

# Deep Generative Models: Latent Diffusion Models

Fall Semester 2024

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS)

 Rachleff University Professor, University of Pennsylvania 

Amazon Scholar & Chief Scientist at NORCE



# Diffusion Models

- **Derivation of Diffusion Models (Last Lecture)**
  - Markov Hierarchical Variational Auto Encoders (MHVAE)
  - Diffusion Models are VAEs with Linear Gaussian Autoregressive latent space
    - Forward Process
    - Conditional Distributions for the Forward Process
    - Reverse Process
  - ELBO for Diffusion Models is a particular case of ELBO for VAEs with extra structure
  - Implementation Details
- **Application of Diffusion Models (Today's Lecture)**
  - Stable Diffusion: Text-Conditioned Diffusion Model
  - ControlNet: Multimodal Control for Consistent Synthesis

# Stable Diffusion

- DDPM operates in pixel space: optimization takes **hundreds of GPU days** and **inference is expensive**.
  - 50k sample takes around 5 days on a single A100 GPU.
- To enable training of diffusion models on limited computational resources while retaining their quality and flexibility: SD performs **denoising in the latent space** of powerful pretrained autoencoders.
- Contribution:
  1. Denoising in the latent space enables spatial **complexity reduction** and **detail preservation**.
  2. Introducing cross-attention layers enables **conditional input such as texts**.

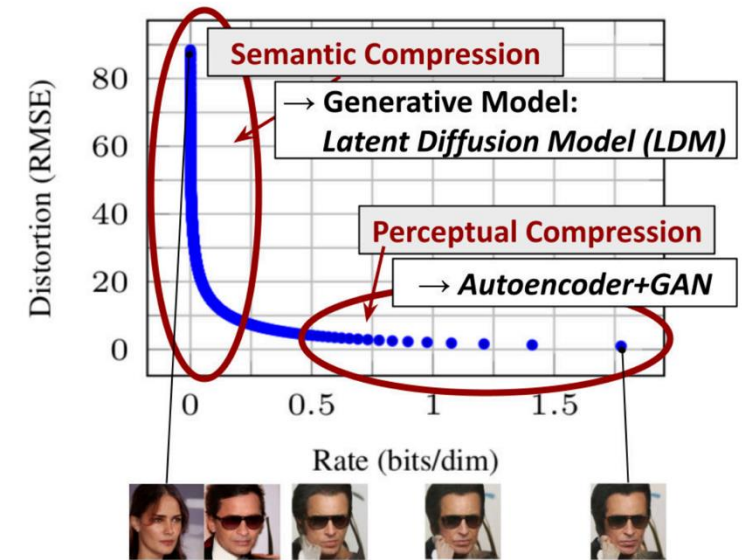
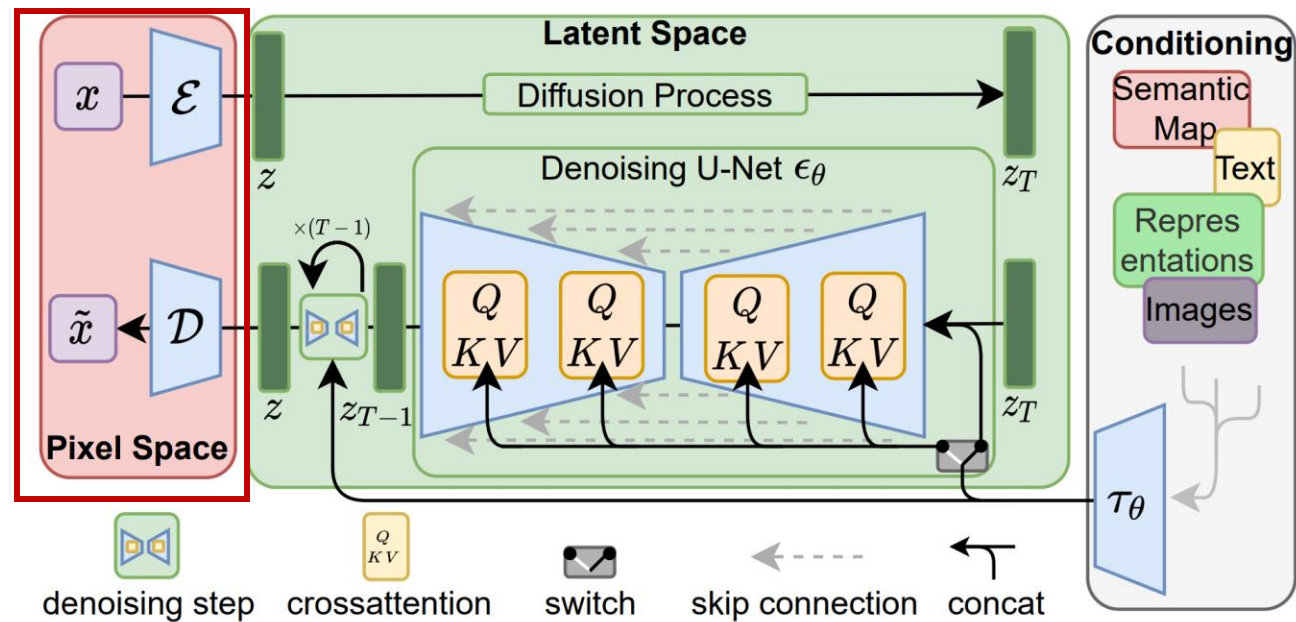


Figure 2. Illustrating perceptual and semantic compression: Most bits of a digital image correspond to imperceptible details. While DMs allow to suppress this semantically meaningless information by minimizing the responsible loss term, gradients (during training) and the neural network backbone (training and inference) still need to be evaluated on all pixels, leading to superfluous computations and unnecessarily expensive optimization and inference. We propose *latent diffusion models (LDMs)* as an effective generative model and a separate mild compression stage that only eliminates imperceptible details. Data and images from [29].

# Stable Diffusion: Two-Stage Image Synthesis

- (Stage 1) **Perceptual Image Compression**: The SD framework uses a pre-trained VAE to map data into a low-dimensional space and back to pixel space.
- Compared to the high-dimensional pixel space, the low-dimensional latent space is more suitable for likelihood-based generative models, as they can now (I) **focus on the important, semantic bits of the data** and (II) train in a **lower dimensional and more computationally efficient space**.

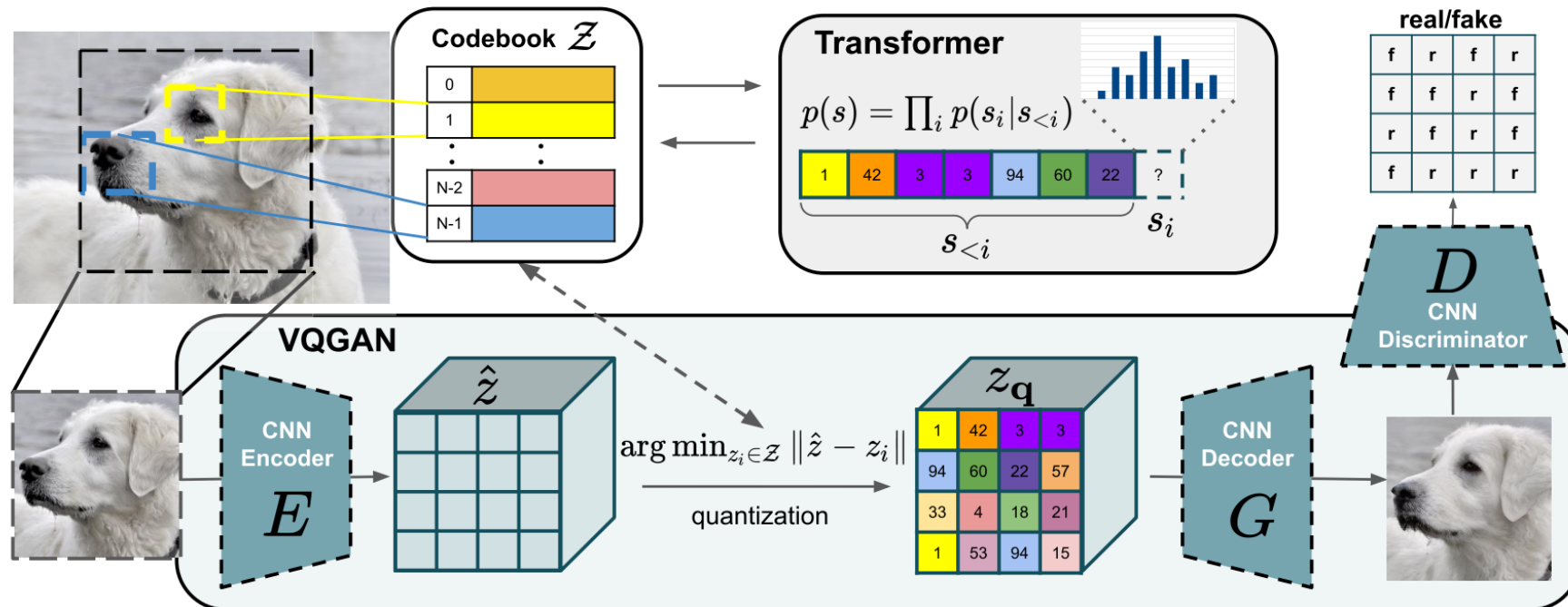


# Stable Diffusion: Two-Stage Image Synthesis

- (Stage 1) **Perceptual Image Compression**: The VAE is trained by a combination of a reconstruction loss and a patch-based adversarial loss.
- It deploys a VQGAN to learn a codebook of context-rich visual parts, whose composition is modeled with an autoregressive transformer architecture.

learn to reconstruct the image      differentiate original images from reconstructions

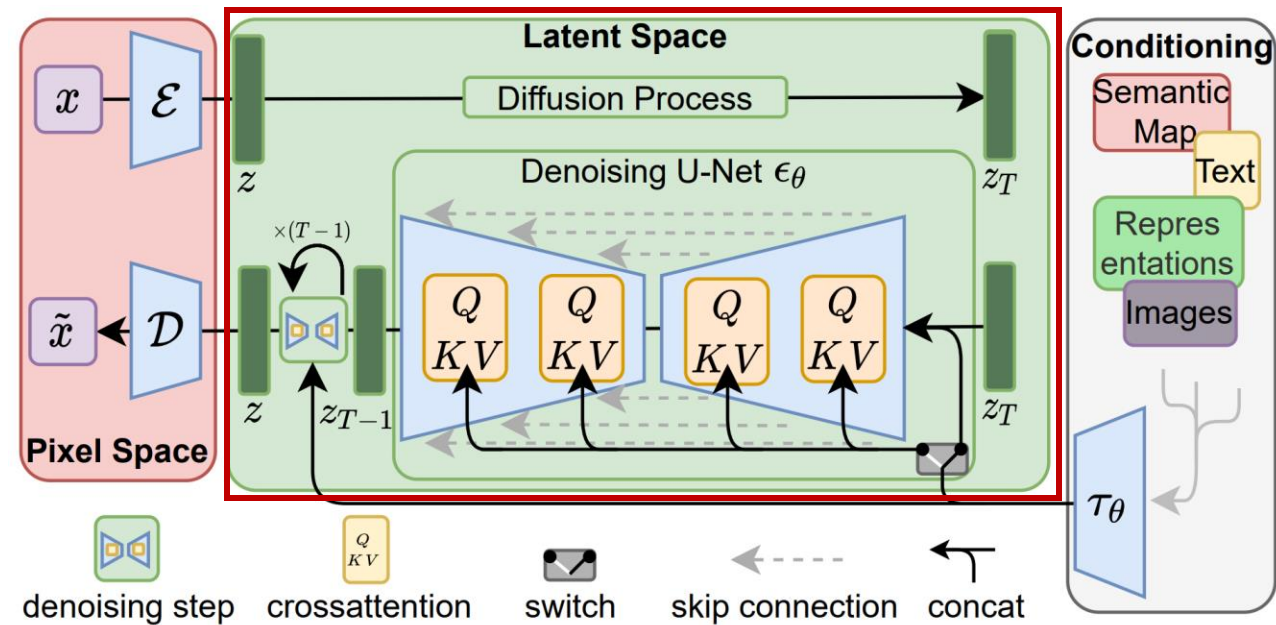
$$L_{VQGAN} = \min_{E,G} \max_D \left( L_{rec}(x, G(E(x))) - L_{synthetic}(D(G(E(x)))) + L_{real} D(x) + L_{reg}(x; E, G) \right)$$



# Stable Diffusion: Two-Stage Image Synthesis

- (Stage 2) **Denoising Latent Representations**: the denoising happens in this compressed latent space given by Stage 1.
- Recall DDPM where we denoise in the image space. The SD is nothing more but denoising for the latent code.
- The forward process is to deterministically add  $T$  Gaussian noises to the original latent code  $z$ , and **the reverse process learns to denoise**.

Encoder and Decoder are from the Stage 1.



# Stable Diffusion: Two-Stage Image Synthesis

- The neural backbone of SD is a time-conditional U-Net  $\epsilon_\phi(z_t, t, \tau_\theta(y))$ , trained to **predict the noise to be removed from the latent code  $z_t$** .
- SD augments the U-Net backbone with the cross-attention mechanism to receive conditions  $\tau_\theta(y)$  such as textual prompts or semantic segmentations.

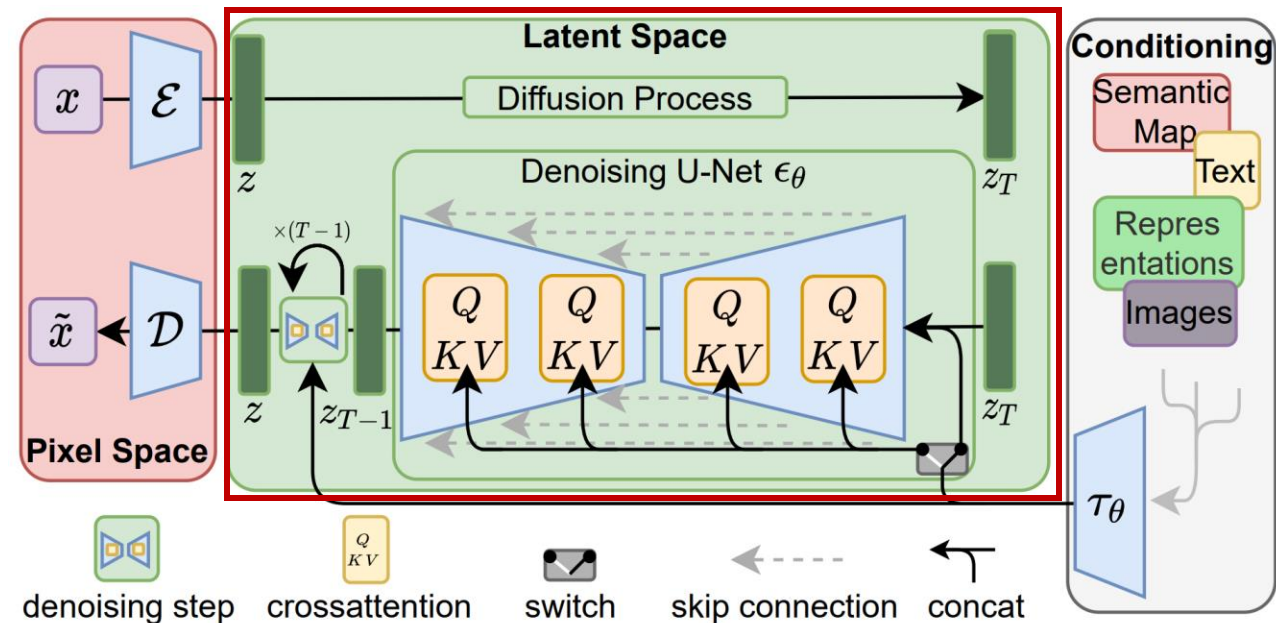
$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t} \left[ \left\| \epsilon - \epsilon_\phi(z_t, t, \tau_\theta(y)) \right\|_2^2 \right]$$

(flattened) visual representation from the U-Net  $\epsilon_\phi$

$$\begin{aligned} Q &= W_Q^{(i)} \cdot \varphi_i(z_t), \\ K &= W_K^{(i)} \cdot \tau_\theta(y), \\ V &= W_V^{(i)} \cdot \tau_\theta(y) \end{aligned}$$

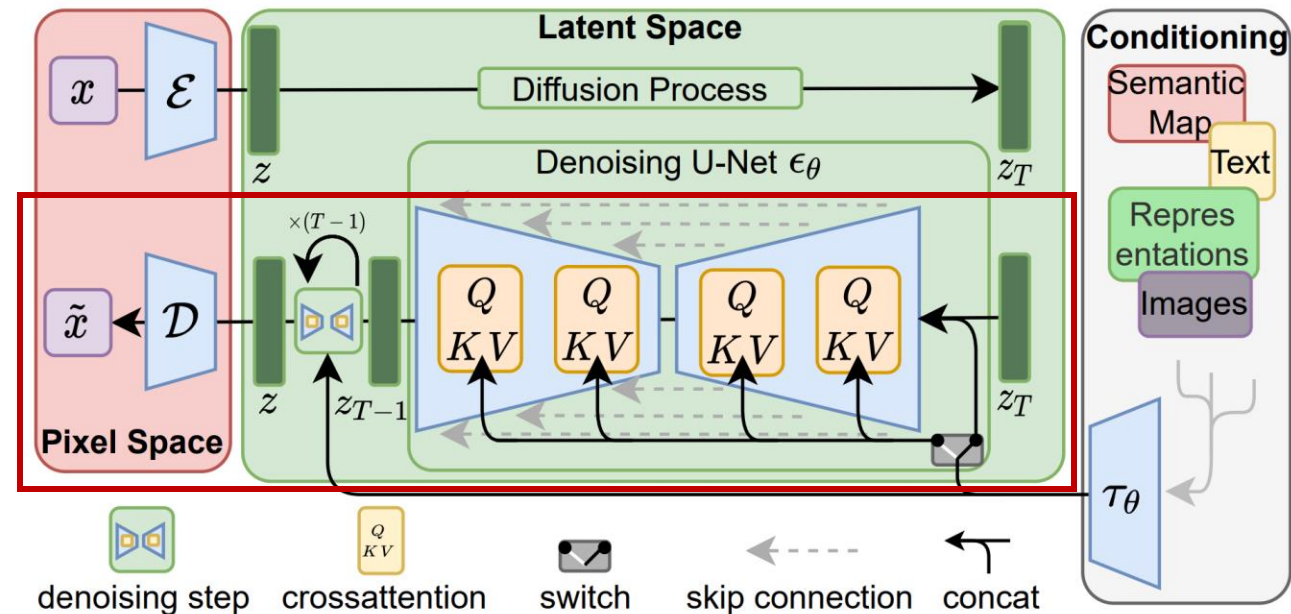
Cross-attention Modules

conditional representation such as textual prompt



# Stable Diffusion: Two-Stage Image Synthesis

- (Stage 2) **Generative Modeling of Latent Representations:** the denoising happens in this compressed latent space given by Stage 1.
- Sampling of latent code:  $z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_{\phi}(z_t, t, \tau_{\theta}(y)) \right) + \sigma_t \epsilon_t$  where  $y$  is the conditioning (e.g., textual prompt, semantic segmentation).
- Generating (decoding) the image:  $\tilde{x} = D(z_0)$





# Unconditional Generation

One can train a SD on a dataset of a visual domain without imposing extra conditions. Under such scenarios, the training of SD is similar to DDPM except that SD learns the denoising in quantized latent space.

Here are samples from five unconditional stable diffusion models, each trained on a separate dataset.

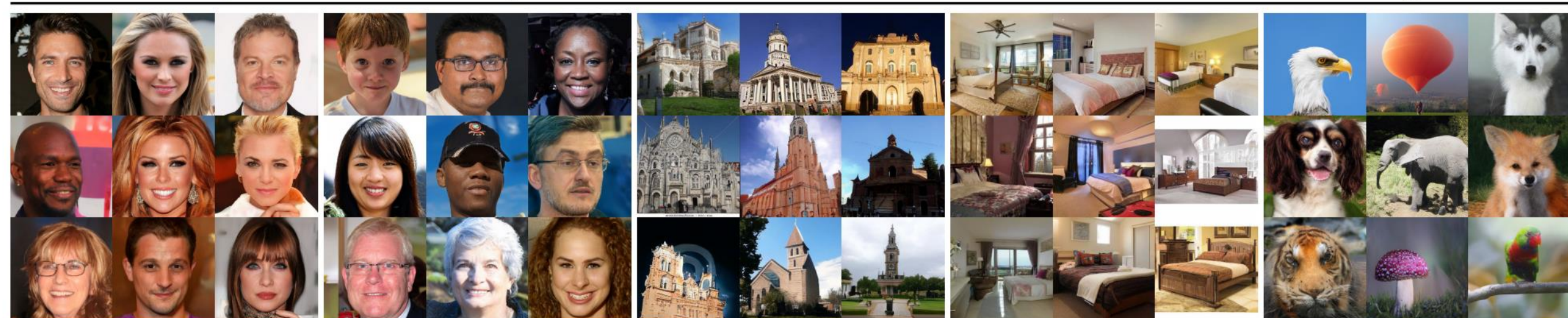
CelebAHQ

FFHQ

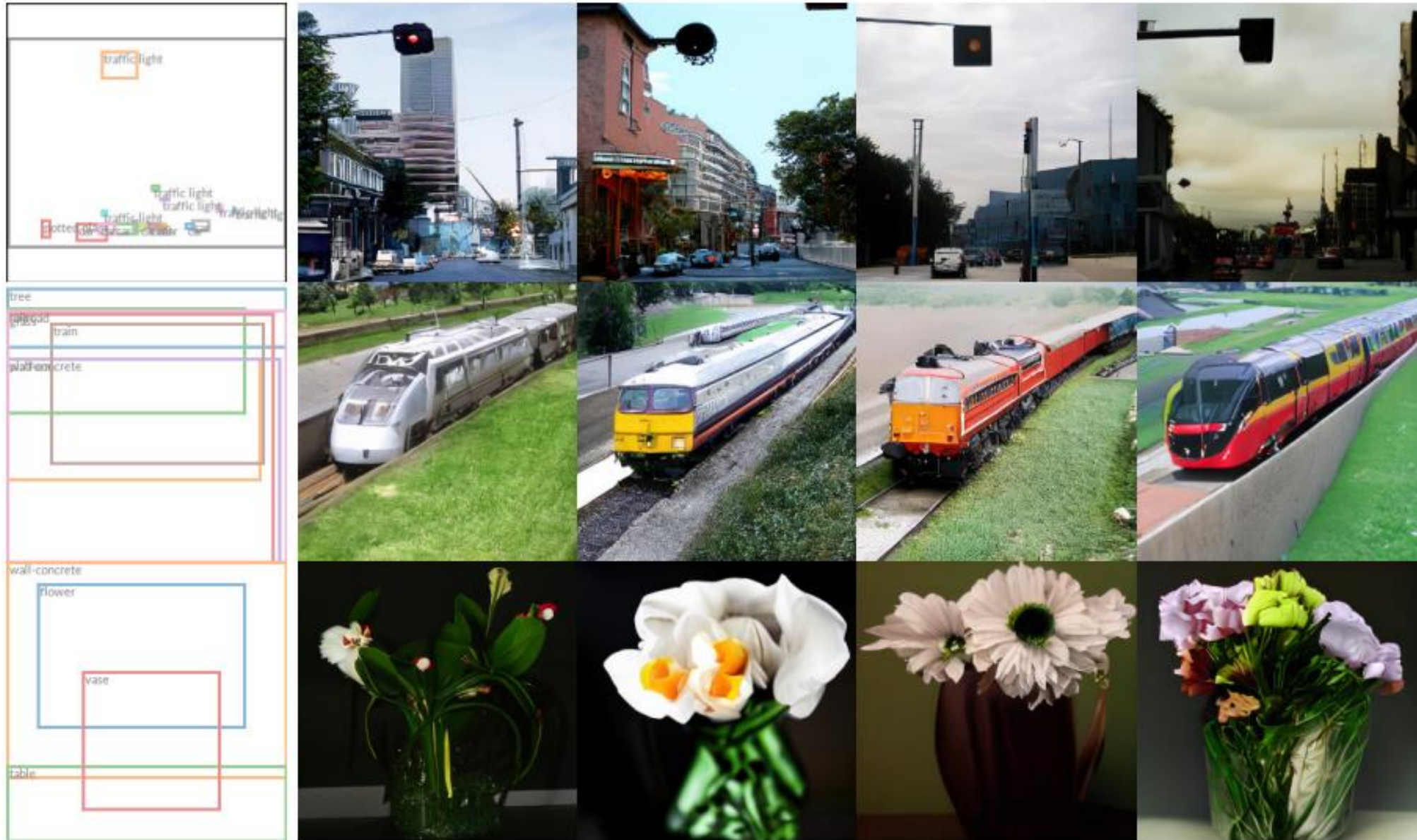
LSUN-Churches

LSUN-Beds

ImageNet



# Image Generation Conditioned on Layout



# Image Generation Conditioned on Text

Text-to-Image Synthesis on LAION. 1.45B Model.

'A zombie in the style of Picasso'

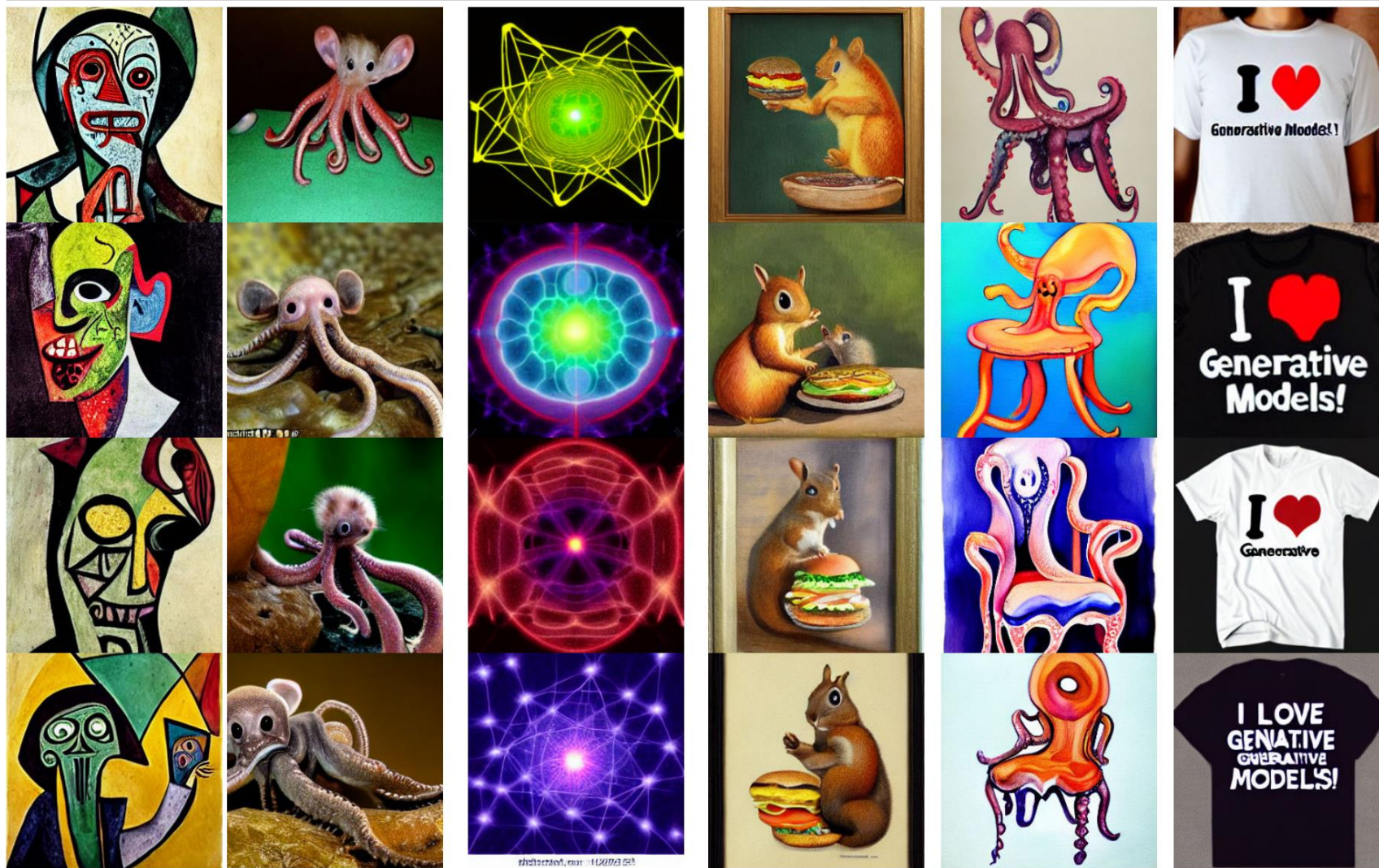
'An image of an animal half mouse half octopus'

'An illustration of a slightly conscious neural network.'

'A painting of a squirrel eating a burger.'

'A watercolor painting of a chair that looks like an octopus.'

'A shirt with the inscription: "I love generative models!".'



Details of this 1.45B Model  
**Training:** KL-regularized  
**Text Conditioner:** CLIP-like  
**Dataset:** LAION-400M

# Hyper-Parameters for Implementation

Task	Text-to-Image	Layout-to-Image	Class-Label-to-Image	Super Resolution	Inpainting	Semantic-Map-to-Image	
Dataset	LAION	OpenImages	COCO	ImageNet	ImageNet	Places	Landscapes
$f$	8	4	8	4	4	4	8
$z$ -shape	$32 \times 32 \times 4$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$32 \times 32 \times 4$
$ \mathcal{Z} $	-	8192	16384	8192	8192	8192	16384
Diffusion steps	1000	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear	linear
Model Size	1.45B	306M	345M	395M	169M	215M	215M
Channels	320	128	192	192	160	128	128
Depth	2	2	2	2	2	2	2
Channel Multiplier	1,2,4,4	1,2,3,4	1,2,4	1,2,3,5	1,2,2,4	1,4,8	1,4,8
Number of Heads	8	1	1	1	1	1	1
Dropout	-	-	0.1	-	-	-	-
Batch Size	680	24	48	1200	64	128	48
Iterations	390K	4.4M	170K	178K	860K	360K	360K
Learning Rate	1.0e-4	4.8e-5	4.8e-5	1.0e-4	6.4e-5	1.0e-6	4.8e-5
Conditioning	CA	CA	CA	CA	concat	concat	concat
(C)A-resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	-	-	-
Embedding Dimension	1280	512	512	512	-	-	-
Transformer Depth	1	3	2	1	-	-	-

# Stable Diffusion as the Foundation Model

Representatives of generative vision works that take SD as the backbone:

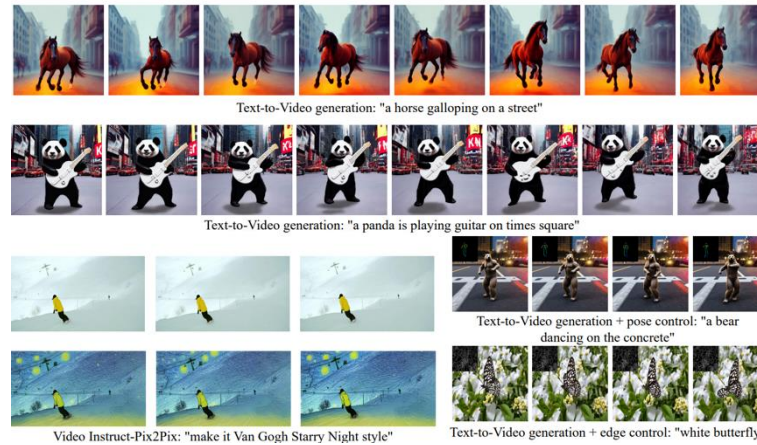
## DREAMFUSION: TEXT-TO-3D USING 2D DIFFUSION

Ben Poole<sup>1</sup>, Ajay Jain<sup>2</sup>, Jonathan T. Barron<sup>1</sup>, Ben Mildenhall<sup>1</sup>  
<sup>1</sup>Google Research, <sup>2</sup>UC Berkeley  
 {pooleb, barron, bmild}@google.com, ajayj@berkeley.edu



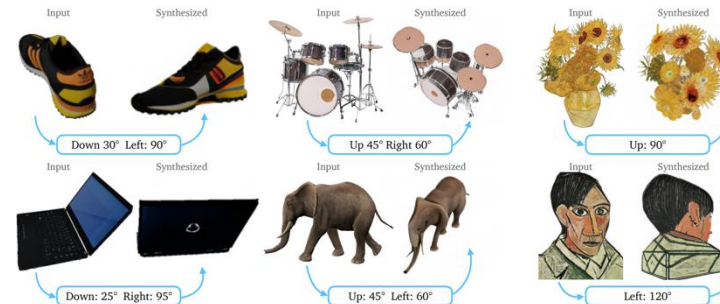
## Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators

Levon Khachatryan<sup>1\*</sup>, Andranik Movsisyan<sup>1\*</sup>, Vahram Tadevosyan<sup>1\*</sup>, Roberto Henschel<sup>1\*</sup>,  
 Zhangyang Wang<sup>1,2</sup>, Shant Navasardyan<sup>1</sup>, Humphrey Shi<sup>1,3,4</sup>  
<sup>1</sup>Picsart AI Resarch (PAIR) <sup>2</sup>UT Austin <sup>3</sup>U of Oregon <sup>4</sup>UIUC  
<https://github.com/Picsart-AI-Research/Text2Video-Zero>



## Zero-1-to-3: Zero-shot One Image to 3D Object

Ruoshi Liu<sup>1</sup>, Rundi Wu<sup>1</sup>, Basile Van Hoorick<sup>1</sup>, Pavel Tokmakov<sup>2</sup>, Sergey Zakharov<sup>2</sup>, Carl Vondrick<sup>1</sup>  
<sup>1</sup> Columbia University <sup>2</sup> Toyota Research Institute  
[zero123.cs.columbia.edu](https://zero123.cs.columbia.edu)



## SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn

Jonas Müller, Joe Penna, Robin Rombach

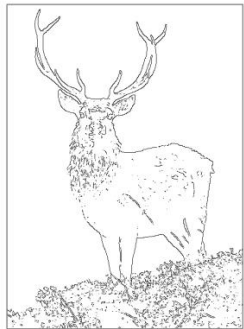
Stability AI, Applied Research

Code: <https://github.com/Stability-AI/generative-models> Model weights: <https://huggingface.co/stabilityai/>



# Adding Conditional Control to Text-to-Image Diffusion Models

- **ControlNet** is a neural network architecture to add spatial conditioning to large pre-trained text-to-image diffusion models (e.g., Stable Diffusion).
- ControlNet allows users to add conditions like Canny edges or human pose to control the text-to-image synthesis.



Input Canny edge



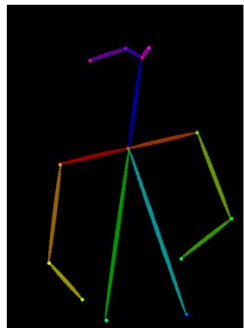
Default



"masterpiece of fairy tale, giant deer, golden antlers"



"..., quaint city Galic"



Input human pose



Default



"chef in kitchen"

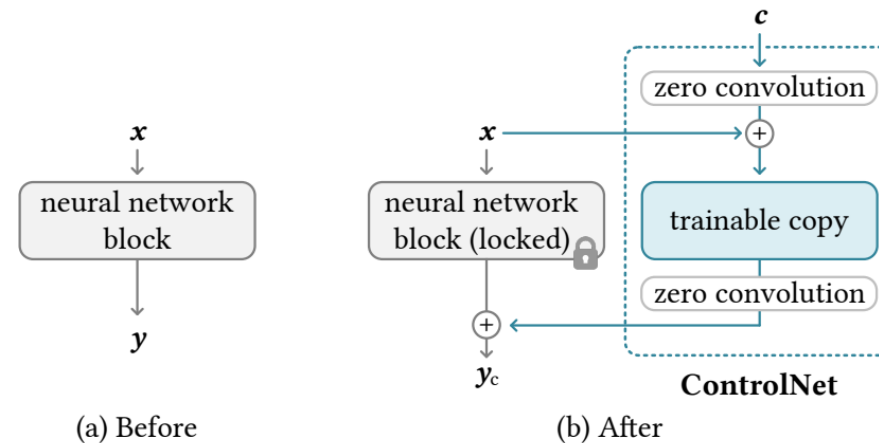


"Lincoln statue"



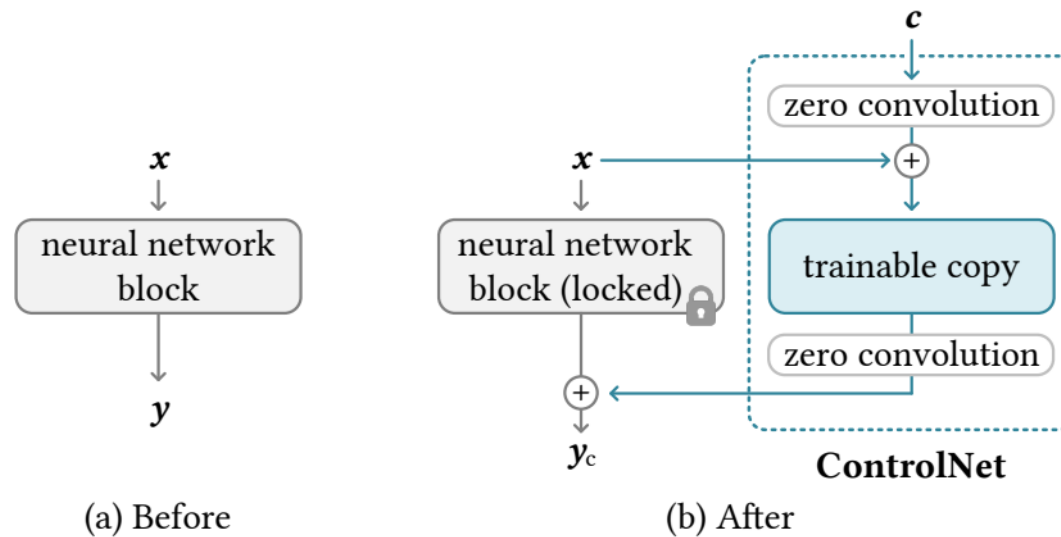
# Key Intuitions of ControlNet

- Core idea:
  - ControlNet **freezes the parameters  $\Theta$**  of the original generative neural block  $F_{\Theta}$  and simultaneously clones the block to **a trainable copy with parameters  $\Theta_c$** .
  - The trainable copy takes an external conditioning vector  $c$  as input.
- Why it works well:
  - The locked parameters of a large model **preserve the production-ready information** with billions of images, while the trainable copy establishes a flexible learning paradigm for **handling diverse input conditions**.



# MLP Layers with Zero Initialization

- The trainable copy is connected to the locked model with zero convolution layers, denoted  $Z(\cdot; \cdot)$ .
- Specifically,  $Z(\cdot; \cdot)$  is a convolution layer or MLP layer with both weight and bias initialized to zeros.



$$y_c = \mathcal{F}(x; \Theta) + \mathcal{Z}(\mathcal{F}(x + \mathcal{Z}(c; \Theta_{z1}); \Theta_c); \Theta_{z2})$$



# Gradient Calculation of A Zero Initialization Layer

- Consider a linear layer with weight  $W$  and bias  $B$ , at spatial position  $p$  and channel-wise index  $i$ . Given an input map  $I \in R^{h \times w \times c}$ , the forward pass can be written as

$$\mathcal{Z}(I; \{W, B\})_{p,i} = B_i + \sum_j^c I_{p,j} W_{i,j}$$

- A zero Initialization layer is initialized with  $W = 0$  and  $B = 0$ . The gradient for anywhere that  $I_{p,i} \neq 0$  is that:

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{Z}(I; \{W, B\})_{p,i}}{\partial B_i} = 1, \\ \frac{\partial \mathcal{Z}(I; \{W, B\})_{p,i}}{\partial I_{p,i}} = \sum_j^c W_{i,j} = 0, \\ \frac{\partial \mathcal{Z}(I; \{W, B\})_{p,i}}{\partial W_{i,j}} = I_{p,j} \neq 0. \end{array} \right.$$

- We see that the gradients for updating weight and bias are not always zero.

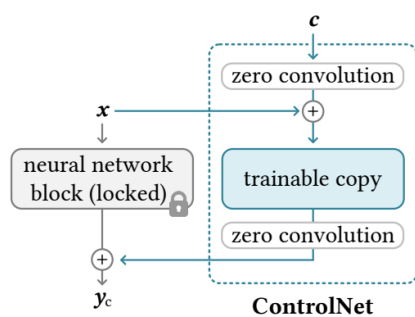
# ControlNet for Stable Diffusion

- The ControlNet structure is applied to each encoder level of the U-Net.

- An analogy with Residual Learning:

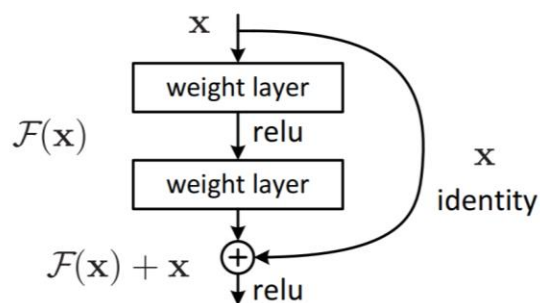
Original SD:  $Y = \text{Generator}(X)$

ControlNet:  $Y = \text{Generator}(X) + \text{ControlNet}(X, C)$

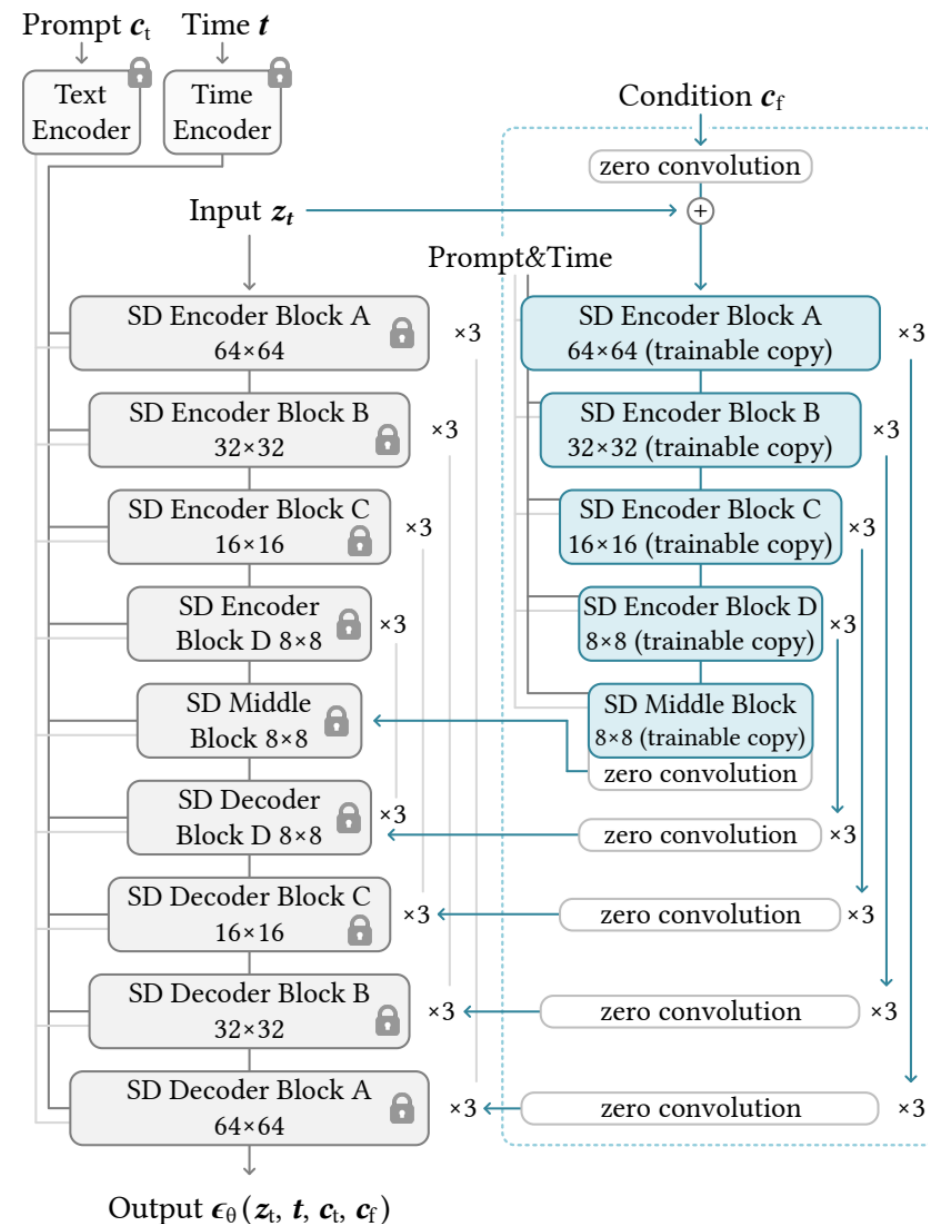


(b) After

ControlNet

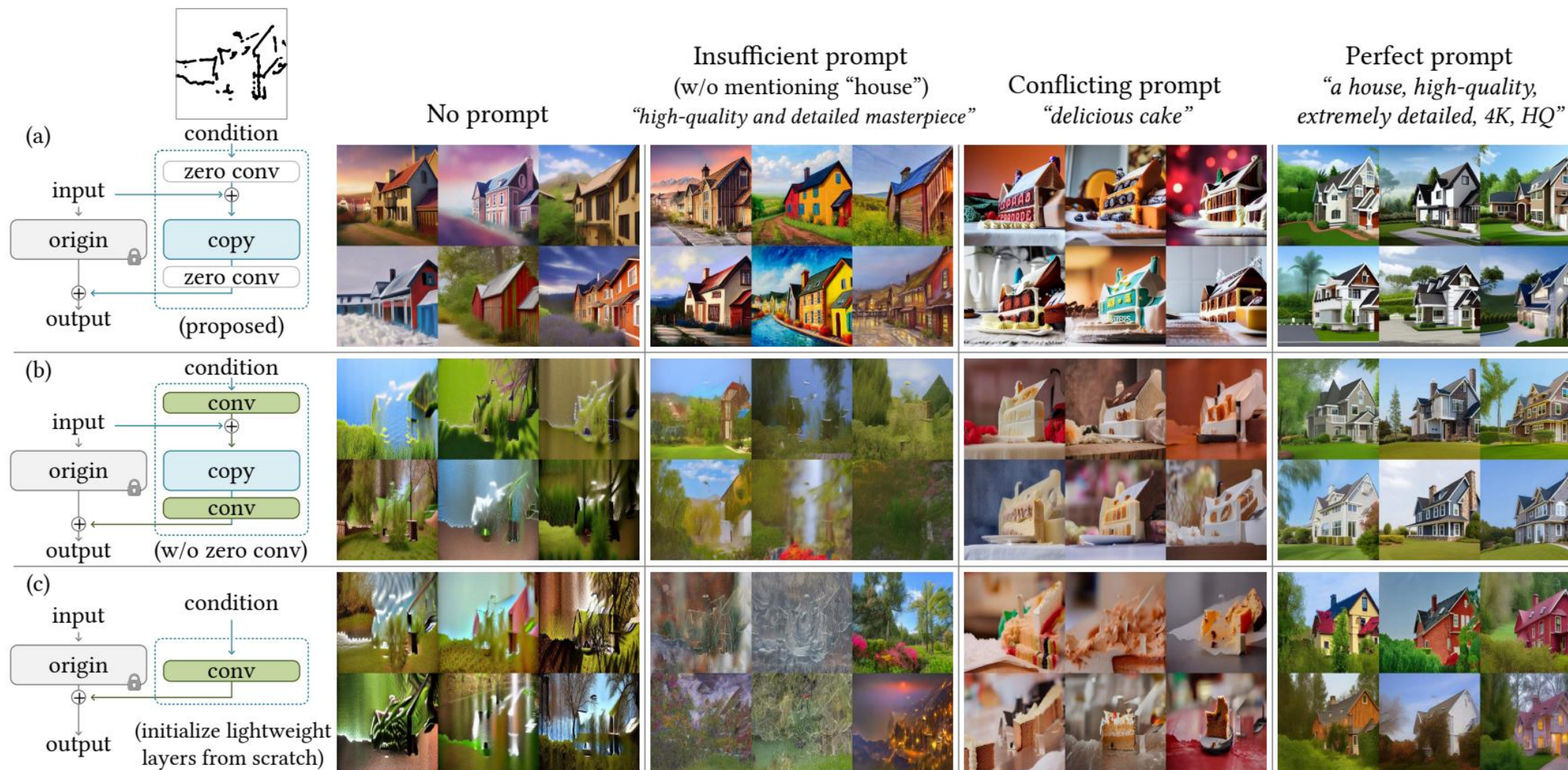


Residual Learning  
[He et al., 2015]



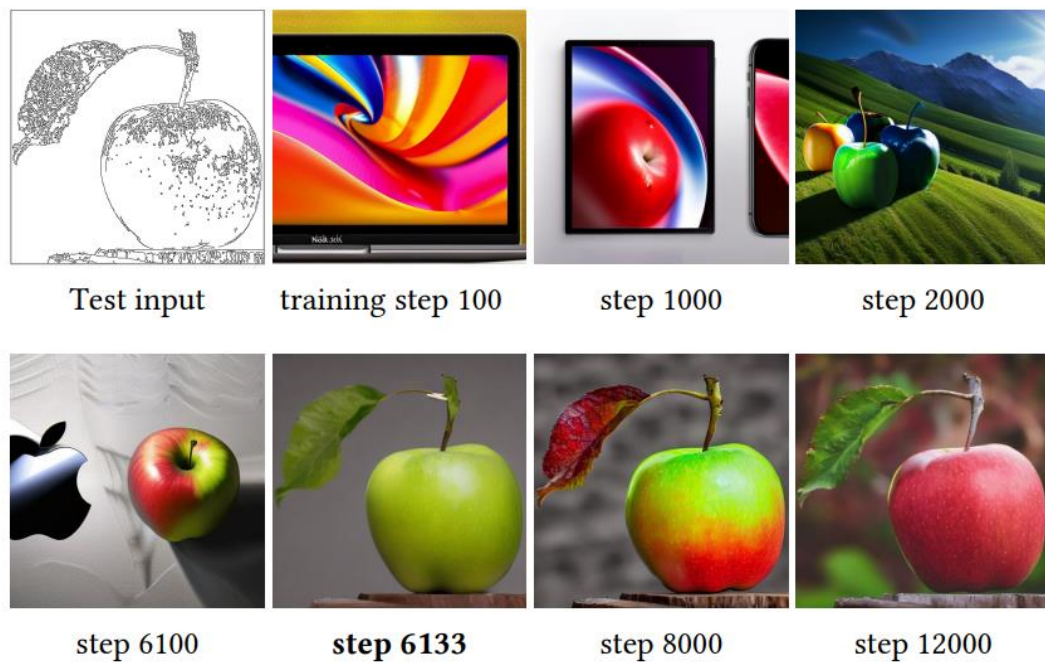
# Does Zero Initialization Help?

- The ablation study indicates that zero initialization is beneficial for ControlNet.



# Training Results

- ControlNet supports conditioning in multiple modalities.
- Examples: Canny edge, human pose.



(a) Input Canny map    (b) W/o CFG    (c) W/o CFG-RW    (d) Full (w/o prompt)

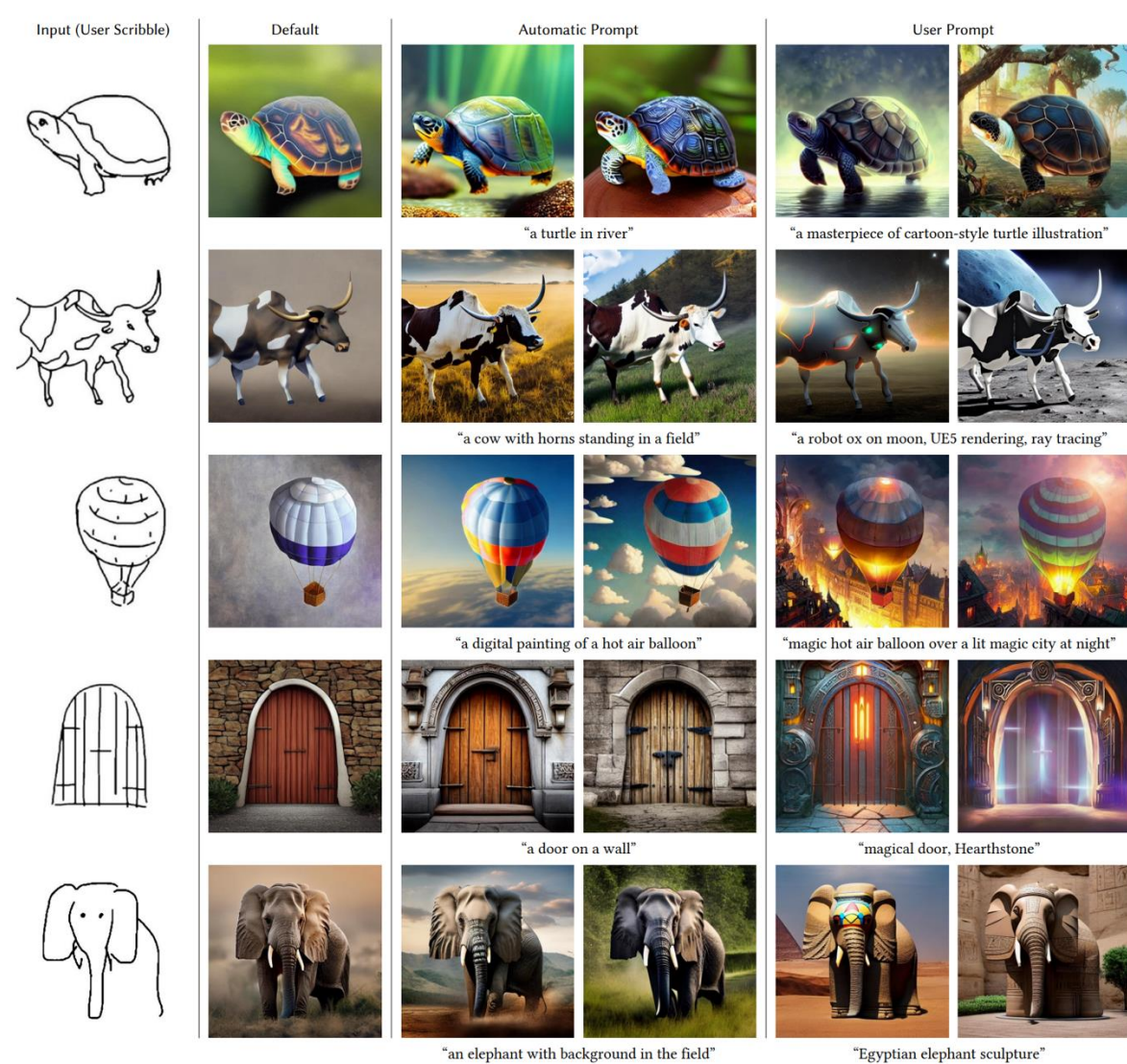
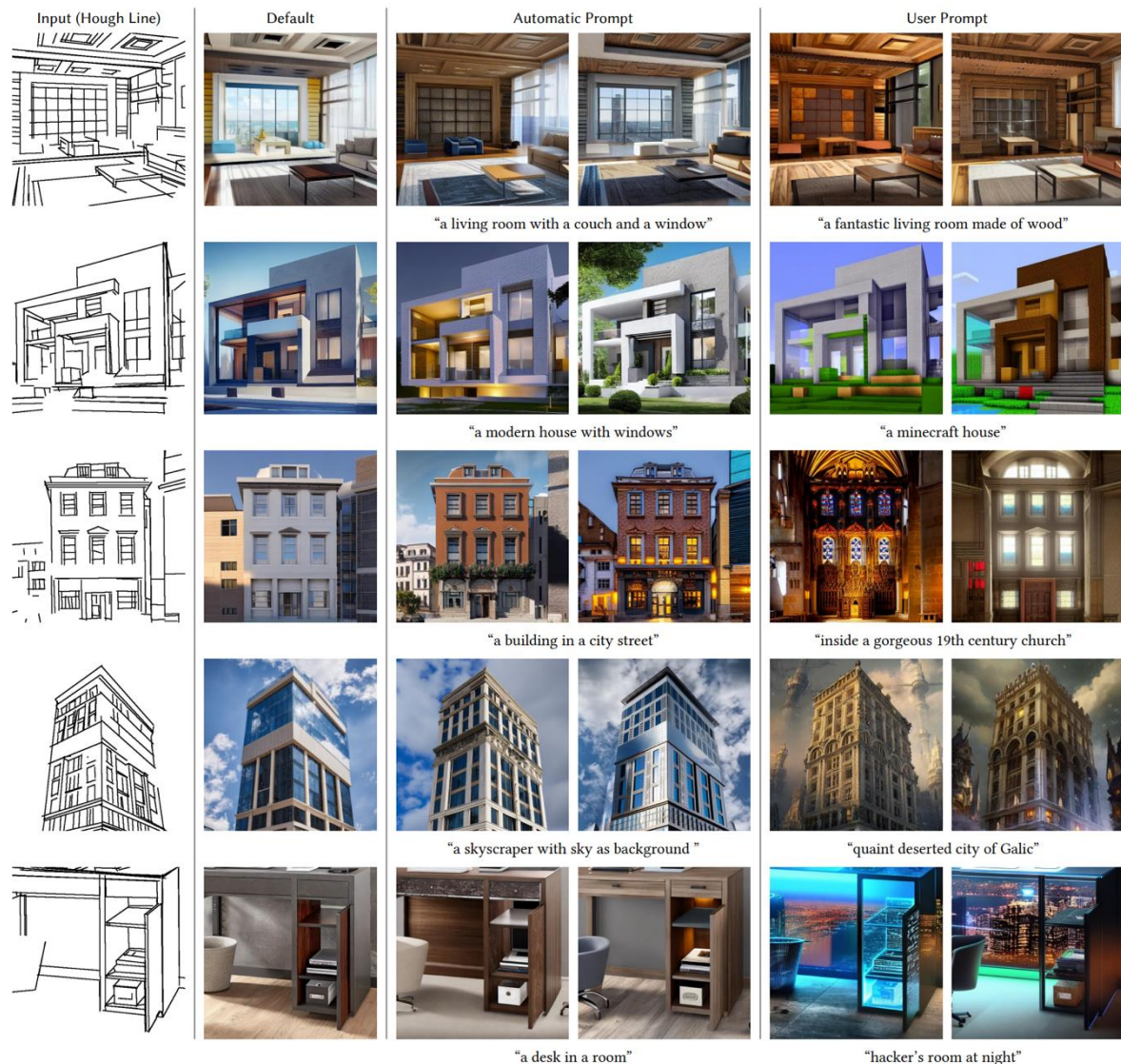
Figure 5: Effect of Classifier-Free Guidance (CFG) and the proposed CFG Resolution Weighting (CFG-RW).



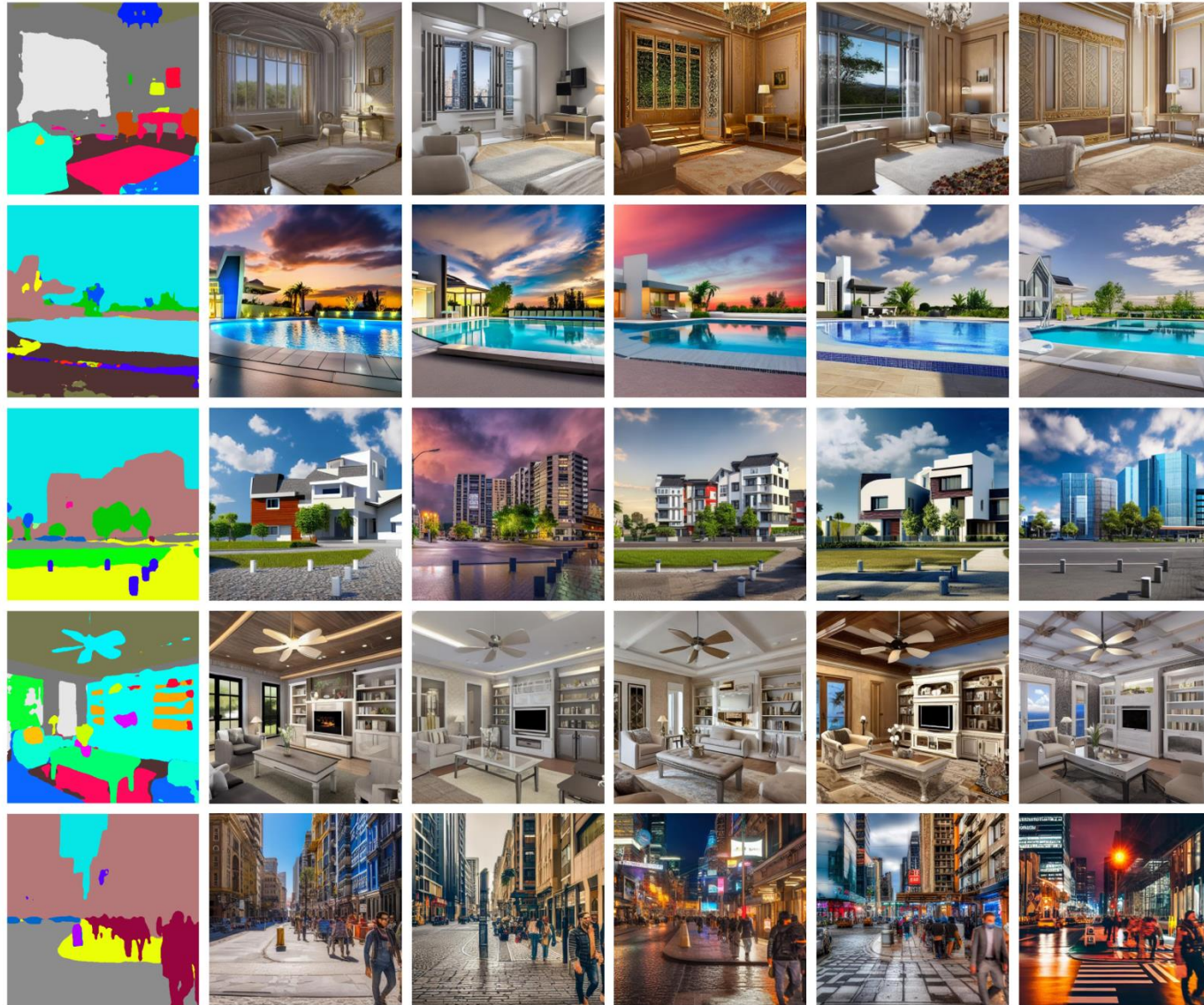
Multiple condition (pose&depth)    “boy”    “astronaut”

Figure 6: Composition of multiple conditions. We present the application to use depth and pose simultaneously.

# Hough Lines and User Scribble



# Segmentation and Human Pose



"Michael Jackson's concert"

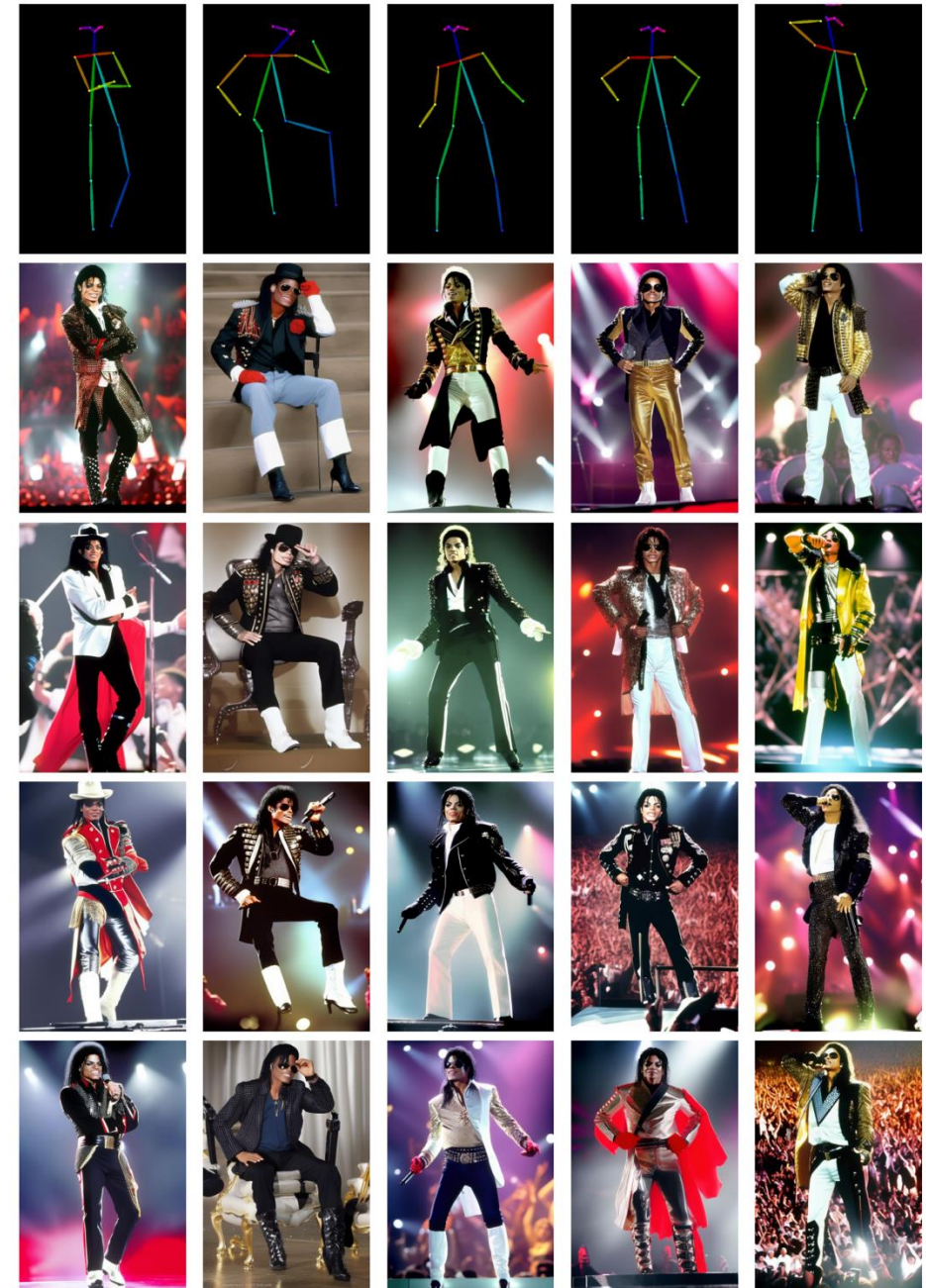


Figure 15: Controlling Stable Diffusion with ADE20K segmentation map. All results are achieved with default prompt. See also the Appendix for source images for semantic segmentation map extraction.

# Conclusion for Lectures in Diffusion Models

In the last lecture, we elaborate on the derivation of Diffusion Models from a VAE perspective. In this lecture, we introduce two mainstream applications of Diffusion Models with multimodal conditions:

- **Stable Diffusion**

1. Denoising in the latent space enables spatial **complexity reduction** and **detail preservation**.
2. Introducing cross-attention layers enables **conditional input such as texts**.

- **ControlNet**

1. Enabling fine-grained control through **versatile conditions** (e.g., sketches, outlines, keypoints)
2. Preserving **pre-trained model knowledge** while extending functionality.