VAEs and diffusion models

Andrey Stotskiy, Vlad Shakhuro

11 December 2025

Outline

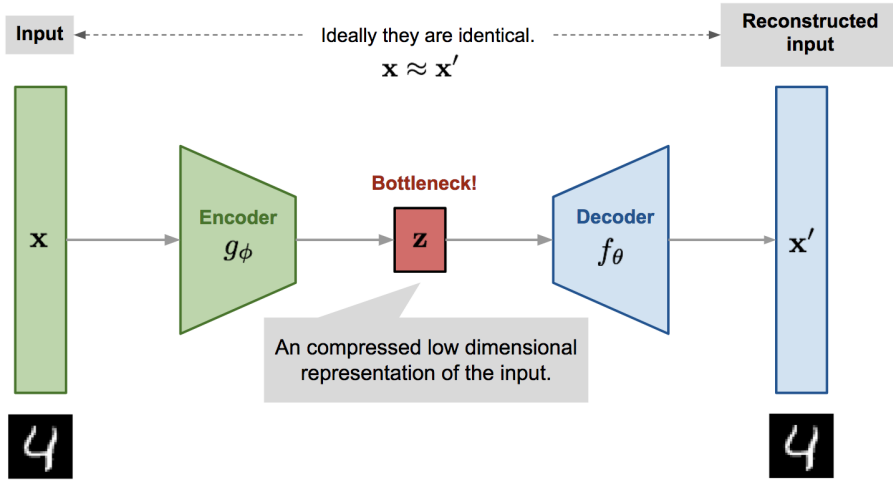
1. Variational autoencoders

2. Vector quantized VAEs

3. Denoising diffusion probabilistic models

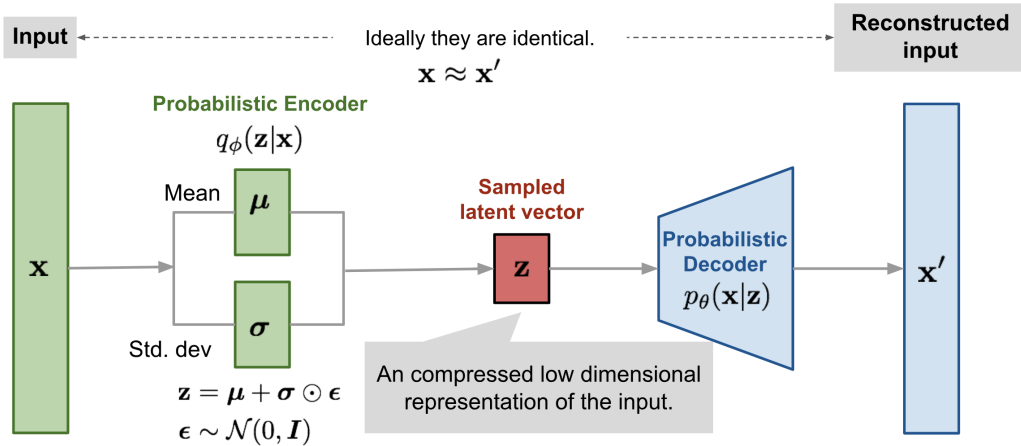
4. High-res and conditional generation with diffusion models

Autoencoder



Hinton et al. Reducing the dimensionality of data with neural networks. Science 2006

VAE

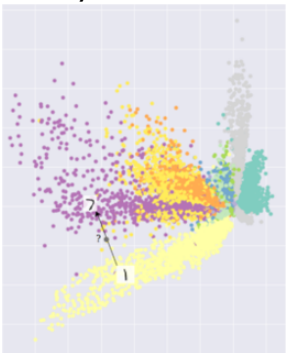


Kingma, Welling. Auto-Encoding Variational Bayes. ICLR 2014

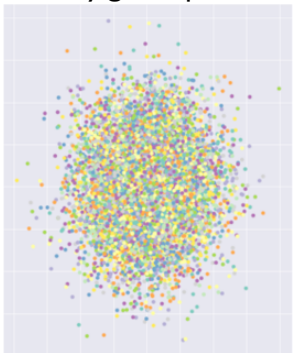
VAE loss function

$$\mathscr{L} = \underbrace{d(\boldsymbol{x}, \boldsymbol{x'})}_{\text{reconstr. err}} + \underbrace{KL(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}))}_{\text{gauss. prior on latent distribution}}$$

only reconstr. err



only gauss. prior



combination

Reparametrization trick

To compute gradients ∇_ϕ through sampling, substitute sampling

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi^2)$$

with deterministic function that depends on separate noise variable

$$\boldsymbol{z} = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

This way sampled values $\boldsymbol{\varepsilon}$ during backpropagation may be seen as constant that don't depend on encoder parameters ϕ

Outline

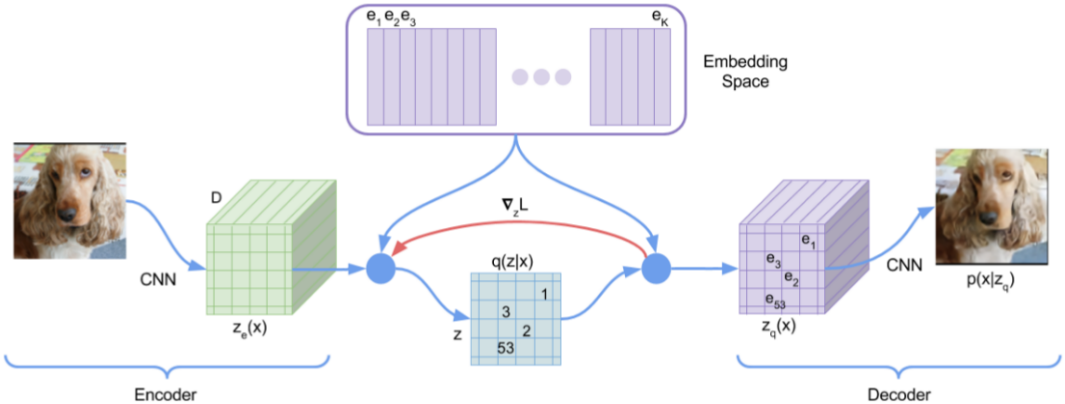
1. Variational autoencoders

2. Vector quantized VAEs

3. Denoising diffusion probabilistic models

4. High-res and conditional generation with diffusion models

VQ-VAE



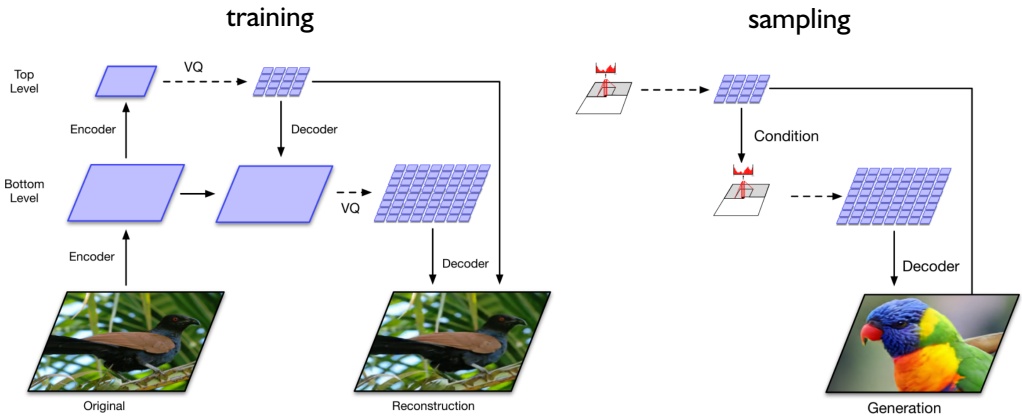
Oord et al. Neural Discrete Representation Learning. NeurIPS 2017

VQ-VAE learning

$$\mathcal{L} = \underbrace{d(x, x')}_{\text{reconstr. err}} + \underbrace{\left\|\text{sg}\left[z_e(x)\right] - e\right\|_2^2}_{\text{dict learning}} + \underbrace{\beta \left\|z_e(x) - \text{sg}\left[e\right]\right\|_2^2}_{\text{regularization of image features}}$$

Here sg is a stop gradient function, which makes operand non-updated constant. k-means step with EMA may be used to update dictionary instead of the second term

VQ-VAE-2

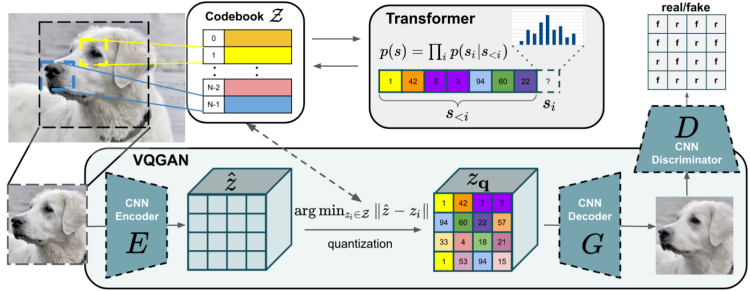


Razavi et al. Generating Diverse High-Fidelity Images with VQ-VAE-2. NeurIPS 2019

VQ-VAE-2 reconstruction example

h_{top} | $h_{\text{top}}, h_{\text{middle}}$ | $h_{\text{top}}, h_{\text{middle}}, h_{\text{bottom}}$ | Original

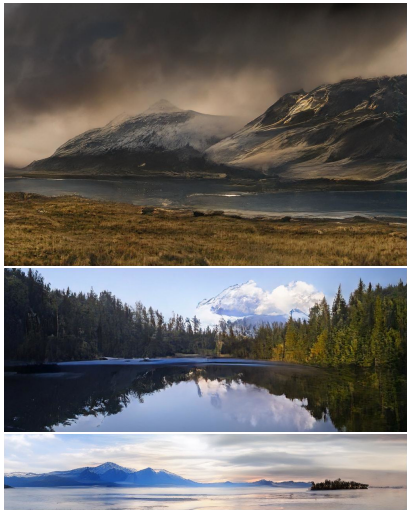
VQ-GAN



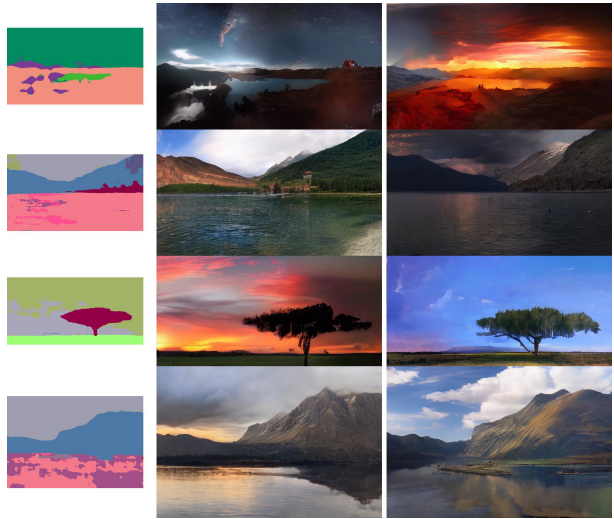
Esser et al. Taming Transformers for High-Resolution Image Synthesis. CVPR 2021

VQ-GAN results

unconditional



conditional



DALL·E

Large-scale dVAE (similar to VQ-VAE) + transformer prior:

- 250M (image, text) pairs
- image VAE was trained on 64×V100 GPUs
- ResNets as image encoder and decoder
- image dict size K = 8192, 256×256 → 32×32
- transformer (12B params) was trained on 1024×V100 GPUs
- efficient implementation of distributed mixed precision training (params, activations and Adam moments are stored in 16 bits)

Ramesh et al. Zero-Shot Text-to-Image Generation. arXiv:2102.12092

DALL·E

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

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

(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

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

Quality of samples may be improved using CLIP scoring

Ramesh et al. Zero-Shot Text-to-Image Generation. arXiv:2102.12092

Outline

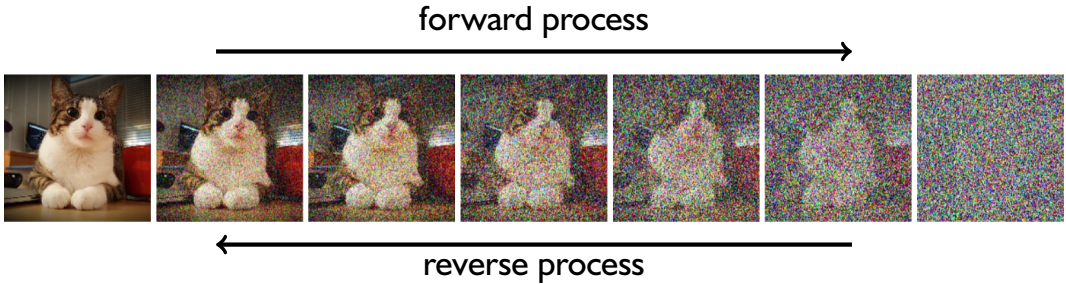
1. Variational autoencoders

2. Vector quantized VAEs

3. Denoising diffusion probabilistic models

4. High-res and conditional generation with diffusion models

Denoising diffusion probabilistic models



DDPM consists of two processes:

- forward diffusion process gradually adds gaussian noise to input
- reverse denoising process that learns to generate data by denoising

Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS 2020

Diffusion (forward process)

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I})$$

Forward sampling using reparametrization trick:

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\,\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}, \quad \bar{\alpha}_t = \Pi_{i=1}^{t}(1-\beta_i)$$

β_t is a noise schedule designed s.t.

$$q(\boldsymbol{x}_T|\boldsymbol{x}_0) \approx \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Generation (reverse process)

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \sigma_t^2 \boldsymbol{I})$$

Substitute $\boldsymbol{\mu}_\theta$ with $\boldsymbol{\varepsilon}_\theta$ due to reparametrization trick. Sampling:

$\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

for $t = T, \dots, 1$:

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

$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \right) + \sigma_t \boldsymbol{z}$

$\boldsymbol{\varepsilon}_\theta$ — trained UNet

α_t, σ_t — diff. parameters

return $\boldsymbol{x}_0$

Learning to denoise at any step

repeat until converged:

$\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$

$t \sim \mathcal{U}\{1, T\}$

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

Take gradient step on

$$\nabla_\theta \left\| \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\, \boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\varepsilon}, t \right) \right\|^2$$

Connection to SDEs

Forward sampling

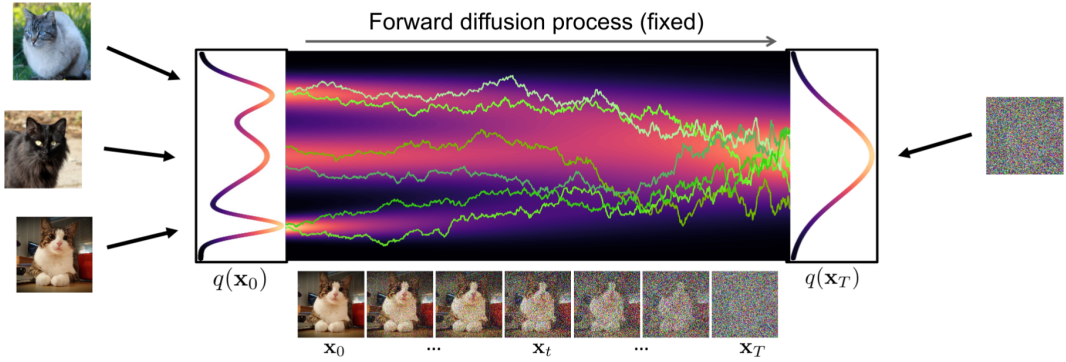
$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\,\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

is a discretization of a stochastic differential equation (SDE):

$$d\boldsymbol{x}_t = \underbrace{-\frac{1}{2}\beta(t)\boldsymbol{x}_t dt}_{\substack{\text{drift term} \\ \text{(pulls towards mode)}}} + \underbrace{\sqrt{\beta(t)}d\boldsymbol{\omega}_t}_{\substack{\text{diffusion term} \\ \text{(injects noise)}}}, \quad d\boldsymbol{\omega}_t = \boldsymbol{\varepsilon}\sqrt{dt}$$

Song et al. Score-Based Generative Modeling through Stochastic Differential Equations. ICLR 2021

Connection to SDEs



$$d\boldsymbol{x}_t = \underbrace{-\frac{1}{2}\beta(t)\boldsymbol{x}_t dt}_{\text{drift term (pulls towards mode)}} + \underbrace{\sqrt{\beta(t)}d\boldsymbol{\omega}_t}_{\text{diffusion term (injects noise)}}, \quad d\boldsymbol{\omega}_t = \boldsymbol{\varepsilon}\sqrt{dt}$$

Generation (reverse process)

Forward SDE has corresponding reverse SDE:

$$d\boldsymbol{x}_t = \left(-\frac{1}{2}\beta(t)\boldsymbol{x}_t - \beta(t)\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \right) dt + \sqrt{\beta(t)}\, d\,\bar{\boldsymbol{\omega}}_t$$

and reverse ODE:

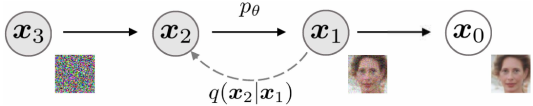
$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\left(\boldsymbol{x}_t + \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\right) dt$$

Consequences:

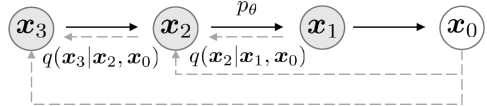
- diffusion process may be generalized through SDE to get other parameterizations
- high-order ODE solvers may be used for accelerating inference
- deterministic encoding and generation

DDIM

DDPM process

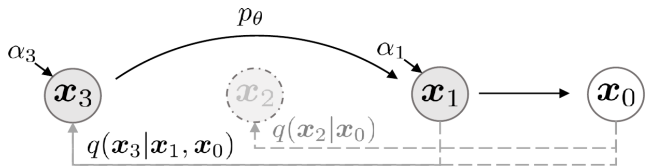


DDIM process



Song et al. Denoising Diffusion Implicit Models. ICLR 2021

DDIM



$$\boldsymbol{x}_j = \sqrt{\alpha_j}\underbrace{\left(\frac{\boldsymbol{x}_i - \sqrt{1-\alpha_i}\cdot\boldsymbol{\epsilon}(\boldsymbol{x}_i)}{\sqrt{\alpha_i}}\right)}_{\text{predicted } \boldsymbol{x}_0} + \underbrace{\sqrt{1-\alpha_j-{\sigma_i}^2}\cdot\boldsymbol{\epsilon}(\boldsymbol{x}_i)}_{\text{move back towards } \boldsymbol{x}_t} + \underbrace{\sigma_i \boldsymbol{z}}_{\text{random noise}}$$

Song et al. Denoising Diffusion Implicit Models. ICLR 2021

Outline

1. Variational autoencoders

2. Vector quantized VAEs

3. Denoising diffusion probabilistic models

4. High-res and conditional generation with diffusion models

Conditional generation

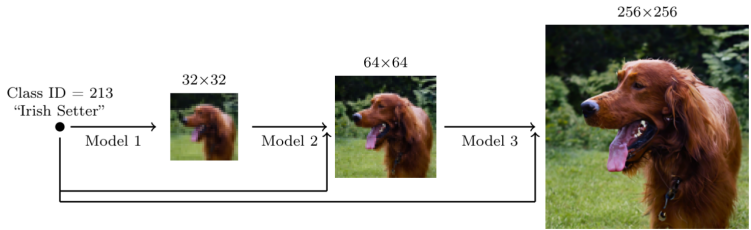
Conditional sampling may be achieved using three methods:

1. Explicit conditional training. Train on pairs $(\boldsymbol{x}, y)$ and modify $\boldsymbol{\varepsilon}_\theta$ to depend on y
2. Classifier guidance. Guide sampling process to class y by adding classifier gradient $\nabla_{\boldsymbol{x}_t} \log p_\phi(y \mid \boldsymbol{x}_t)$ to $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$. Need classifier trained on noisy images
3. Classifier-free guidance. Train conditional and unconditional diffusion model in a single neural network: $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, y, t)$ and $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, y = \varnothing, t)$. Disable conditioning using special null token $\varnothing$. During inference mix noise from conditional and unconditional models. Better than 2 because DM doesn't train to generate adversarial attacks for classifier

Dhariwal, Nichol. Diffusion models beat gans on image synthesis. NeurIPS 2021
Ho, Salimans. Classifier-Free Diffusion Guidance. NeurIPS 2021

Cascaded diffusion models

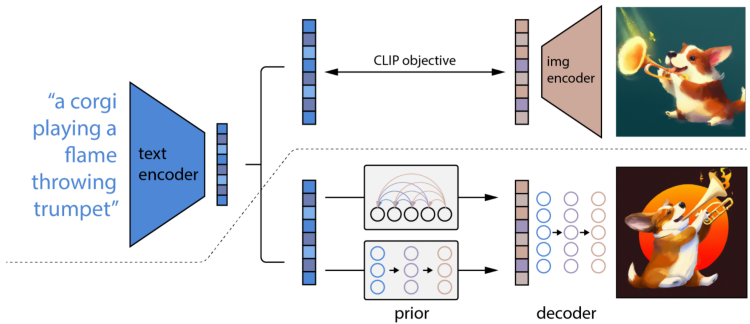
Train 3 conditional models, one for low-res generation from class label and two superresolution models. Augment images with gaussian blur or more complex superres augmentations. Models may be trained in parallel



Condition on class using normalizations, condition on low-res image via concatenation to input tensor

Ho et al. Cascaded Diffusion Models for High Fidelity Image Generation. JMLR 2022
Zhang et al. Designing a Practical Degradation Model for Deep Blind Image Super-Resolution. ICCV 2021

DALL·E 2



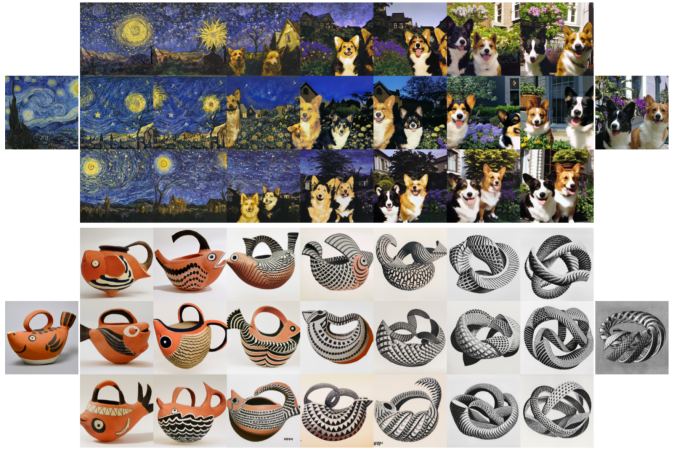
Given pretrained CLIP model, train 2 models:

- diffusion decoders (UNet, 64→256→1024) that produce image x conditioned on CLIP image features z_i using classifier-free guidance
- diffusion prior (transformer) that generates CLIP image features z_i from caption y and CLIP text features z_t

Ramesh et al. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125

Image variations

Image interpolation



Rotate between CLIP image features z_{i_1} and z_{i_2} using spherical interpolation

Text diffs

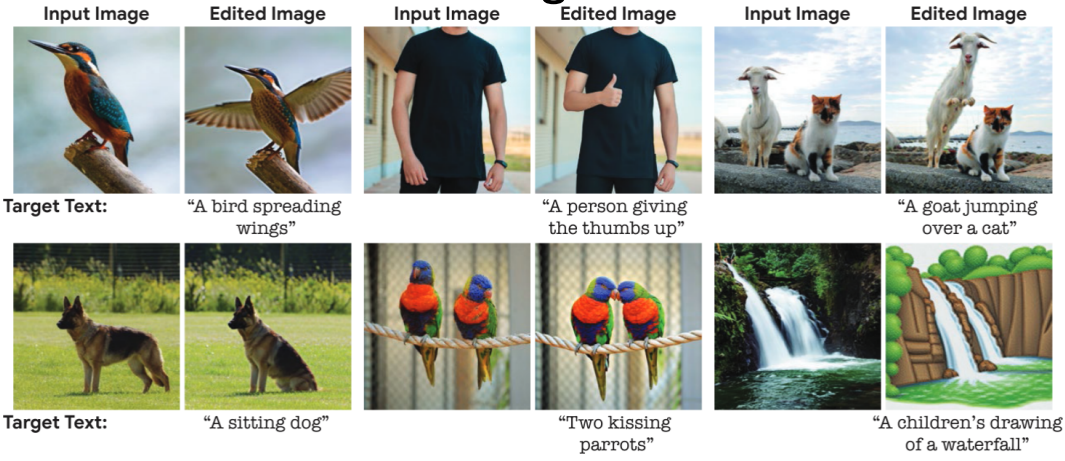
a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

a photo of a landscape in winter → a photo of a landscape in fall

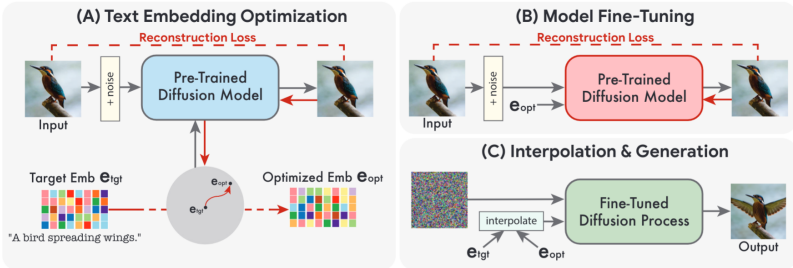
Compute text diff $z_d = \text{norm}(z_t - z_{t_0})$. Spherically interpolate z_i towards z_d

Imagic



Kawar et al. Imagic: Text-Based Real Image Editing with Diffusion Models. CVPR 2023

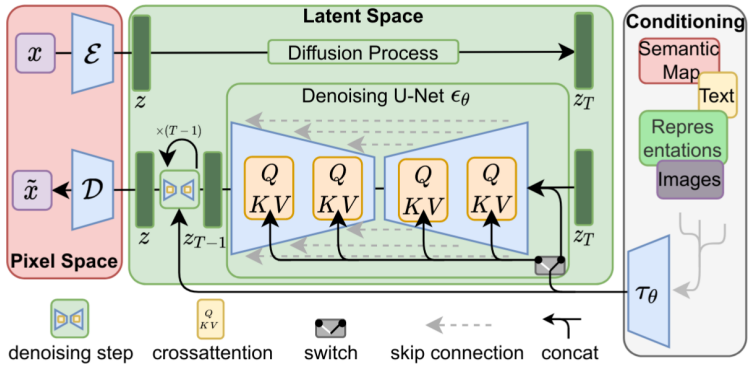
Imagic



- optimize e_{opt} for 100 steps using 64×64 diffusion model
- finetune 64×64 diffusion model and 64→256 SR diffusion model for 1500 steps
- overall it takes 8 minutes on 2×TPUv4 (≈3×A100 GPUs) to transform a single image

Kawar et al. Imagic: Text-Based Real Image Editing with Diffusion Models. CVPR 2023

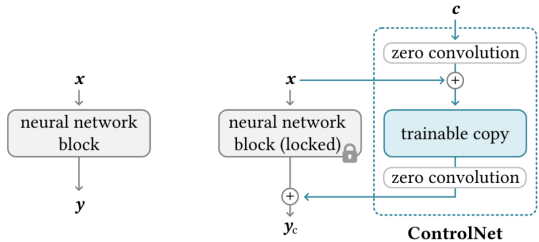
Latent diffusion models



Train an autoencoder (similar to VQ-GAN), fit denoising UNet with attention conditioning to model prior

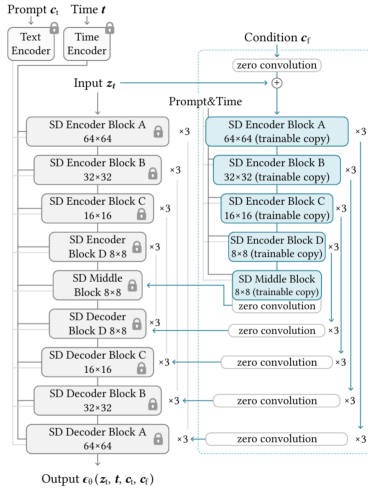
Rombach et al. High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022

ControlNet



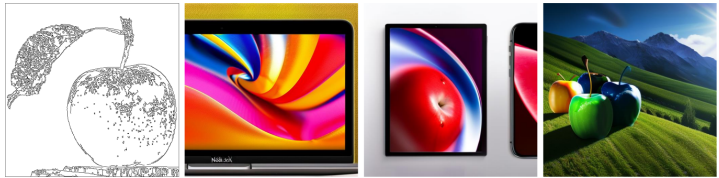
Zhang, Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. ICCV 2023

ControlNet



Zhang, Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. ICCV 2023

ControlNet

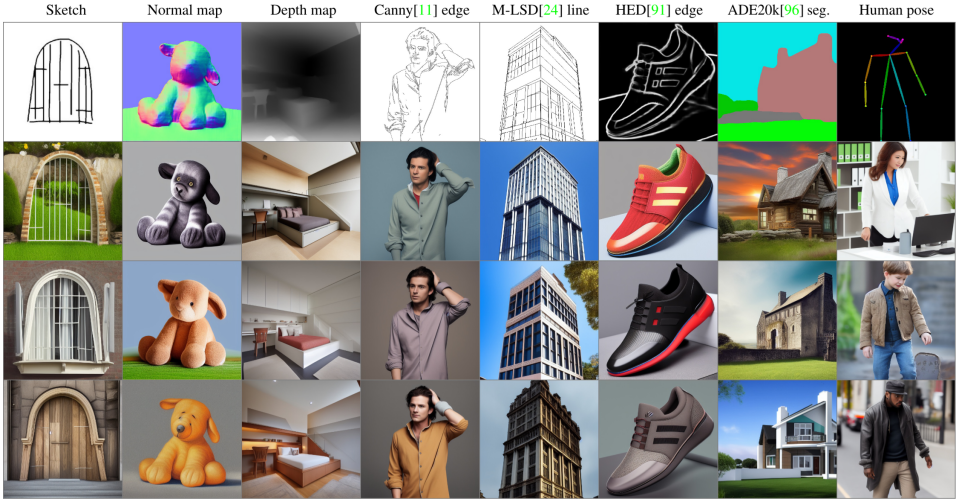


Test input | training step 100 | step 1000 | step 2000

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

Zhang, Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. ICCV 2023

ControlNet



Zhang, Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. ICCV 2023

ControlNet

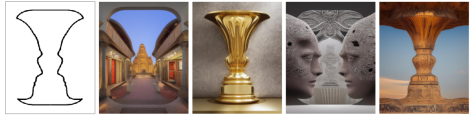


Figure 11: Interpreting contents. If the input is ambiguous and the user does not mention object contents in prompts, the results look like the model tries to interpret input shapes.

Figure 12: Transfer pretrained ControlNets to community models [16, 61] without training the neural networks again.

Zhang, Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. ICCV 2023

Summary

We reviewed following topics:

- variational autoencoders — an important class of probabilistic models that highly influenced the field
- vector-quantized variational autoencoders that improve basic VAE towards high-quality generation
- diffusion models — a modern class of generative models that borrowed several important ideas from GANs and VAEs
- most influential diffusion models are cascaded DMs and LDMs. Basic approaches for conditional generation are classifier-free guidance and ControlNet adapter for lightweight finetuning