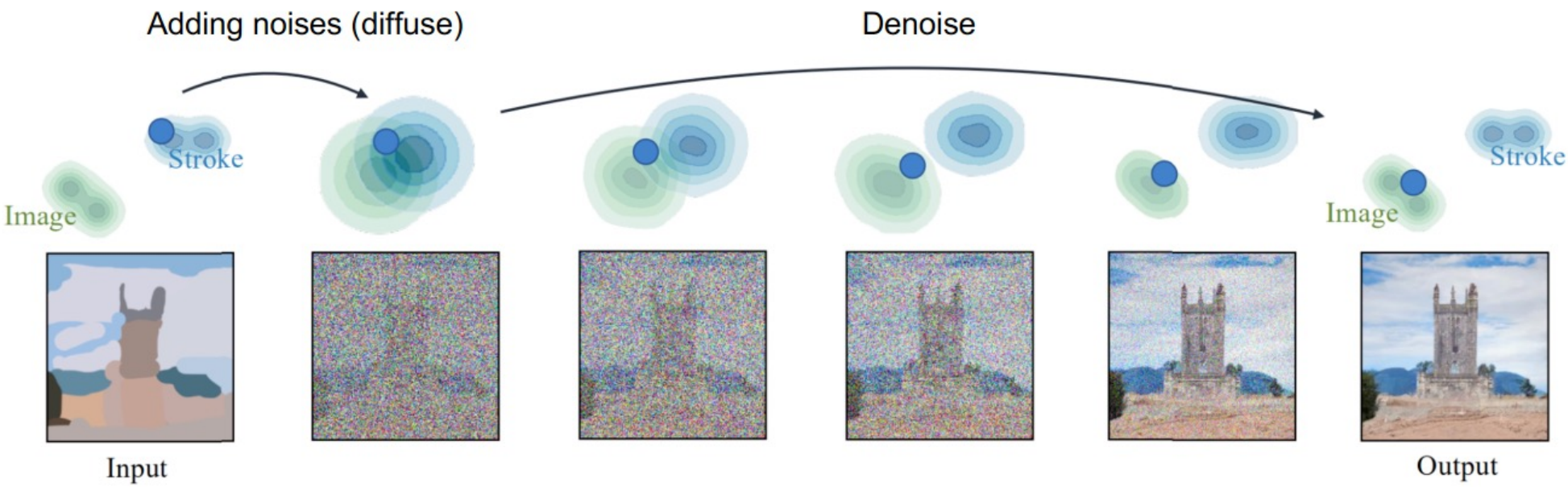
"The boulevards are crowded today."

https://www.cs.cmu.edu/~mgormley/courses/10423/

Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023

Image Customization: Stroke-guided Image-to-image

SDEdit (https://arxiv.org/abs/2108.01073) recipe: diffuse → denoise



Slide credit: Robin Rombach

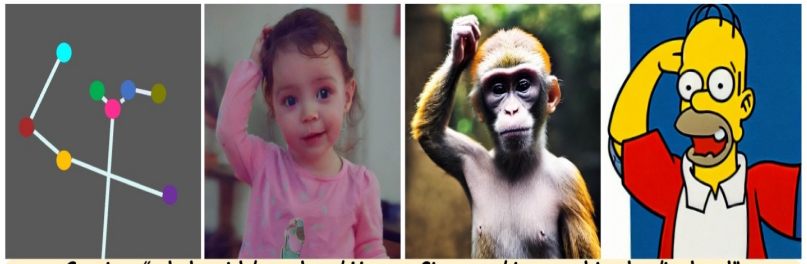
Other Image Applications

GLIGEN: Open-Set Grounded Text-to-Image Generation

Caption: "A woman sitting in a restaurant with a pizza in front of her "
Grounded text: table, pizza, person, wall, car, paper, chair, window, bottle, cup

Caption: "Elon Musk and Emma Watson on a movie poster"
Grounded text: Elon Musk, Emma Watson; Grounded style image: blue inset

Caption: "A dog / bird / helmet / backpack is on the grass"
Grounded image: red inset



Caption: "a baby girl / monkey / Hormer Simpson / is scratching her/its head"
Grounded keypoints: plotted dots on the left image

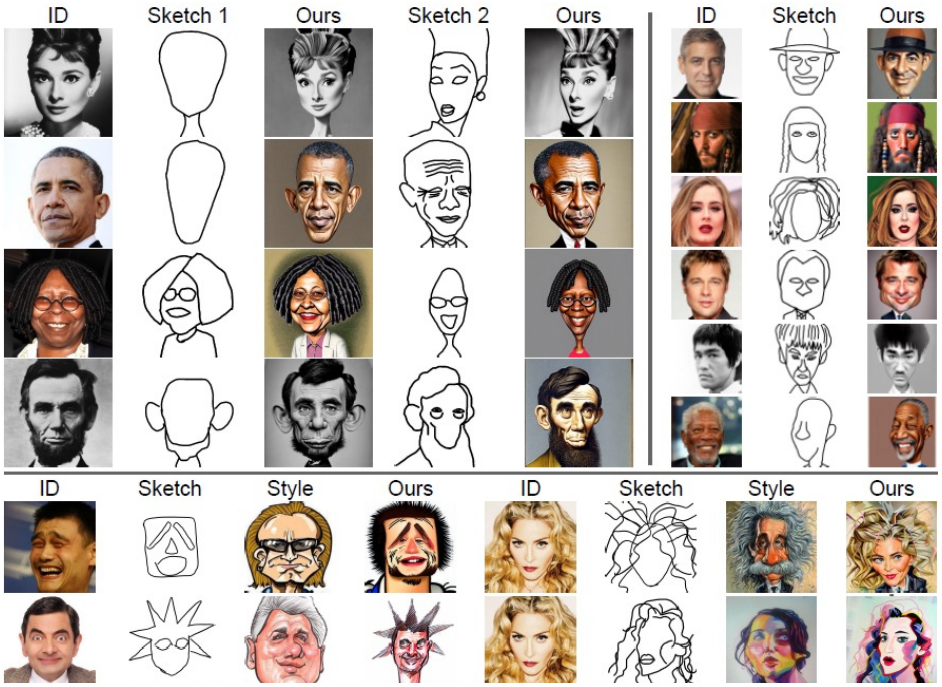
https://huggingface.co/spaces/gligen/demo, https://gligen.github.io/

GLIGEN: Open-Set Grounded Text-to-Image Generation

Yong Jae Lee, https://huggingface.co/spaces/gligen/demo, https://gligen.github.io/

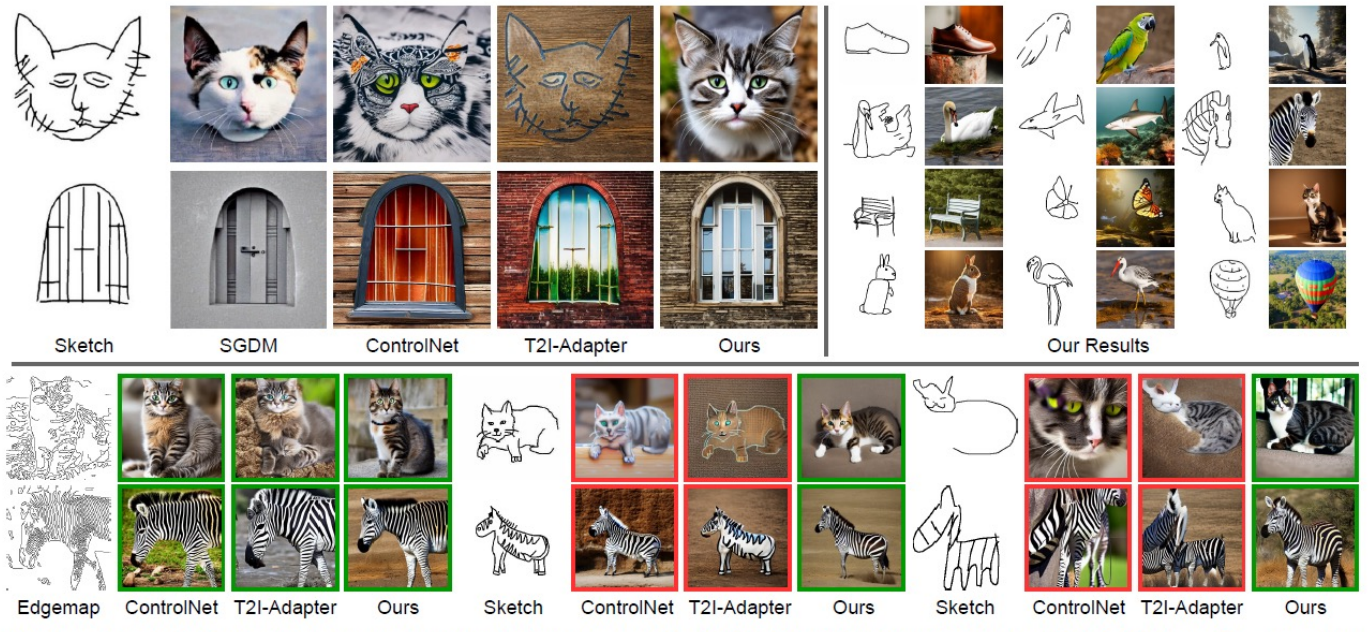
DemoCaricature: Democratising Caricature Generation with a Rough Sketch

Dar-Yen Chen Subhadeep Koley Aneeshan Sain Pinaki Nath Chowdhury
Tao Xiang Ayan Kumar Bhunia Yi-Zhe Song
SketchX, CVSSP, University of Surrey, United Kingdom.
{s.koley, a.sain, p.chowdhury, t.xiang, a.bhunia, y.song}@surrey.ac.uk
https://democaricature.github.io



It's All About Your Sketch: Democratising Sketch Control in Diffusion Models

Subhadeep Koley[1,2] Ayan Kumar Bhunia[1] Deeptanshu Sekhri[1] Aneeshan Sain[1,2]
Pinaki Nath Chowdhury[1,2] Tao Xiang[1,2] Yi-Zhe Song[1,2]
[1]SketchX, CVSSP, University of Surrey, United Kingdom.
[2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.
{s.koley, a.bhunia, d.sekhri, a.sain, p.chowdhury, t.xiang, y.song}@surrey.ac.uk



RAVE: Randomized Noise Shuffling for Fast and Consistent Video Editing with Diffusion Models

Ozgur Kara[1*] Bariscan Kurtkaya[2*†] Hidir Yesiltepe[4] James M. Rehg[1,3] Pinar Yanardag[4]

[1]Georgia Tech [2]KUIS AI Center [3]UIUC [4]Virginia Tech

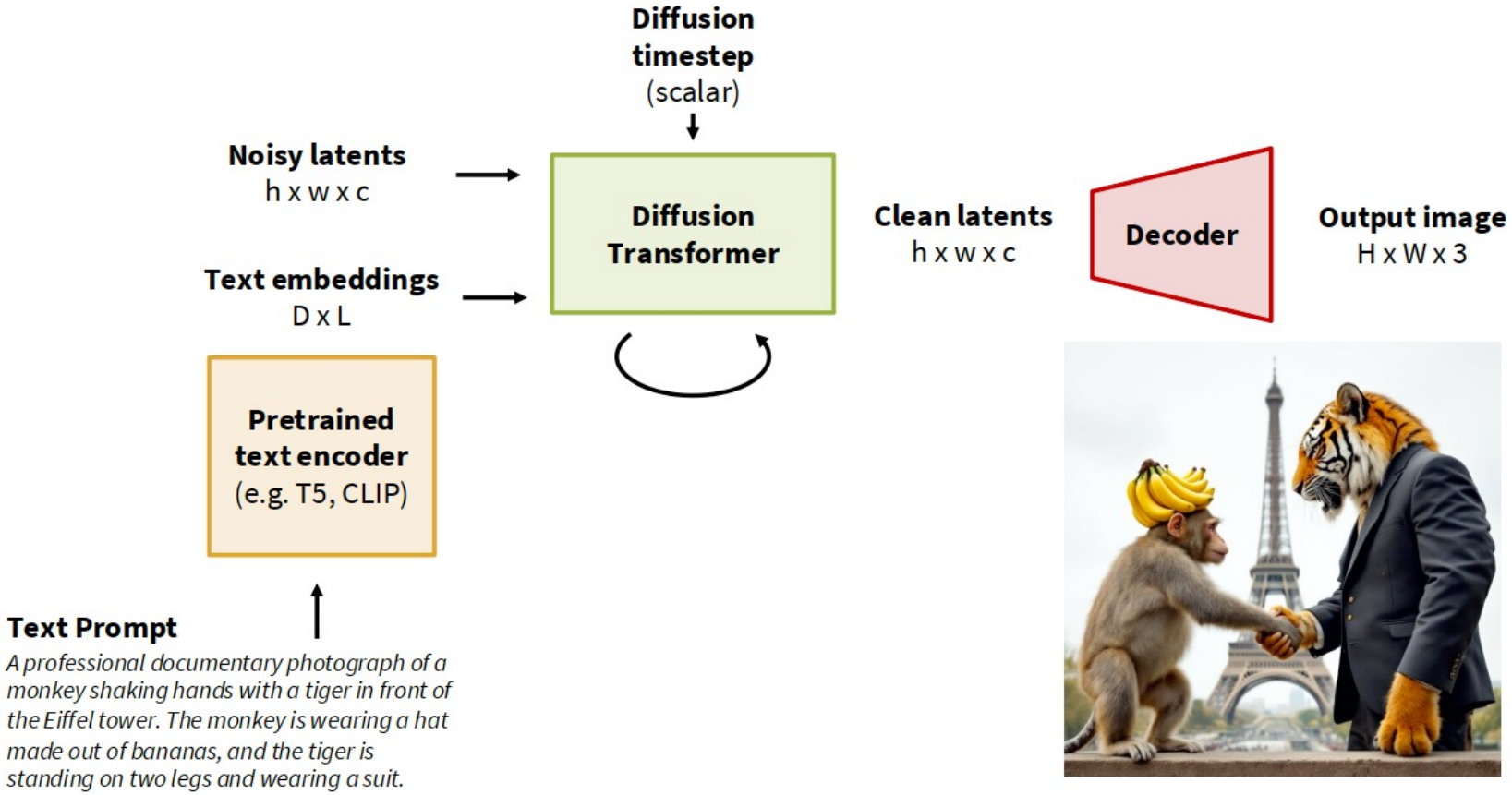
okara7@gatech.edu, bkurtkaya23@ku.edu.tr, hidir@vt.edu, jrehg@uiuc.edu, pinary@vt.edu

Project Webpage: https://rave-video.github.io

Video Diffusion Models

Text-conditional: Diffusion Transformer (DiT)

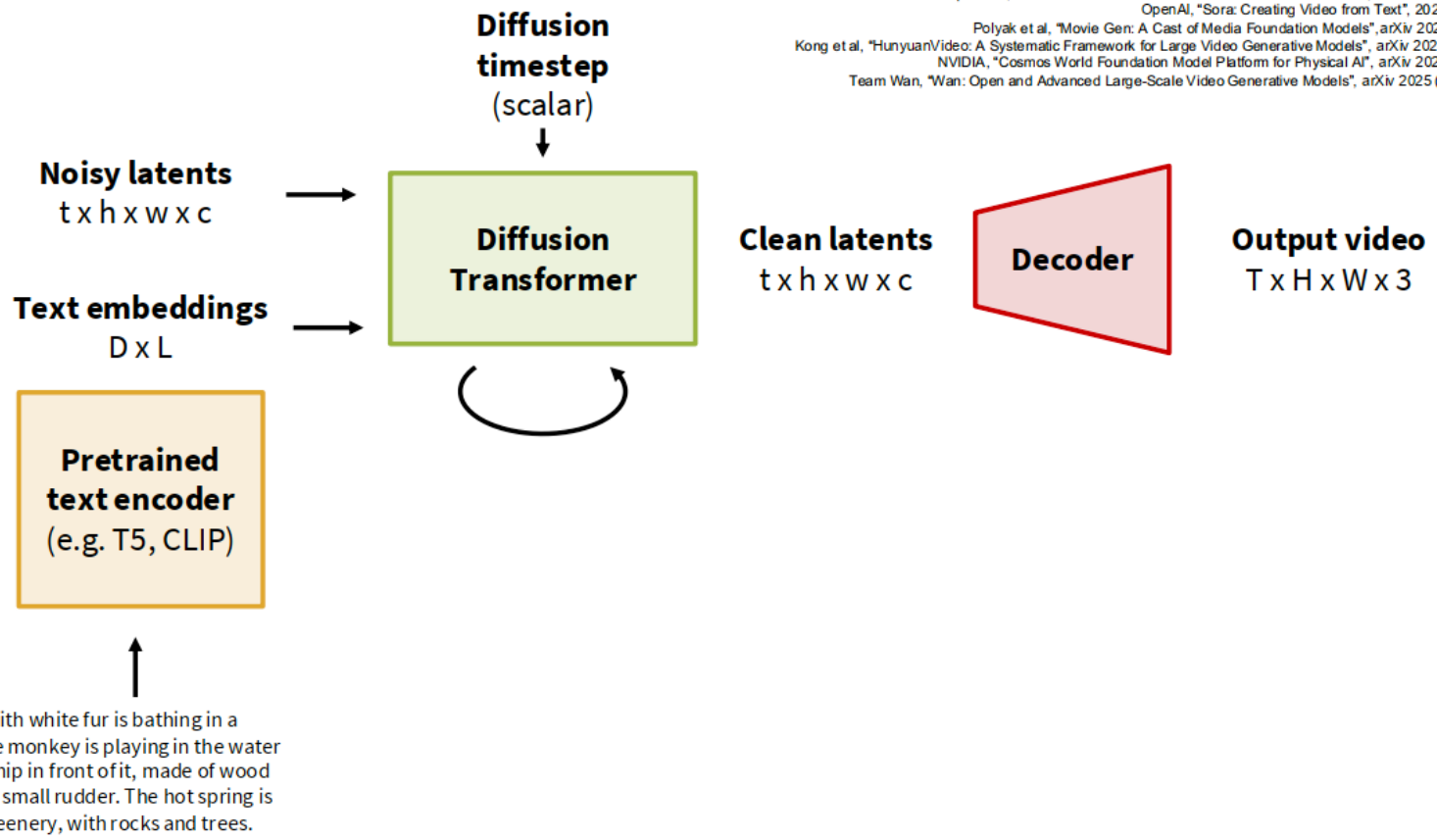
Recall



https://cs231n.stanford.edu/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023

Text to Video

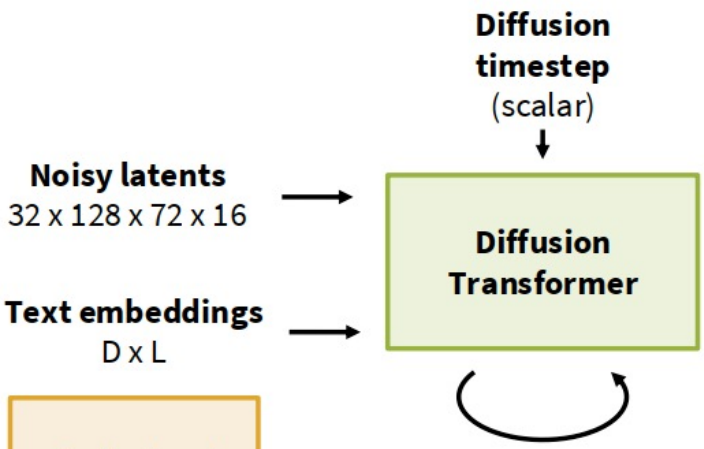


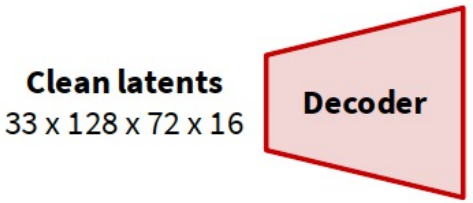
Gupta et al, "Photorealistic Video Generation with Diffusion Models", arXiv 2023 (Dec)
OpenAI, "Sora: Creating Video from Text", 2024 (Feb)
Polyak et al, "Movie Gen: A Cast of Media Foundation Models", arXiv 2024 (Oct)
Kong et al, "HunyuanVideo: A Systematic Framework for Large Video Generative Models", arXiv 2024 (Dec)
NVIDIA, "Cosmos World Foundation Model Platform for Physical AI", arXiv 2025 (Jan)
Team Wan, "Wan: Open and Advanced Large-Scale Video Generative Models", arXiv 2025 (March)

https://cs231n.stanford.edu/

Text to Video

Gupta et al, "Photorealistic Video Generation with Diffusion Models", arXiv 2023 (Dec)
OpenAI, "Sora: Creating Video from Text", 2024 (Feb)
Polyak et al, "Movie Gen: A Cast of Media Foundation Models", arXiv 2024 (Oct)
Kong et al, "HunyuanVideo: A Systematic Framework for Large Video Generative Models", arXiv 2024 (Dec)
NVIDIA, "Cosmos World Foundation Model Platform for Physical AI", arXiv 2025 (Jan)
Team Wan, "Wan: Open and Advanced Large-Scale Video Generative Models", arXiv 2025 (March)



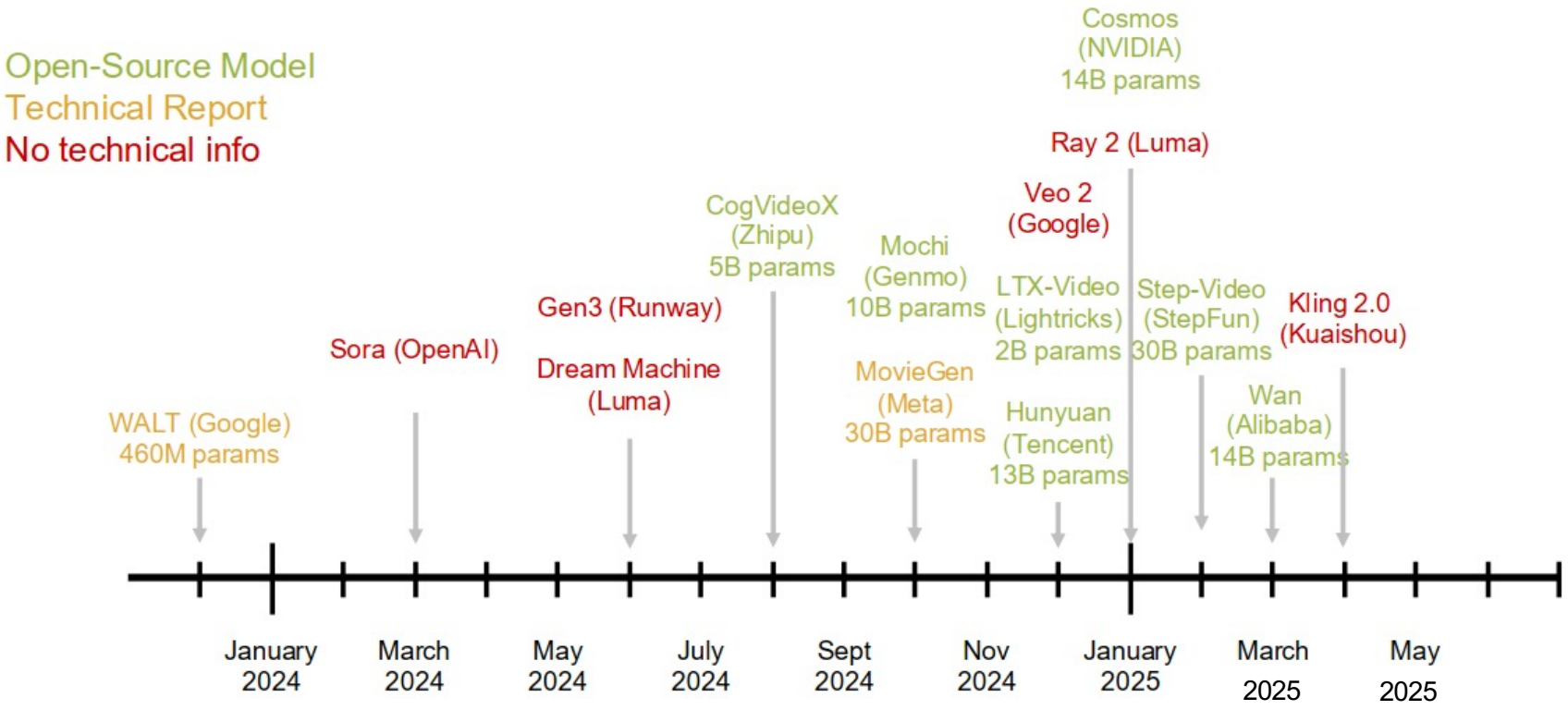


Example: Meta MovieGen

Text Encoder: UL2, ByT5, MetaCLIP
Encoder/Decoder: 8x8x8 downsample
Diffusion model: 30B param DiT
1x2x2 patchify => 76K tokens

https://cs231n.stanford.edu/

Video Diffusion Era



Gupta et al, "Photorealistic Video Generation with Diffusion Models", arXiv 2023 (Dec)
OpenAI, "Sora: Creating Video from Text", 2024 (Feb)
Polyak et al, "Movie Gen: A Cast of Media Foundation Models",arXiv 2024 (Oct)
Kong et al, "HunyuanVideo: A Systematic Framework for Large Video Generative Models", arXiv 2024 (Dec)
NVIDIA, "Cosmos World Foundation Model Platform for Physical AI", arXiv 2025 (Jan)
Team Wan, "Wan: Open and Advanced Large-Scale Video Generative Models", arXiv 2025 (March)

Stable Video Diffusion

Stable Video Diffusion: link

Large-scale Diffusion Models

Large-scale diffusion models - Stable Diffusion 3

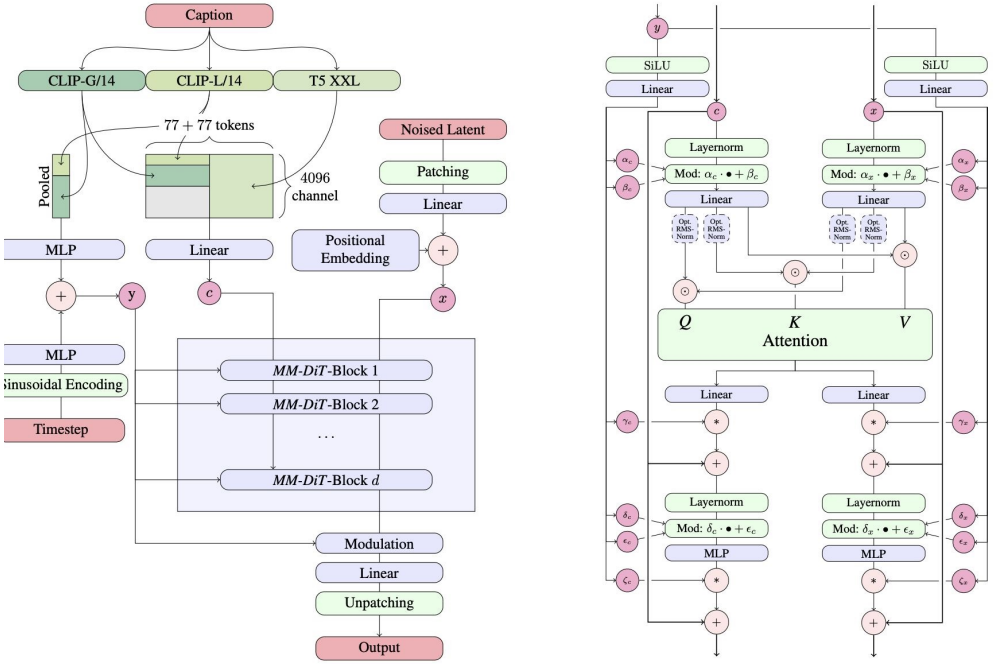
- **Flow Matching** model with “straight line” schedulers (CondOT path)
- **Classifier-free guidance** with weight 2.0 - 5.0
- **Flow Matching** in latent space (use pre-trained **VAE**)
- Number of parameters of model: 8 billion
- Number of simulation/sampling steps: 50
- Dataset: LAION

Image source: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis [1]

https://diffusion.csail.mit.edu/2026/

Large-scale diffusion models - Stable Diffusion 3

- Conditions on **CLIP** (coarse-grained) and **T5-XXL** (sequence-level) text embeddings via cross-attention.
- **MM-DiT** architecture: Extends **DiT** from class-conditioning to text-conditioning and processes text and images through the entire network via cross-attention



https://diffusion.csail.mit.edu/2026/

Image source: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis [1]

Large-scale diffusion models - Meta MovieGen

- **Flow Matching** model with “straight line” schedulers (CondOT path)
- **Classifier-free guidance**
- **Flow Matching in latent space** (use pre-trained VAE)
- Really **crucial** for videos because of added time dimension
- Neural network architecture: **DiT** adapted to videos
- Number of parameters of model: 30 billion
- **6,144 H100 GPUs!**

https://diffusion.csail.mit.edu/2026/

Image source: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis [1]

Extra

CFG: Classifier-free Guidance

- Diffusion models (unlike GANs) are great at generating diverse samples
- But when **diffusion** is conditioned on some input (text, label, etc.) that diversity may cause it to stray away from the prompt
- **Classifier-free guidance** helps diffusion to adhere to the prompt, yielding higher quality images

Algorithm 1 Sampling from DDPM with Classifier-free Guidance

$w = 7.5$

$\mathbf{c} = \texttt{tokenize("a cat with green eyes")}$

$\mathbf{c}' = \texttt{tokenize("")}$

$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t \in \{T, \ldots, 1\}$ **do**

  $\boldsymbol{\epsilon}_\theta \leftarrow (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c}')$

  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

  $\hat{\mathbf{z}}_0 \leftarrow \left(\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta\right) / \sqrt{\bar{\alpha}_t}$

  $\hat{\boldsymbol{\mu}}_t \leftarrow \alpha_t^{(0)}\hat{\mathbf{z}}_0 + \alpha_t^{(t)}\mathbf{z}_t$

  $\mathbf{z}_{t-1} \leftarrow \hat{\boldsymbol{\mu}}_t + \sigma_t^2\boldsymbol{\epsilon}$

return $\mathbf{x}_0$

Generator
Noisy Sample

https://www.cs.cmu.edu/~mgormley/courses/10423/

Peebles and Xie, "Scalable Diffusion Models with Transformer", ICCV 2023