Deep Generative Models: Latent Diffusion Models

Fall Semester 2026

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

Penn
University of Pennsylvania

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 samples take 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.

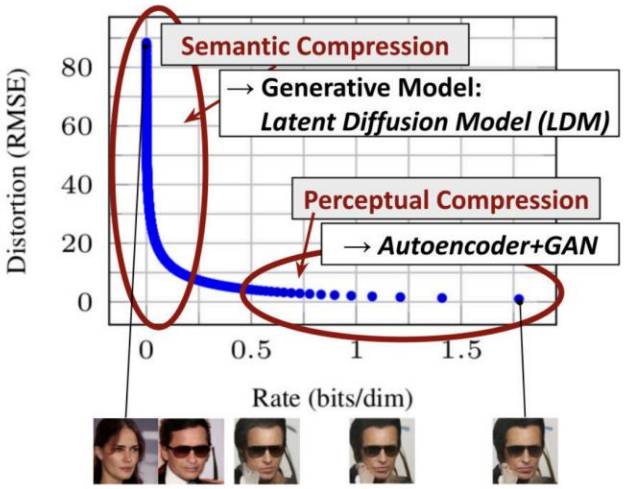
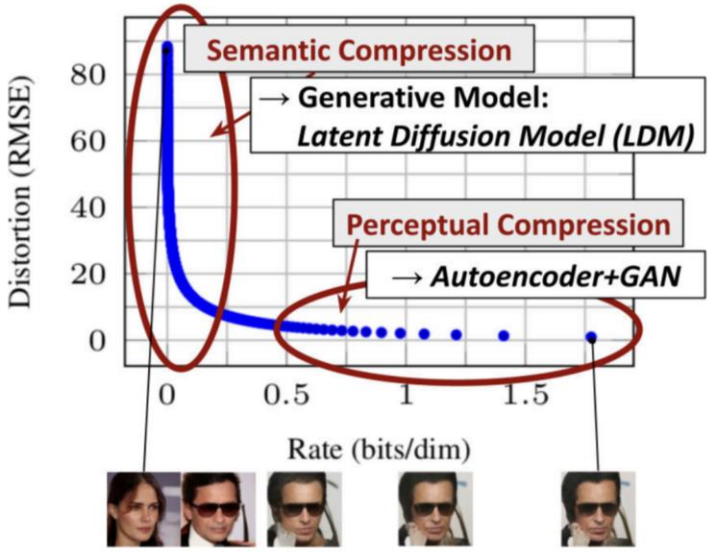


Figure 2. Illustrating perceptual and semantic compression: Most bits of a digital image correspond to imperceptible details. While DMs allow to suppress this semantically meaningless information by minimizing the responsible loss term, gradients (during training) and the neural network backbone (training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

Perceptual vs Semantic Compression

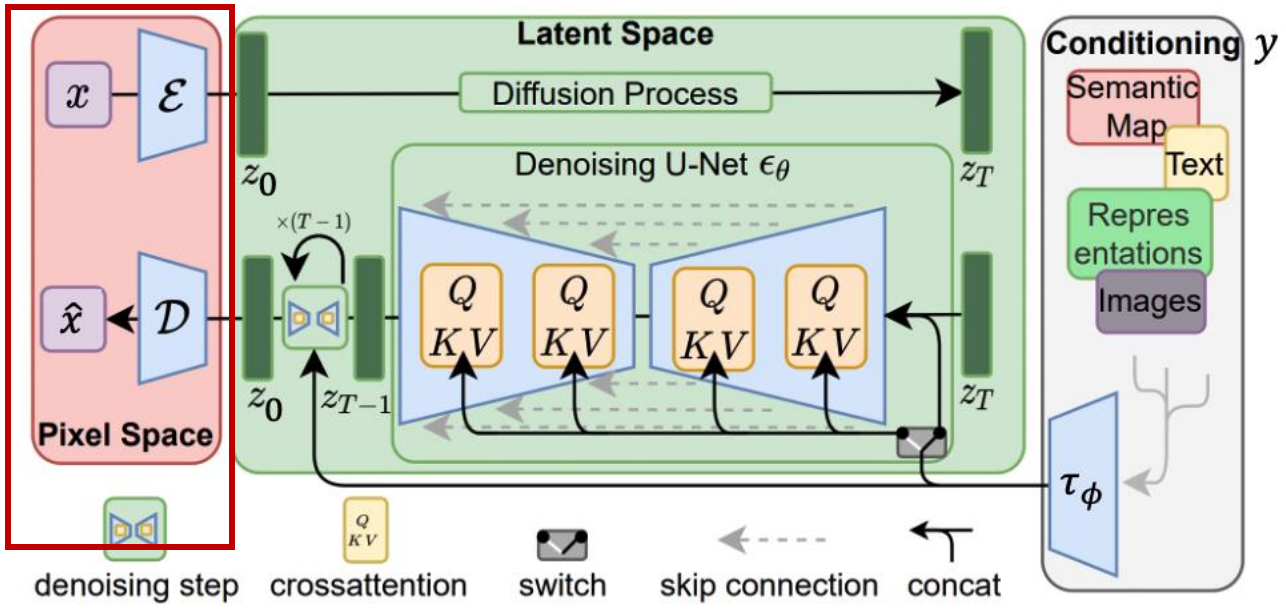
- Perceptual Compression
  - Removes perceptually insignificant high-frequency details (e.g., fine skin textures) while preserving important visual information (e.g., recognizable facial features).
  - Maps images into a lower-dimensional latent space for efficient generative modeling.



- Semantic Compression
  - Learns the semantic and conceptual composition of images (e.g., the arrangement of eyes, nose, and mouth in human faces).
  - Models the distribution of the perceptually compressed representations.
- Two-Stage Image Synthesis
  - Stage 1: A pretrained autoencoder performs **perceptual compression**.
  - Stage 2: A diffusion model performs **semantic compression** by learning the distribution and semantic composition of compressed latent representations.

Stable Diffusion: Two-Stage Image Synthesis

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



Stable Diffusion: Two-Stage Image Synthesis

- **(Stage 1) Perceptual Image Compression**: A pretrained VAE maps images into a lower-dimensional continuous latent space for efficient reconstruction.
  - Encoder $\mathcal{E}_\phi$: Maps an image $\boldsymbol{x}$ to a Gaussian latent distribution:

$$q_\phi(\boldsymbol{z}_0|\boldsymbol{x}) = \mathcal{N}\left(\mu_\phi(\boldsymbol{x}), \mathrm{diag}\left(\sigma_\phi^2(\boldsymbol{x})\right)\right)$$

  - Decoder $\mathcal{D}_\theta$: Reconstructs an image from a sampled of the latent diffusion model p_{LDM} :

$$\boldsymbol{z}_0 \sim p_{\mathrm{LDM}}, \qquad \hat{\boldsymbol{x}} = D_\theta(\boldsymbol{z}_0)$$

  - **VAE Objective**: Encourages image reconstruction while regularizing the latent distribution toward a standard Gaussian prior

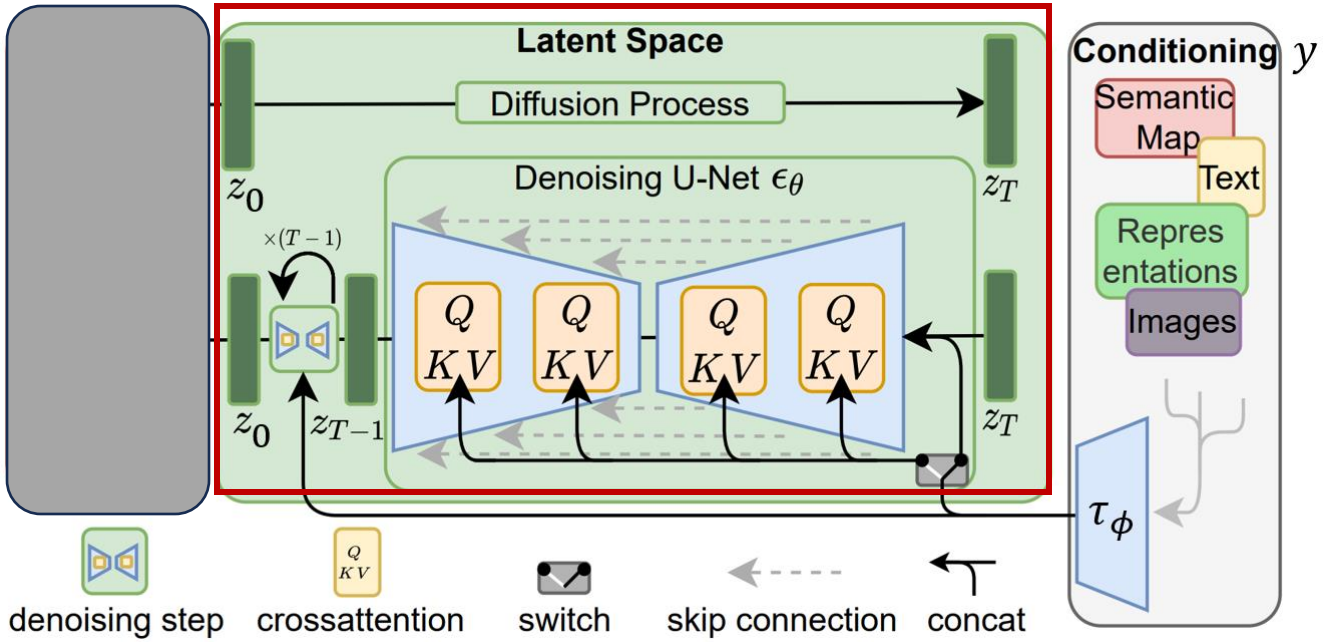
$$\min_{\phi,\theta} \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}} \underbrace{\mathbb{E}_{z\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\, L_{\mathrm{rec}}(\boldsymbol{x},\hat{\boldsymbol{x}}) + \lambda_{\mathrm{per}} L_{\mathrm{per}}\left(\varphi(\boldsymbol{x}), \varphi(\hat{\boldsymbol{x}})\right) + \lambda_{\mathrm{adv}} L_{\mathrm{adv}}}_{\text{Reconstruction + Perceptual}} + \underbrace{\beta D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))}_{\text{KL regularization}}$$

    - Learned Perceptual Image Patch Similarity (LPIPS) loss (a.k.a. perceptual loss) measures reconstruction error in a perceptual space induced by a frozen VGG-16 or AlexNet encoder φ rather than raw pixel space
    - Patch adversarial loss is used to enforce sharp textures and realistic local detail
    - $\beta \ll 1$ is used since decoder $\mathcal{D}_\theta$ maps latent from diffusion model to image space, rather than a standard Gaussian, hence less strong regularization is needed

Stable Diffusion: Two-Stage Image Synthesis

- **(Stage 2) Denoising Latent Representations**: the denoising process operates in the compressed latent space obtained from Stage 1.
- Recall that DDPM performs denoising in the image space, while the Stable Diffusion performs denoising in the latent code.
- The forward process progressively adds Gaussian noises to the original latent code $\boldsymbol{z_0}$ over T steps, while the reverse process learns to remove the noise.

Encoder and Decoder are from the Stage 1.



$\boldsymbol{z_0} = \boldsymbol{E}(\boldsymbol{x})$: Encoding of image from stage 1

$$\boldsymbol{z}_t = \sqrt{\overline{\alpha_t}}\boldsymbol{z}_0 + \sqrt{1-\overline{\alpha_t}}\,\bar{\boldsymbol{\epsilon}}_t,$$
$$\bar{\boldsymbol{\epsilon}}_t \sim \mathcal{N}(0, \boldsymbol{I})$$

Stable Diffusion: Two-Stage Image Synthesis

- The neural backbone of SD is a time-conditional U-Net $\hat{\epsilon}_\theta\left(\boldsymbol{z}_t, t, \tau_\phi(y)\right)$ trained to predict the noise to be removed from the latent code $\boldsymbol{z}_t$.
- SD augments the U-Net backbone with cross-attention to incorporate conditioning inputs y (e.g., text or semantic maps), encoded by a conditioning encoder $\tau_\phi(\cdot)$.

$$L_{\text{LDM}} = \mathbb{E}_{\boldsymbol{x},y,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}),t}\left[\| \bar{\boldsymbol{\epsilon}}_t - \hat{\boldsymbol{\epsilon}}_\theta\left(\boldsymbol{z}_t, t, \tau_\phi(y)\right) \|_2^2\right]$$

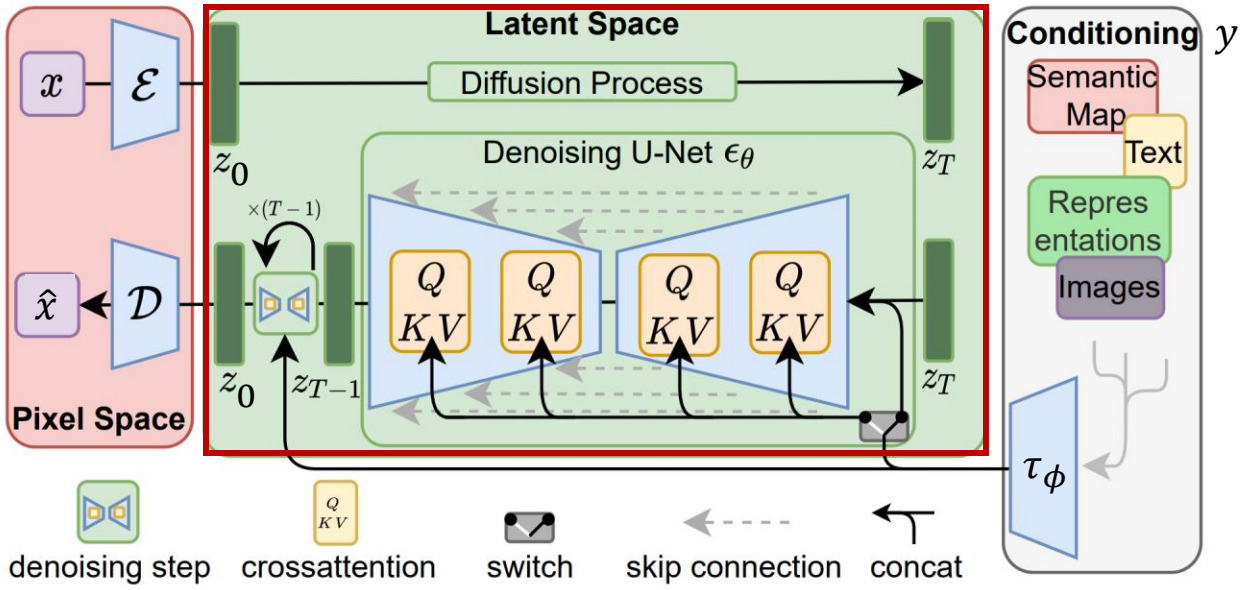
(flattened) visual representation from the U-Net

Cross-attention Modules

$$\boldsymbol{Q} = \boldsymbol{W}_Q^{(i)} \cdot \psi_i(\boldsymbol{z}_t),$$
$$\boldsymbol{K} = \boldsymbol{W}_K^{(i)} \cdot \tau_\phi(y),$$
$$\boldsymbol{V} = \boldsymbol{W}_V^{(i)} \cdot \tau_\phi(y)$$

conditional representation such as textual prompt

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V}$$

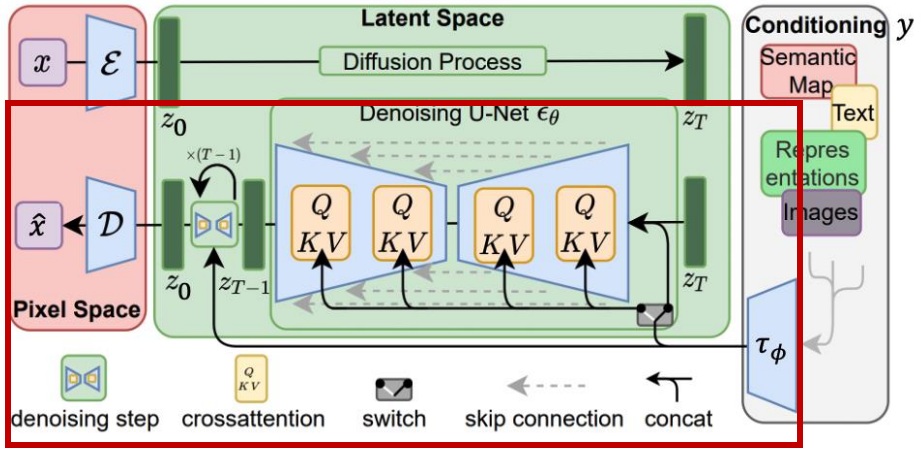


Stable Diffusion: Two-Stage Image Synthesis

- (Stage 2) **Generative Modeling of Latent Representations**: denoising happens in the compressed latent space given by Stage 1.
- **Reverse Diffusion Process**: Starting from $\boldsymbol{z}_T \sim \mathcal{N}(0, \boldsymbol{I})$, the reverse diffusion process iteratively denoises the latent representation to obtain $\boldsymbol{z}_0$.

$$\boldsymbol{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\hat{\boldsymbol{\epsilon}}_\theta\left(\boldsymbol{z}_t, t, \tau_\phi(y)\right)\right) + \sqrt{\beta_t}\,\boldsymbol{\epsilon}_t, \qquad \text{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

- **Image Generation**: The pretrained decoder from Stage 1 generates the image from the denoised latent representation $\hat{x} = D(\boldsymbol{z}_0)$.



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 latent space.
- Exemplary samples are provided from five unconditional stable diffusion models, each trained on a separate dataset.



Image Generation Conditioned on Text

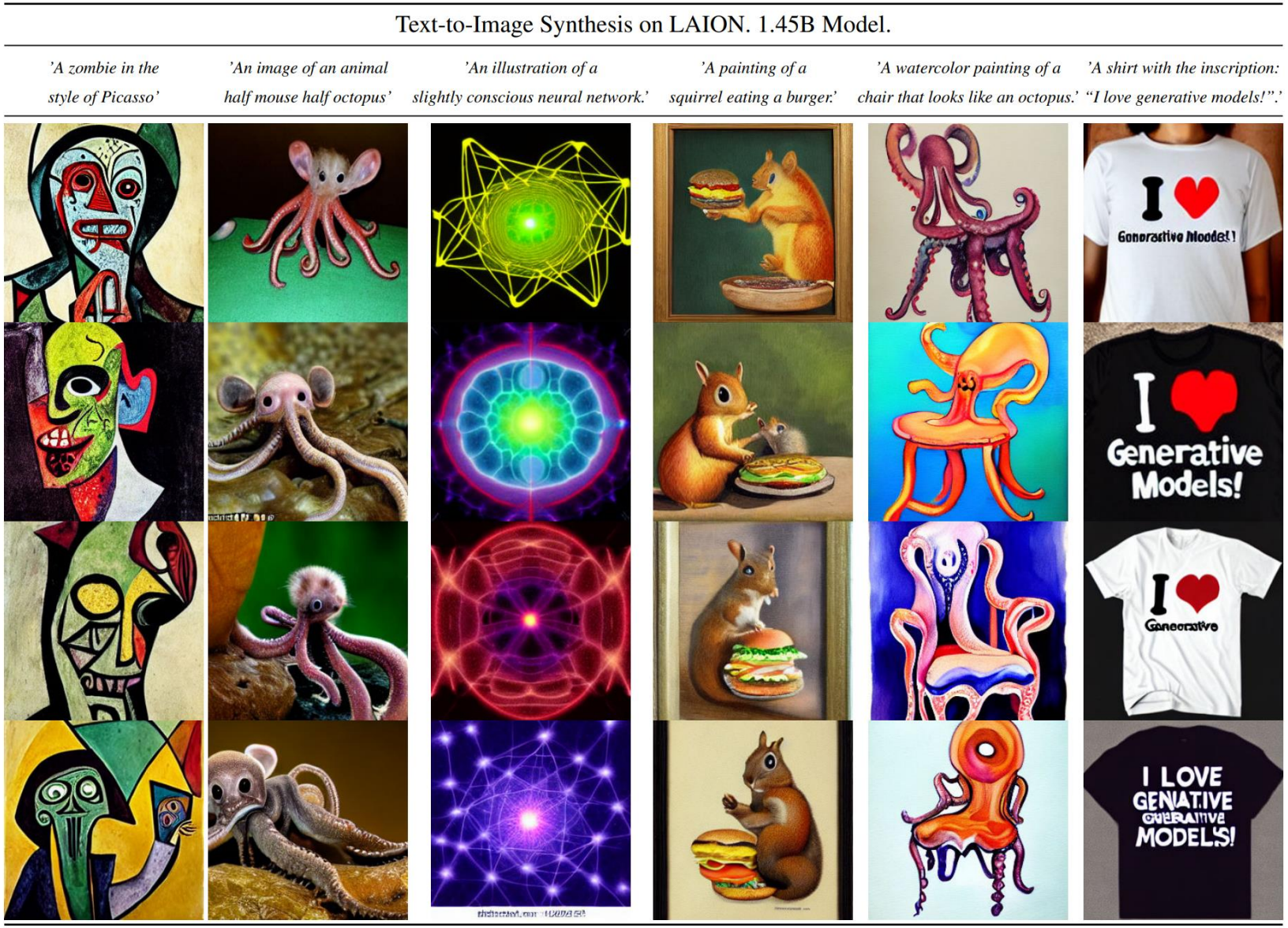
Details of this 1.45B Model

Autoencoder: KL-regularized

Text Conditioner: Transformer-based

Training Dataset: LAION-400M

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



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:

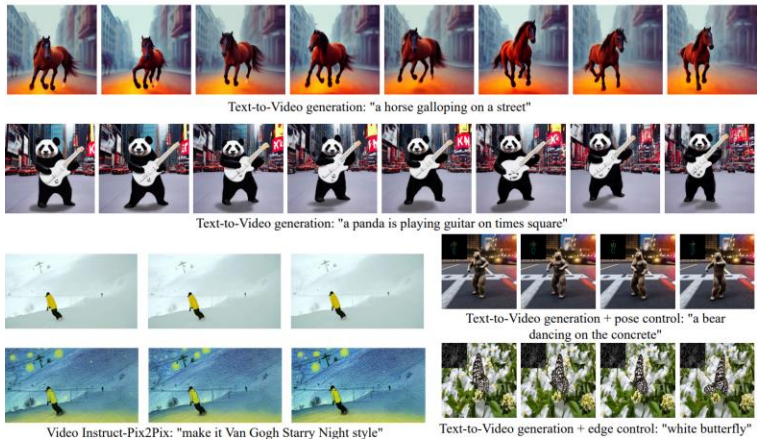
DREAMFUSION: TEXT-TO-3D USING 2D DIFFUSION

Ben Poole[1], **Ajay Jain**[2], **Jonathan T. Barron**[1], **Ben Mildenhall**[1]
[1]Google Research, [2]UC Berkeley
{pooleb, barron, bmild}@google.com, ajayj@berkeley.edu



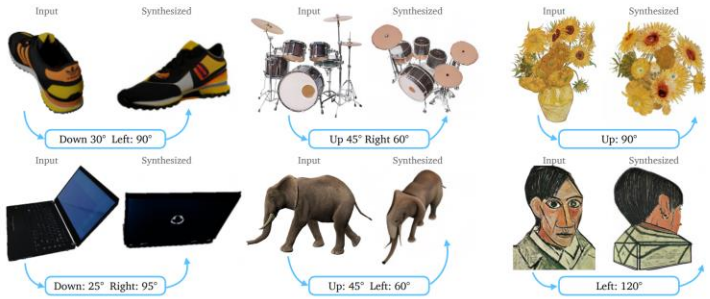
Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators

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



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

Ruoshi Liu[1] Rundi Wu[1] Basile Van Hoorick[1] Pavel Tokmakov[2] Sergey Zakharov[2] Carl Vondrick[1]
[1] Columbia University [2] Toyota Research Institute
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

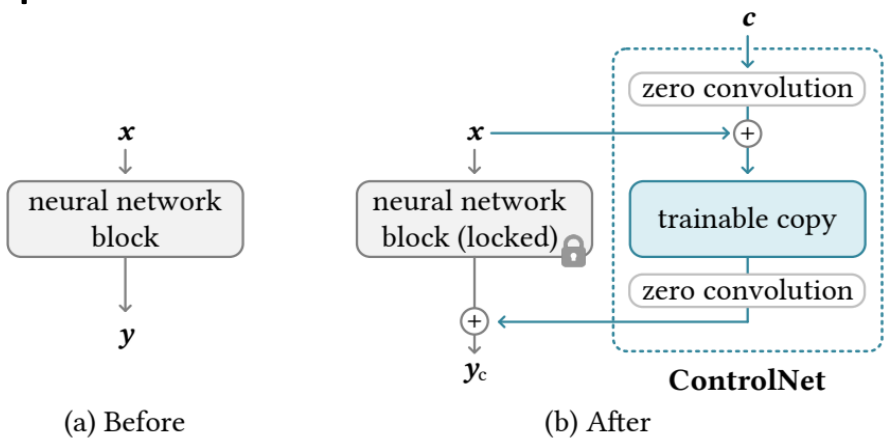
- Text prompts provide limited spatial control, motivating conditions such as edges, depth maps, and human poses.
- ControlNet is a neural network architecture that adds spatial conditioning to large pre-trained text-to-image diffusion models (e.g., Stable Diffusion).
- ControlNet allows users to add conditions like Canny edges or human pose to control the text-to-image synthesis.





Key Intuitions of ControlNet

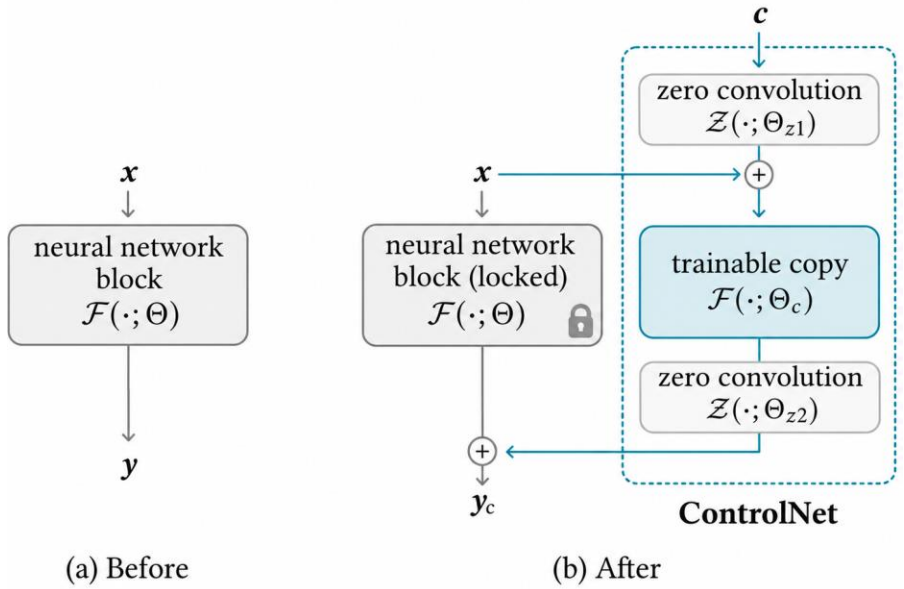
- Core idea:
  - ControlNet freezes the parameters Θ of the original generative neural block $\mathcal{F}_\Theta$ and create a trainable copy with parameters Θ_c.
  - The trainable copy incorporates an external spatial condition $\boldsymbol{c}$ (e.g., edge, depth, or pose maps) through a zero-convolution layer.
- Why it works well?
  - **Frozen parameters** preserve the pretrained model's generative capabilities, while the **trainable copy** learns to incorporate diverse spatial conditions.
  - **Zero-initialized convolution** ensures that the trainable branch initially contributes no correction, preserving the pretrained model's behavior.



Zero-Initialized Convolution Layers

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

Θ: frozen pretrained parameters
Θ_c: trainable parameters of the cloned block
Θ_{z_1}: trainable parameters of the first zero convolution
Θ_{z_2}: trainable parameters of the second zero convolution



$$\boldsymbol{y}_c = \mathcal{F}(\boldsymbol{x};\Theta) + \mathcal{Z}(\mathcal{F}\left(\boldsymbol{x} + \mathcal{Z}\left(\boldsymbol{c};\Theta_{z_1}\right);\Theta_c\right);\Theta_{z_2})$$

- $\boldsymbol{y}_c$ combines the frozen block output $\mathcal{F}(\boldsymbol{x};\Theta)$ with a trainable correction conditioned on $\boldsymbol{c}$ (e.g., edge or depth maps), where zero initialization preserves the pretrained model's behavior.

Gradient Calculation of A Zero Initialization Layer

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

$$\mathcal{Z}(\boldsymbol{x}; \{\boldsymbol{W}, \boldsymbol{B}\})_{p,i} = B_i + \sum_{j=1}^{c} \boldsymbol{x}_{p,j} W_{i,j}$$

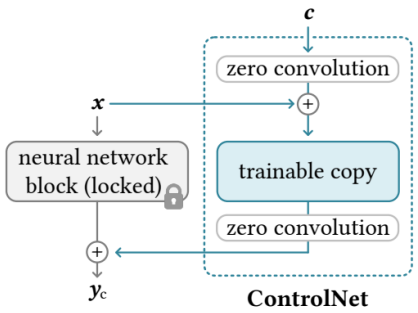
- Each layer is initialized with $\boldsymbol{W} = \boldsymbol{0}$ and $\boldsymbol{B} = \boldsymbol{0}$. However, the gradient at a point such that $\boldsymbol{x}_{p,i} \neq 0$ need not be zero, allowing these weights to be learned:

$$\begin{cases} \dfrac{\partial \mathcal{Z}(\boldsymbol{x}; \{\boldsymbol{W}, \boldsymbol{B}\})_{p,i}}{\partial B_i} = 1, \\ \dfrac{\partial \mathcal{Z}(\boldsymbol{x}; \{\boldsymbol{W}, \boldsymbol{B}\})_{p,i}}{\partial x_{p,j}} = W_{i,j} = 0 \\ \dfrac{\partial \mathcal{Z}(\boldsymbol{x}; \{\boldsymbol{W}, \boldsymbol{B}\})_{p,i}}{\partial W_{i,j}} = x_{p,j} \neq 0 \end{cases}$$

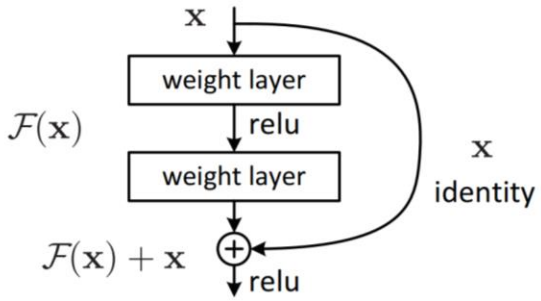
ControlNet for Stable Diffusion

- The ControlNet structure is applied to each encoder level of the U-Net.
- An analogy with Residual Learning:
- Original SD: $h_i = h_i^{\text{SD}}$
- ControlNet: $h_i = h_i^{\text{SD}} + \Delta h_i^{\text{ControlNet}}$

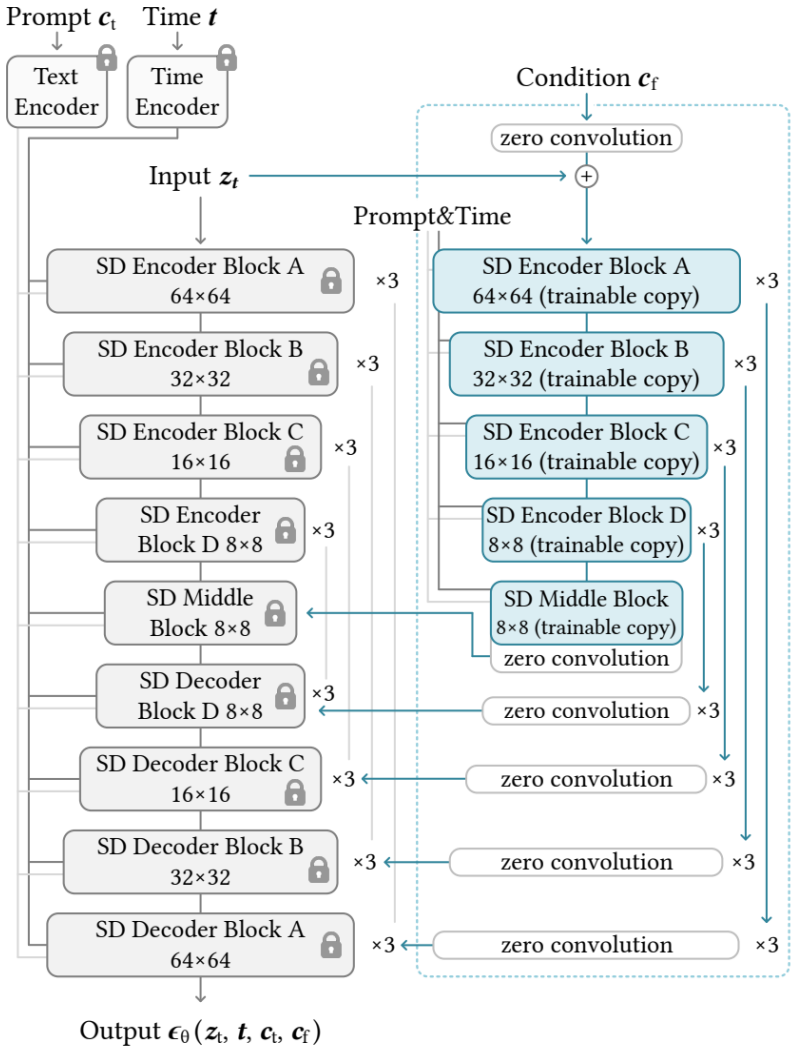
Where h_i^{SD} denotes an intermediate U-Net feature, and $\Delta h_i^{\text{ControlNet}}$ is the learned correction from ControlNet.



ControlNet

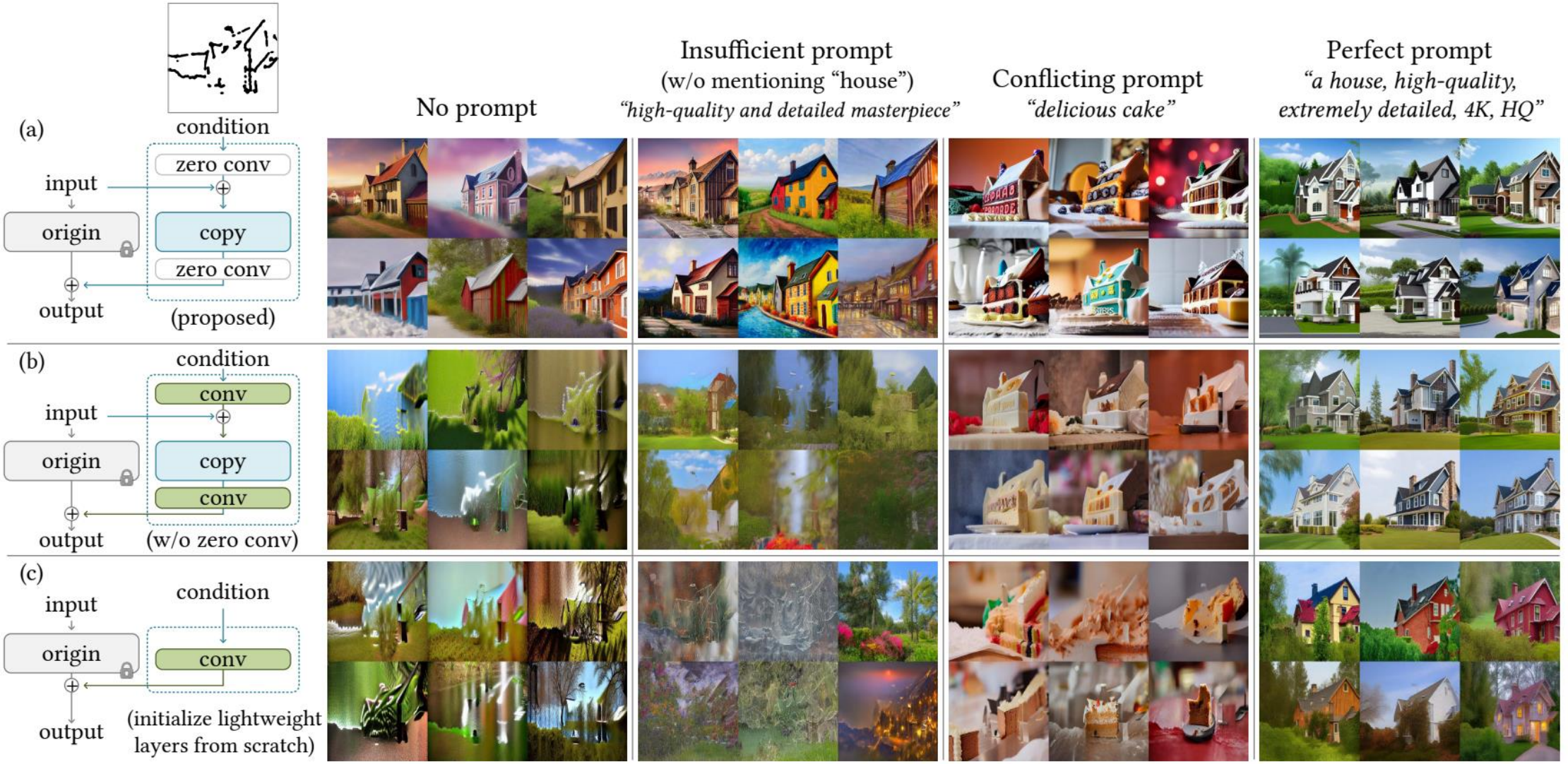


Residual Learning
[He et al., 2015]



Does Zero Initialization Help?

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



Training Results

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



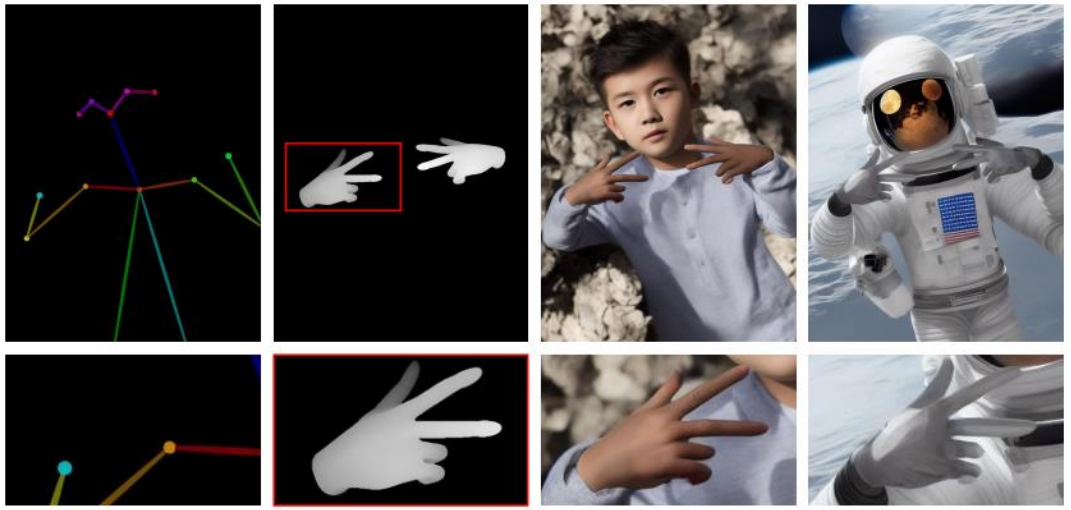
Test input | training step 100 | step 1000 | step 2000

step 6100 | **step 6133** | step 8000 | step 12000



(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

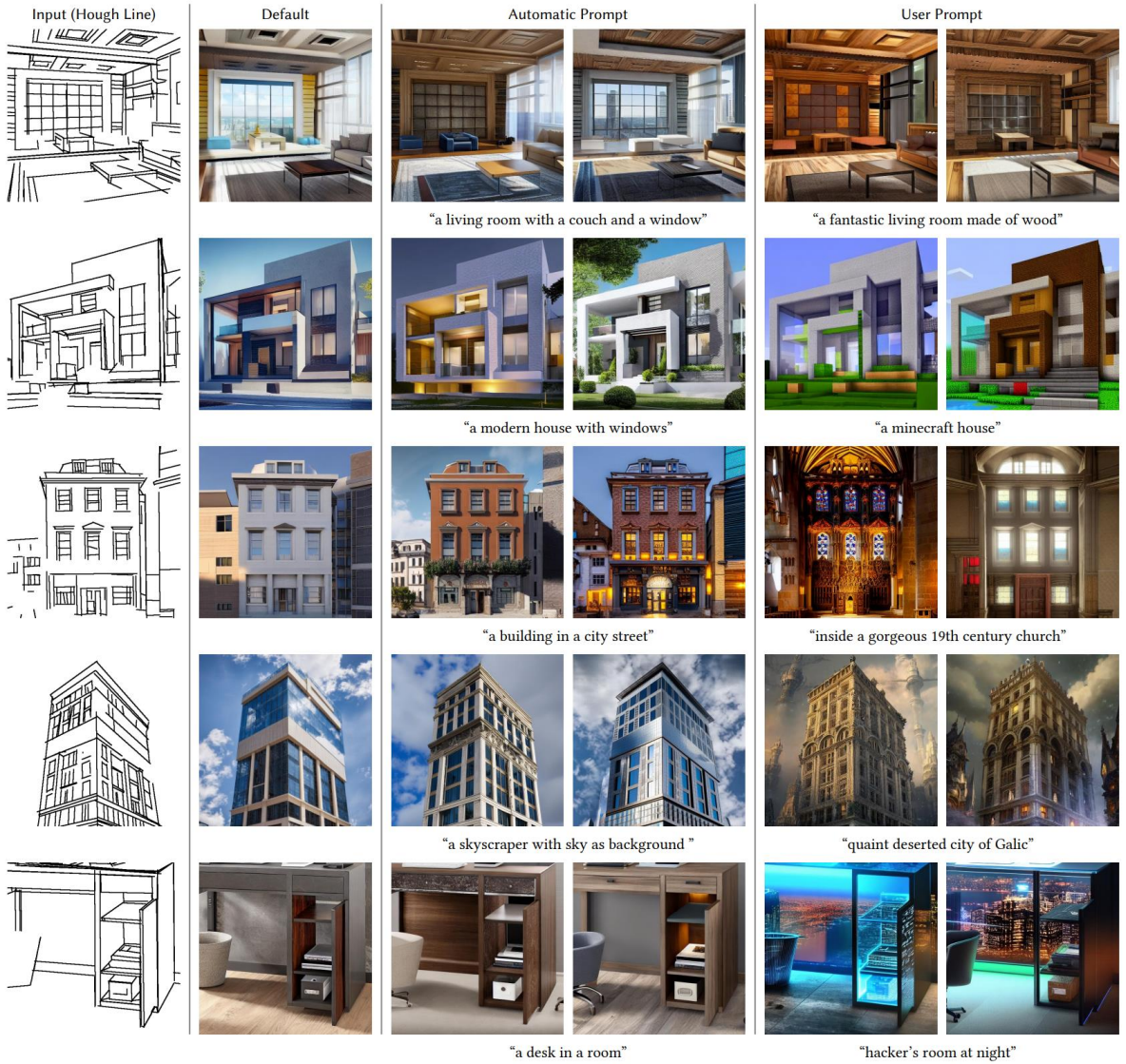


Figure 9: Controlling Stable Diffusion with Hough lines (M-LSD). The "automatic prompts" are generated by BLIP based on the default result images without using user prompts. See also the Appendix for source images for line detection.

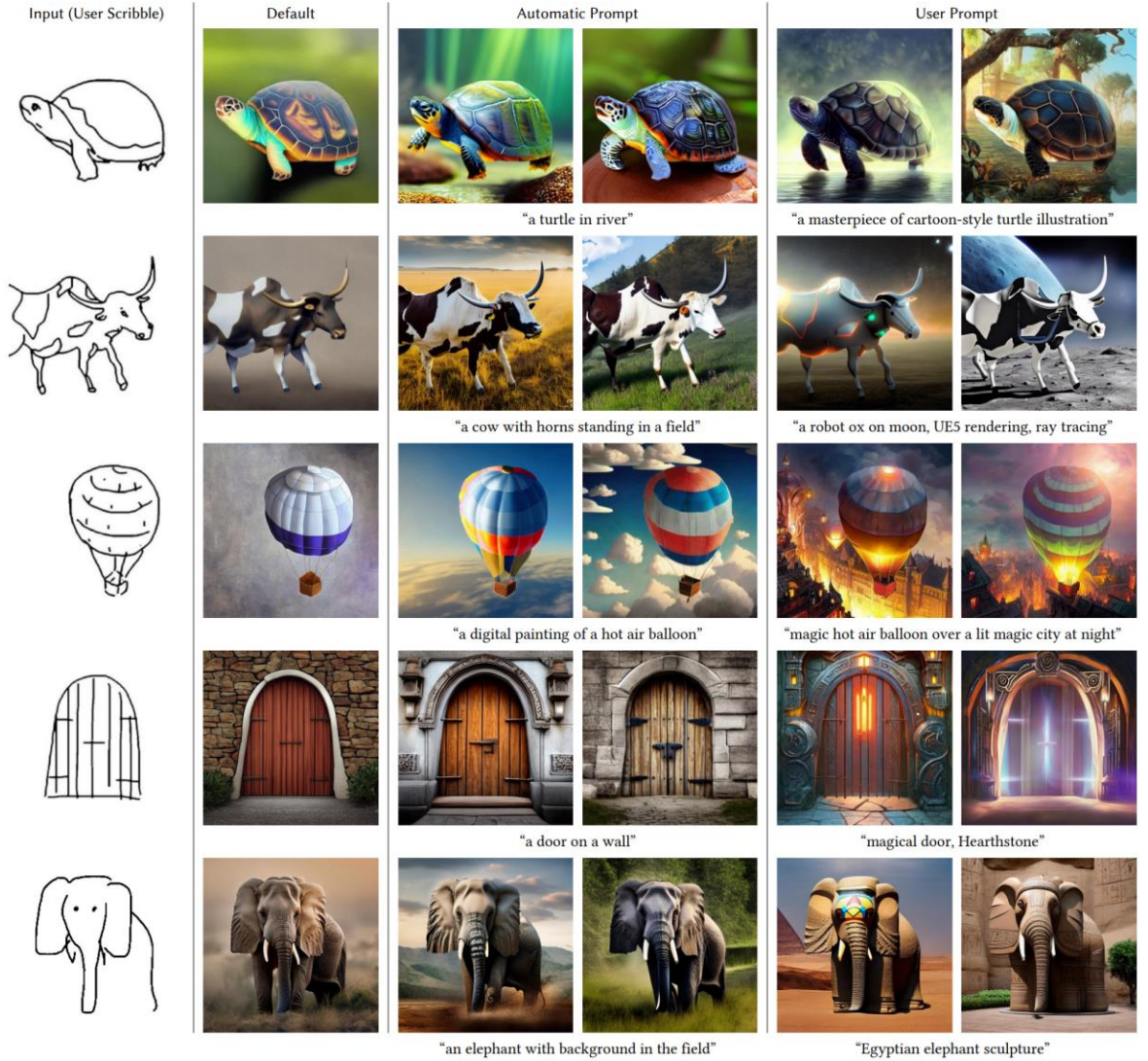


Figure 10: Controlling Stable Diffusion with Human scribbles. The "automatic prompts" are generated by BLIP based on the default result images without using user prompts. These scribbles are from [19].

Segmentation and Human Pose

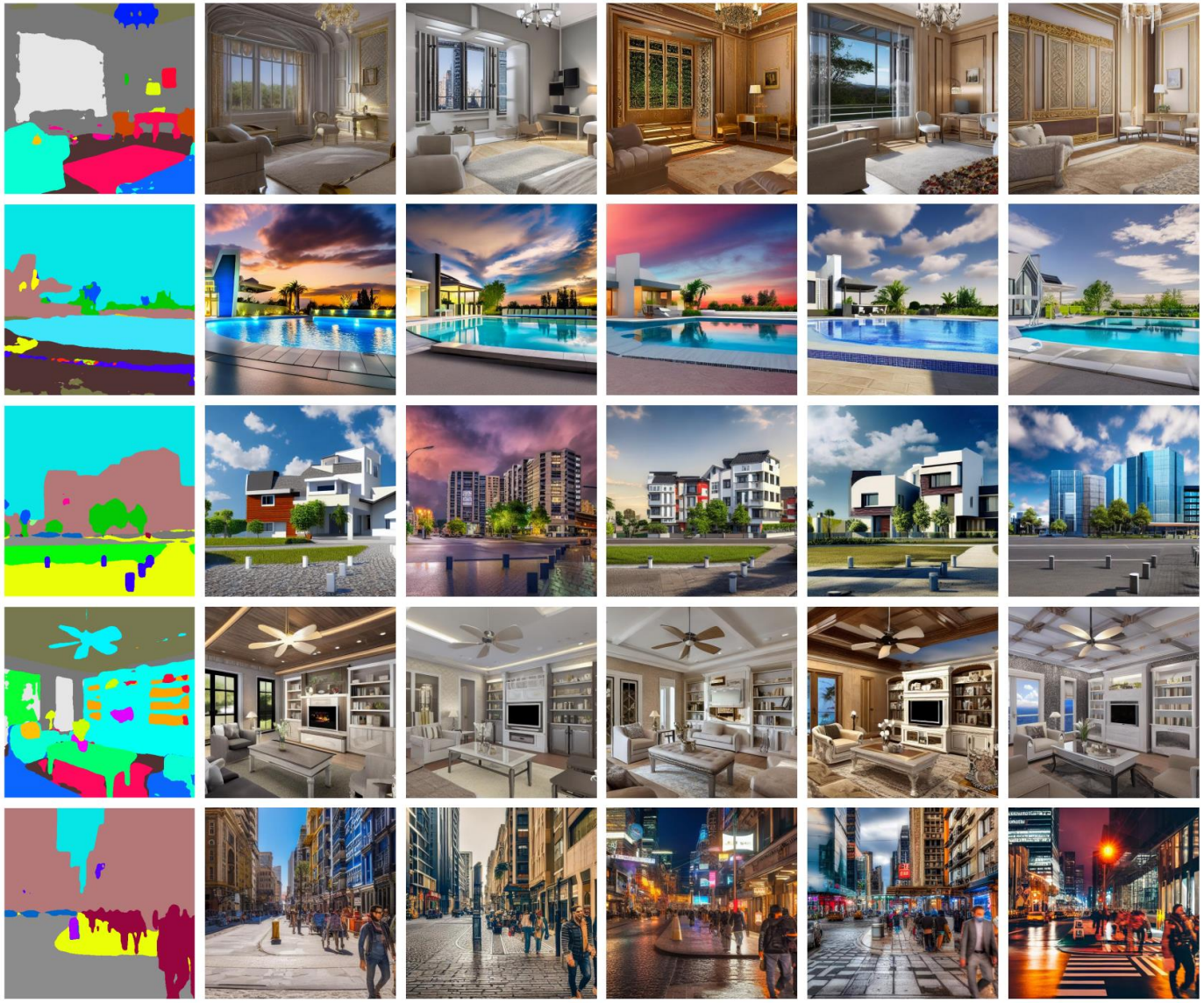
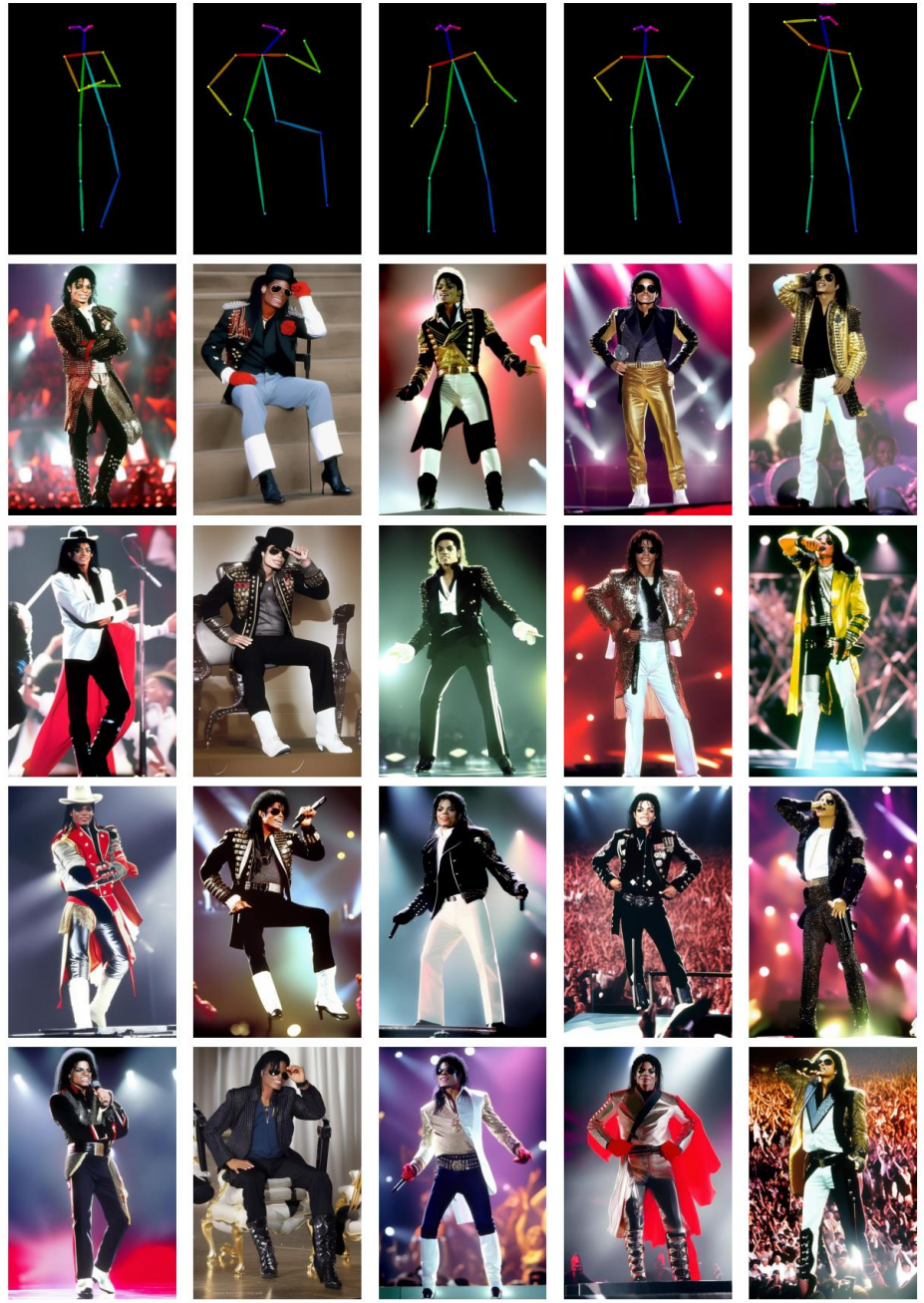


Figure 15: Controlling Stable Diffusion with ADE20K segmentation map. All results are achieved with default prompt. See also the Appendix for source images for semantic segmentation map extraction.

"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 diffusion-based frameworks: **Stable Diffusion** and **ControlNet**.

- **Stable Diffusion**
  - Denoising in a compressed latent space reduces computational complexity while preserving perceptually important visual information.
  - Introducing cross-attention layers enables conditional input such as texts.
- **ControlNet**
  - Enabling fine-grained spatial control through diverse conditions (e.g., edges, depth maps, and human poses)
  - Preserving pre-trained model capabilities while learning new spatial conditioning.