

# Deep Generative Models: Latent Diffusion Models

Fall Semester 2025

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



# Outline

- Markov Hierarchical Variational Auto Encoders (MHVAEs)
  - Autoregressive Encoder and Autoregressive Decoder of an MHVAE
  - Derivation of the ELBO of an MHVAE
- Diffusion Models: MHVAEs with a Linear Gaussian Autoregressive Latent Space
  - Forward Diffusion Process
  - Reverse Diffusion Process
  - ELBO for Diffusion Models as a particular case of the ELBO for MHVAEs
- Implementation Details: UNet architecture, Training and Sampling Strategies
- **Application of Diffusion Models**
  - Text-Conditioned Diffusion Model: Stable Diffusion
  - Multimodal Control for Consistent Synthesis: ControlNet
  - Image Editing: DDIM, P2P

# 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 on limited computational resources, while retaining quality and flexibility, Stable Diffusion performs **denoising in the latent space** of powerful pretrained autoencoders.
- Benefits of 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**.

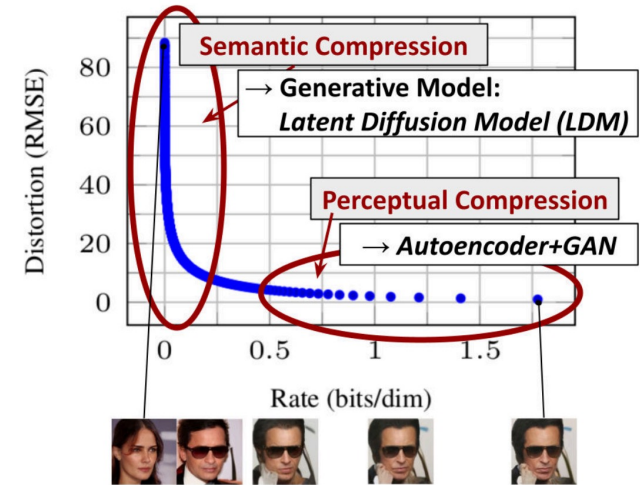
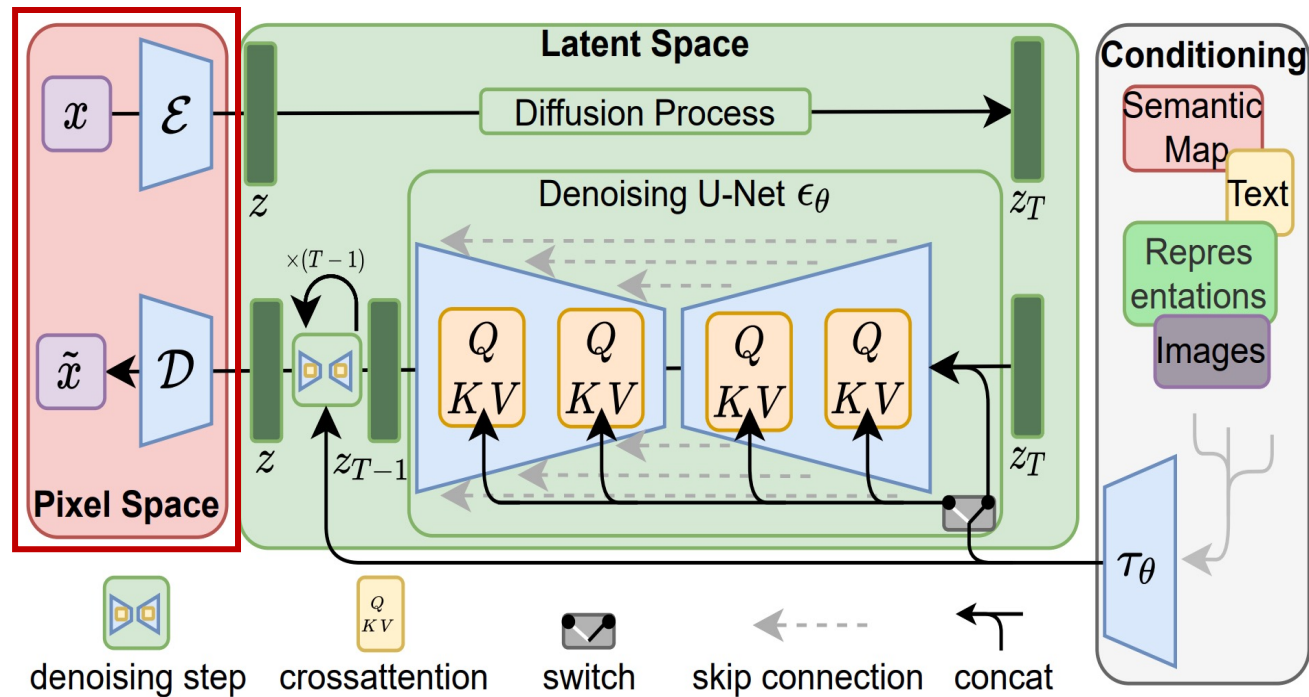


Figure 2. Illustrating perceptual and semantic compression: Most bits of a digital image correspond to imperceptible details. While DMs allow to suppress this semantically meaningless information by minimizing the responsible loss term, gradients (during training) and the neural network backbone (training 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

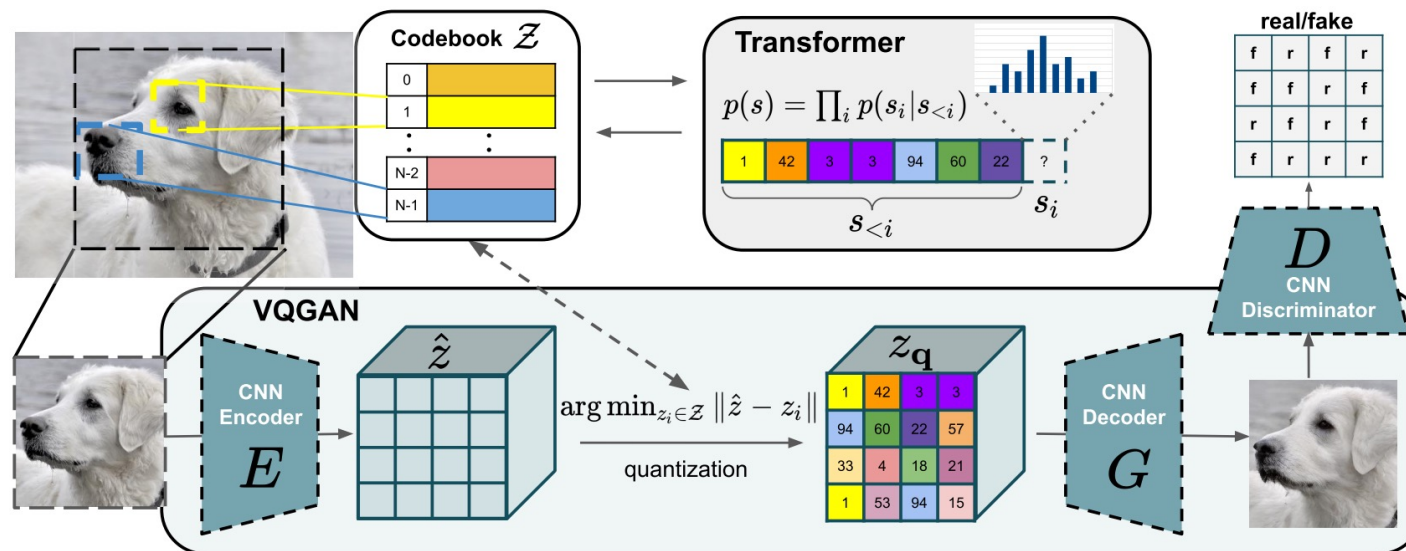
- **(Stage 1) Perceptual Image Compression:** The SD framework uses a pre-trained VAE to map data into a low-dimensional space and back to the pixel space.
- Compared to the high-dimensional pixel space, the low-dimensional latent space is more suitable for likelihood-based generative models, as
  -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



# Stable Diffusion: Two-Stage Image Synthesis

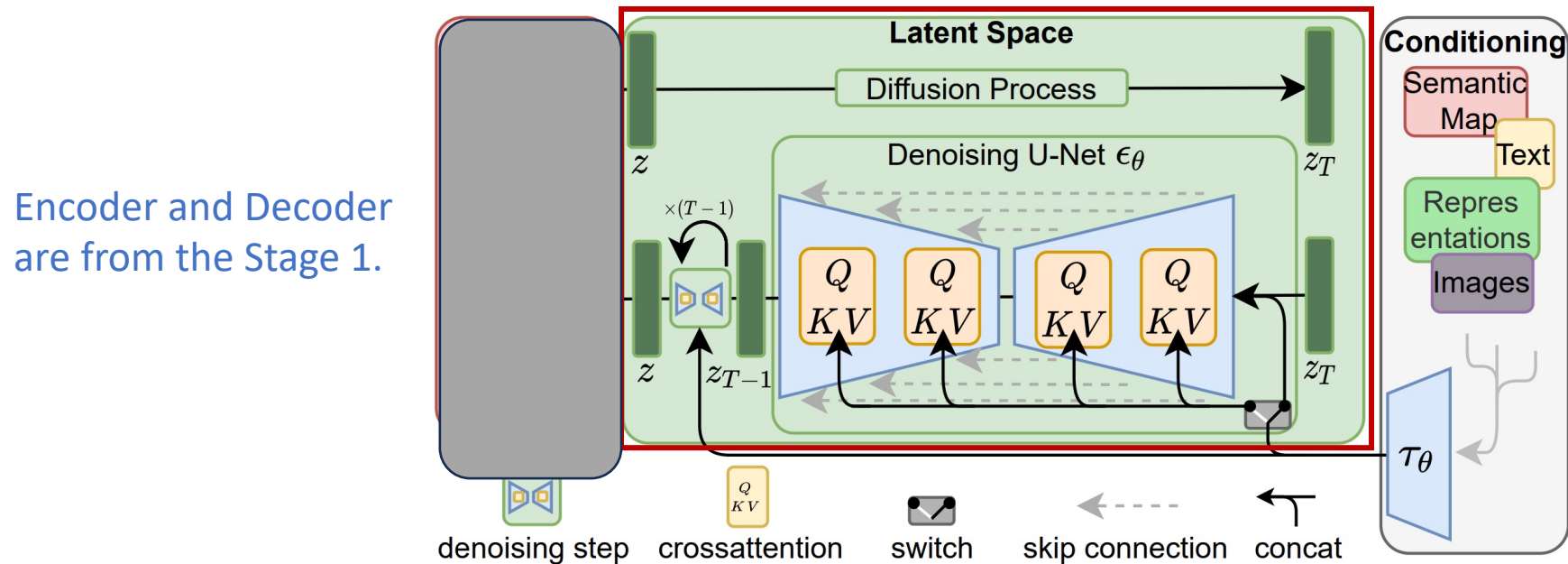
- **(Stage 1) Perceptual Image Compression:** The VAE is trained by a combination of a reconstruction loss and a patch-based adversarial loss.
- It deploys a VQGAN to learn a codebook of context-rich visual parts, whose composition is modeled with an autoregressive transformer architecture.

$$L_{VQGAN} = \min_{E,G} \max_D \left( \underbrace{L_{rec}(x, G(E(x)))}_{\text{image reconstruction}} - \underbrace{L_{synthetic}(D(G(E(x)))) + L_{real} D(x)}_{\text{differentiate original images from reconstructions}} + L_{reg}(x; E, G) \right)$$



# Stable Diffusion: Two-Stage Image Synthesis

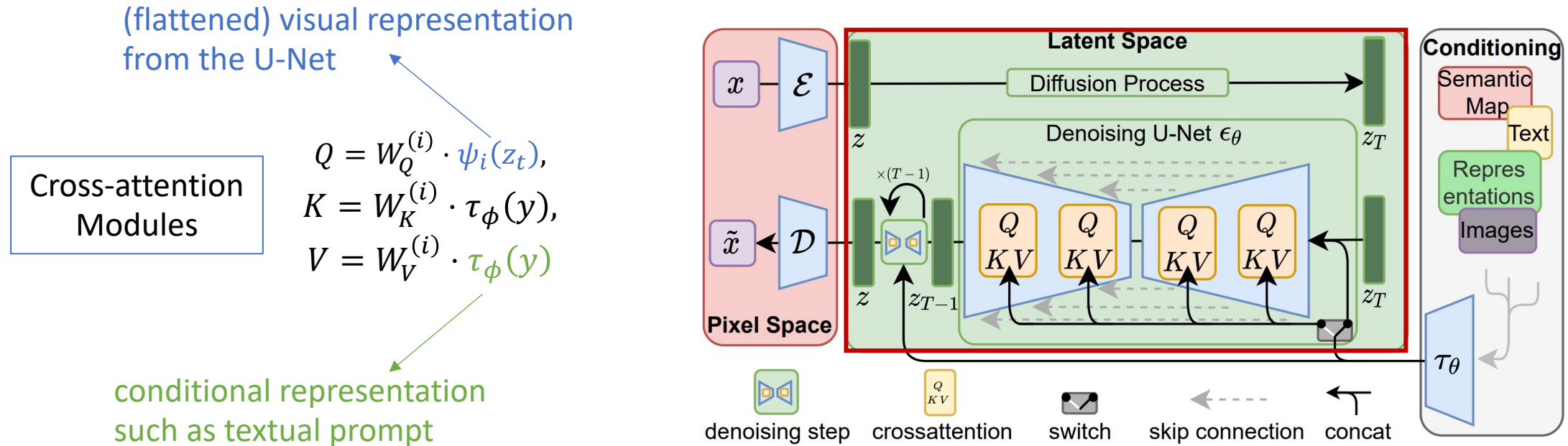
- **(Stage 2) Denoising Latent Representations:** the denoising process happens in the compressed latent space of Stage 1.
- Recall that DDPM denoises the input in the image space, while the SD performs denoising for the latent code.
- The forward process deterministically adds  $T$  Gaussian noises to the original latent code  $z$ , and **the reverse process learns to denoise**.



# Stable Diffusion: Two-Stage Image Synthesis

- The neural backbone of SD is a time-conditional U-Net  $\epsilon_\theta(z_t, t, \tau_\phi(y))$  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  $\tau_\phi(y)$  such as textual prompts or semantic segmentations.

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



# Stable Diffusion: Two-Stage Image Synthesis

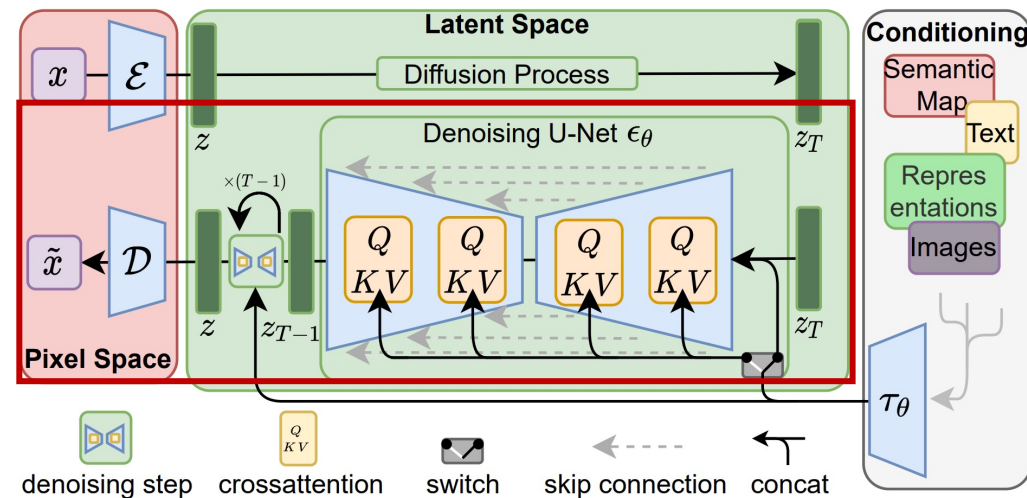
- **(Stage 2) Generative Modeling of Latent Representations:** denoising happens in the compressed latent space given by Stage 1.

- Sampling of the latent code

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( z_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(z_t, t, \tau_{\phi}(y)) \right) + \sqrt{\beta_t} \epsilon_t$$

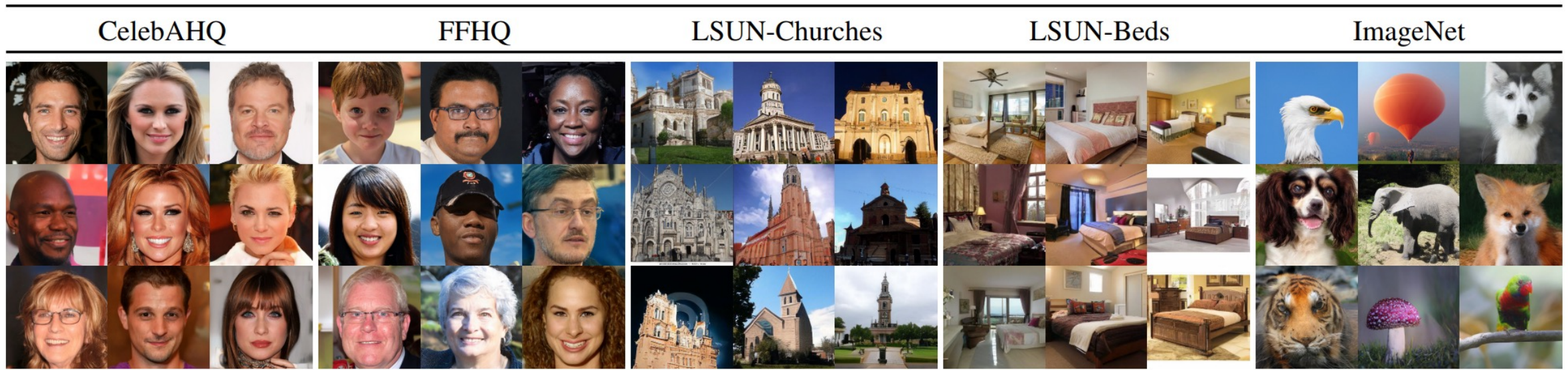
where  $y$  is the conditioning (e.g., textual prompt, semantic segmentation).

- Generating (decoding) the image is performed using  $\tilde{x} = D(z_0)$ .



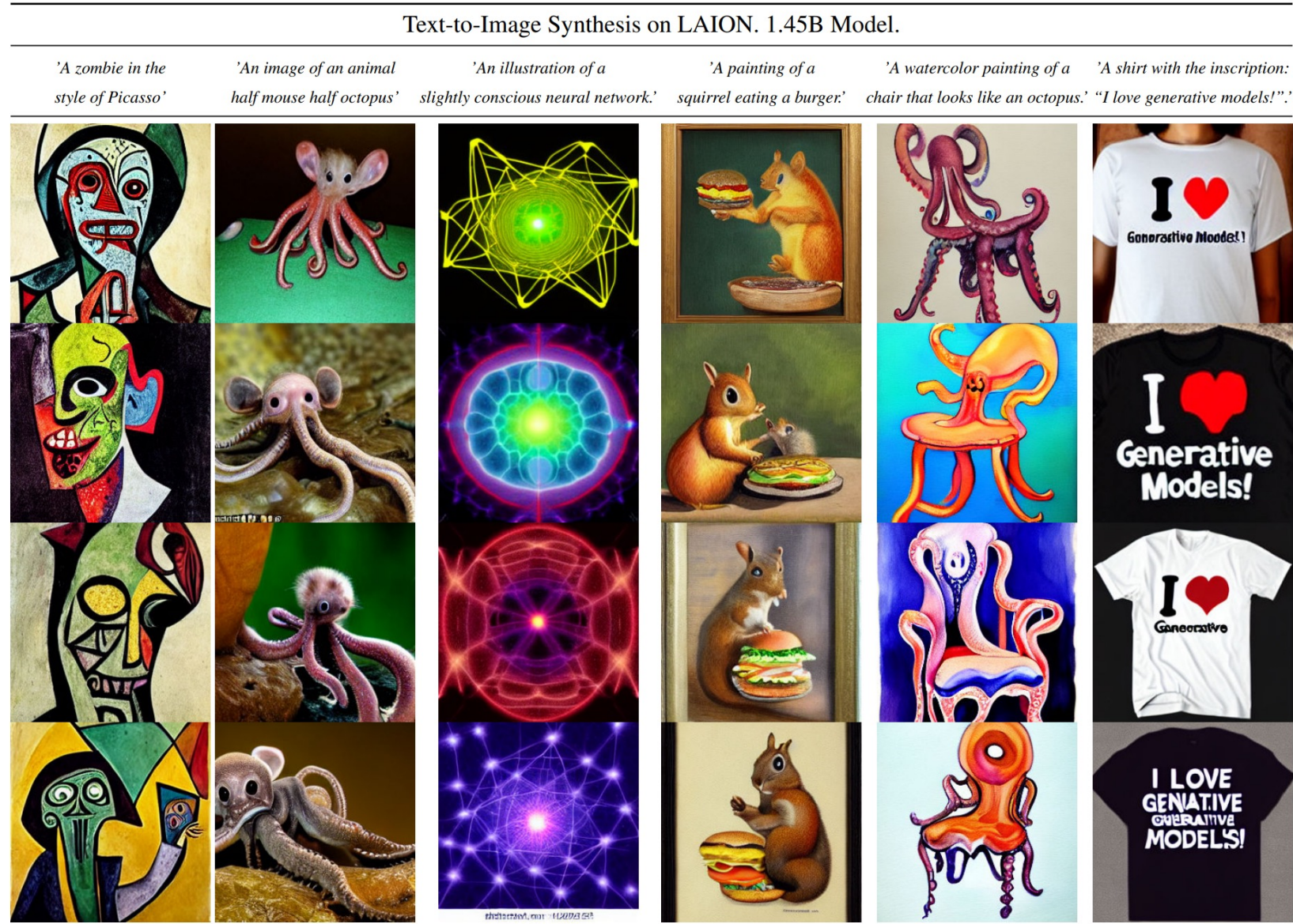
# Unconditional Generation

- One can train a SD on a dataset of a visual domain without imposing extra conditions.
- In this case, the training of SD is similar to DDPM except that SD learns the denoising in a **quantized latent space**.
- Exemplary samples are provided from five unconditional stable diffusion models, each trained on a separate dataset.



# Image Generation Conditioned on Text

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



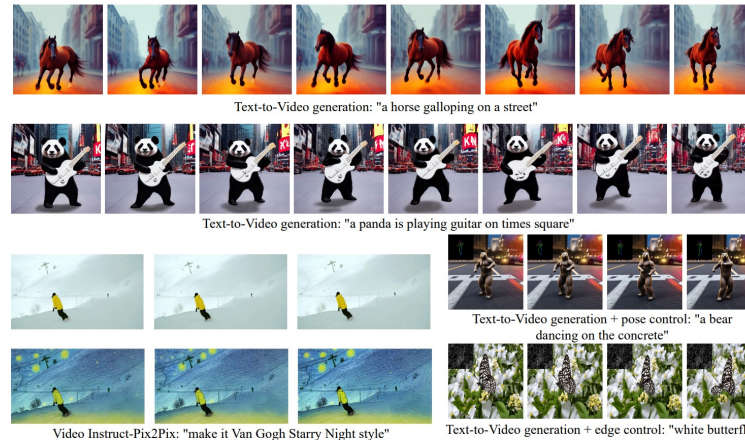
### Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators

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 Research (PAIR) <sup>2</sup>UT Austin <sup>3</sup>U of Oregon <sup>4</sup>UIUC

<https://github.com/Picsart-AI-Research/Text2Video-Zero>

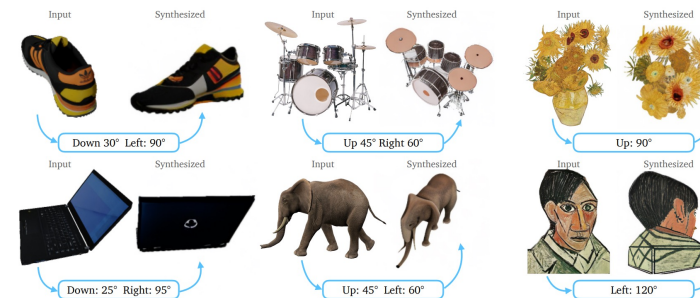


### Zero-1-to-3: Zero-shot One Image to 3D Object

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](mailto: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/>

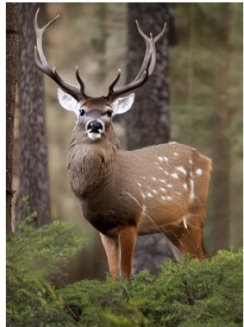


# Adding Conditional Control to Text-to-Image Diffusion Models

- **ControlNet** is a neural network architecture that adds 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



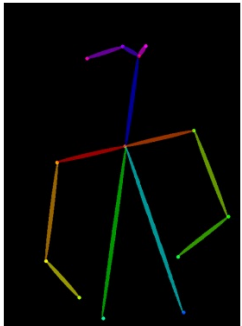
Default



“masterpiece of fairy tale, giant deer, golden antlers”



“..., quaint city Galic”



Input human pose



Default



“chef in kitchen”

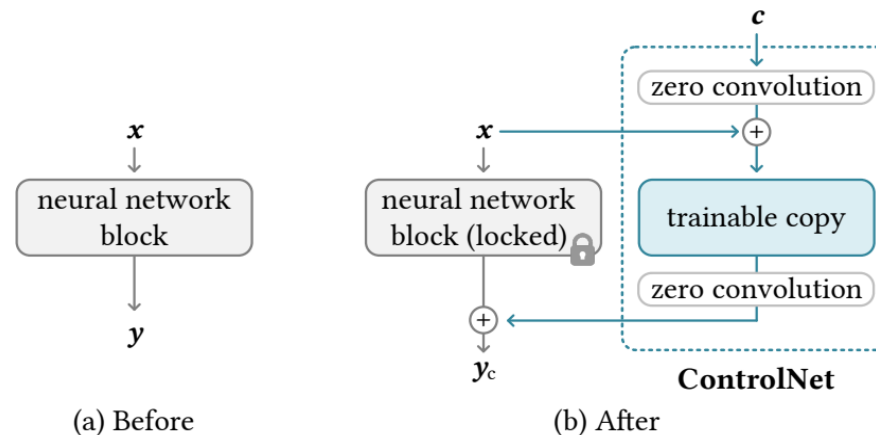


“Lincoln statue”



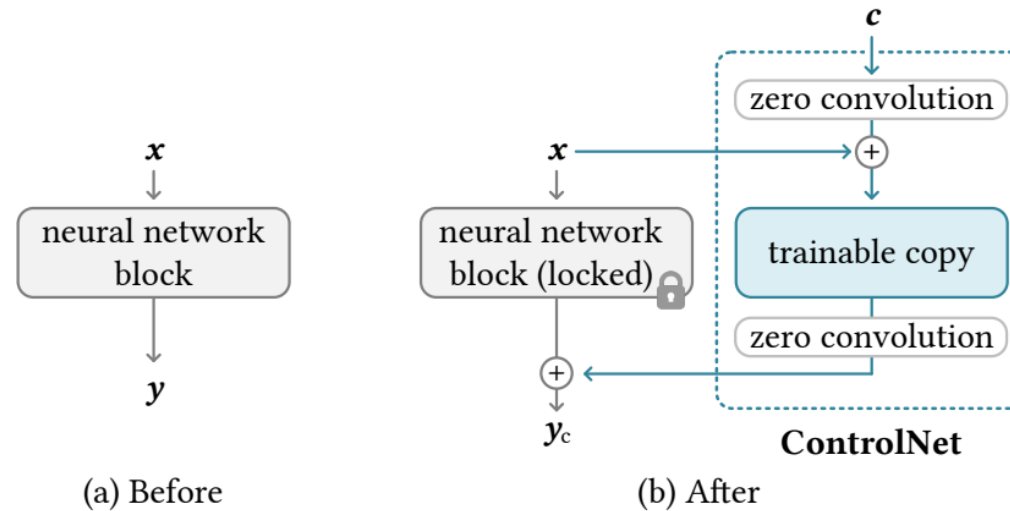
# Key Intuitions of ControlNet

- 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 or MLP layer with both weight and bias initialized to zeros.



$$y_c = \mathcal{F}(x; \Theta) + \mathcal{Z}(\mathcal{F}(x + \mathcal{Z}(c; \Theta_{Z_1}); \Theta_c); \Theta_{Z_2})$$

# Gradient Calculation of A Zero Initialization Layer

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

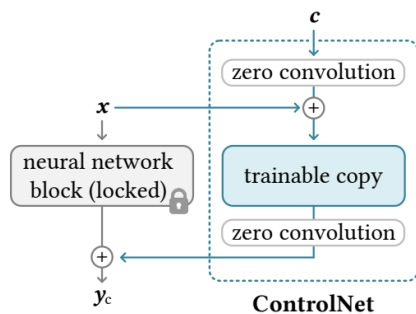
$$\mathcal{Z}(x; \{W, B\})_{p,i} = B_i + \sum_j^c x_{p,j} W_{i,j}$$

- Each layer is initialized with  $W = 0$  and  $B = 0$ . However, the gradient at a point such that  $x_{p,i} \neq 0$  need not be zero, allowing for the training of these weights:

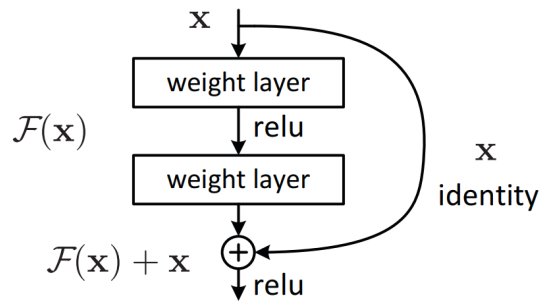
$$\left\{ \begin{array}{l} \frac{\partial \mathcal{Z}(x; \{W, B\})_{p,i}}{\partial B_i} = 1, \\ \frac{\partial \mathcal{Z}(x; \{W, B\})_{p,i}}{\partial x_{p,i}} = \sum_j^c W_{i,j} = 0 \\ \frac{\partial \mathcal{Z}(x; \{W, B\})_{p,i}}{\partial W_{i,j}} = x_{p,j} \neq 0 \end{array} \right.$$

# ControlNet for Stable Diffusion

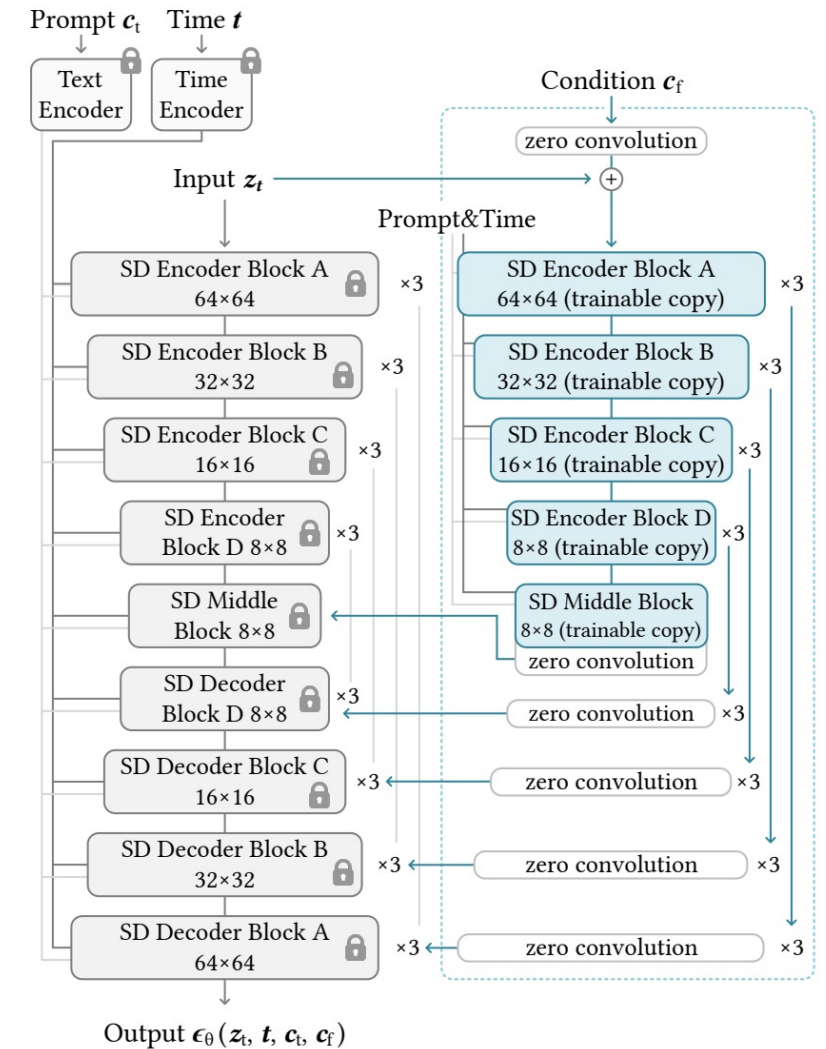
- The ControlNet structure is applied to each encoder level of the U-Net.
- An analogy with Residual Learning:  
**Original SD:**  $y = \text{Generator}(x)$   
**ControlNet:**  $y = \text{Generator}(x) + \text{ControlNet}(x, c)$



ControlNet



Residual Learning  
[He et al., 2015]





# Training Results

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

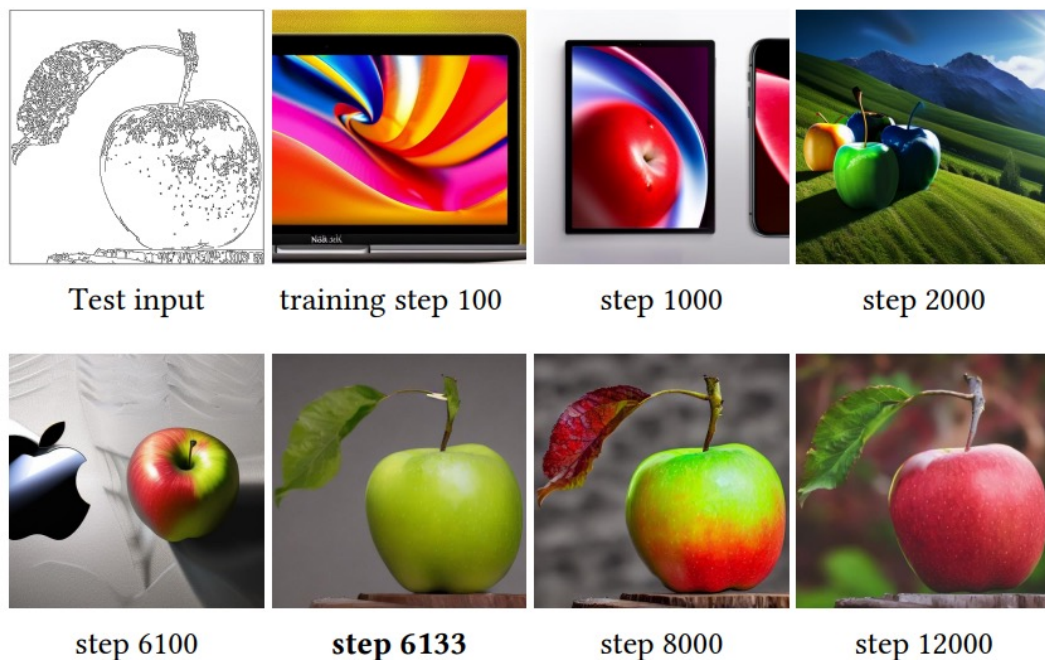


Figure 5: Effect of Classifier-Free Guidance (CFG) and the proposed CFG Resolution Weighting (CFG-RW).

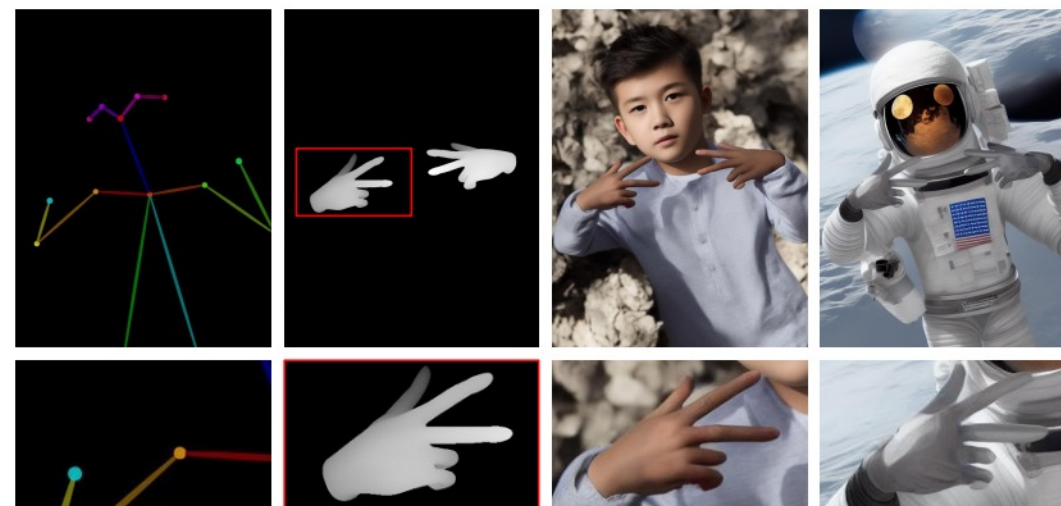


Figure 6: Composition of multiple conditions. We present the application to use depth and pose simultaneously.

# Hough Lines and User Scribble

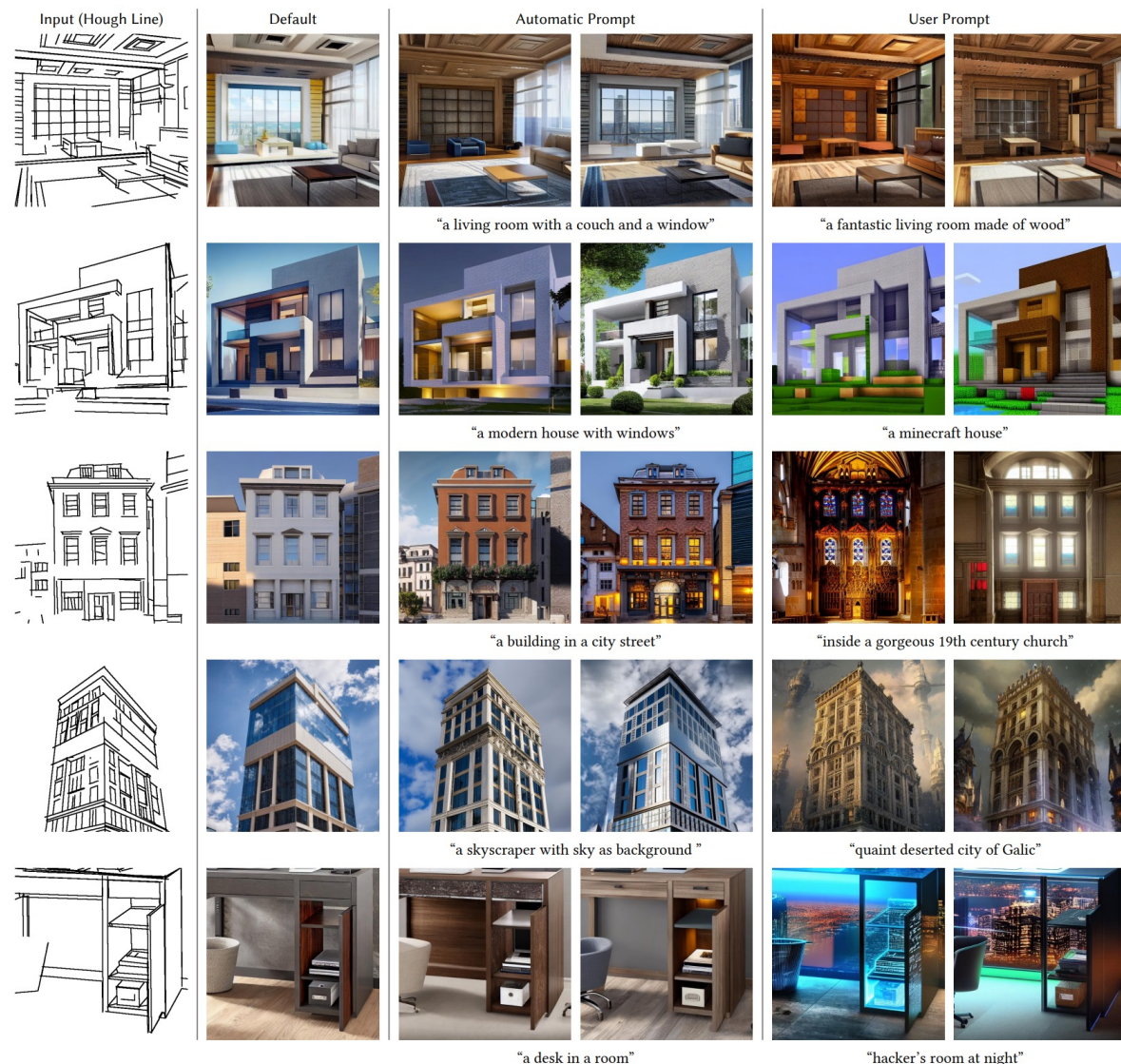


Figure 9: Controlling Stable Diffusion with Hough lines (M-LSD). The “automatic prompts” are generated by BLIP based on the default result images without using user prompts. See also the Appendix for source images for line detection.

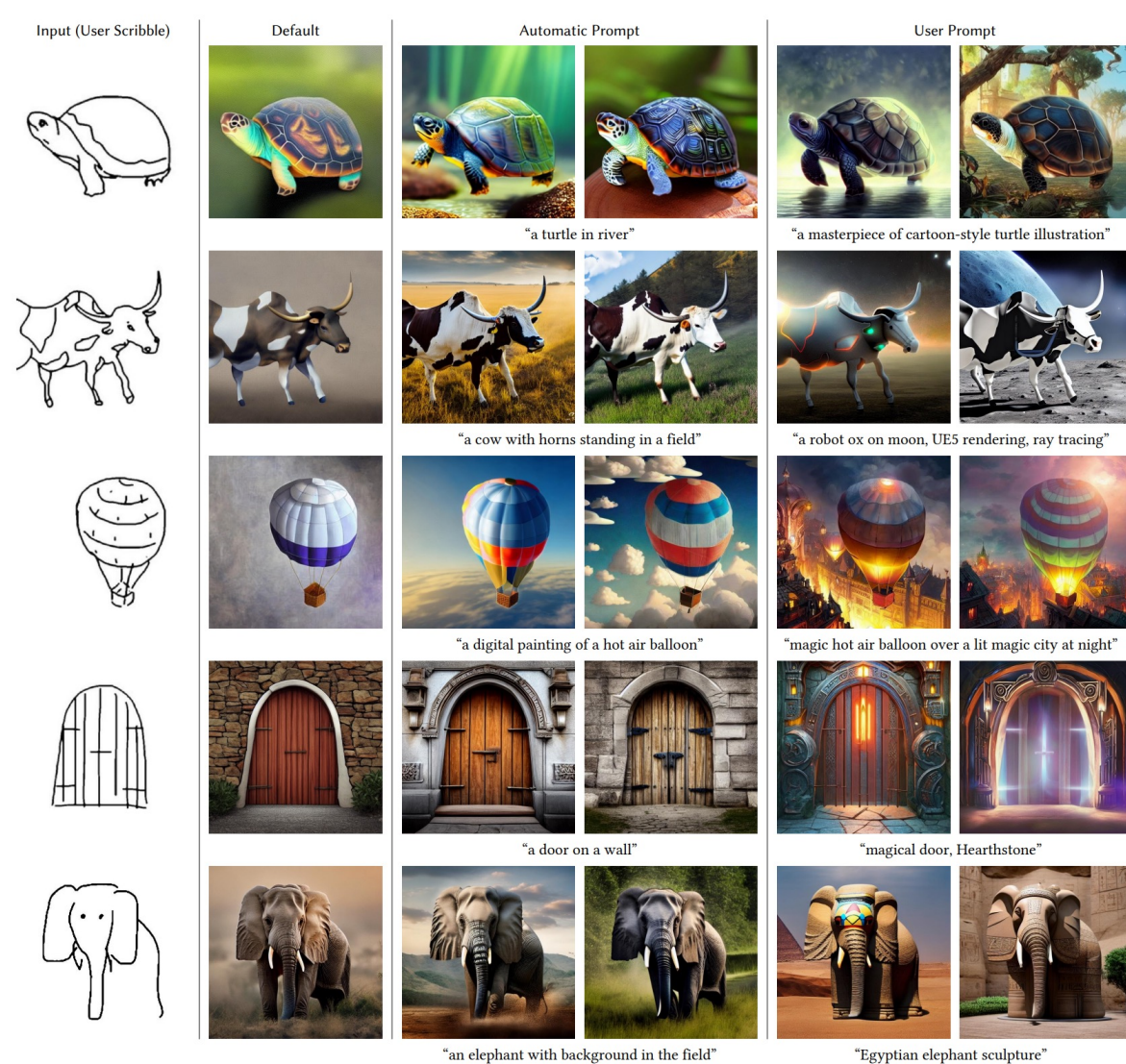


Figure 10: Controlling Stable Diffusion with Human scribbles. The “automatic prompts” are generated by BLIP based on the default result images without using user prompts. These scribbles are from [19].

# Segmentation and Human Pose

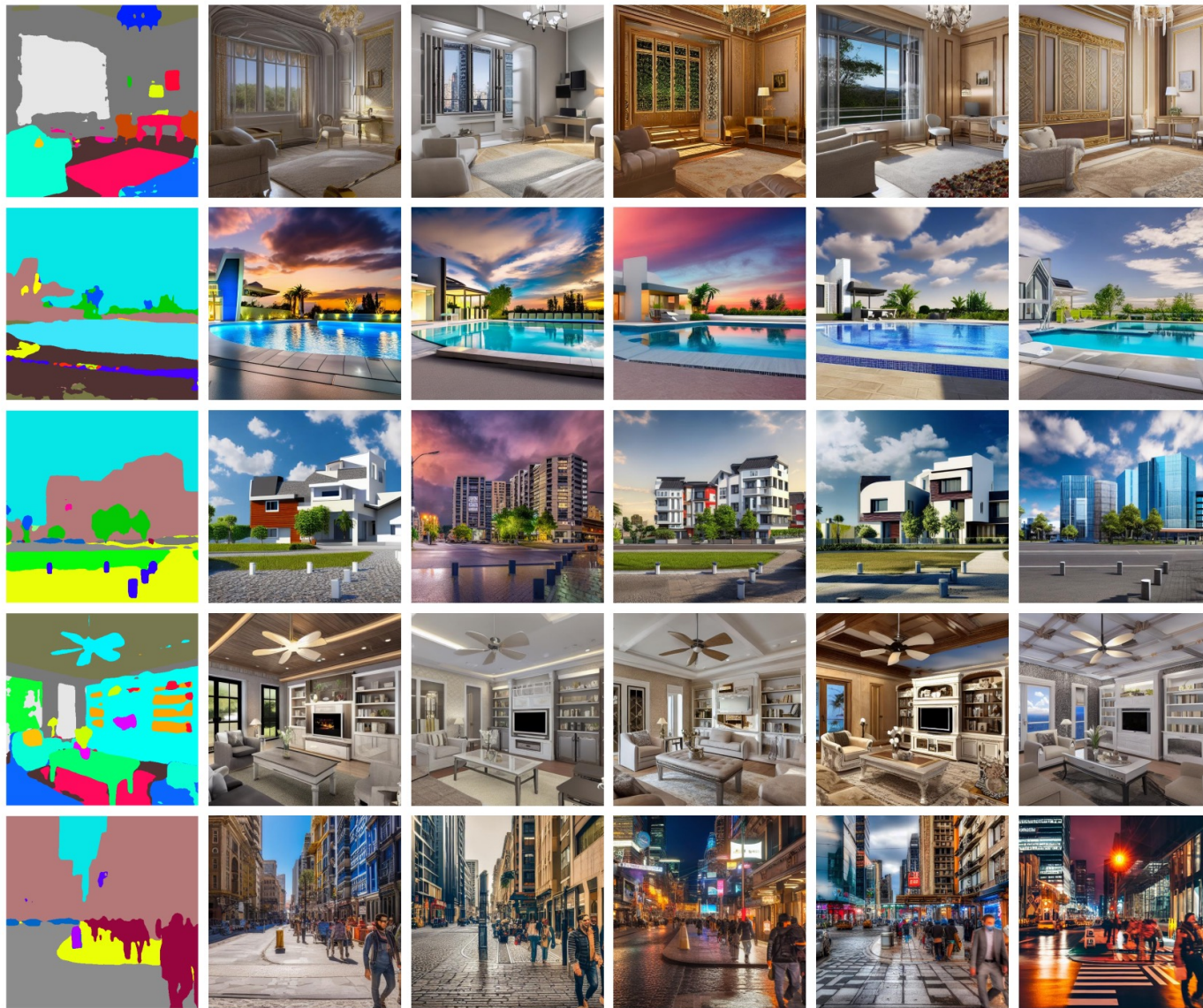
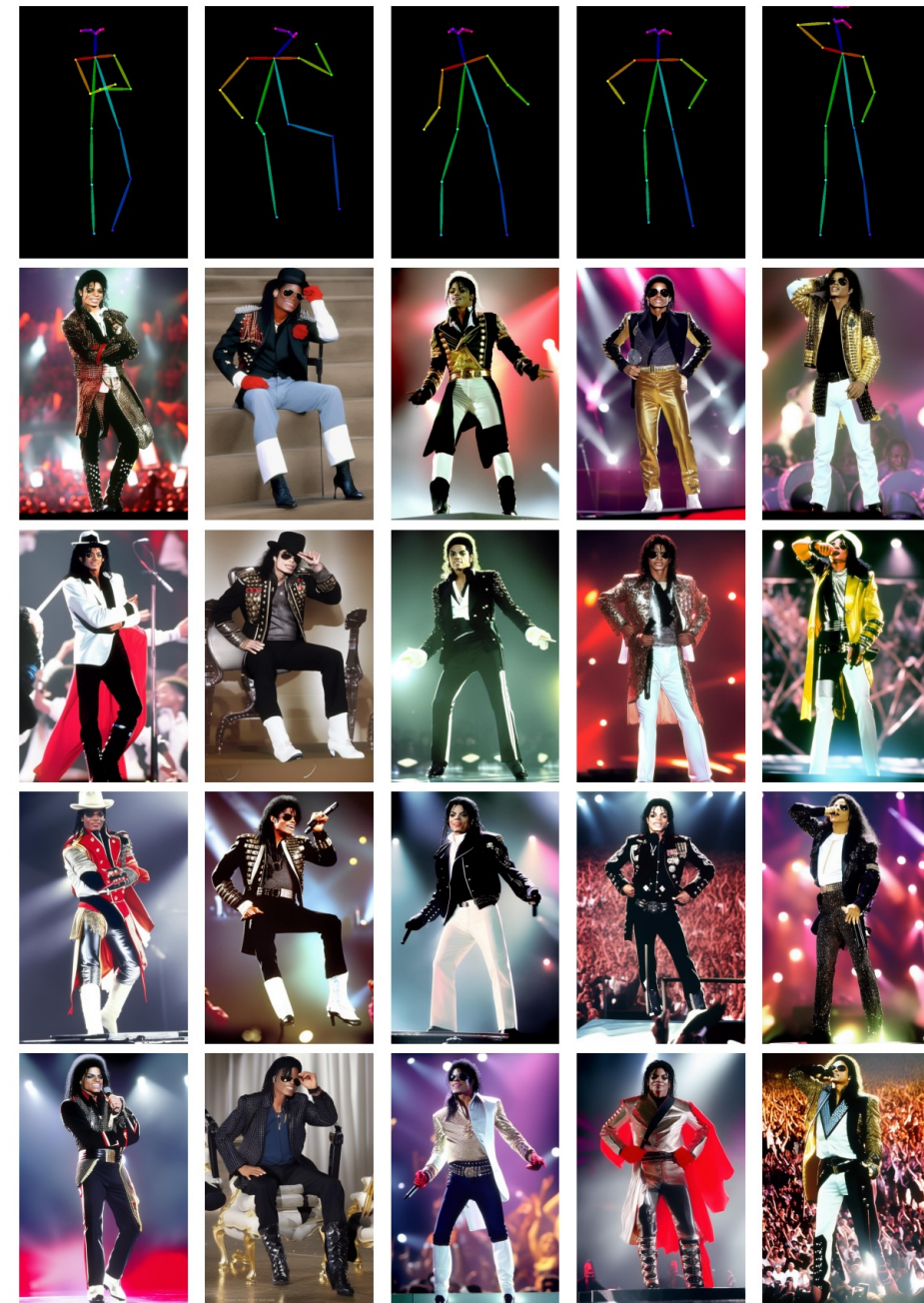


Figure 15: Controlling Stable Diffusion with ADE20K segmentation map. All results are achieved with default prompt. See also the Appendix for source images for semantic segmentation map extraction.

“Michael Jackson’s concert”



# Conclusion for Lectures in Diffusion Models

- Previous lectures focused 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**
  - Denoising in the latent space enables spatial **complexity reduction** and **detail preservation**.
  - Introducing cross-attention layers enables **conditional input such as texts**.
- **ControlNet**
  - Enabling fine-grained control through **versatile conditions** (e.g., sketches, outlines, keypoints)
  - Preserving **pre-trained model knowledge** while extending functionality.