Lecture 23: Image Generation
CS486/686 Intro to Artificial Intelligence

2026-3-31

Pascal Poupart
David R. Cheriton School of Computer Science

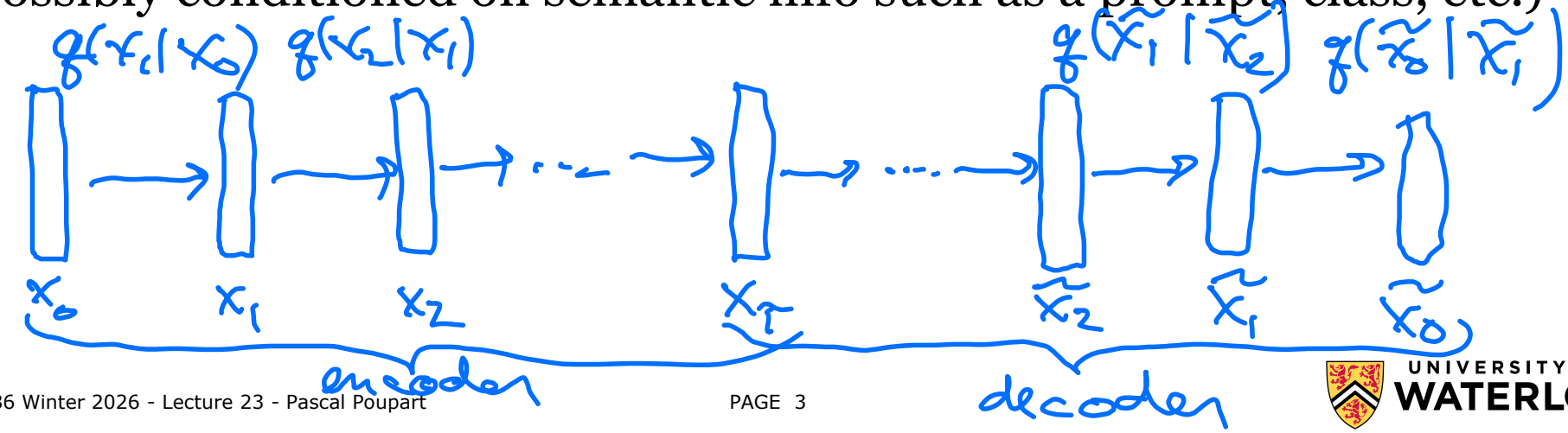
UNIVERSITY OF
WATERLOO

Outline

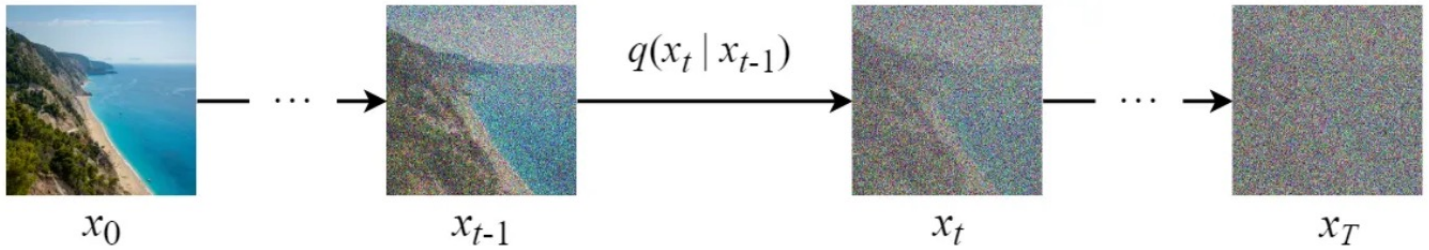
- Diffusion models
- Text-to-Image generation
- Image-Poser: composition of Text-to-Image and Image-to-Image models

Diffusion Model

- Leading model to generate images
- Stochastic autoencoder
  - encoder introduces noise
  - decoder denoises the data
- Stochastic decoder transforms a noisy vector into a desired image (possibly conditioned on semantic info such as a prompt, class, etc.)



Forward Diffusion Process



Distribution of the noised images — Output — Mean μ_t — Variance Σ_t

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t;\ \sqrt{1-\beta_t}x_{t-1},\ \beta_t I)$$

Notations:

t : time step (from 0 to T)

x_0 : a data sampled from the real data distrition $q(x)$ (i.e. $x_0 \sim q(x)$)

β_t : variance schedule ($0 \leq \beta_t \leq 1$, and β_0 = small number, β_T = large number)

I : identity matrix

From Steins (medium.com)

Stochastic Transformation

- Recall the reparameterization trick:
  - When $x \sim P(x) = N(x|\mu, \sigma^2)$
    then $x = \sigma\epsilon + \mu$ where $\epsilon \sim N(\epsilon|0,1)$

- Since $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = N\left(\boldsymbol{x}_t\middle|\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}\right)$
  Then $\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$

Stochastic Transformation

- We can speed up the noise process by computing x_t in one step:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}|\mathbf{0},\mathbf{1})$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i \text{ and } \alpha_i = 1 - \beta_i$$

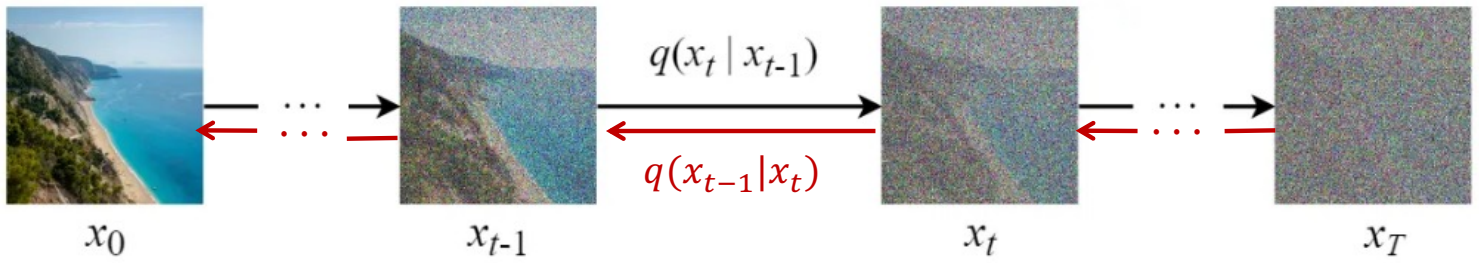
See https://medium.com/@steinsfu/diffusion-model-clearly-explained-cd331bd41166 for derivation

- In the limit, $\boldsymbol{x}_\infty$ is a random vector from an isotropic Gaussian

$$\lim_{t\to\infty} \boldsymbol{x}_t = \sqrt{\bar{\alpha}_\infty}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_\infty}\boldsymbol{\epsilon} = \boldsymbol{\epsilon} \text{ since } \bar{\alpha}_\infty \to 0$$

Reverse Denoising Process

From Steins (medium.com)



- Forward factorization: $q(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = q(\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$
- Reverse factorization: $q(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = \prod_{t=1}^{T} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\, q(\boldsymbol{x}_T)$
  - Since joint distribution is Gaussian then $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is also Gaussian
  - $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(\boldsymbol{x}_{t-1}|\tilde{\mu}_t(\boldsymbol{x}_t, t), \sigma_t \boldsymbol{I})$

Reverse Conditional Gaussian

- The reverse conditional $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(\boldsymbol{x}_{t-1}|\tilde{\mu}_t(\boldsymbol{x}_t, t), \sigma_t \boldsymbol{I})$ does not have a closed form, but Ho, Jain and Abbeel (2020) derived the following approximation for $\tilde{\mu}_t$:

$$\tilde{\mu}_t(\boldsymbol{x}_t, t) \approx \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$$

where $\boldsymbol{\epsilon}_t = N(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$ is the noise introduced at step t

- We do not know $\boldsymbol{\epsilon}_t$, but we can train a neural network $\epsilon_\theta(\boldsymbol{x}_t, t)$ to approximate it:

$$\text{Minimize } L(\theta) = \left|\left|\boldsymbol{\epsilon}_t - \epsilon_\theta(\boldsymbol{x}_t, t)\right|\right|_2^2$$

Training Algorithm

Repeat

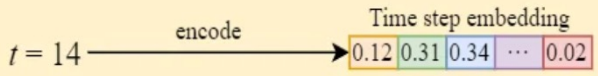
- $\boldsymbol{x}_0 \sim q(\boldsymbol{x_0})$
- $t \sim uniform(\{1, \dots, T\})$
- $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{I})$
- $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$
- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} \left|\left|\epsilon - \epsilon_\theta(\boldsymbol{x}_t, t)\right|\right|_2^2$

Until convergence

Training Algorithm

For each training step:

1. Randomly select a time step & encode it



2. Add noise to image

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon$$

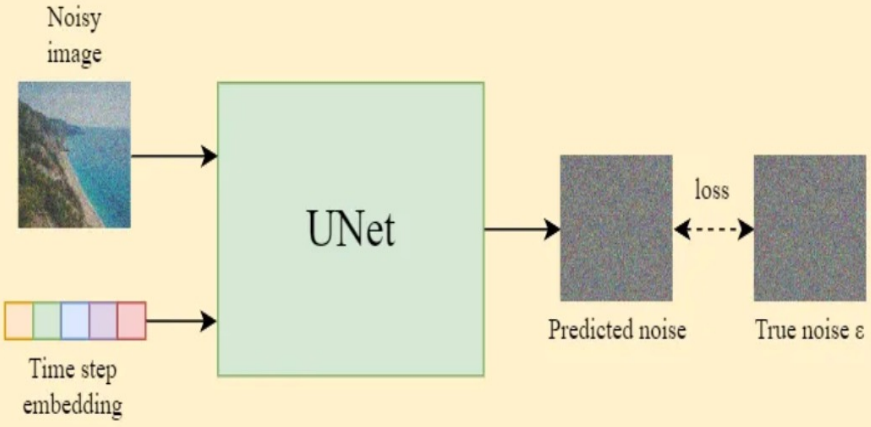
Adjust the amount of noise according to the time step t

$$\varepsilon \sim \mathcal{N}(0, 1)$$

$$\alpha_t = 1 - \beta_t$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

3. Train the UNet

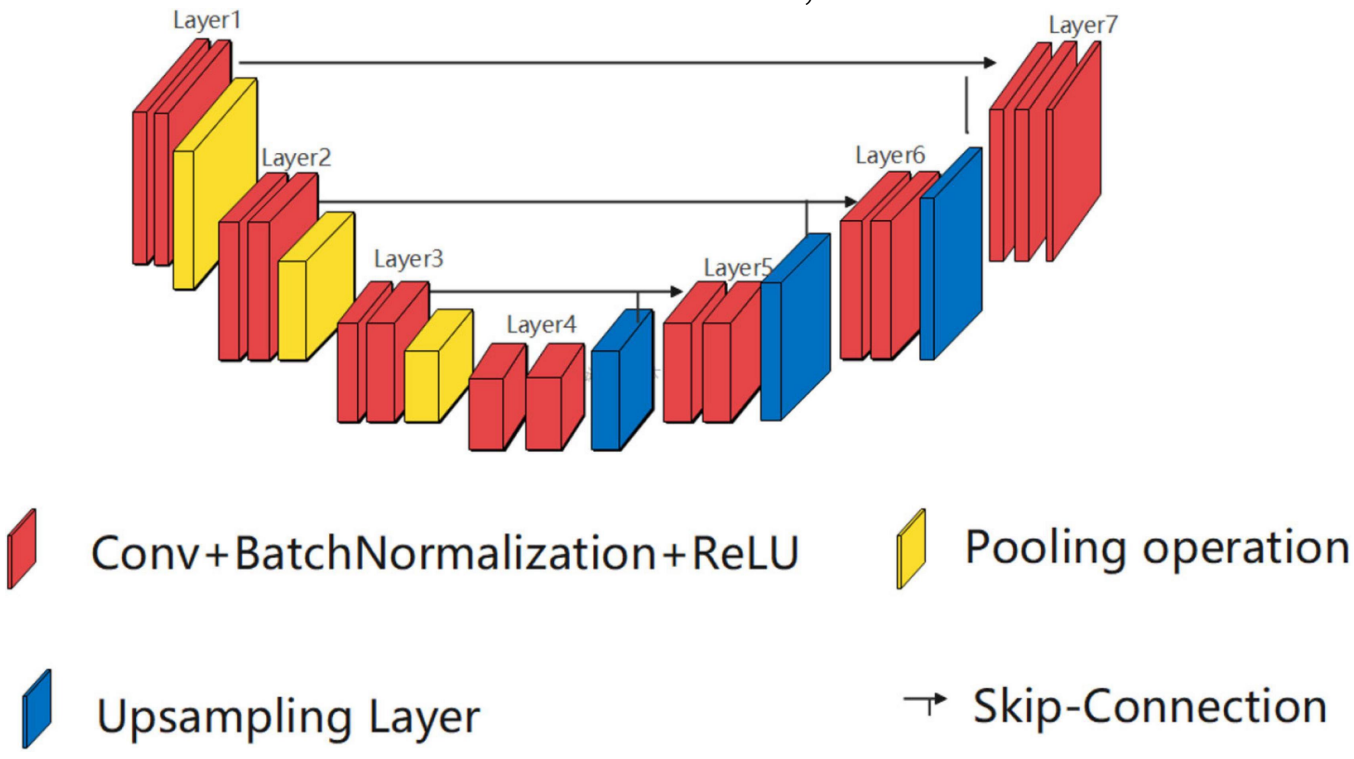


From Steins (medium.com)

UNet

Special type of fully convolutional neural network

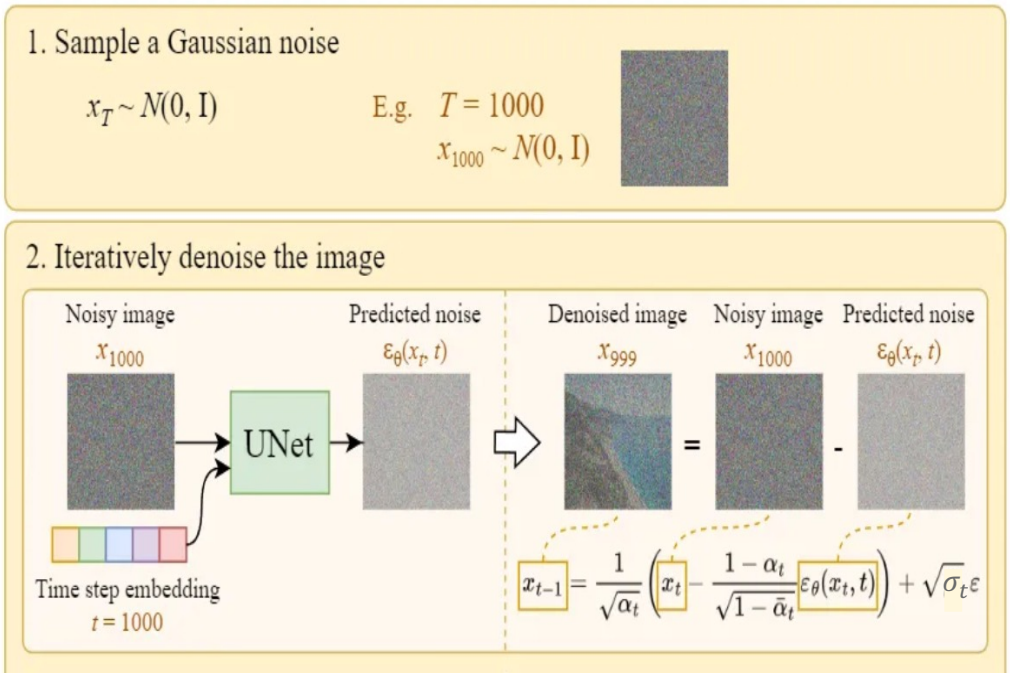
From Cai, Wu and Chen 2022



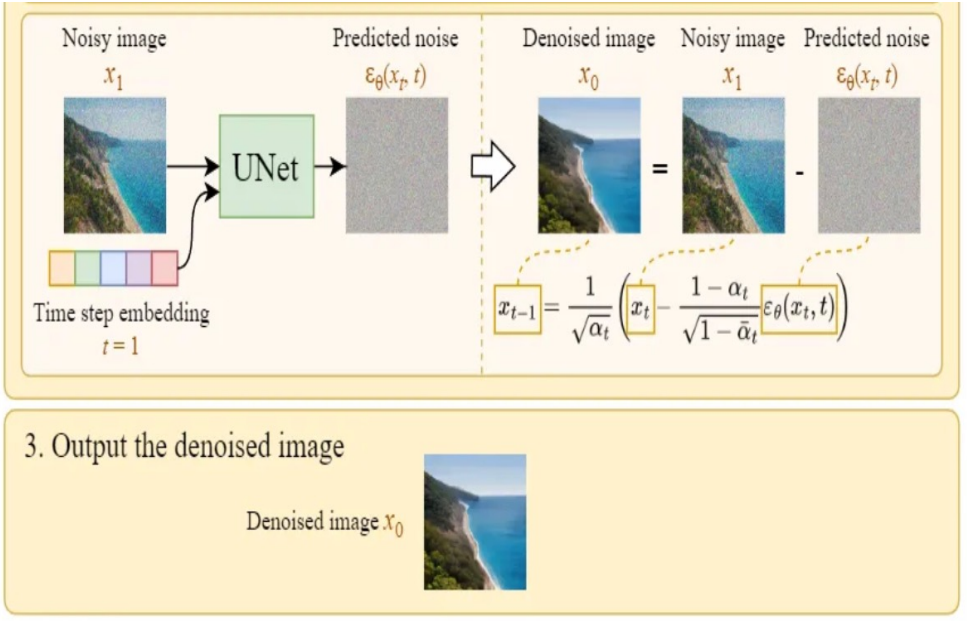
Data Generation Algorithm

- $\boldsymbol{x}_T \sim N(\boldsymbol{x}|\boldsymbol{0}, \boldsymbol{I})$
- For $t = T, \ldots, 1$ do
  - $\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{\epsilon} = \boldsymbol{0}$
  - $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t)\right) + \sqrt{\sigma_t}\boldsymbol{\epsilon}$
- Return $\boldsymbol{x}_0$

Data Generation Algorithm



. . .



From Steins (medium.com)

Results

Ho, Jain and Abbeel (2020)

Figure 3: LSUN Church samples. FID=7.89

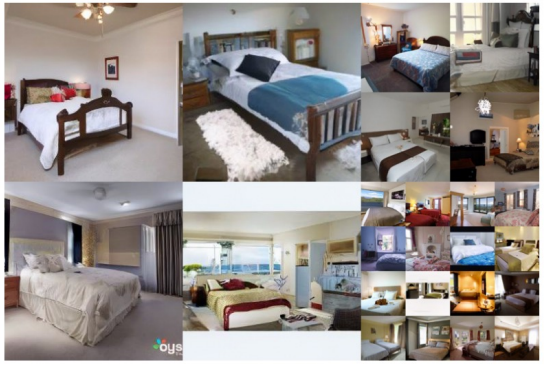
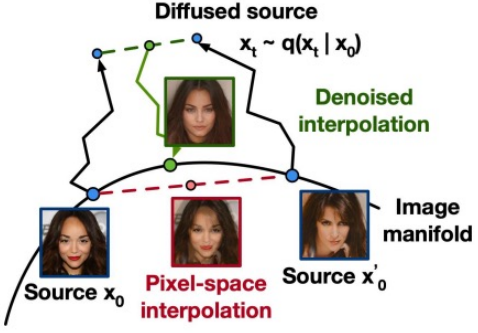


Figure 4: LSUN Bedroom samples. FID=4.90



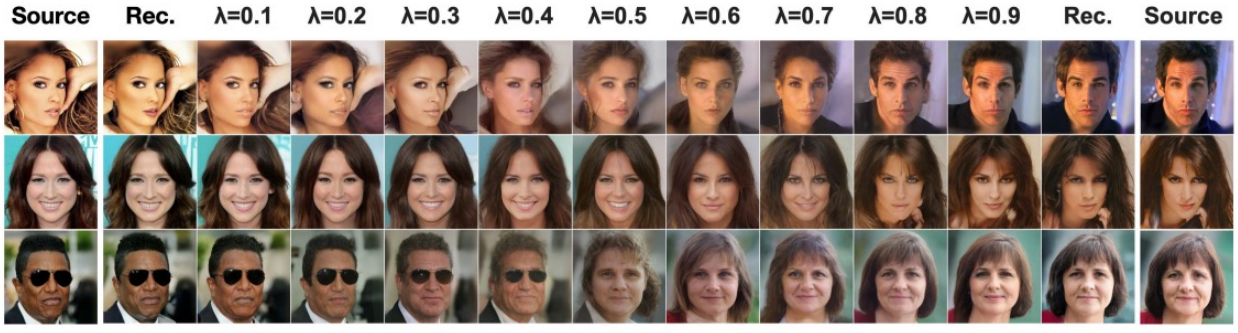
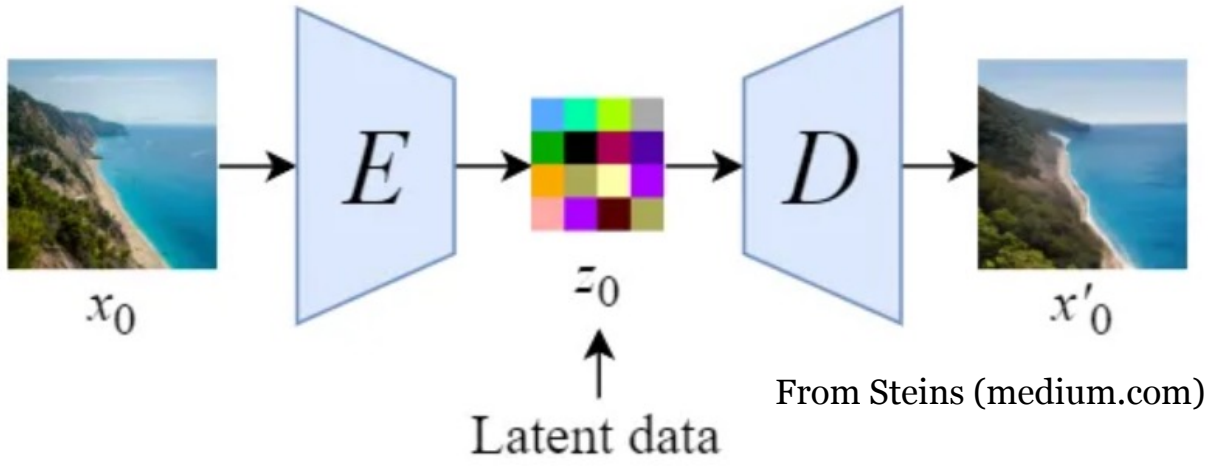


Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

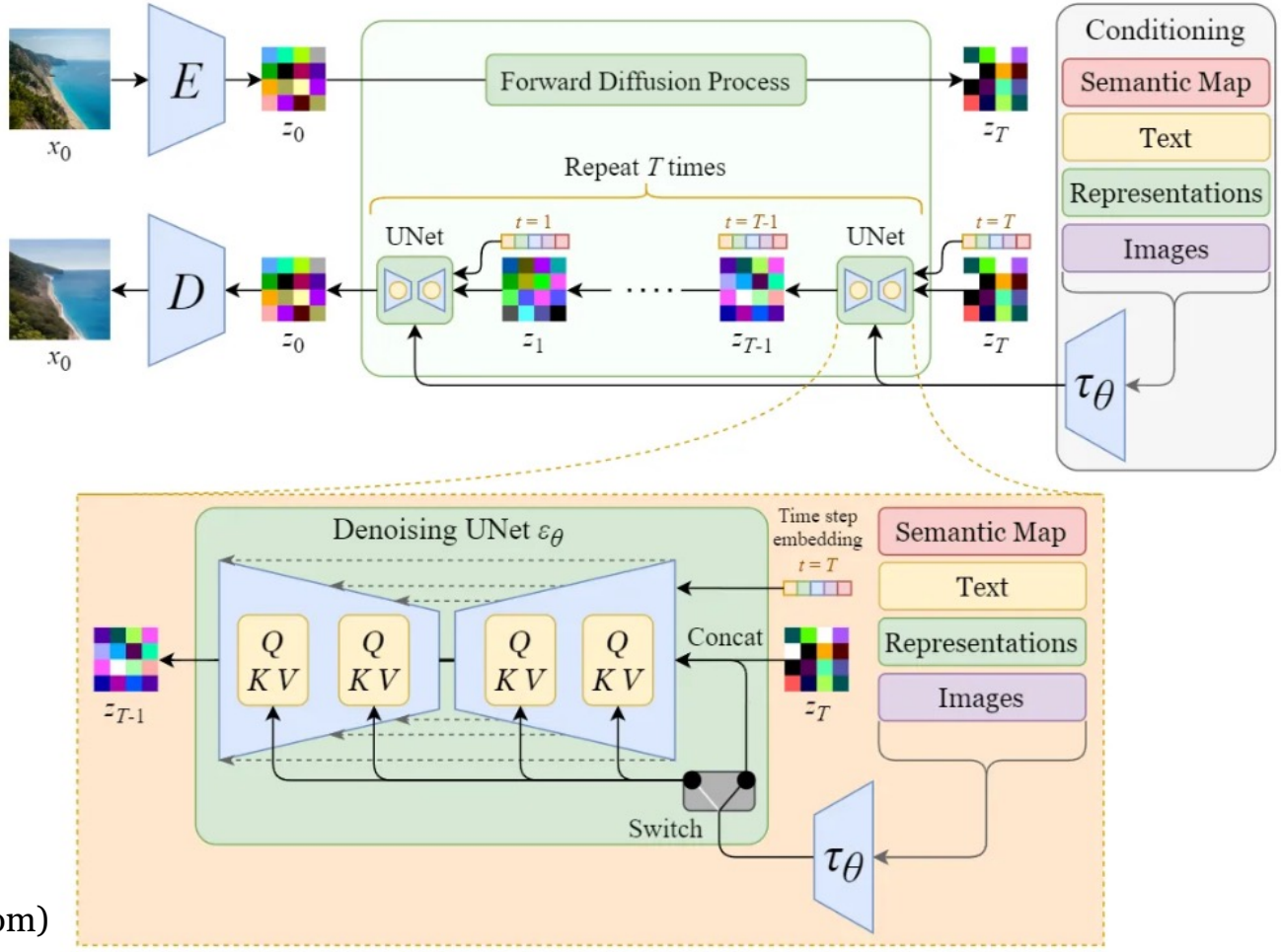
Latent Diffusion Model (a.k.a. Stable Diffusion)

- Rombach, Blattman et al., 2022
- **Speed up**: performing the diffusion in a low dimensional latent space
- **Conditional generation:** condition denoising on text, images, etc.



From Steins (medium.com)

Full Architecture



From Steins (medium.com)

Results

- Rombach, Blattman et al., 2022

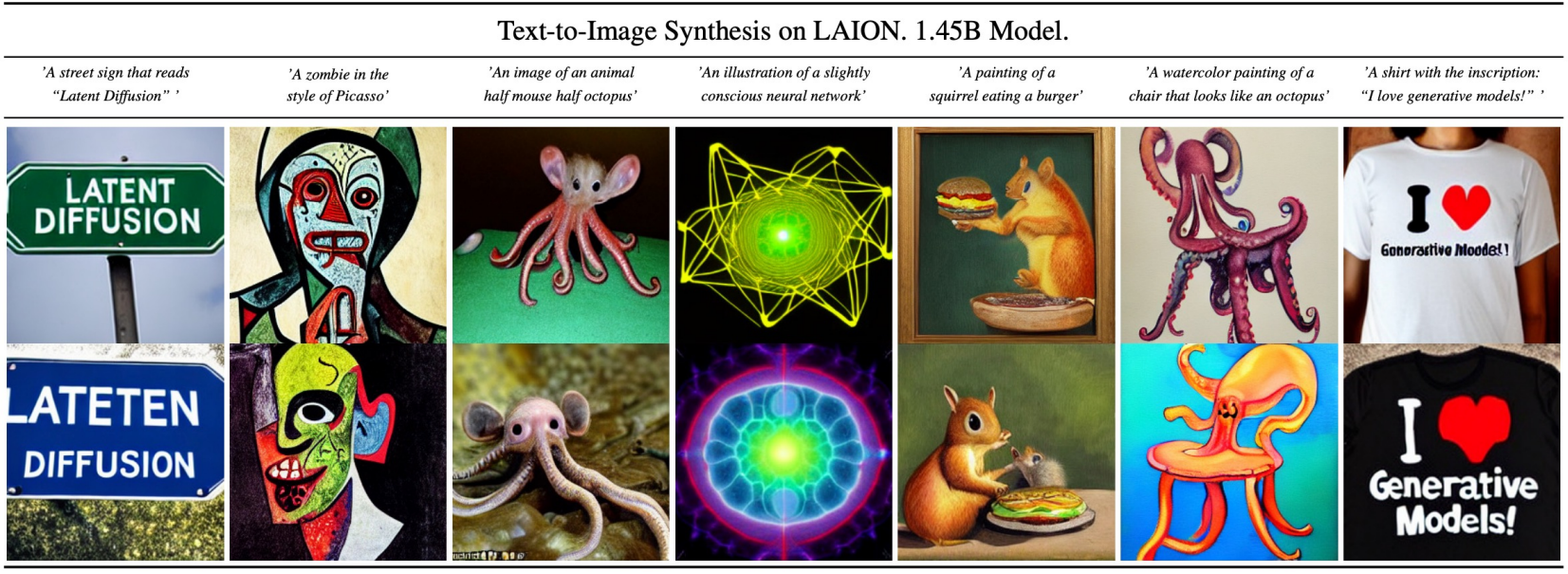
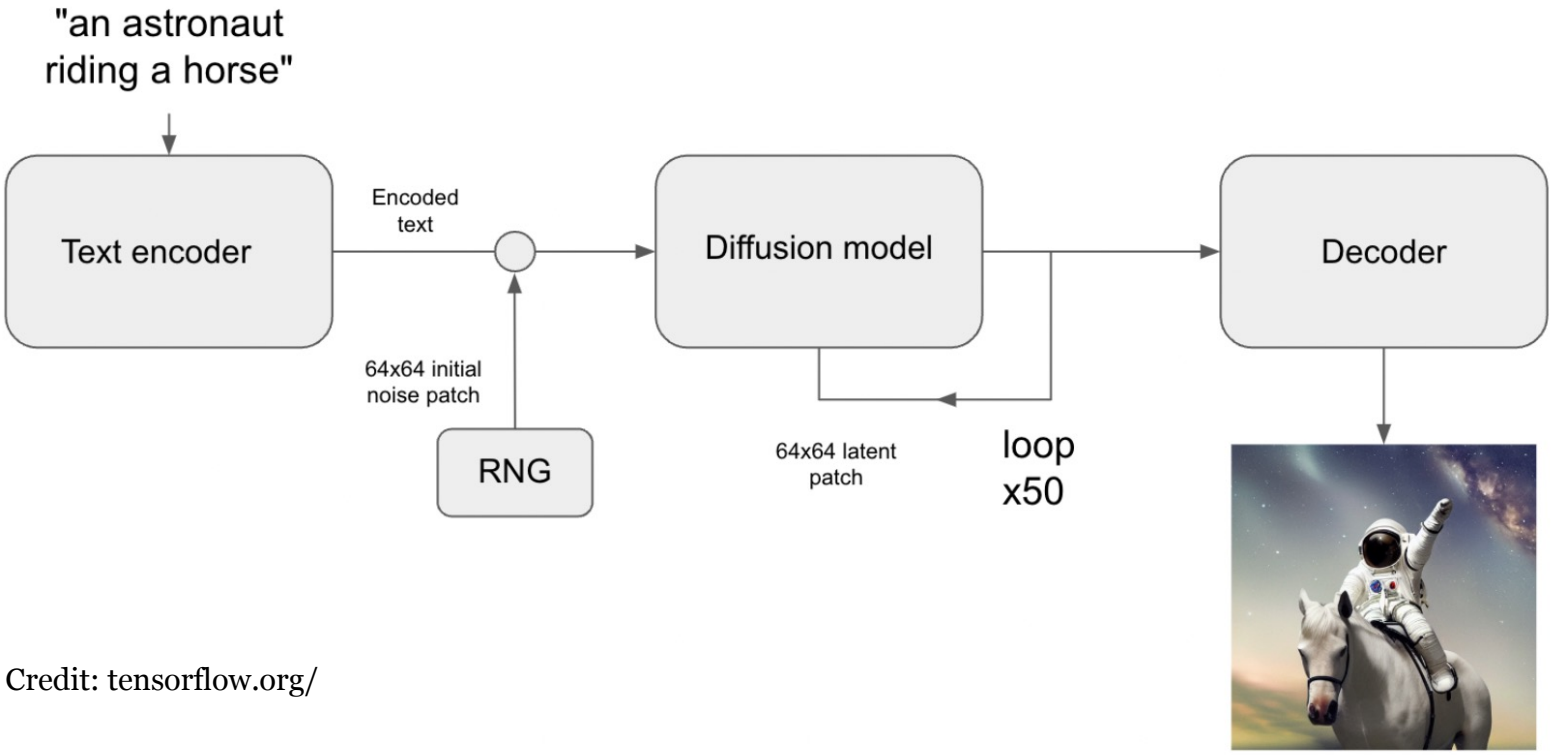


Figure 5. Samples for user-defined text prompts from our model for text-to-image synthesis, *LDM-8 (KL)*, which was trained on the LAION [78] database. Samples generated with 200 DDIM steps and $\eta = 1.0$. We use unconditional guidance [32] with $s = 10.0$.

Text to Image Generation



Credit: tensorflow.org/

Many Image Models

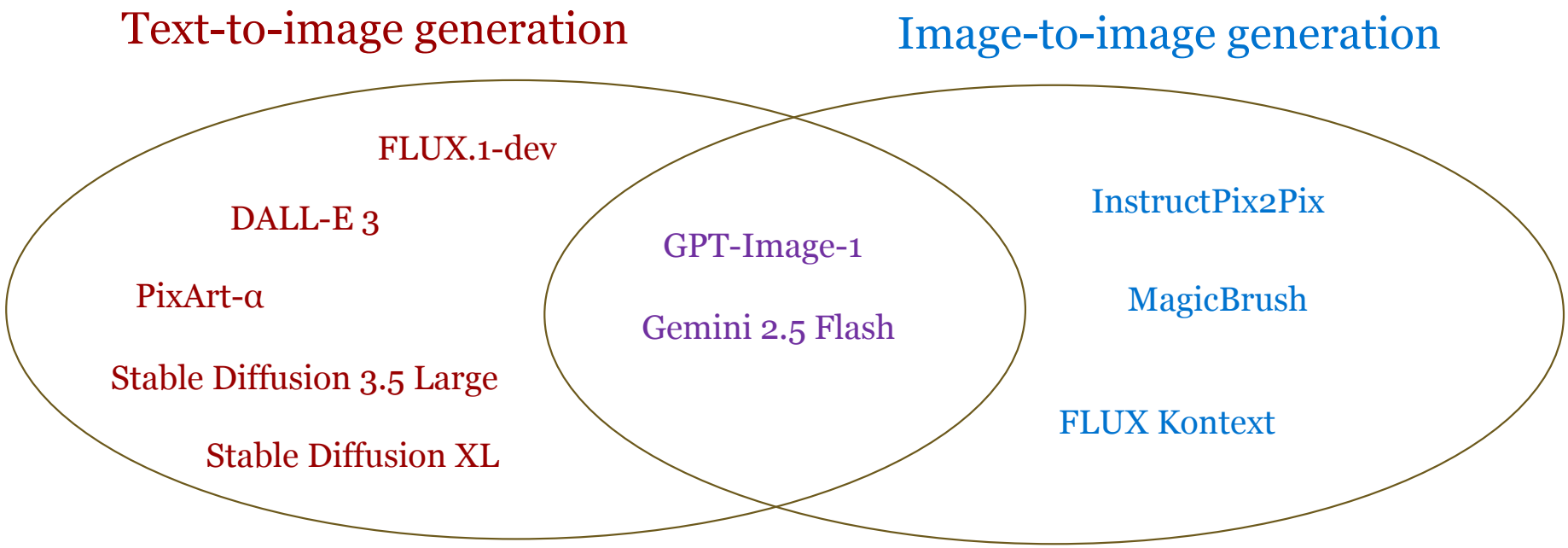
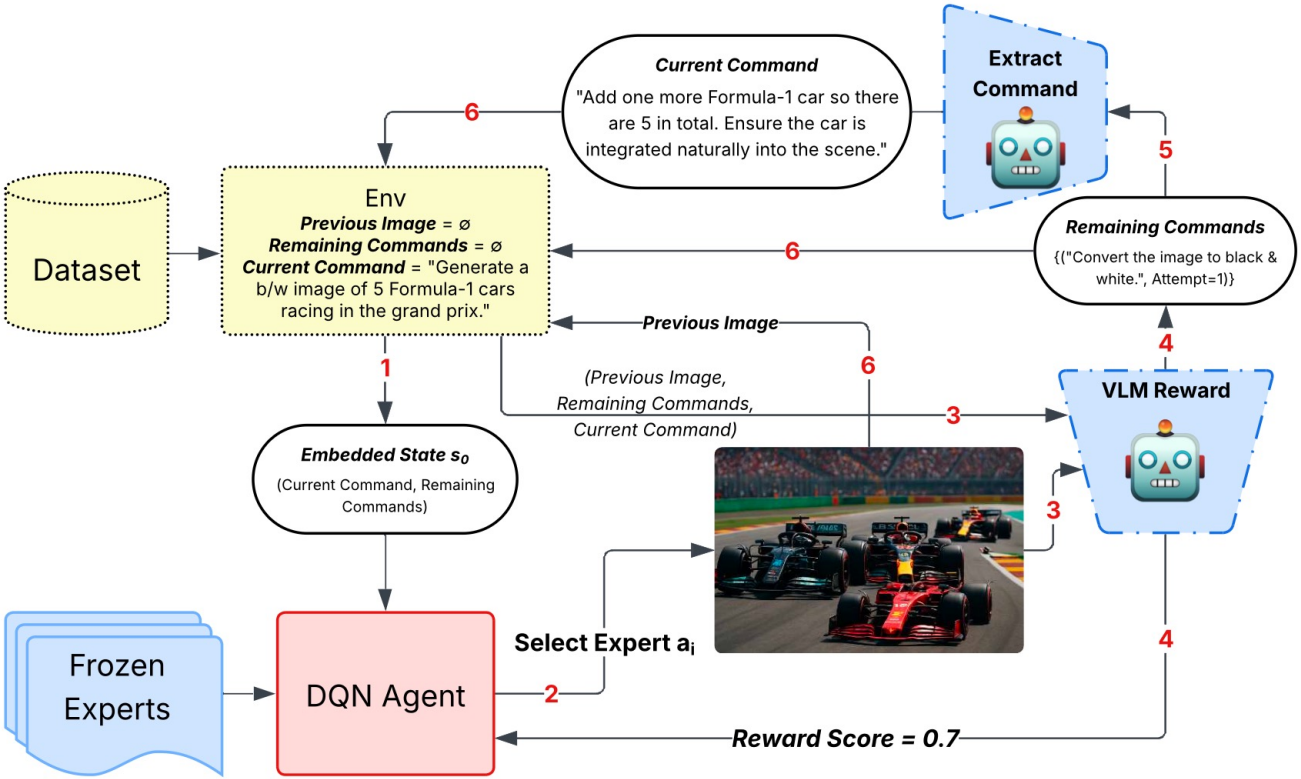


Image-POSER

Mohebbi, Abdulrahman, Miao, Poupart, Kothawade (2026) **Image-POSER: Reflective RL for Multi-Expert Image Generation and Editing,** under review.



UNIVERSITY OF WATERLOO

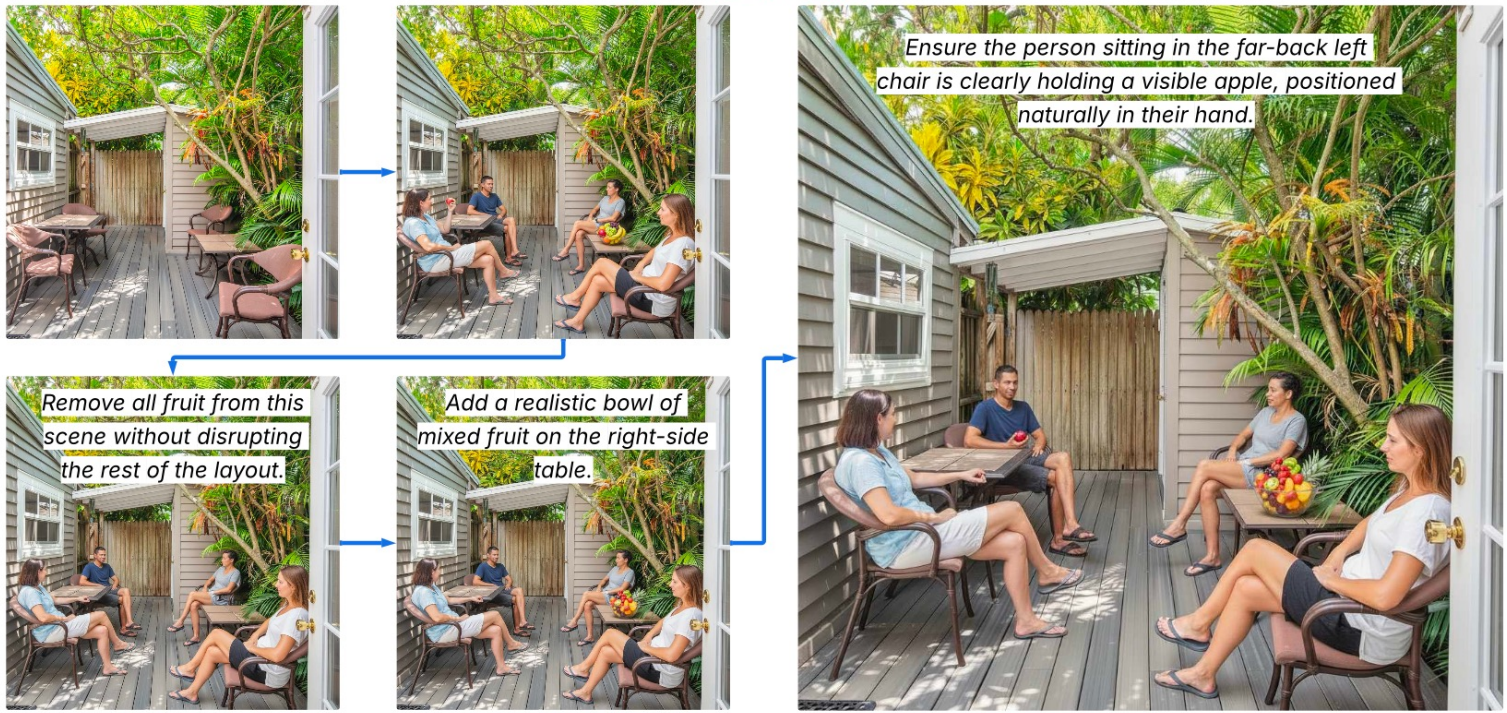
RL for model selection

- State: embedding([current command, remaining commends])
- Actions: set of text-to-image and image-to-image models
  - E.g., {FLUX-1.dev, DALL-E, Pix-Art, Stable Diffusion, GPT-Image, Gemini, InstructPix2Pix, Magic Brush, FLUX-Kontext}
- Reward function: Vision Language Model that critiques and rates each image

- DQN: (state, action) → total reward

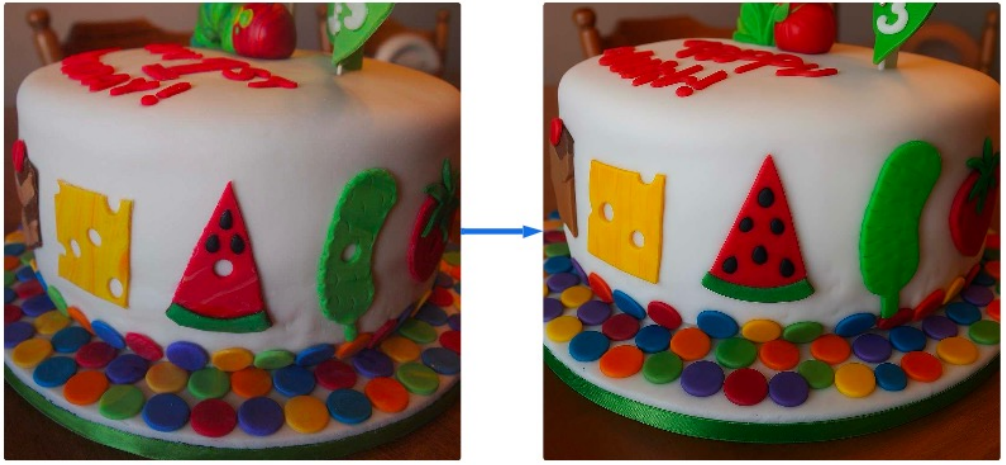
Image-to-Image Generation Example

Add a person sitting in each of the four empty chairs without changing the layout of any of the chairs, tables or background. Add a bowl of fruit on top of the table to the right. Make only the person sitting in the far back left chair hold an apple.

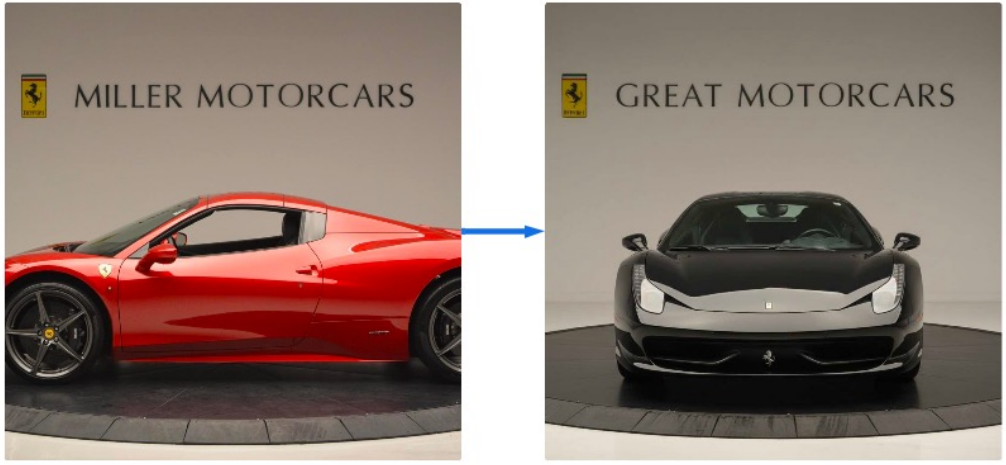


Additional Examples

Fill in the empty white hole on the center watermelon slice as well as the cucumber on the right of the cake. Add three more black seeds to the watermelom for a total of six. Maintain the rest of the image and keep the cake and other compositions the same.



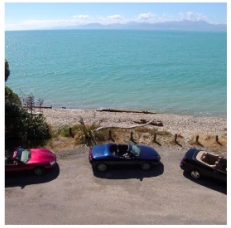
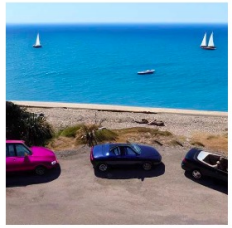
Rotate the Ferrari so that the car is facing forwards and we see the front. Make the car black instead of red. Change the title above it to be 'GREAT MOTORCARS' instead of 'MILLER MOTORCARS' while keeping the same font.



Text-to-Image Qualitative Comparison

Prompt	PixArt-α	SD 3.5 Large	FLUX.1-dev	Gemini 2.5 Flash	GPT Image 1	Image-POSER
A basketball court in the middle of New York City. Six players are playing 3v3. There are also two spectators watching. One player is mid-dunk in this sequence.						
A soccer field under a bright noon sun. The scoreboard in the back reads 'HOME 3 : AWAY 1' in amber LEDs. A midfielder in red jersey #7 is mid-kick, right leg fully extended; a defender in blue jersey slides in from the lower left.						
Six paper boats float in a baking tray full of marbles, while a red pepper mill to the right of a blue mug watches the regatta.						

Image-to-Image Qualitative Comparison

Input Image	Instruction	MagicBrush	FLUX Kontext	Gemini 2.5 Flash	GPT Image 1	Image-POSER
	Change the colors of the cars in the middle and on the right to match the car on the left. Add six sailboats out in the distant ocean background.					
	Change the color of the kitchen cabinets to be vintage green. Add a stuffed turkey cooking inside the oven visible from the window. Add a stack of exactly three pots on the kitchen countertop and one single frying pan on the stove.					
	Apply an animated style to this image. Remove the existing bills and replace them with three new bills poking out of the wallet. The bills should read, '$100', '$20', and '$50' in that sequence. All the bills should be green.					

Human Evaluation

Table 3: **Win Rates from User Study.** Table shows the average rate at which annotators preferred Image-POSER's outputs over a given baseline (higher is better for Image-POSER).

Text-to-Image								
SD 3.5 Large	**SD XL**	**DALL-E 3**	**FLUX.1**	**GoT-R1-7B**	**PixArt-α**	**GenArtist**	**GPT Image 1**	**Gemini 2.5 Flash**
0.80 ± 0.07	0.97 ± 0.03	0.80 ± 0.07	0.80 ± 0.07	0.97 ± 0.03	0.93 ± 0.05	0.93 ± 0.05	0.67 ± 0.09	0.57 ± 0.09

Image-to-Image				
MagicBrush	**InstructPix2Pix**	**FLUX Kontext**	**GPT Image 1**	**Gemini 2.5 Flash**
0.97 ± 0.03	0.97 ± 0.03	0.80 ± 0.07	0.67 ± 0.09	0.60 ± 0.09