Multimodal AI

Lecture 6.2 – More Multimodal Generative AI

Paul Liang
Assistant Professor
MIT Media Lab & MIT EECS

https://pliang279.github.io
ppliang@mit.edu
@pliang279

mit media lab

MIT EECS

Assignments for This Coming Week

David sent out announcement regarding course credits.

Edgar sent out announcement regarding midterm conflicts/accommodations.

Midterm review next Tuesday 3/17. Midterm exam next Thursday 3/19.

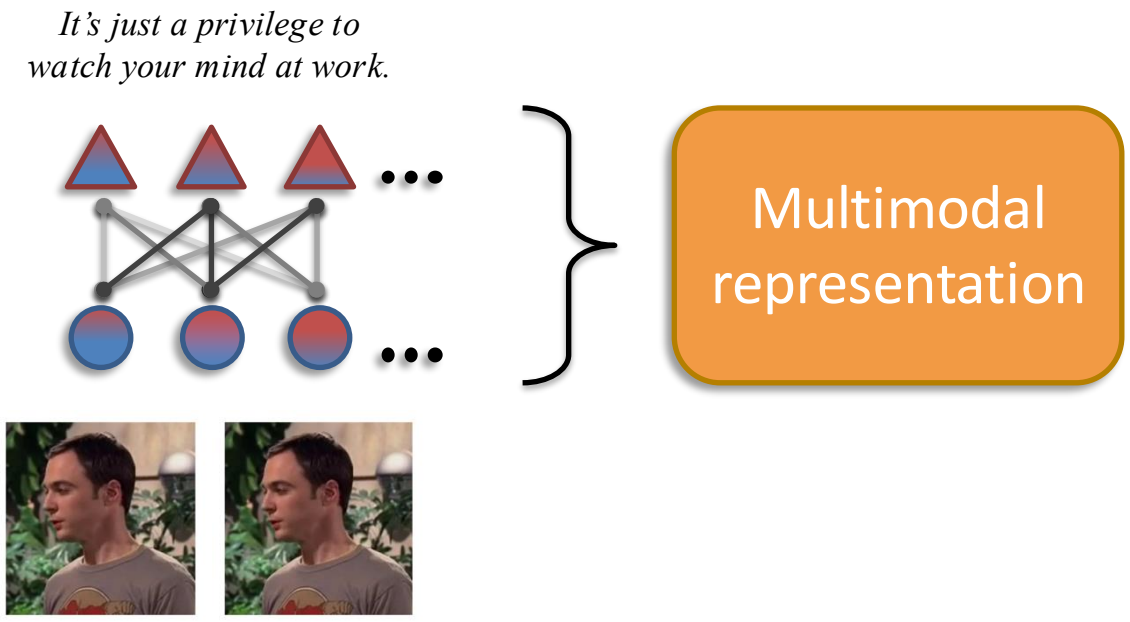
Project mentors released. Try to meet as often as you can. Meet with me today.

Today's lecture

1. Text-to-image generation
2. Introduction to diffusion and flow matching
3. Diffusion models for other modalities
4. Controlling generative models

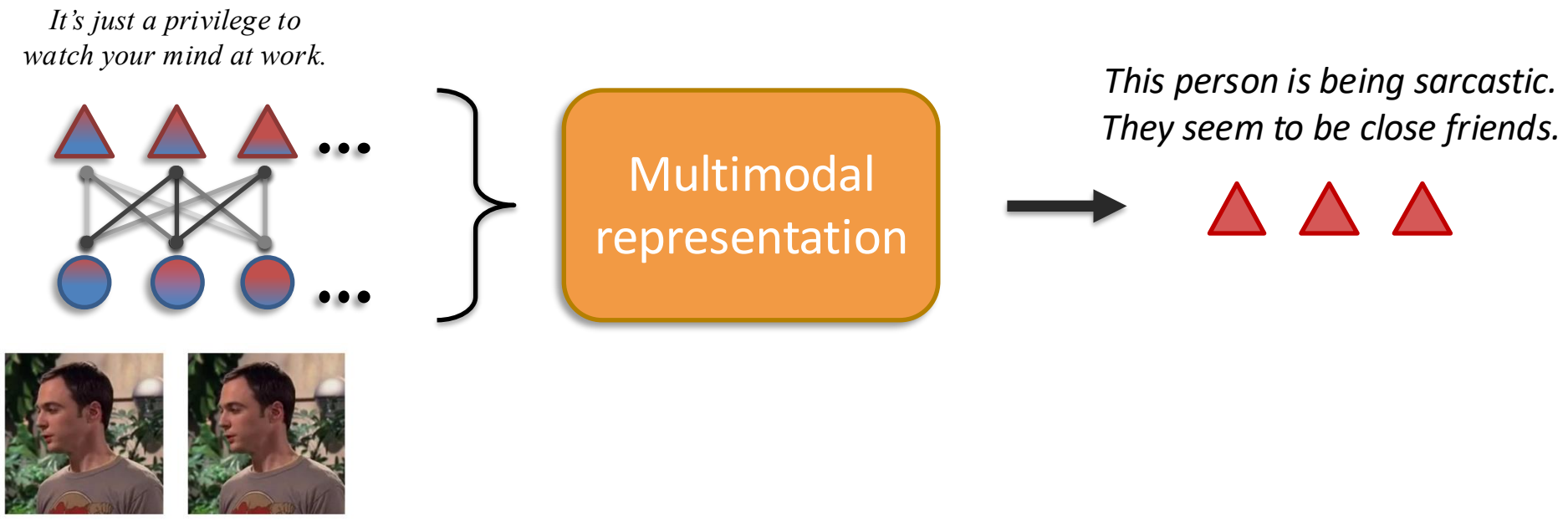
Lecture outline

Part 1: Multimodal foundation model representations of text, video, audio



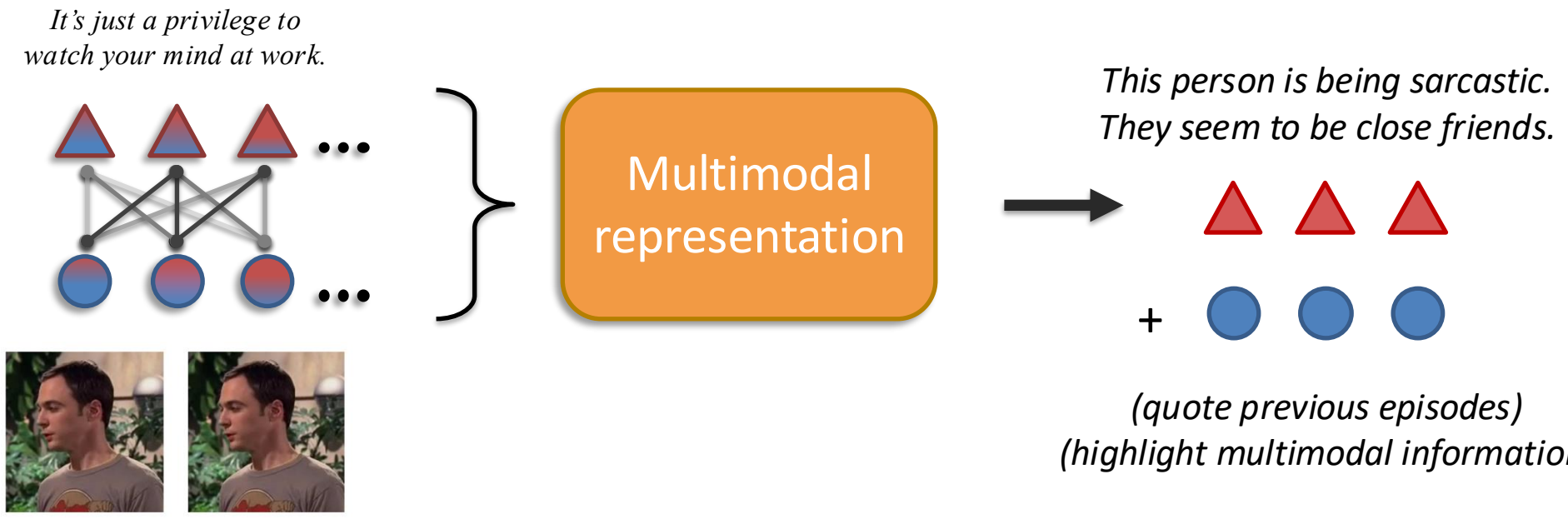
Lecture outline

Part 2: Adapting large language models for multimodal text generation



Lecture outline

Part 3: Enabling text and image generation

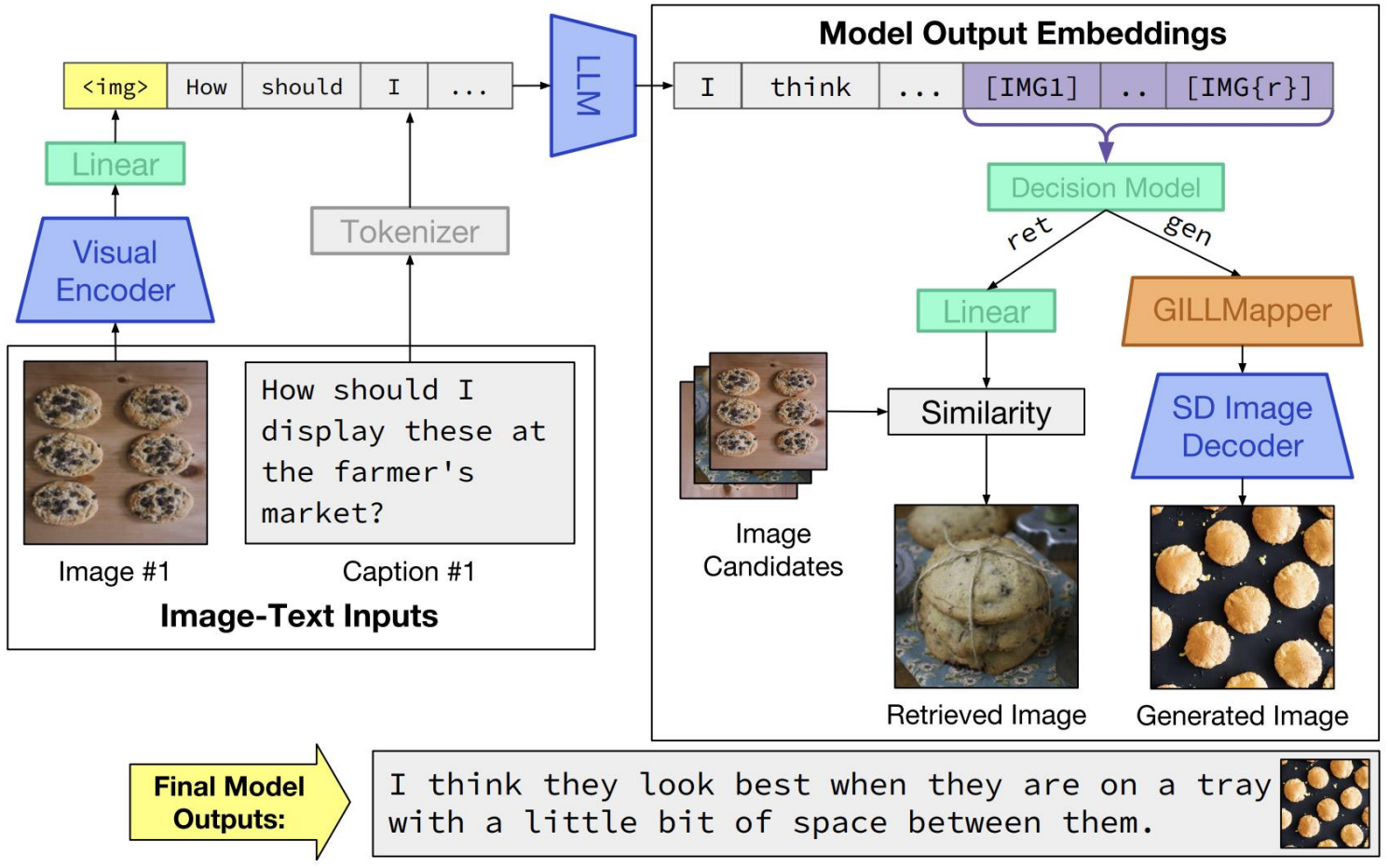


1. Text-to-image retrieval

2. Text-to-image generation

Grounding LMs for Multimodal Generation

Large multimodal models with image generation



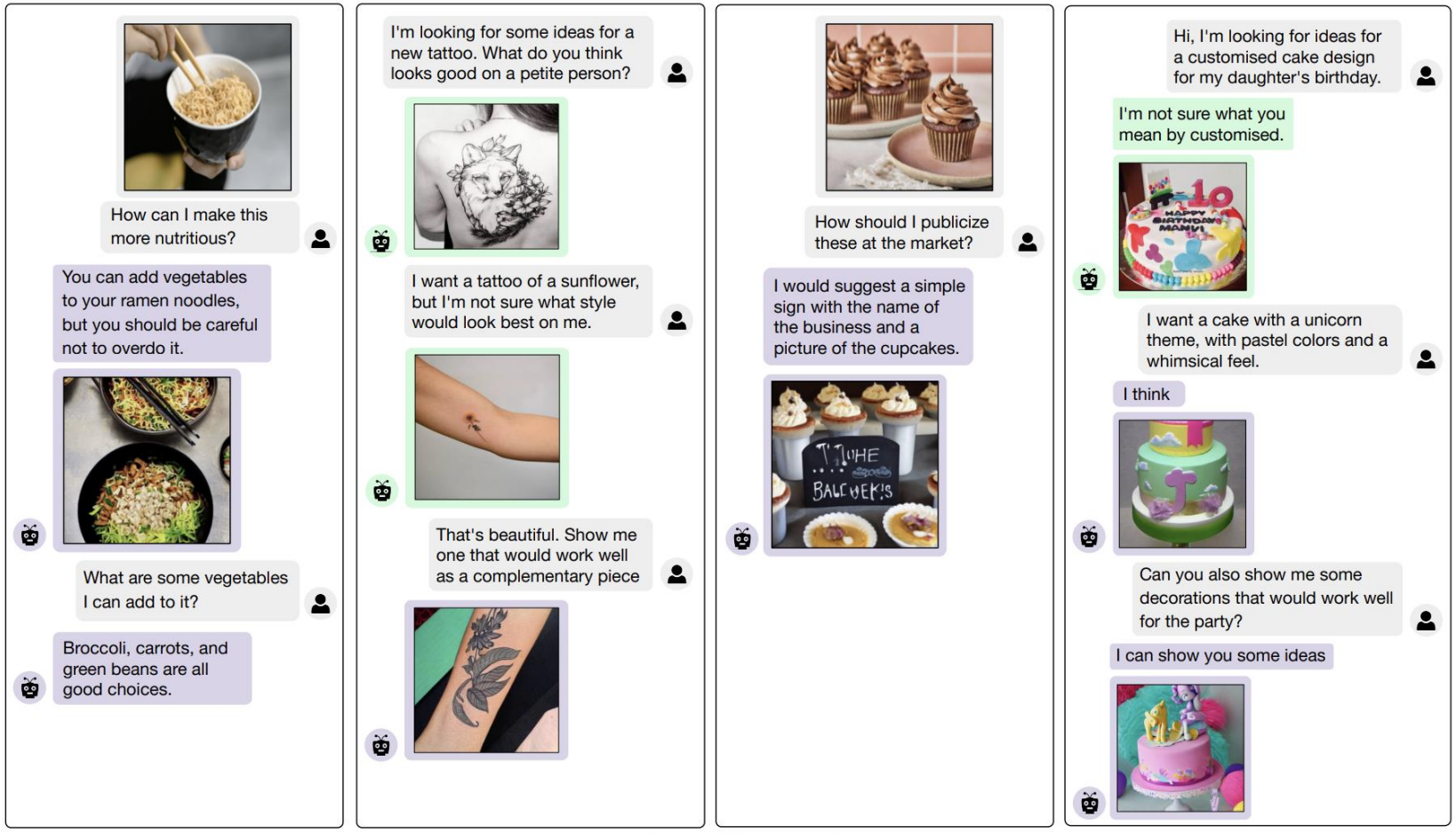
[Koh et al., Generating Images with Multimodal Language Models. NeurIPS 2023]

multisensory intelligence

Grounding LMs for Multimodal Generation

Visual Storytelling

Our model can condition on interleaved image-and-text inputs to generate more relevant images compared to non-LLM based text-to-image generation models.



Multimodal Dialogue

Our model can generate multimodal dialogue, weaving together text, retrieved images, and generated images.

User prompts Retrieved Generated

multisensory intelligence

Text to Image Generation

An armchair in the shape of an avocado →

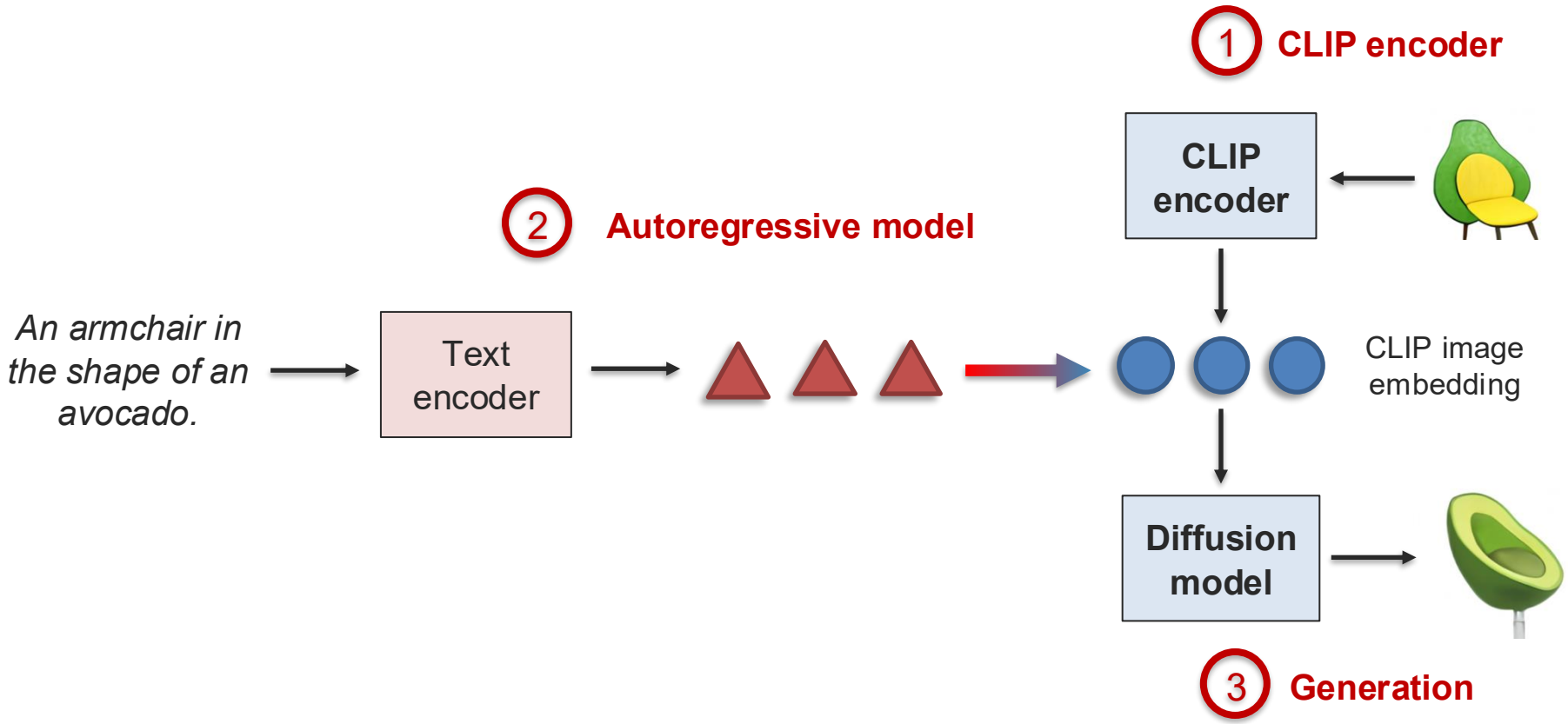
[*DALL-E.* Ramesh et al., Zero-Shot Text-to-Image Generation. ICML 2021]
[see also, Esser et al. Taming Transformers for High Resolution Image Synthesis. CVPR 2021]

multisensory intelligence

From Text to Multimodal Generation

Directly training diffusion models with conditional information

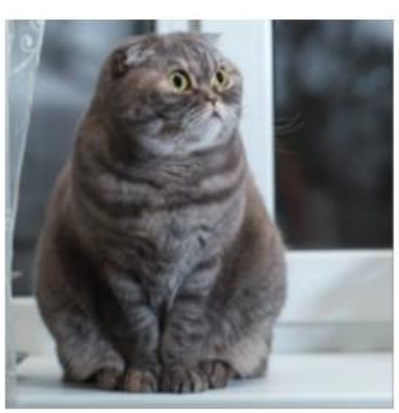
Conditional latent variables are pretrained CLIP embeddings, then diffusion model to generate image.

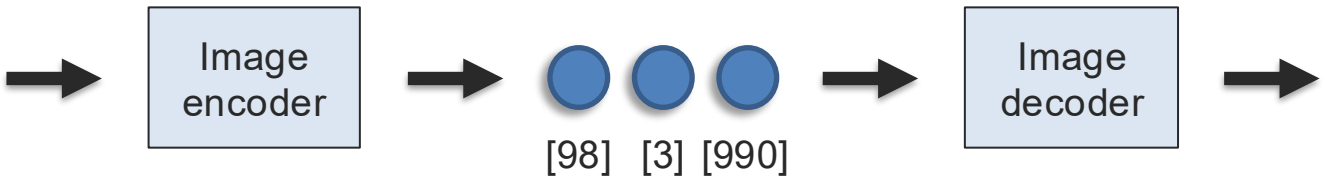


[Ramesh et al., Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv 2022]

From Text to Multimodal Generation

(1) Using a discrete variational autoencoder to learn discrete visual tokens

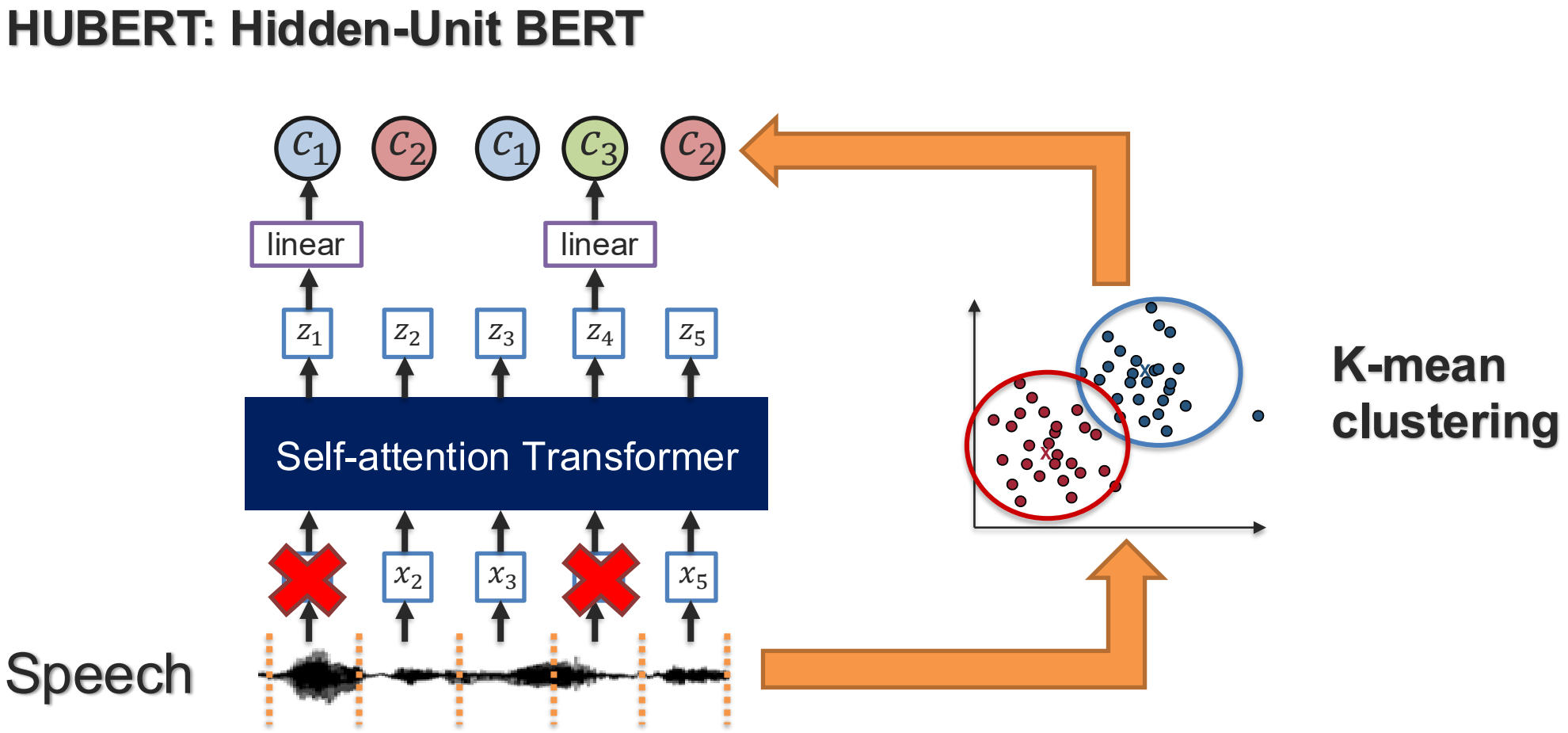




List of digits, [0… 8192]
Each digit is a “visual token”

[Ramesh et al., Zero-Shot Text-to-Image Generation. ICML 2021] and https://ml.berkeley.edu/blog/posts/vq-vae/

Recall: Hidden-Unit Pre-training

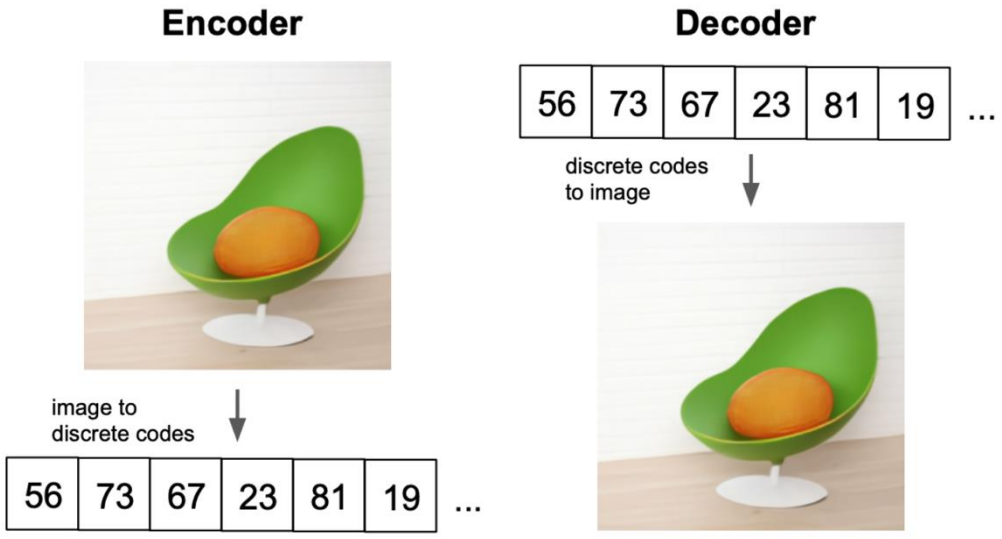


[Hsu et al., HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. IEEE TASLP 2021]

multisensory intelligence

From Text to Multimodal Generation

(2) Autoregressive generation of discrete visual tokens



An armchair in the shape of an avocado. → [56]

An armchair in the shape of an avocado. [56] → [73]

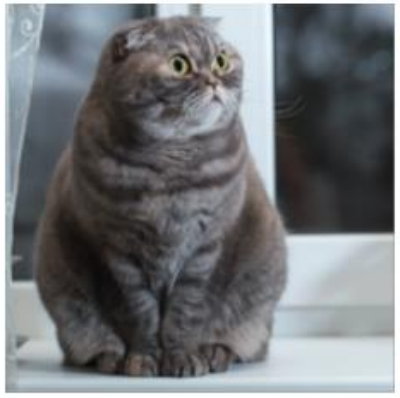
An armchair in the shape of an avocado. [56] [73] → [67]

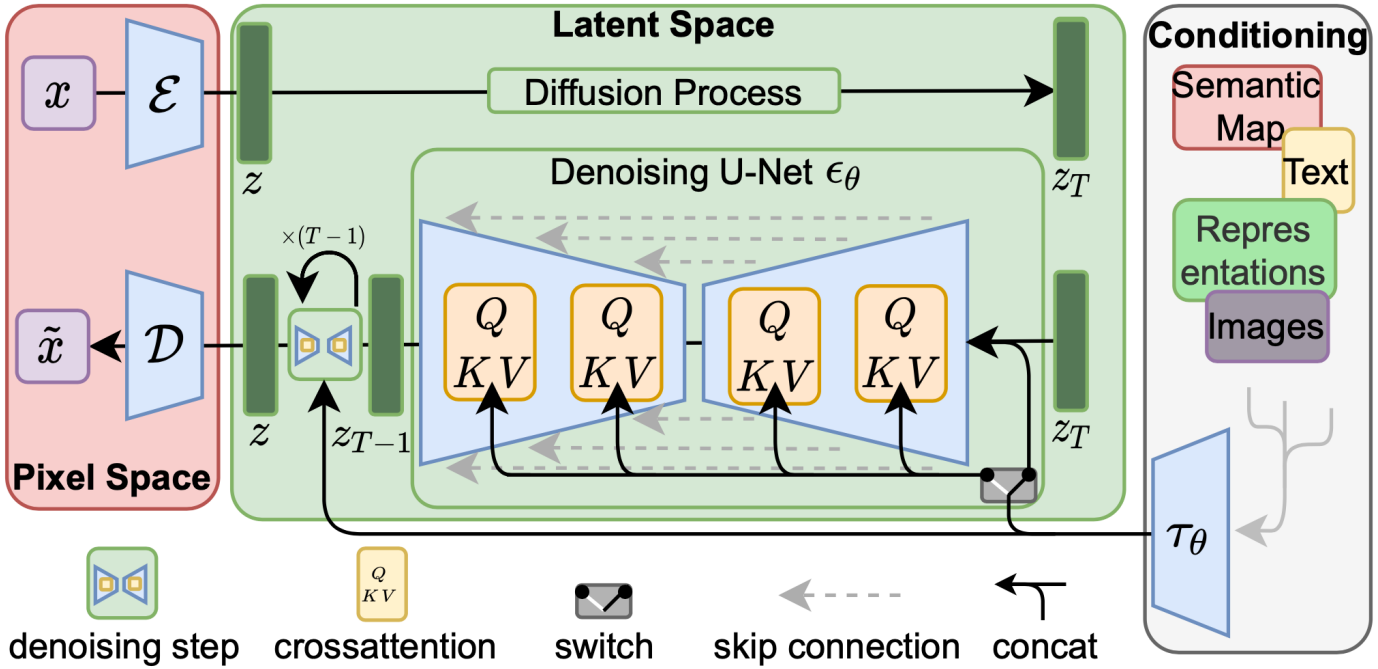
and so on…

[Ramesh et al., Zero-Shot Text-to-Image Generation. ICML 2021] and https://ml.berkeley.edu/blog/posts/vq-vae/

From Text to Multimodal Generation

(3) Image decoding with a diffusion model





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

Crash Course on Modern Generative Models

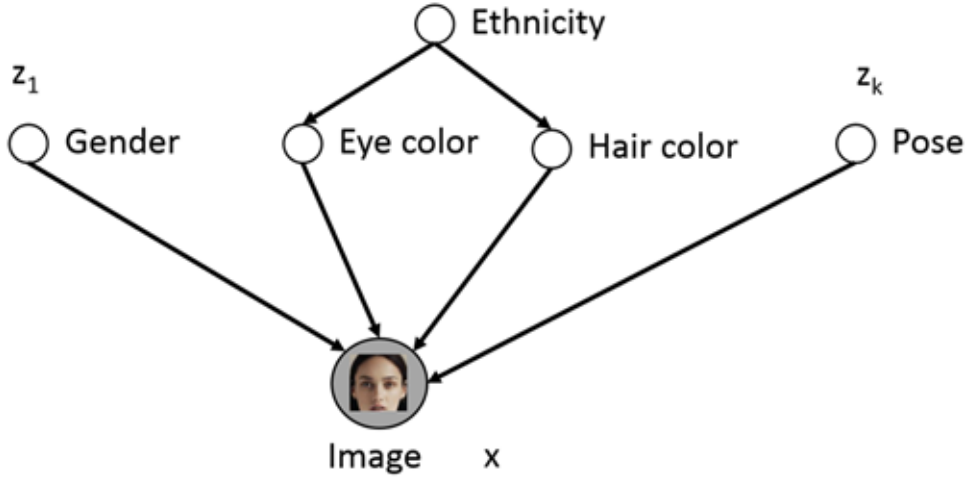
Learn to model **p(x)** where x = text, images, videos, multimodal data (by maximizing likelihood)

- Given x, **evaluate** p(x) - realistic data should have high p(x) and vice versa
- **Sample** new x according to p(x) - sample realistic looking images
- Unsupervised **representation** learning - we should be able to learn what these images have in common, e.g., ears, tail, etc. (features)

INPUT (x)	RECONSTRUCTION (AUTR)	RECONSTRUCTION (Gen-RNN)
unable to stop herself, she briefly, gently, touched his hand.	unable to stop herself, she leaned forward, and touched his eyes.	unable to help her , and her back and her into my way.
why didn't you tell me?	why didn't you tell me?	why didn't you tell me?"
a strange glow of sunlight shines down from above, paper white and blinding, with no heat.	the light of the sun was shining through the window, illuminating the room.	a tiny light on the door, and a few inches from behind him out of the door.
he handed her the slip of paper.	he handed her a piece of paper.	he took a sip of his drink.

Latent Variable Generative Models

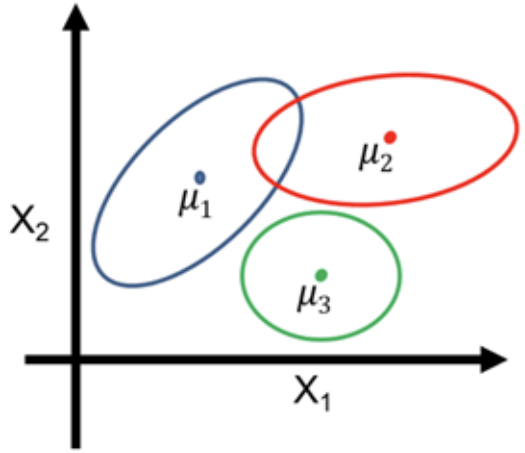
- Lots of variability in images **x** due to gender, eye color, hair color, pose, etc.
- However, unless images are annotated, these factors of variation are not explicitly available (latent).
- Idea: explicitly model these factors using latent variables **z**



Mixture of Gaussians

Mixture of Gaussians (Bayes network z -> x)

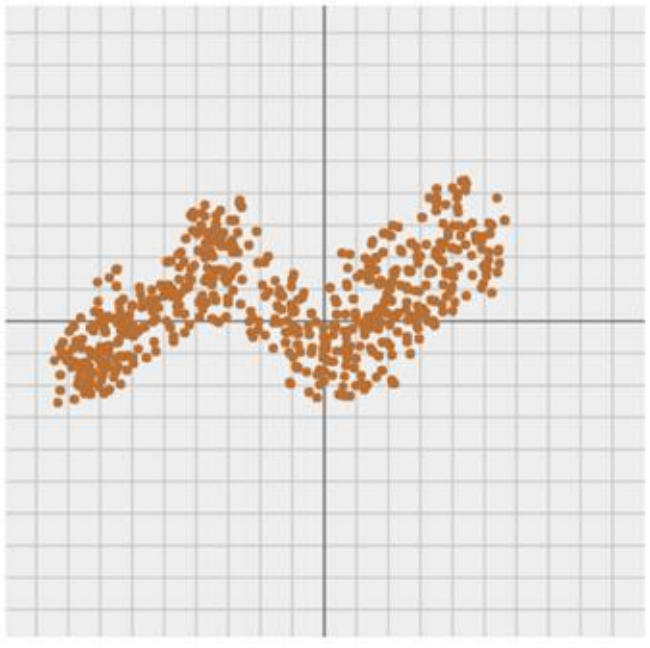
$$\mathbf{z} \sim \text{Categorical}(1, \cdots, K)$$
$$p(\mathbf{x} \mid \mathbf{z} = k) = \mathcal{N}(\mu_k, \Sigma_k)$$

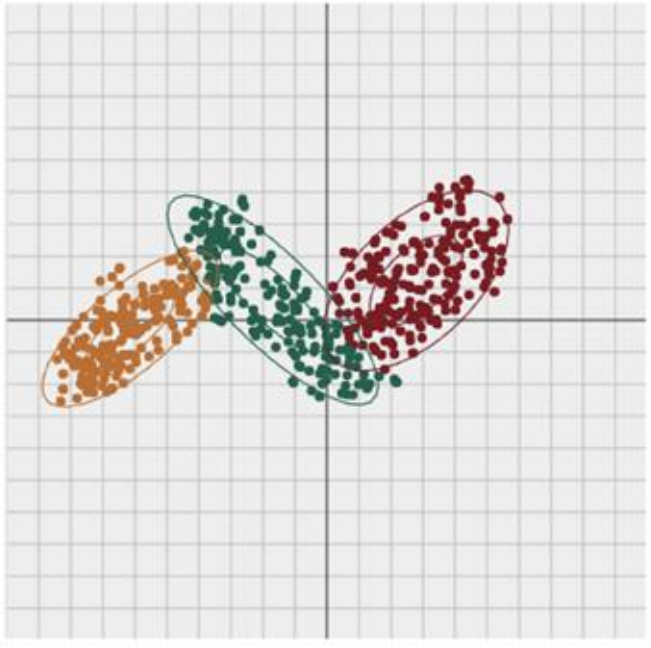


Generative process

1. Pick a mixture component by sampling z
2. Generate a data point by sampling from that Gaussian

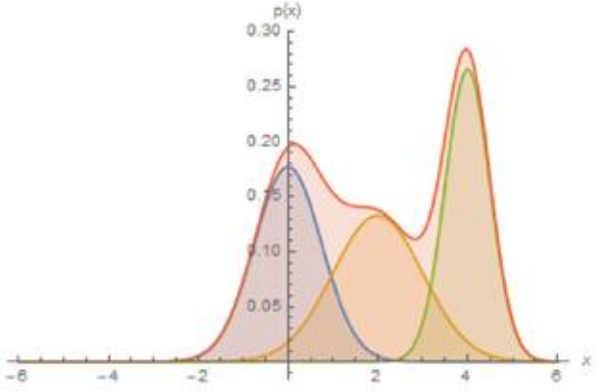
Mixture of Gaussians





Mixture of Gaussians

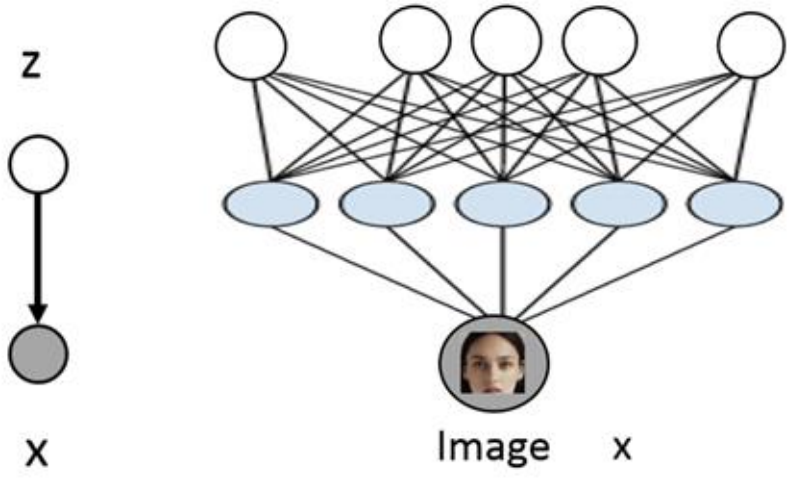
Combining simple models into more expressive ones



$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x} \mid \mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{z} = k) \underbrace{\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)}_{\text{component}}$$

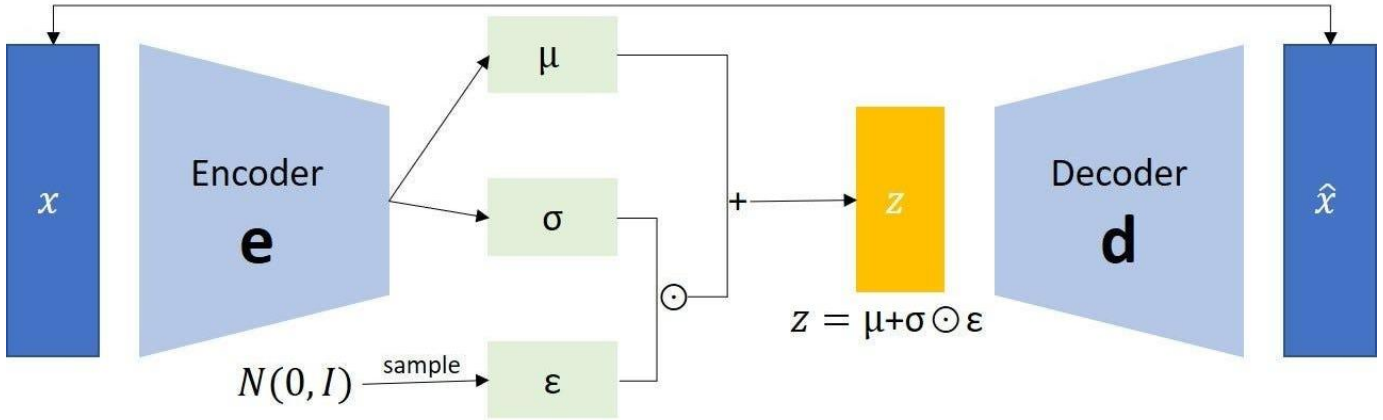
can solve using expectation maximization
Expectation: use mean and variance to estimate p(z=k)
Maximization: use estimate p(z=k) to update mean and variance

From GMMs to VAEs



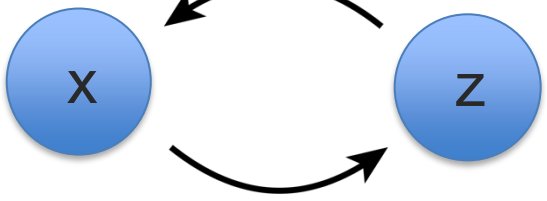
- Put a prior on z $\mathbf{z} \sim \mathcal{N}(0, I)$
  $p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mu_\theta(\mathbf{z}), \Sigma_\theta(\mathbf{z}))$ where μ_θ,Σ_θ are neural networks
- Hope that after training, z will correspond to meaningful latent factors of variation - useful features for unsupervised representation learning
- Given a new image x, features can be extracted via p(z|x)
- Given a random z, a new x can be generated => controllable if z is interpretable
- Even though p(x|z) is simple, marginal p(x) is much richer/complex/flexible

Learning Parameters of VAEs



Generative/decoder

$p(x|z;\theta)$



$q(z|x;\phi)$

Inference/encoder

1. Take a datapoint x_i.
2. Map it to μ, σ using $q_\phi(z|x_i)$. **encoder**
3. Sample $\epsilon \sim N(0, I)$ and compute $\hat{z} = \mu + \sigma\epsilon$. **re-parameterize**
4. Reconstruct $\hat{x}$ by sampling from $p(x|\hat{z};\theta)$. **decoder**

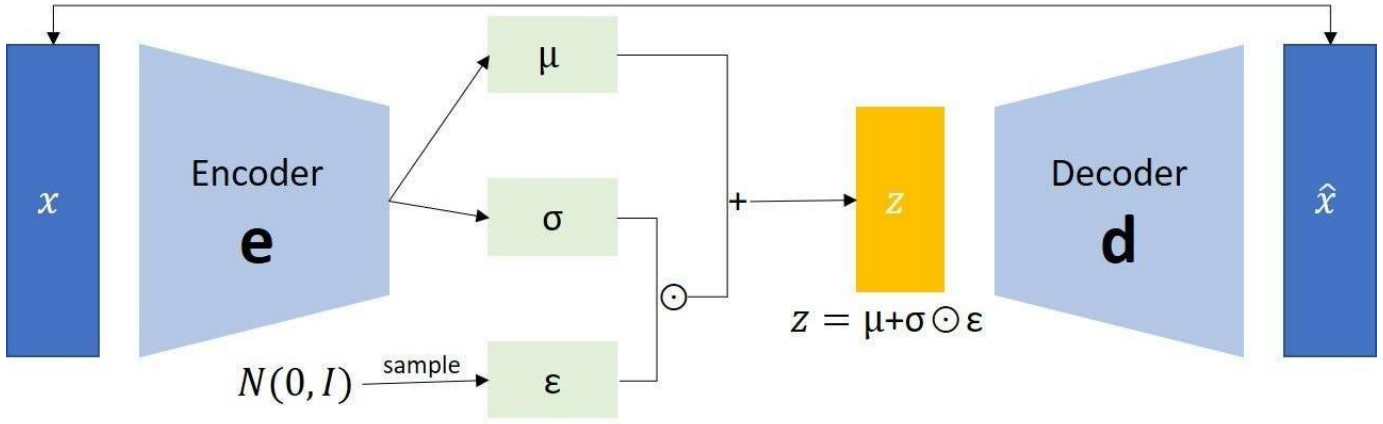
$$\mathcal{L}(\mathbf{x};\theta,\phi) = \underbrace{E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z};\theta)]}_{\text{reconstruction}} - \underbrace{D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))}_{\text{prior over z, standard Gaussian}}$$

reconstruction
x -> z -> x

prior over z,
standard Gaussian

Prior on z: $\mathbf{z} \sim \mathcal{N}(0, I)$

Learning Parameters of VAEs

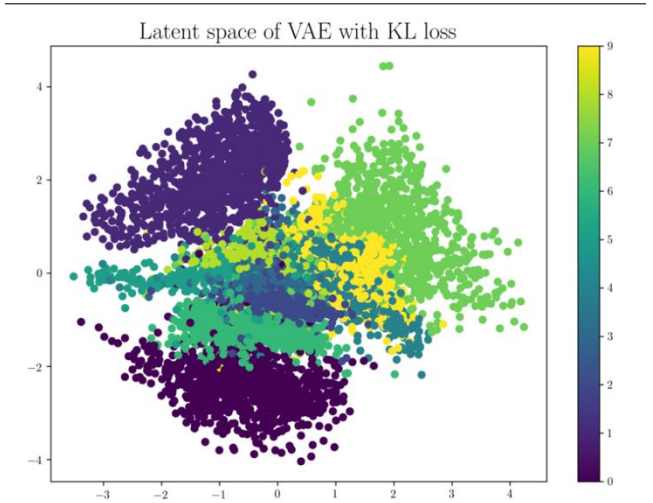


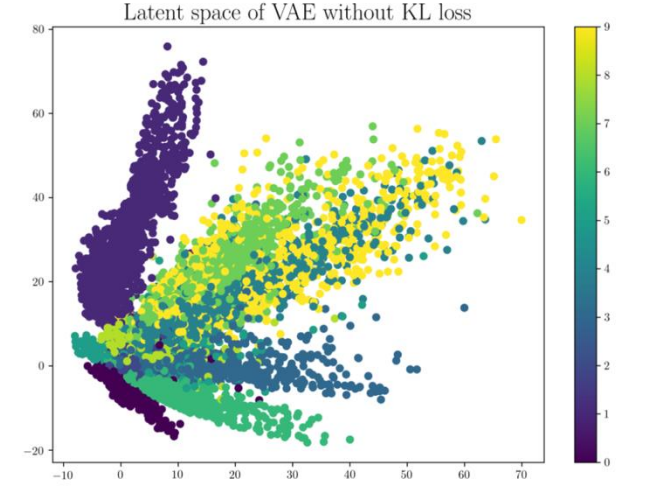
1. Take a datapoint x_i.
2. Map it to μ, σ using $q_\phi(z|x_i)$. **encoder**
3. Sample $\epsilon \sim N(0, I)$ and compute $\hat{z} = \mu + \sigma\epsilon$. **re-parameterize**
4. Reconstruct $\hat{x}$ by sampling from $p(x|\hat{z}; \theta)$. **decoder**

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \underbrace{E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}; \theta)]}_{\text{reconstruction}} - \underbrace{D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}))}_{\text{prior over z, standard Gaussian}}$$

reconstruction
x -> z -> x

prior over z,
standard Gaussian



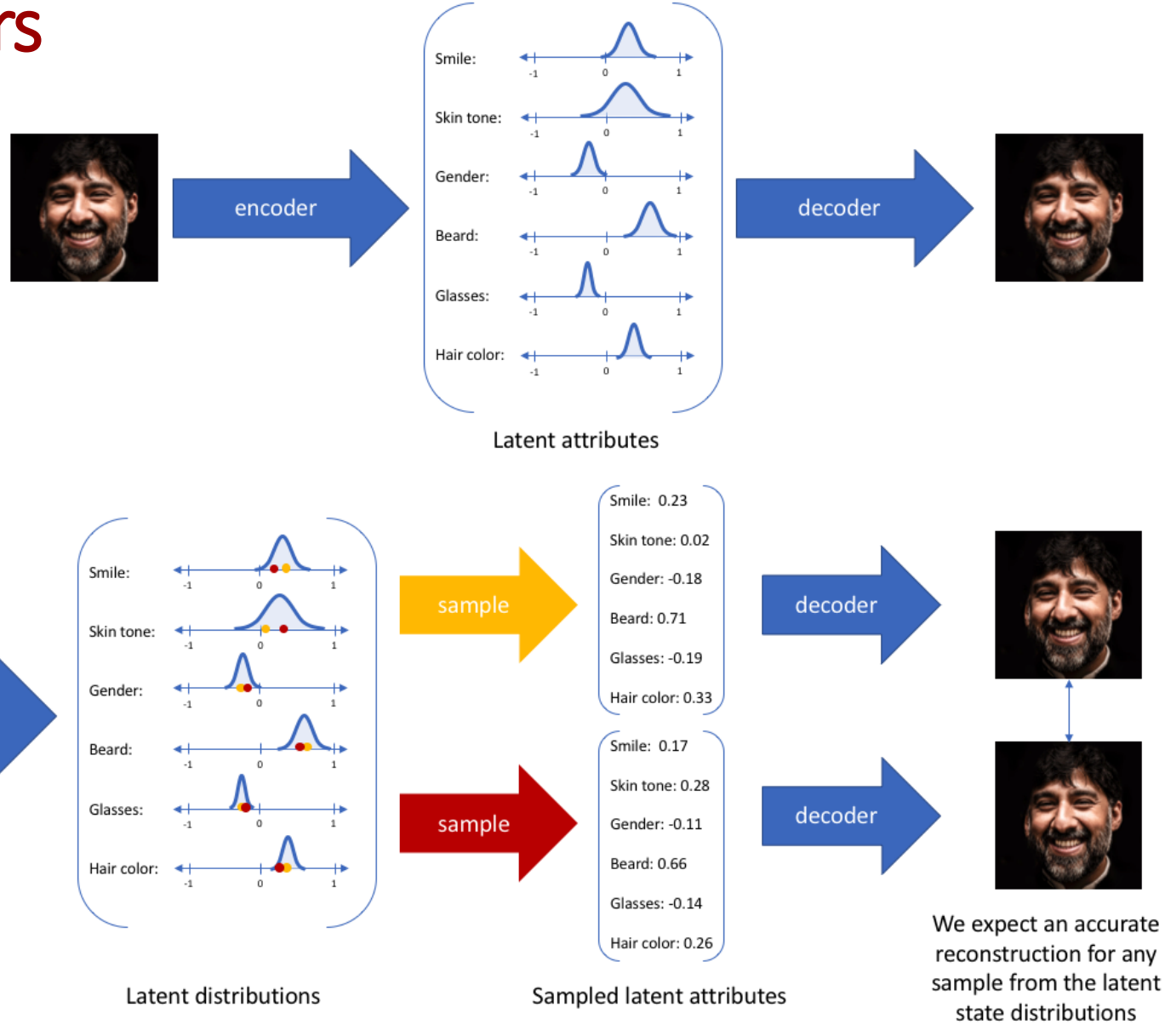


Prior on z: $\mathbf{z} \sim \mathcal{N}(0, I)$

Variational Autoencoders

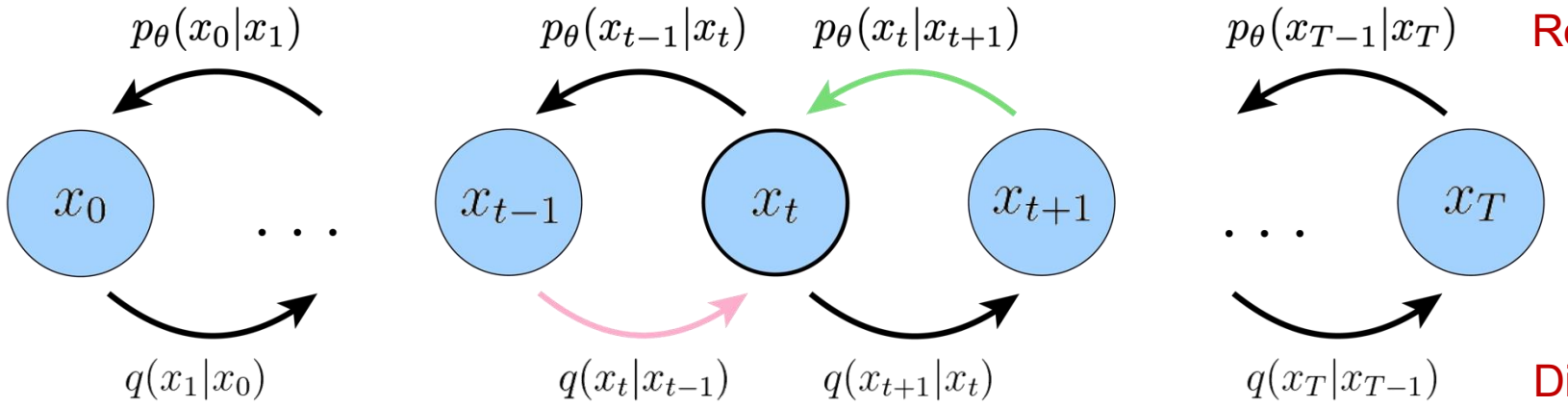
Key ideas:

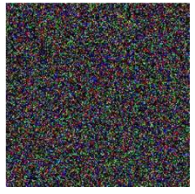
1. Encoder decoder structure.
2. Simple latent variable z.
3. Complex p(x|z) decoder via neural networks.
4. Reconstruction objective.
5. Prior over latent variable z:
   -- smoother latent space permits sampling
   -- disentangles different dimensions of z



From VAEs to Diffusion Models

Generative modeling via denoising





Encoding via adding noise: $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$ Noise parameters

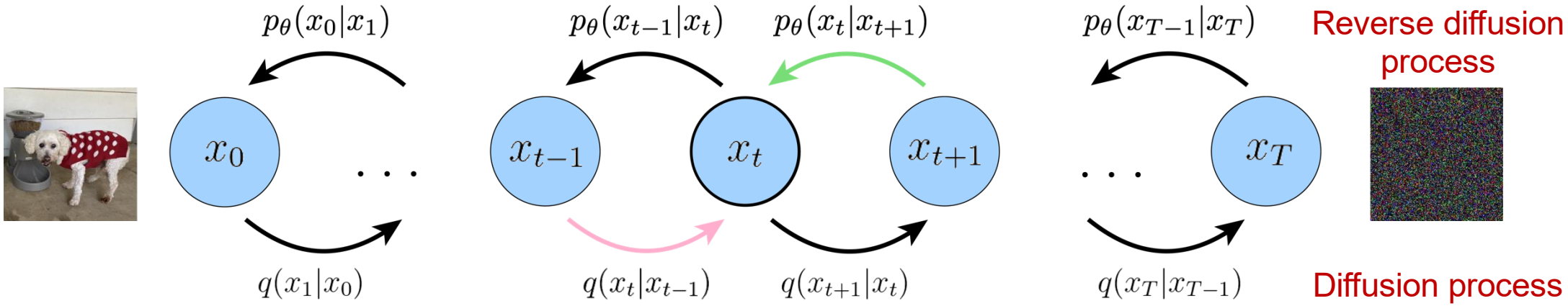
Decoding via denoising: $p(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ where $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \mathbf{I})$

[Tutorial by Calvin Luo and Yang Song]

multisensory intelligence

From VAEs to Diffusion Models

Generative modeling via denoising



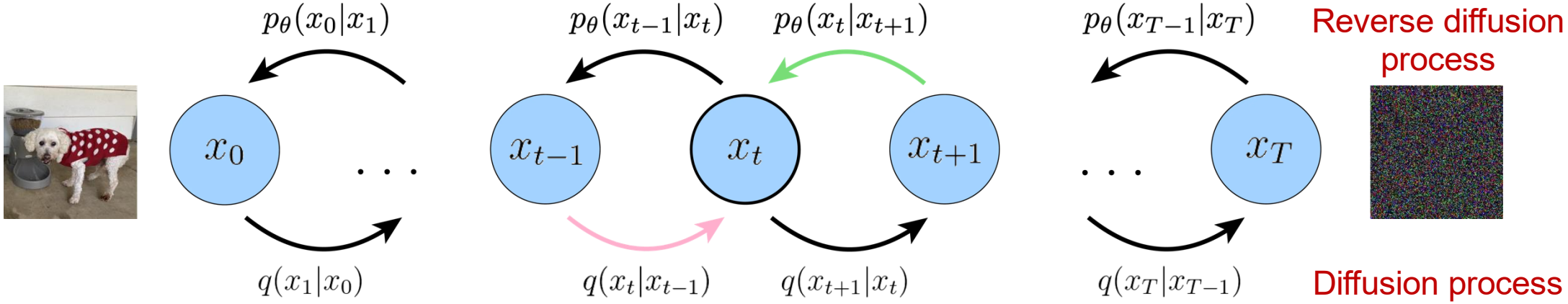
Similar to variational autoencoder, but:

1. The latent dimension is exactly equal to the data dimension.
2. Encoder q is not learned, but pre-defined as a Gaussian distribution centered around the output of previous timestep.
3. Gaussian parameters of latent encoders vary over time such that distribution of final latent is a standard Gaussian.

[Tutorial by Calvin Luo and Yang Song]

Learning Diffusion Models

Key idea: use variational inference



$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right]$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,||\, p(\boldsymbol{x}_T))}_{\text{prior matching term}}$$

Multi-level VAE!

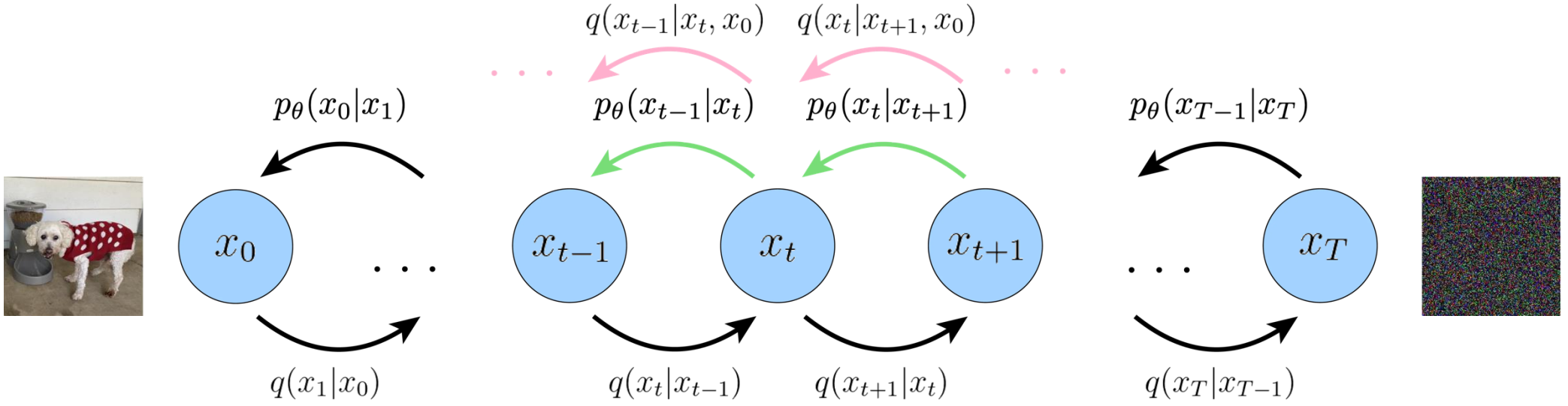
$$- \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))\right]}_{\text{denoising matching term}}$$

[Tutorial by Calvin Luo and Yang Song]

multisensory intelligence

Learning Diffusion Models

Key idea: use variational inference



Intuition: Neural network to predict cleaner image x_{t-1} from noisy image x_t at time t, consistent with the noise adding process.

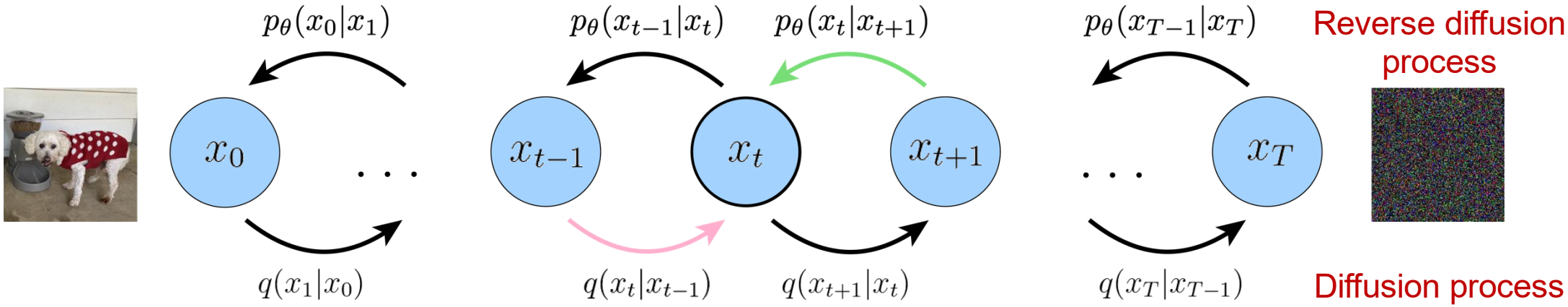
$$-\sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))\right]}_{\text{denoising matching term}}$$

[Tutorial by Calvin Luo and Yang Song]

multisensory intelligence

Learning Noise Parameters

Generative modeling via denoising



Encoding via adding noise: $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$ Noise parameters

Choose $\alpha_1 > \cdots > \alpha_T$

i.e., add smaller noise at the beginning of the diffusion process and gradually increase noise when the samples get noisier.

[Tutorial by Calvin Luo and Yang Song]

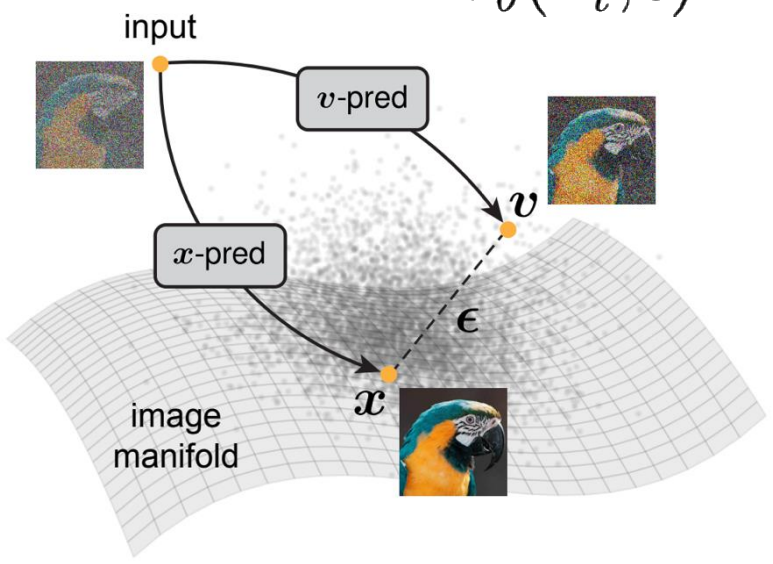
Equivalent Formulations

3 interpretations

$$p(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

1. Original image prediction (over time)
2. Noise prediction (over time)
3. Velocity function estimation (over time)

$$\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) \approx \mathbf{x}_0$$
$$\hat{\epsilon}_\theta(\mathbf{x}_t, t) \approx \epsilon_0$$
$$\hat{\mathbf{v}}_\theta(\mathbf{x}_t, t) \approx \mathbf{v}_0 \qquad \mathbf{v} = \mathbf{x} - \epsilon$$

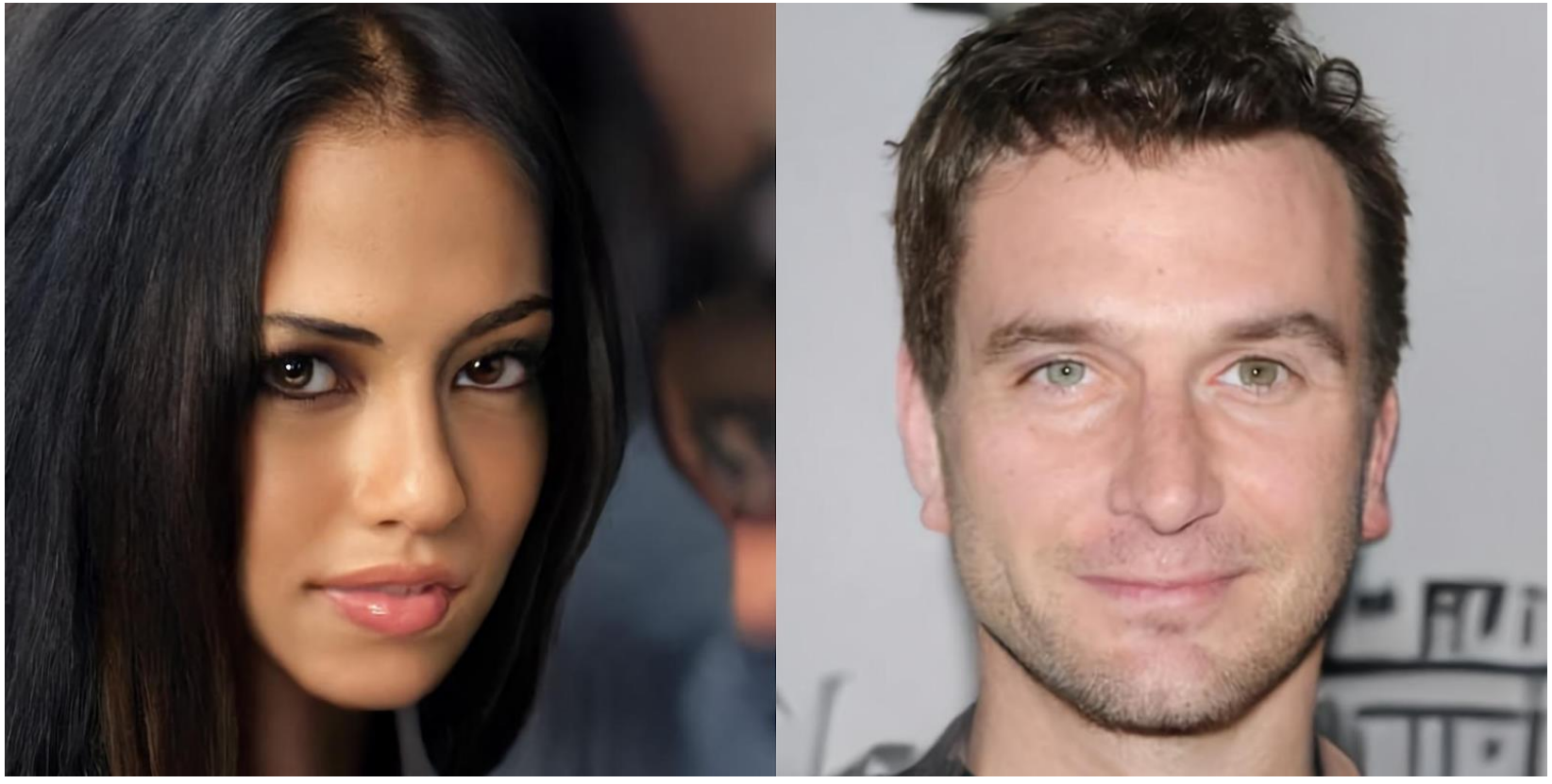


[Li et al., Back to Basics: Let Denoising Generative Models Denoise. arXiv 2025]

multisensory intelligence

Diffusion Models as Differential Equations

From discrete diffusion process to continuous diffusion process

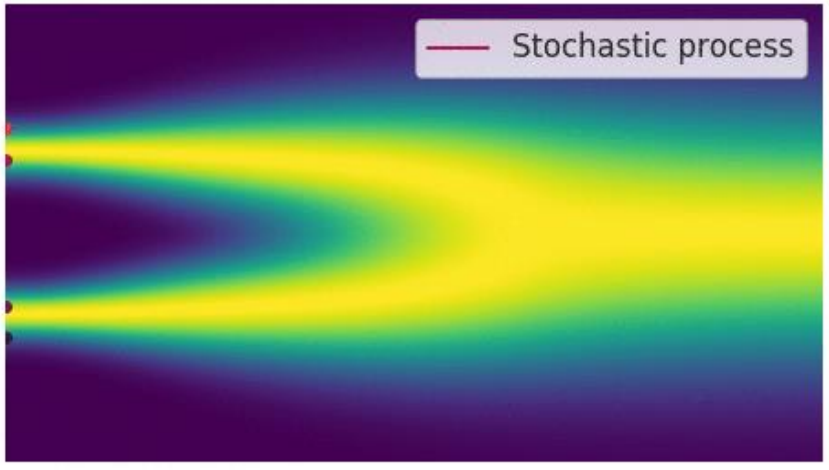


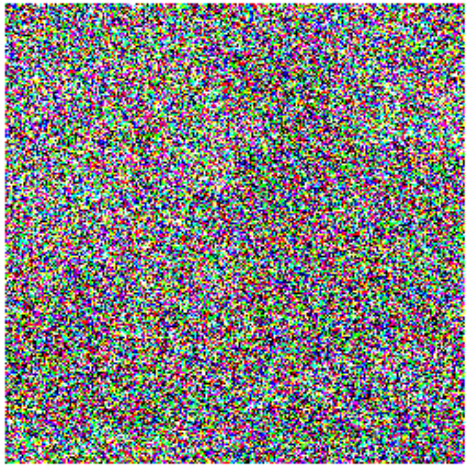
[Tutorial by Calvin Luo and Yang Song]

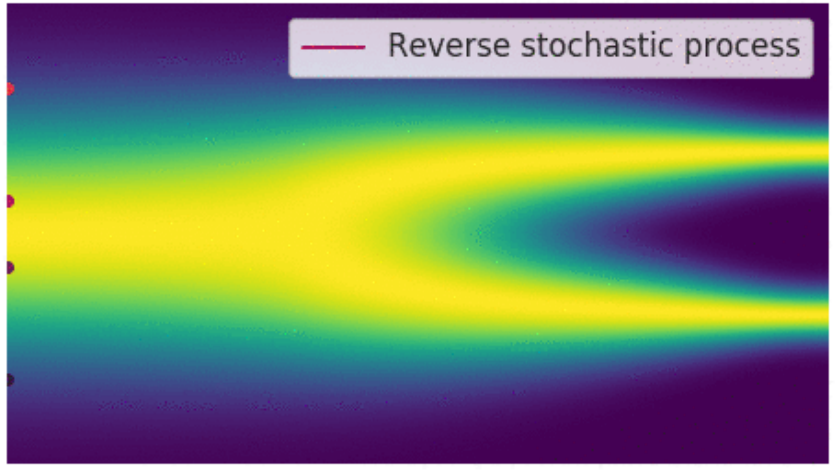
Diffusion Models as Differential Equations

From discrete diffusion process to continuous diffusion process

- Higher quality samples
- Exact log-likelihood
- Controllable generation







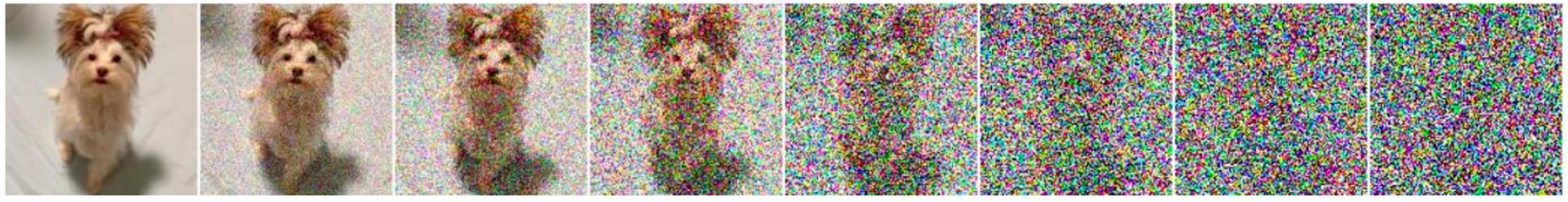
[Tutorial by Calvin Luo and Yang Song]

Diffusion Models as Differential Equations

From discrete diffusion process to continuous diffusion process

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$



score function

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

Think ‘infinite-layer’ latent variable model

[Tutorial by Calvin Luo and Yang Song]

multisensory intelligence

Flow-based Models

Diffusion models generate data by adding noise to data during training and learning to reverse this process step-by-step to denoise pure noise into a sample.

Flow matching learns a continuous vector field that directly transports noise into data through an ODE, avoiding the need for step-by-step forward diffusion process for faster sampling.

Flow-based Models

Flow matching learns a continuous vector field that directly transports noise into data through an ODE, avoiding the need for step-by-step forward diffusion process for faster sampling.

Generation follows an ODE from noise to data:
u is the learned *velocity field.*

$$X_0 \sim p_{\text{init}}$$

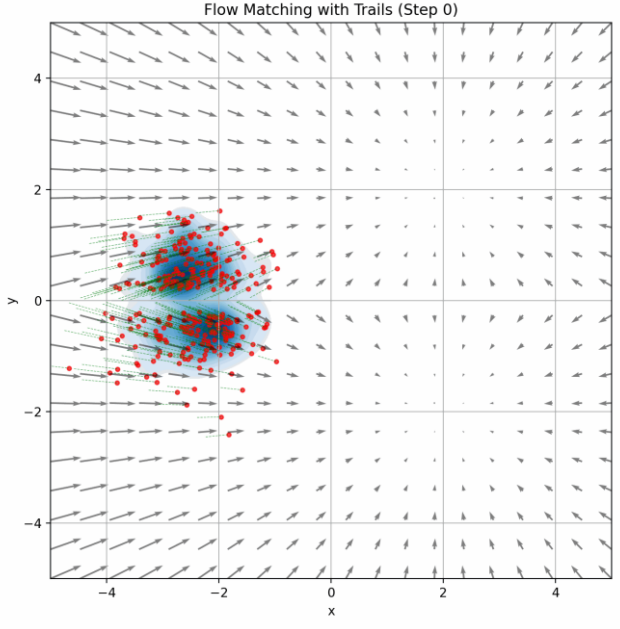
$$\frac{\mathrm{d}}{\mathrm{d}t} X_t = u_t^\theta(X_t)$$

Euler method for integrating an ODE:

Algorithm 1 Sampling from a Flow Model with Euler method

Require: Neural network vector field u_t^θ, number of steps n

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$
4: **for** $i = 1, \ldots, n-1$ **do**
5: $\quad X_{t+h} = X_t + h u_t^\theta(X_t)$
6: $\quad$ Update $t \leftarrow t + h$
7: **end for**
8: **return** X_1



Comparisons of Generative Models

Feature	VAE	Diffusion Models	Flow Matching
Core Idea	Encode/decode with latent noise	Add noise and learn to reverse it	Learn a continuous flow from noise
Training Objective	Minimize reconstruction + KL loss	Learn the score (gradient) of data	Match a vector field (ODE-based)
Noise Handling	Noise in latent space	Progressive noise over time	Start from noise, smooth transform
Sampling Speed	Very fast (one pass)	Slow (many denoising steps)	Faster (solving an ODE)
Advantages	Simple, fast, interpretable	Very high-quality outputs	High quality + faster than diffusion
Disadvantages	Blurry samples, limited expressiveness	Expensive, slow sampling	Newer, still developing
Key Examples	VAE (2013), β-VAE	DDPM, Stable Diffusion	Flow Matching (2023), Rectified Flow

Conditioning Diffusion Models on Text

- DALL-E 2 https://cdn.openai.com/papers/dall-e-2.pdf
- Diffusion on top of frozen CLIP

✗

A black apple and a green backpack.

✗

A horse riding an astronaut.

Imagen https://arxiv.org/pdf/2205.11487.pdf

Diffusion on top of frozen T5 embeddings

✓

✗

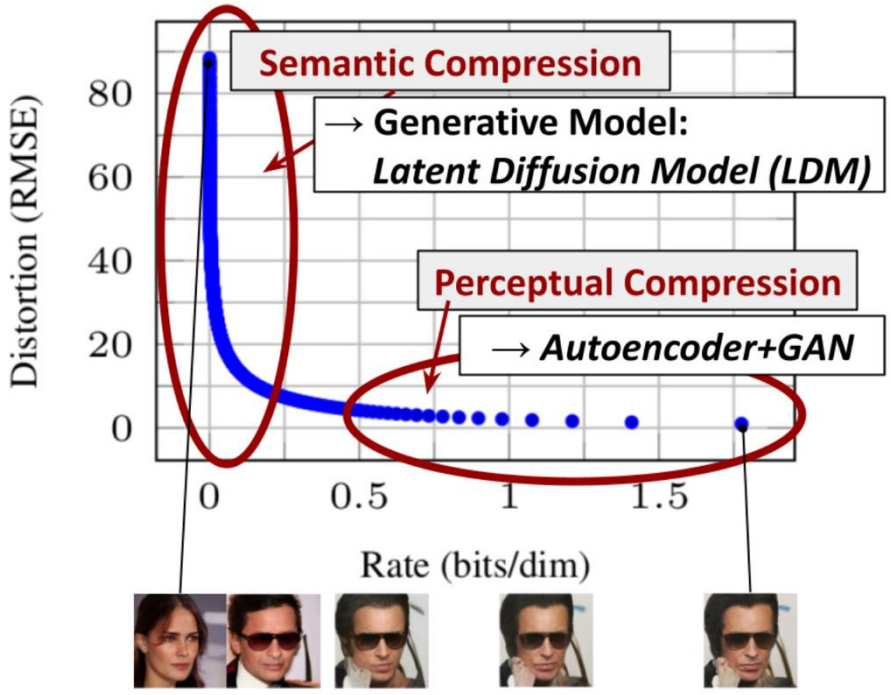
multisensory intelligence

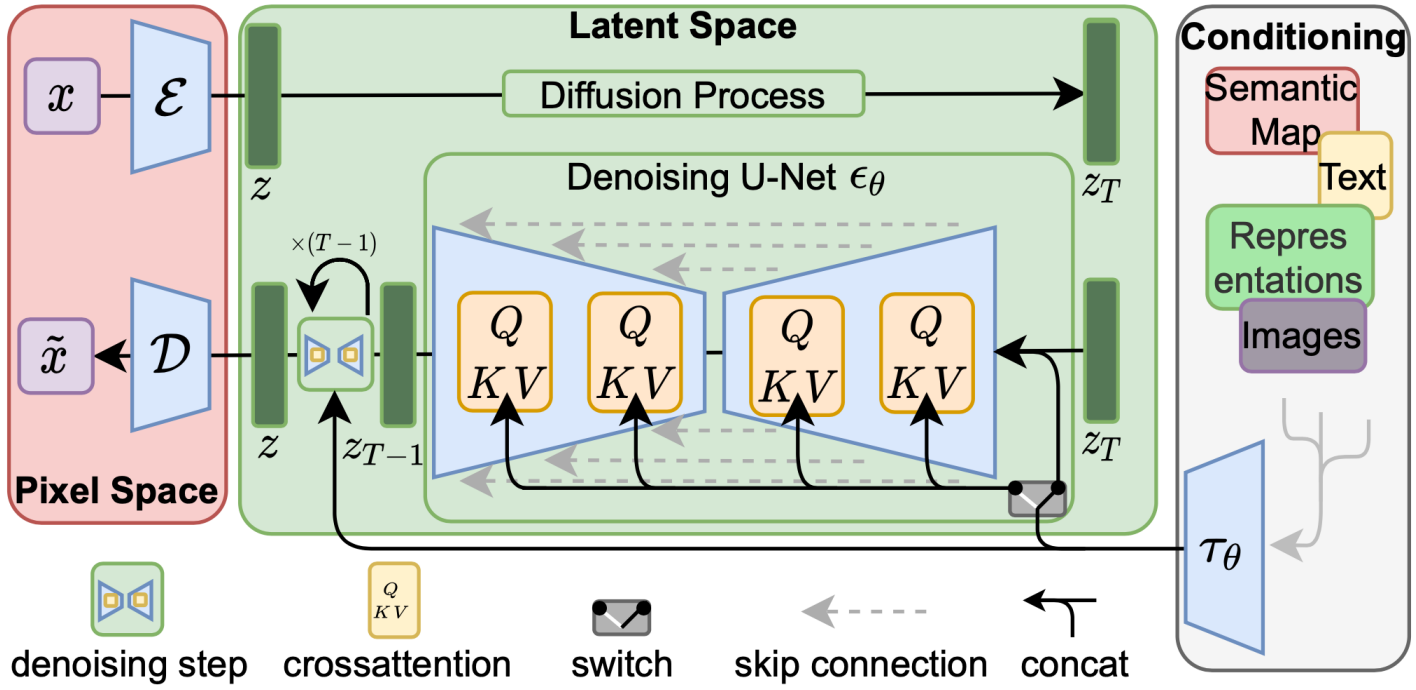
Text-to-Image Generation with Latent Diffusion

1. Directly training diffusion models with conditional information

Diffusion process in latent space instead of pixel space – faster training and inference.

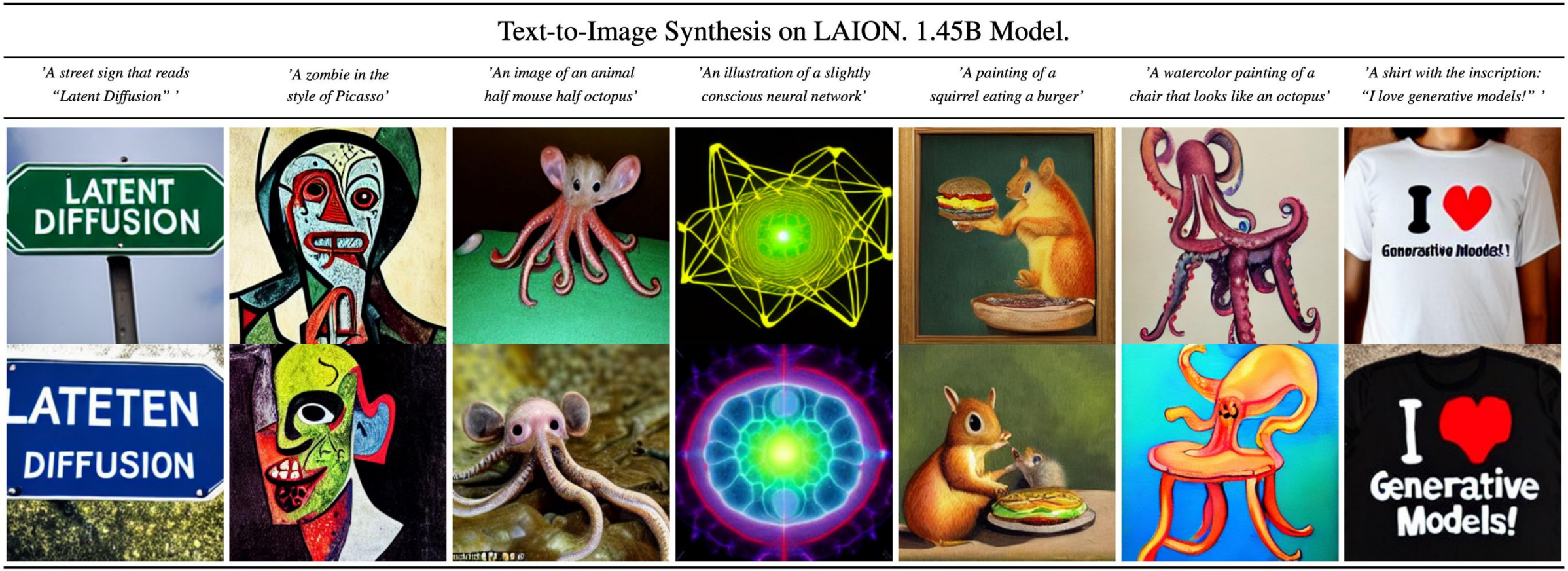
Use autoencoder for perceptual compression, diffusion model for semantic compression.





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

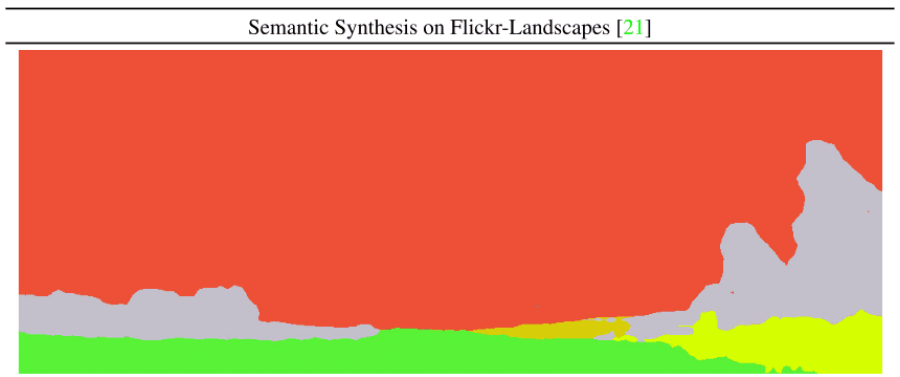
Text-to-Image Generation with Latent Diffusion

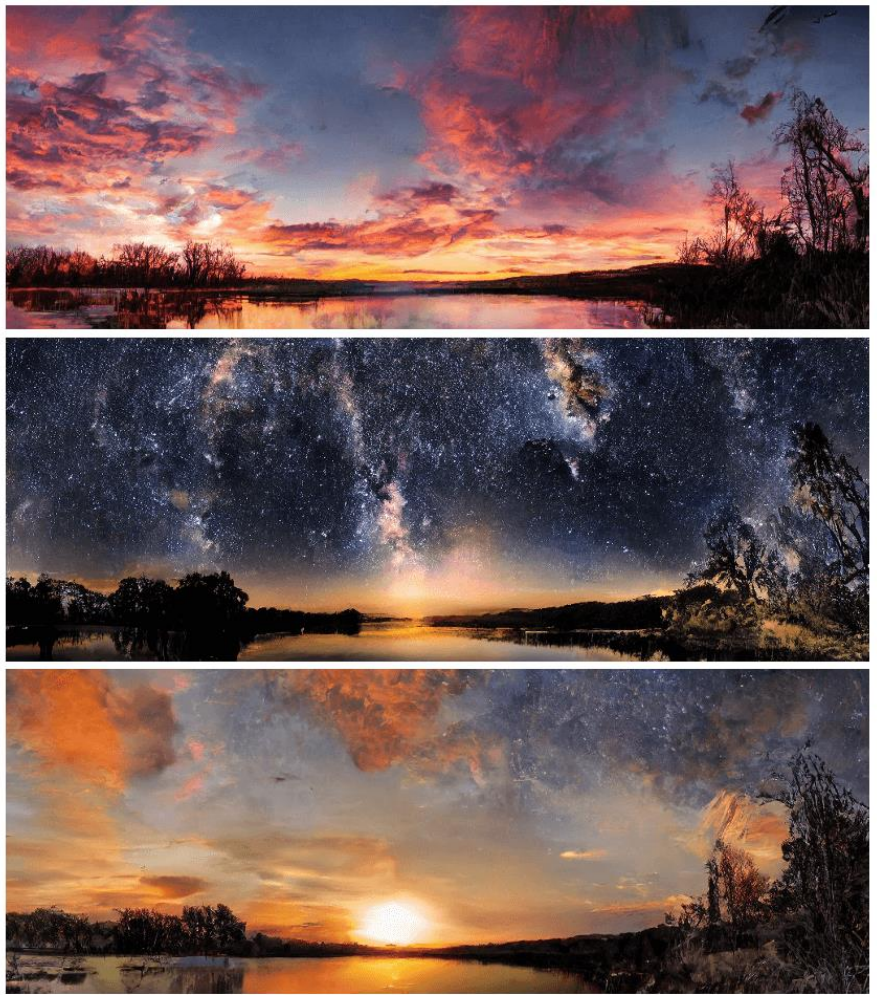


[Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022]

multisensory intelligence

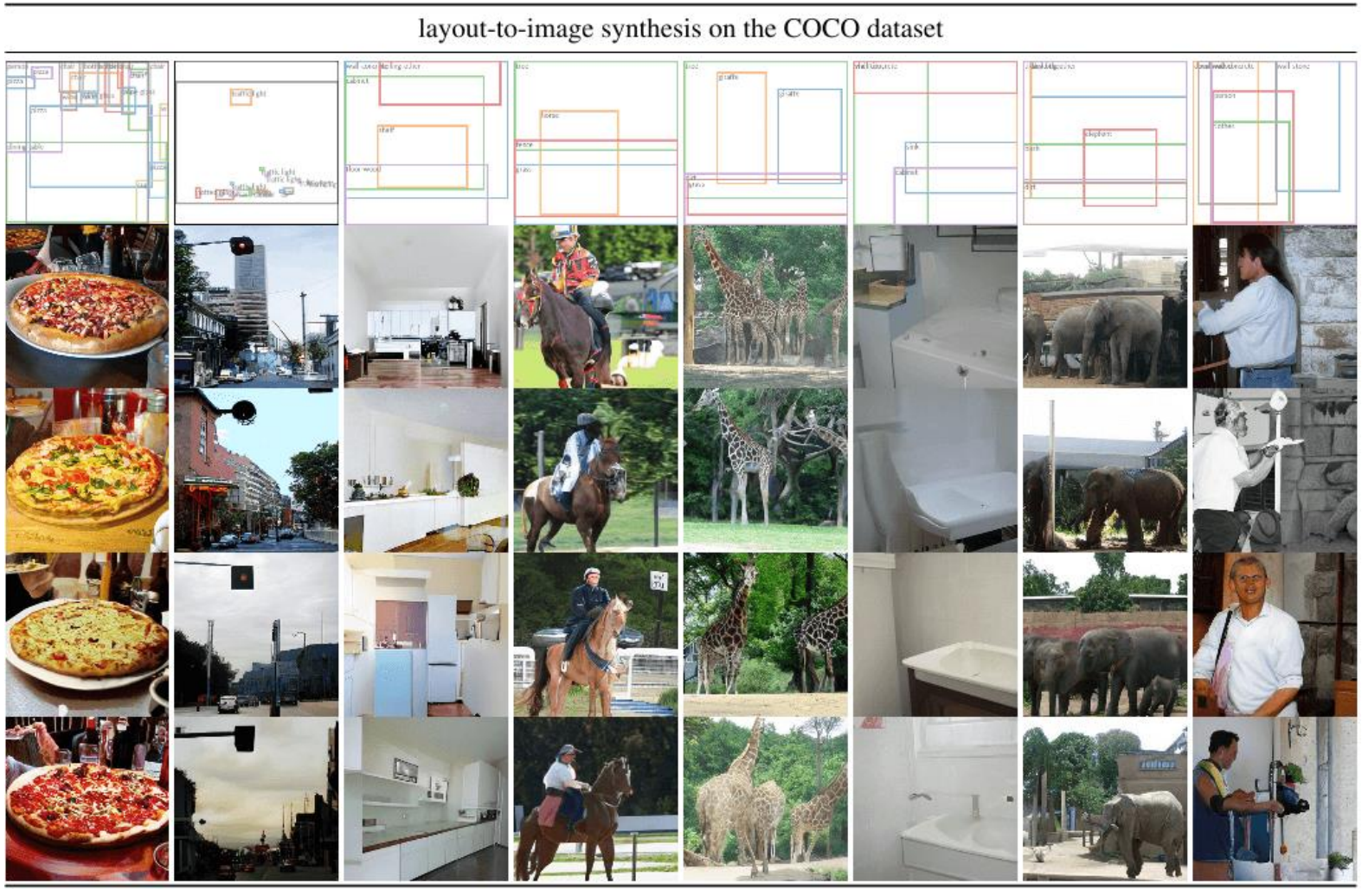
Text-to-Image Generation with Latent Diffusion





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

Text-to-Image Generation with Latent Diffusion



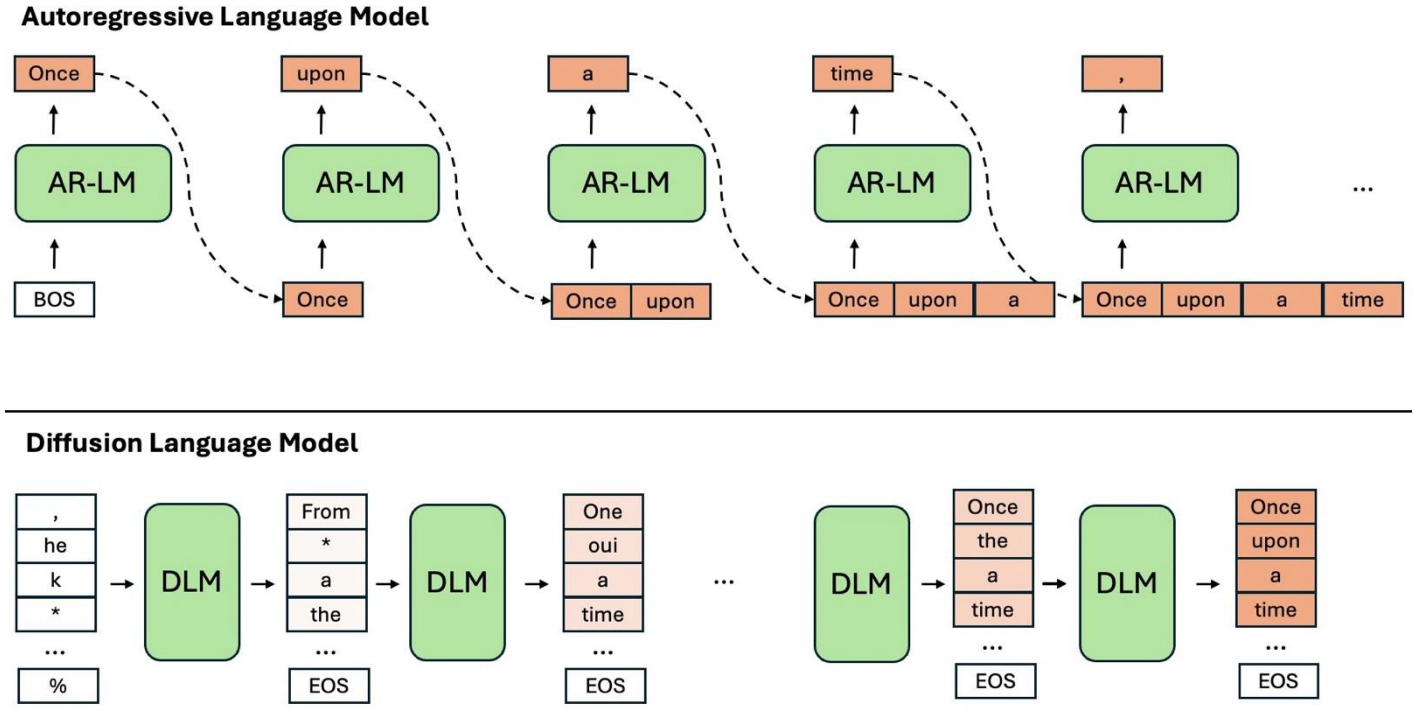
[Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022]

multisensory intelligence

Diffusion Models for Text

Diffusion Language Models: Instead of going token by token, they iteratively refine and predict the *whole* sequence from a noise following a non-autoregressive decoding process.

Key advantages:
- Can fix earlier mistakes
- More controllable
- More diversity
- Faster

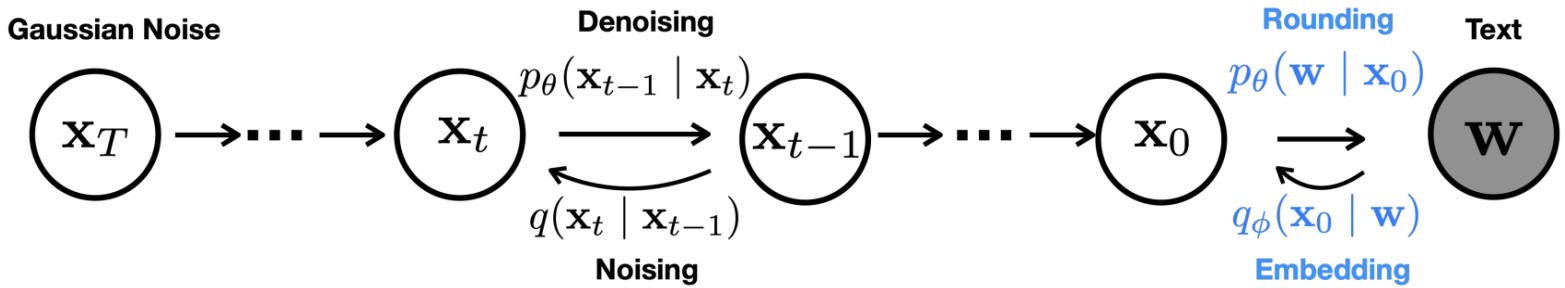


[Zhu et al., What are Diffusion Language Models? 2025]

multisensory intelligence

Diffusion Models for Text

Key differences:



Diffusion for images:

1. Continuous pixels
2. Gaussian noise
3. Directly generates pixels

Diffusion for text:

1. Discrete tokens
2. Categorical noise: masking, token substitution, etc.
3. Rounding and embedding steps
4. Note: discrete diffusion LMs vs continuous diffusion LMs

[Li et al., Diffusion-LM Improves Controllable Text Generation. NeurIPS 2022]

Diffusion Models for Text

Strong results with 8B models, scalable, good at reversal tasks (non-autoregressive)

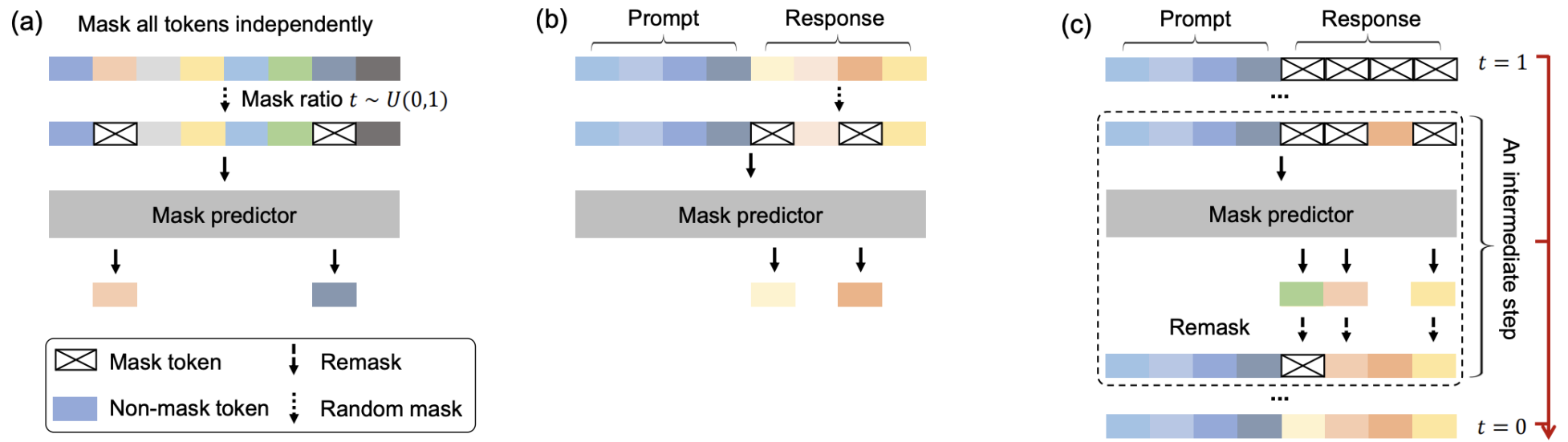
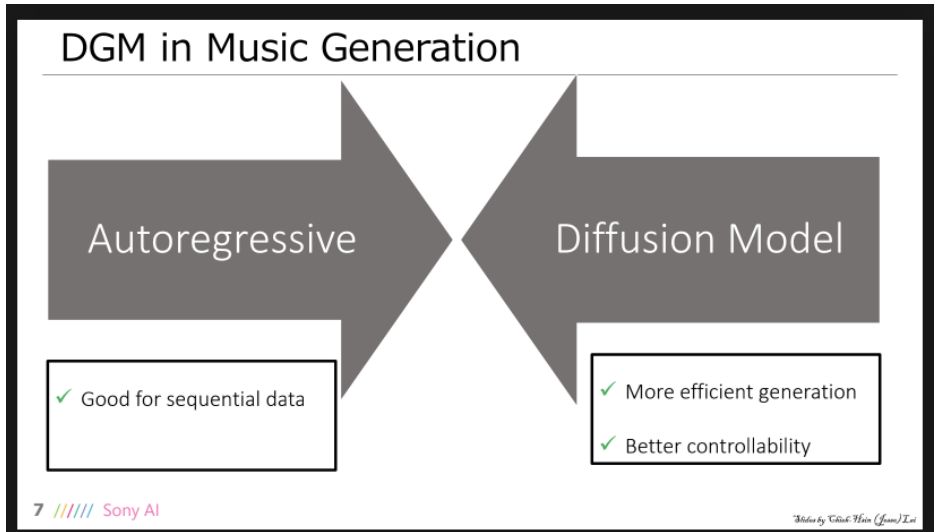


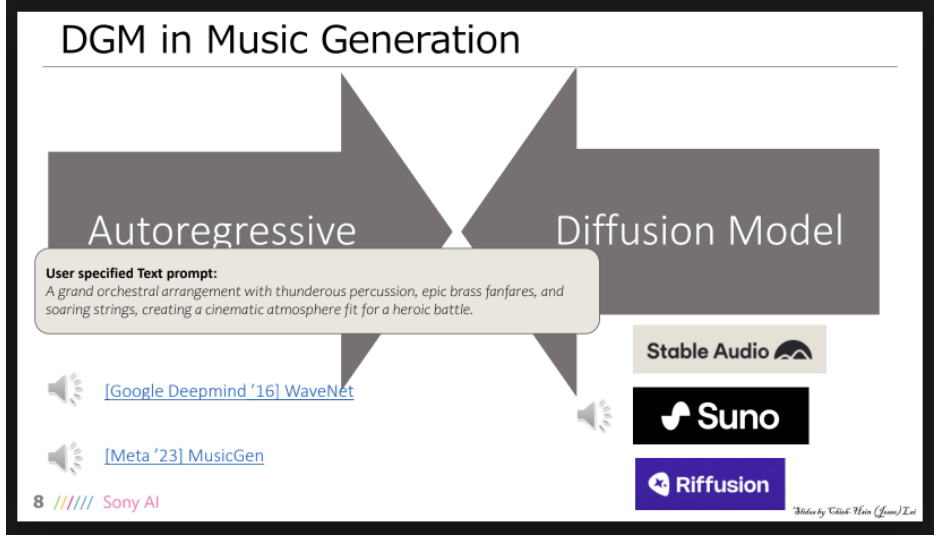
Figure 2: **Overview of LLaDA.** (a) Pre-training. LLaDA is trained on text with random masks applied independently to all tokens at the same ratio $t \sim U[0, 1]$. (b) SFT. Only response tokens are possibly masked. (c) Sampling. LLaDA simulates a diffusion process from $t = 1$ (fully masked) to $t = 0$ (unmasked), predicting all masks simultaneously at each step with flexible remask strategies.

[Nie et al., Large Language Diffusion Models. NeurIPS 2025]

multisensory intelligence

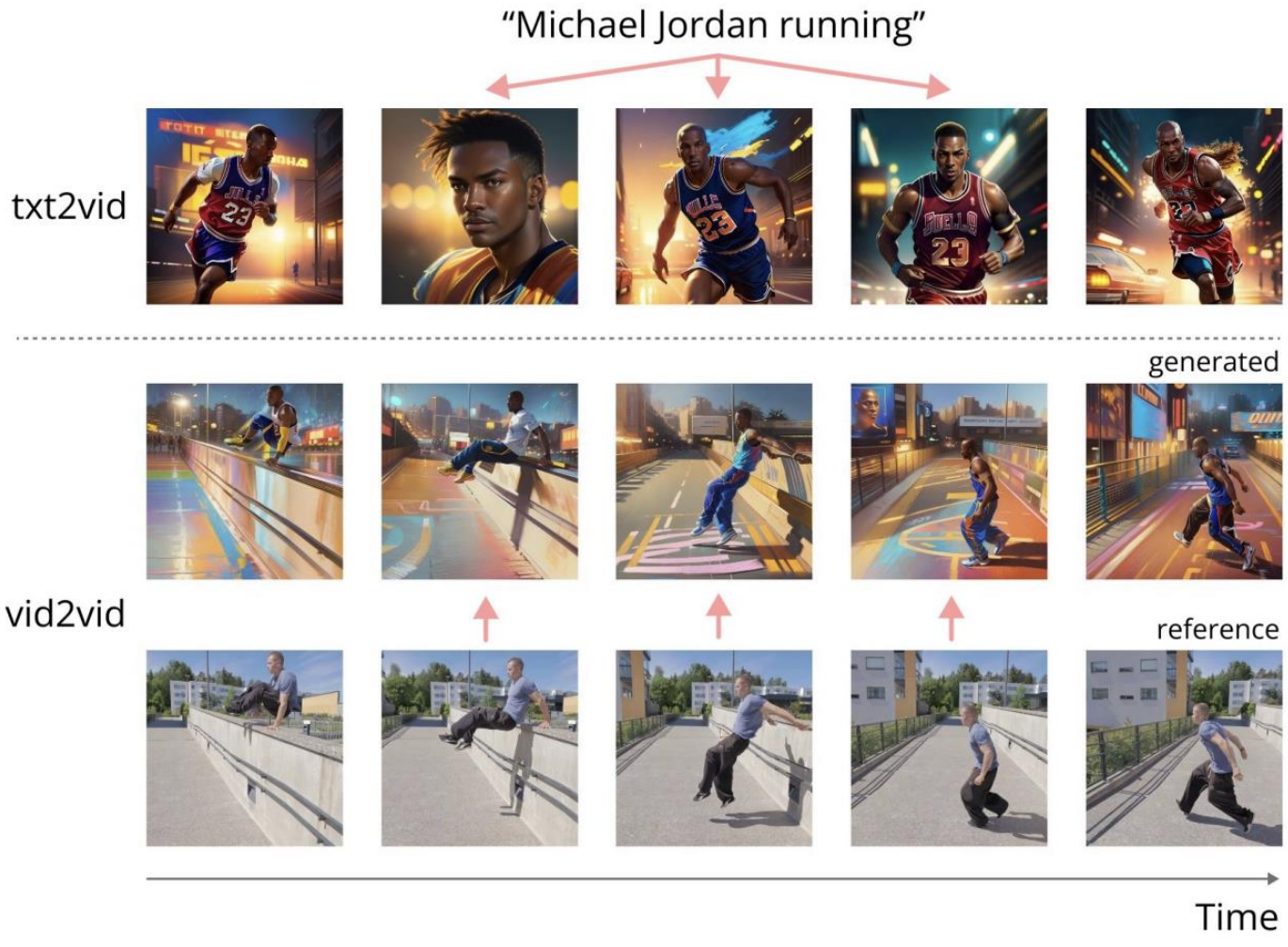
Diffusion Models for Audio





https://sites.google.com/view/diffusionmodeltutorialicassp25/home

Diffusion Models for Video

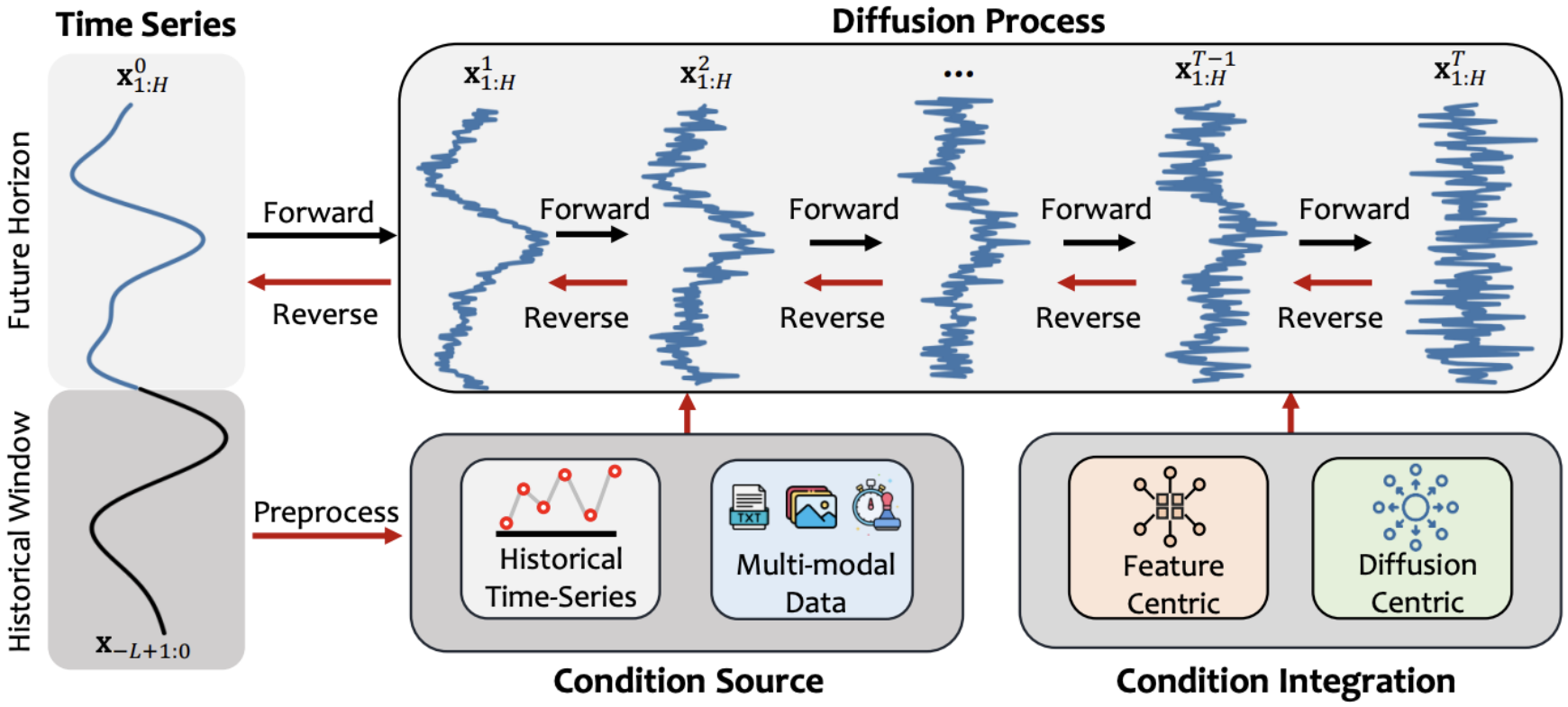


Temporal coherence, physical consistency, viewpoint dynamics are still a challenge

[Melnik et al., Video Diffusion Models: A Survey. arXiv 2024]

multisensory intelligence

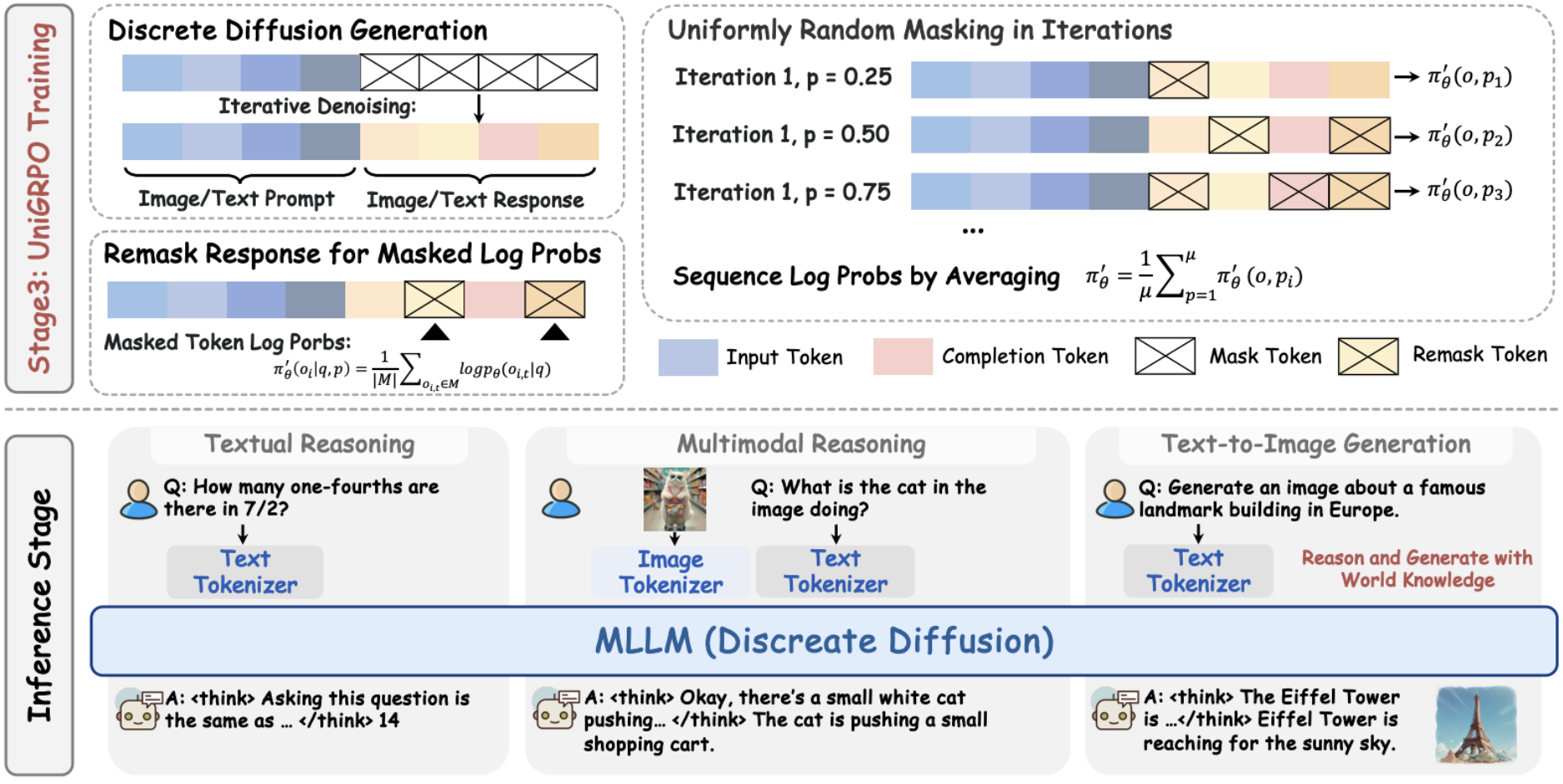
Diffusion Models for Sensors



[Su et al., Diffusion Models for Time Series Forecasting: A Survey. arXiv 2025]

multisensory intelligence

Unified Multimodal Diffusion



[Yang et al., MMaDA: Multimodal Large Diffusion Language Models. arXiv 2025]

multisensory intelligence

Unified Multimodal Diffusion

🤗 Diffusers is the go-to library for state-of-the-art pretrained diffusion models for generating images, audio, and even 3D structures of molecules. Whether you're looking for a simple inference solution or training your own diffusion models, 🤗 Diffusers is a modular toolbox that supports both. Our library is designed with a focus on usability over performance, simple over easy, and customizability over abstractions.

🤗 Diffusers offers three core components:

- State-of-the-art diffusion pipelines that can be run in inference with just a few lines of code.
- Interchangeable noise schedulers for different diffusion speeds and output quality.
- Pretrained models that can be used as building blocks, and combined with schedulers, for creating your own end-to-end diffusion systems.

multisensory intelligence

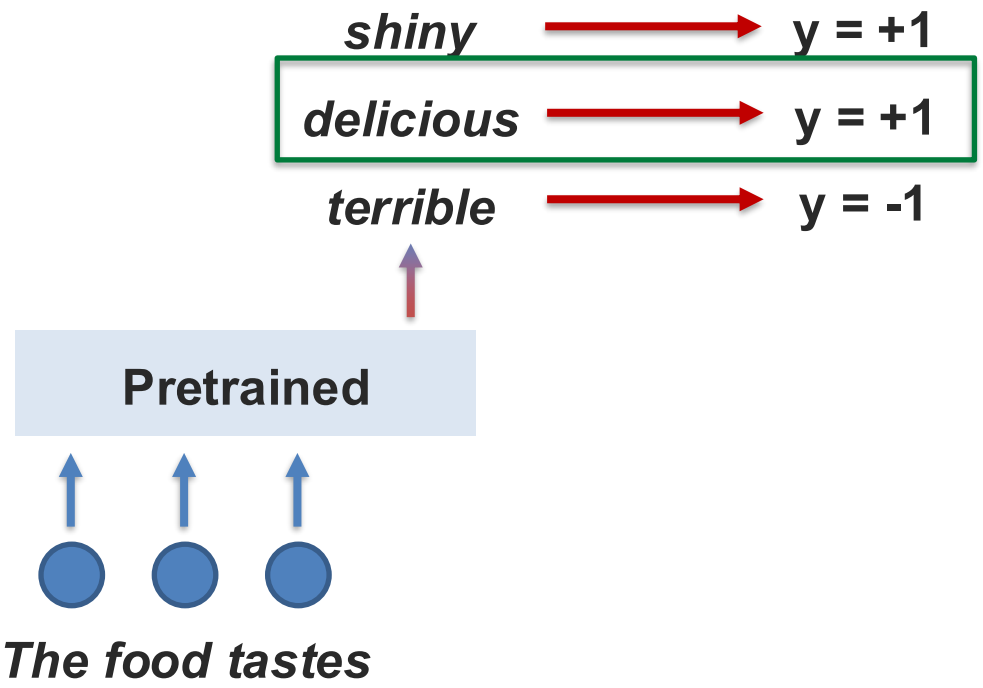
Evaluation Metrics

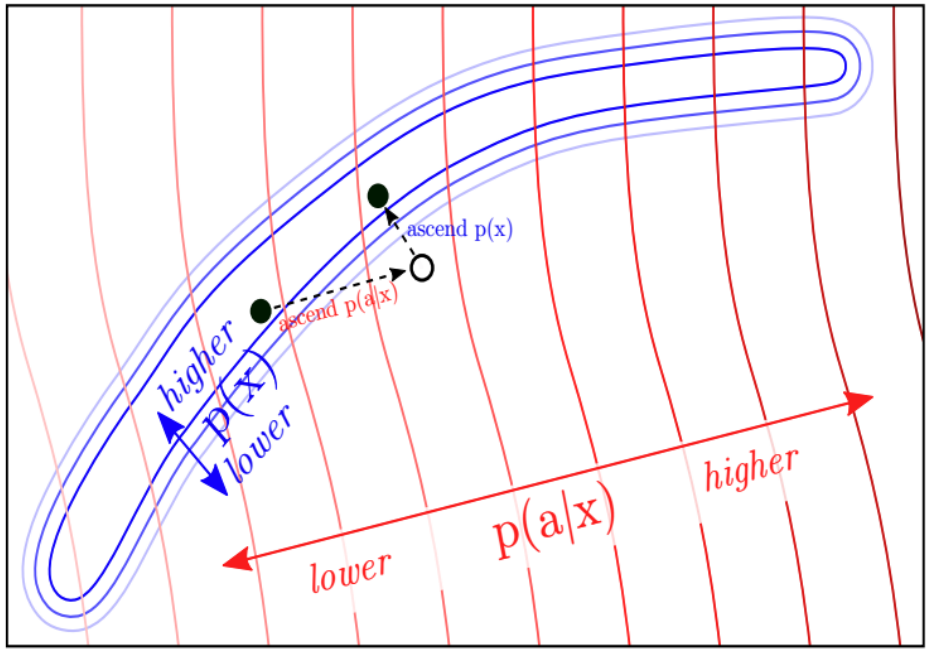
Metric	What It Measures	Intuition	Good Value
FID (Fréchet Inception Distance)	How close generated images are to real images (quality + diversity).	"How much the fake images *feel* like real ones."	Lower is better (e.g., FID < 10 is strong).
CLIP Score	How well the generated image matches the text prompt.	"Did the model generate *what I asked for*?"	Higher is better.
Precision / Recall	Precision = image realism. Recall = variety compared to real images.	"Are the images realistic (precision) and varied (recall)?"	High for both.
Aesthetic Score	How beautiful or professional the images look.	"Would a human think this looks good?"	Higher is better.

Conditioning Generative Models

1. Training unconditional model then classifier guidance

$$\nabla \log p(\boldsymbol{x}_t \mid y) = \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\text{unconditional score}} + \gamma \underbrace{\nabla \log p(y \mid \boldsymbol{x}_t)}_{\text{classifier gradient}}$$





[Dhariwal and Nichol, Diffusion Models Beat GANs on Image Synthesis. arXiv 2021]

multisensory intelligence

Conditioning Generative Models

2. Training unconditional model then classifier-free guidance

$$\begin{aligned}\nabla \log p(\boldsymbol{x}_t \mid y) &= \nabla \log p(\boldsymbol{x}_t) + \gamma \left(\nabla \log p(\boldsymbol{x}_t \mid y) - \nabla \log p(\boldsymbol{x}_t)\right) \\ &= \nabla \log p(\boldsymbol{x}_t) + \gamma \nabla \log p(\boldsymbol{x}_t \mid y) - \gamma \nabla \log p(\boldsymbol{x}_t) \\ &= \underbrace{\gamma \nabla \log p(\boldsymbol{x}_t \mid y)}_{\text{conditional score}} + \underbrace{(1-\gamma) \nabla \log p(\boldsymbol{x}_t)}_{\text{unconditional score}}\end{aligned}$$

2 separate generative model, one conditional and one unconditional?

Just 1 generative model, unconditional training can be seen as setting y=constant

See empirical comparison by GLIDE paper – classifier-free guidance is more preferred

[Nichol et al., GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. arXiv 2022]

Summary: Generative Models

Likelihood-based

1. Autoregressive models – exact inference via chain rule	Easy to train, exact likelihood	Slow to sample from
2. VAEs – approximate inference via evidence lower bound	Fast & easy to train	Lower generation quality
3. Diffusion model – approximate inference via modeling noise	High generation quality	Slow to sample from

Summary: Conditioning Generative Models

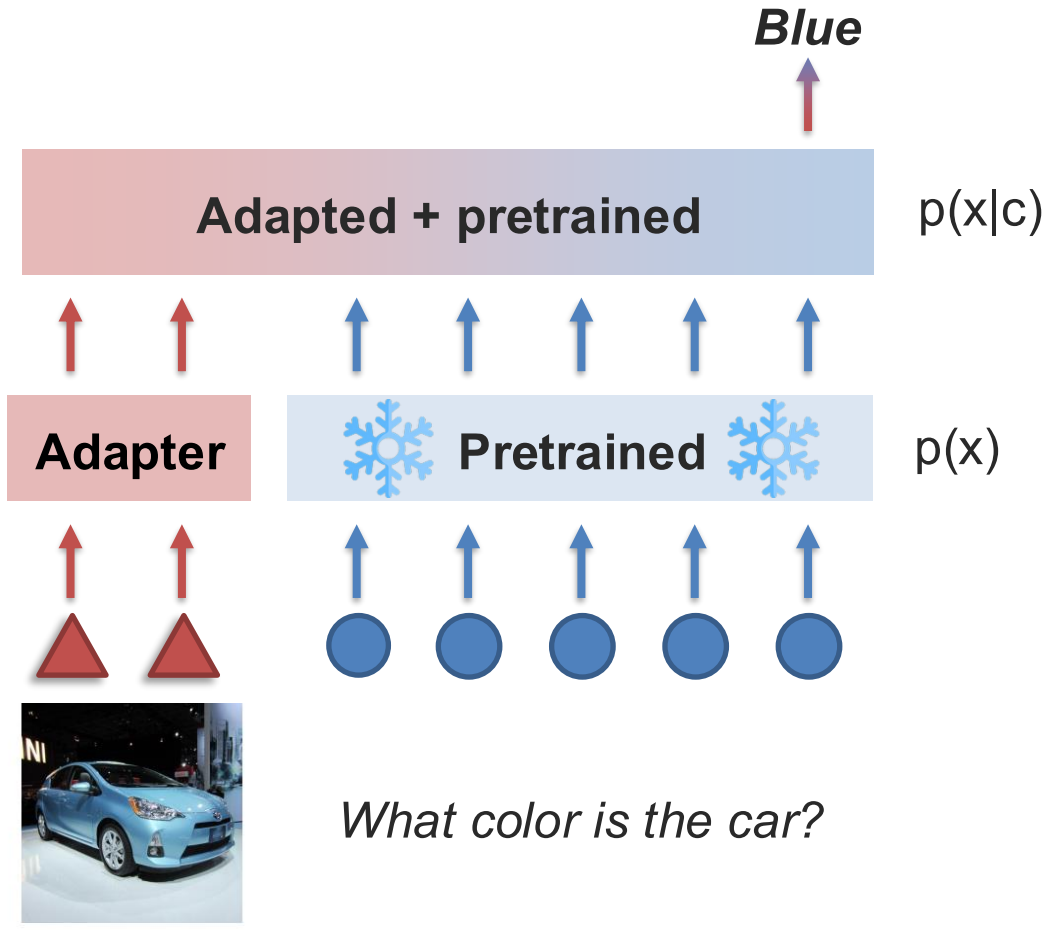
1. Conditioning

$$p(\boldsymbol{x}_{0:T} \mid y) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, y)$$

Summary: Conditioning Generative Models

1. Conditioning

2. Adapting

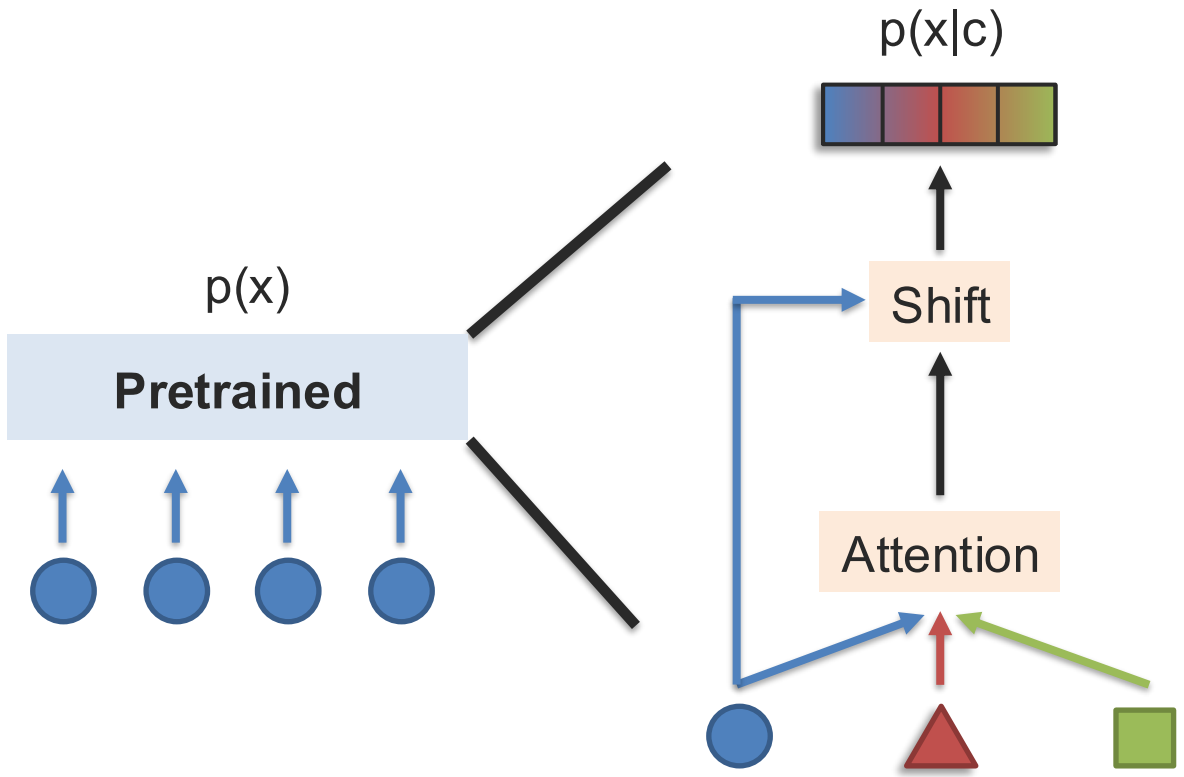


Summary: Conditioning Generative Models

1. Conditioning

2. Adapting

3. Representation tuning



Summary: Conditioning Generative Models

1. Conditioning

$$p(\boldsymbol{x}_{0:T} \mid y) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, y)$$

2. Adapting

3. Representation tuning

4. Classifier gradient tuning

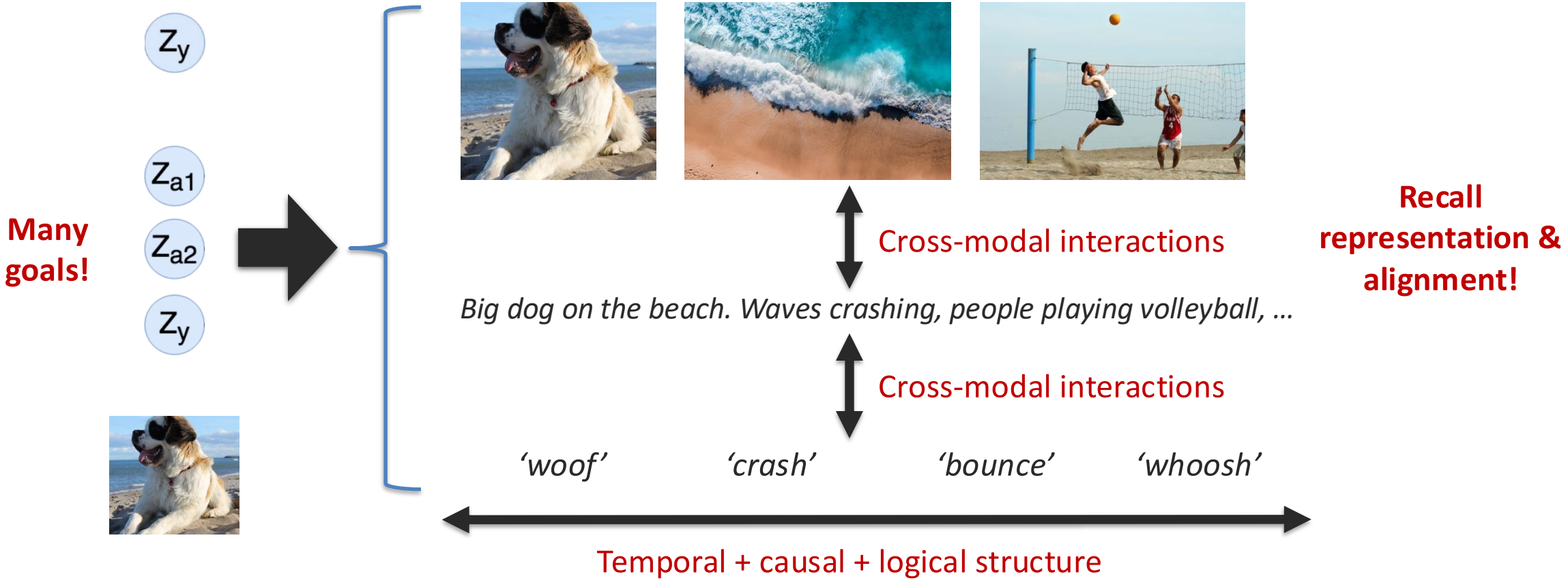
$$\nabla \log p(\boldsymbol{x}_t \mid y) = \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\text{unconditional score}} + \gamma \underbrace{\nabla \log p(y \mid \boldsymbol{x}_t)}_{\text{classifier gradient}}$$

5. Classifier-free tuning

$$\nabla \log p(\boldsymbol{x}_t \mid y) = \underbrace{\gamma \nabla \log p(\boldsymbol{x}_t \mid y)}_{\text{conditional score}} + \underbrace{(1-\gamma) \nabla \log p(\boldsymbol{x}_t)}_{\text{unconditional score}}$$

Open Challenges

Definition: Simultaneously generating multiple modalities to increase information content while maintaining coherence within and across modalities.



Lecture Summary

1. Text-to-image generation
2. Introduction to diffusion and flow matching
3. Diffusion models for other modalities
4. Controlling generative models

Resources

1. Flow Matching for Generative Modelling
2. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior
3. Score-based generative modeling through stochastic differential equation
4. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformer
5. Diffusion Models for Multi-Modal Generative Modelling
6. Multi-Track MusicLDM: Towards Versatile Music Generation with Latent Diffusion Model
7. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
8. https://yang-song.net/blog/

Resources

1. Synchronized generation over multiple modalities.
2. What's special about diffusion models from multimodal perspective?
2. Combining generation with explicit reasoning to enable compositional generation.
3. Better representation fusion and alignment in generation.
4. More control over large-scale generative models, fine-grained + few-shot control.
5. Human-centered evaluation of generative models.

More resources:

https://lilianweng.github.io/tags/generative-model/
https://yang-song.net/blog/2021/score/
https://blog.evjang.com/2018/01/nf1.html & https://blog.evjang.com/2018/01/nf2.html
https://deepgenerativemodels.github.io/syllabus.html
https://www.cs.cmu.edu/~epxing/Class/10708-20/lectures.html
https://cvpr2022-tutorial-diffusion-models.github.io/
https://huggingface.co/blog/annotated-diffusion
https://calvinyluo.com/2022/08/26/diffusion-tutorial.html
https://jmtomczak.github.io/blog/1/1_introduction.html

Assignments for This Coming Week

David sent out announcement regarding course credits.

Edgar sent out announcement regarding midterm conflicts/accommodations.

Midterm review next Tuesday 3/17. Midterm exam next Thursday 3/19.

Project mentors released. Try to meet as often as you can. Meet with me today.