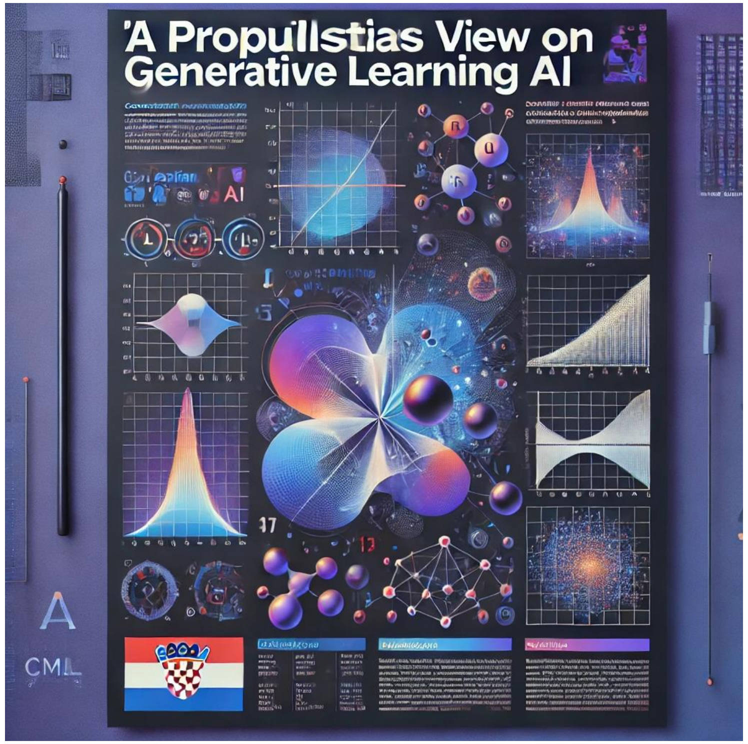
A probabilistic view on Generative AI

Matej Grcić

UniZG-FER

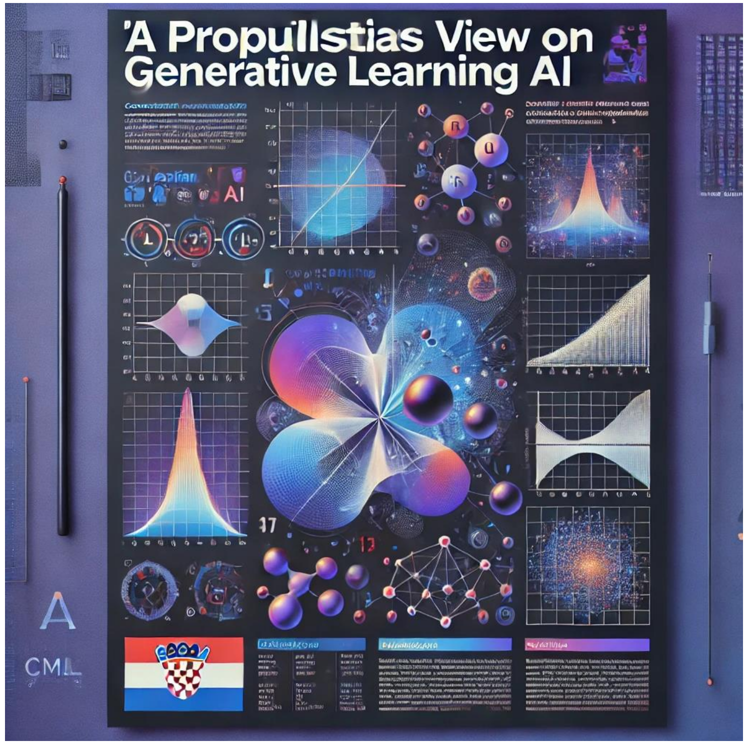
Almighty GenAI

Generate an image for a talk titled "A probabilistic view on Generative AI" at Croatian Machine Learning Workshop (CMLW)



How did we end up here?

Generate an image for a talk titled "A probabilistic view on Generative AI" at Croatian Machine Learning Workshop (CMLW)



Agenda

Preliminaries

Energy-based models

Normalizing flows

Diffusion models

Flow matching

Applications of generative models

Conclusion & outlook

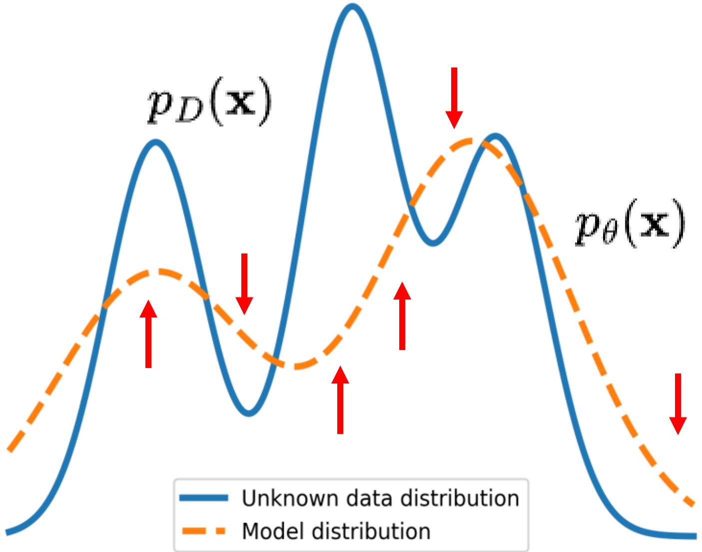
Preliminaries

$(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), \mathbb{P})$	d-dimensional probability space
$\mathbf{x}$	Random variable (d-dimensional vector)
$p(\mathbf{x})$	Probability density function
$\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \dots, \mathbf{x}^i$	Realizations of a random variable
$\mathbf{x}^i \sim p(\mathbf{x})$	Sampling the distribution of **x**





How to learn the unknown data distribution?



Sample

$$\min_\theta \ \mathrm{KL}[p_D || \, p_\theta] \cong \max_\theta \ \mathbb{E}_{\mathbf{x} \sim p_D}[\ln p_\theta(\mathbf{x})] \approx \max_\theta \ \mathbb{E}_{\mathbf{x} \in \mathcal{D}}[\ln p_\theta(\mathbf{x})]$$

$\mathcal{D} = \{$   ...   $\}$

Energy-based models

How it all started..

Boltzmann's energy with deep models

Unnormalized distribution — Energy function

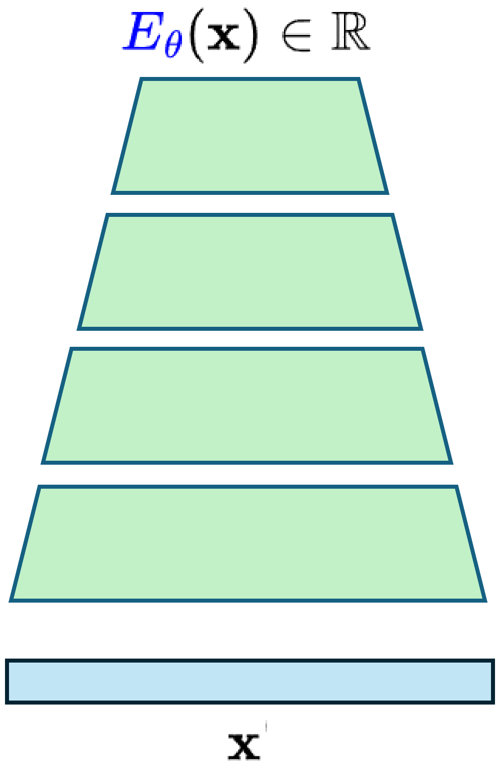
Model distribution

$$p_\theta(\mathbf{x}) = \frac{p'_\theta(\mathbf{x})}{Z(\theta)} = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}$$

Normalization constant

$$Z(\theta) = \int \exp(-E_\theta(\mathbf{x}))\ d\mathbf{x}$$

$$\min_\theta \mathbb{E}_{\mathbf{x}\sim p_D}[-\ln p_\theta(\mathbf{x})]$$ **= ???**

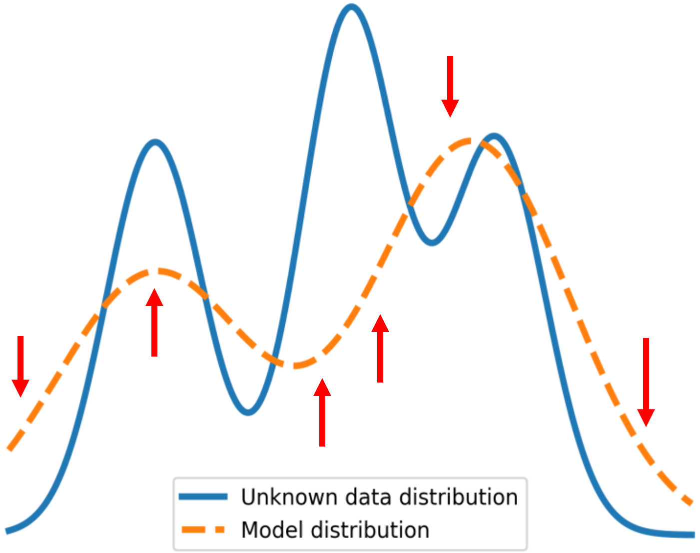


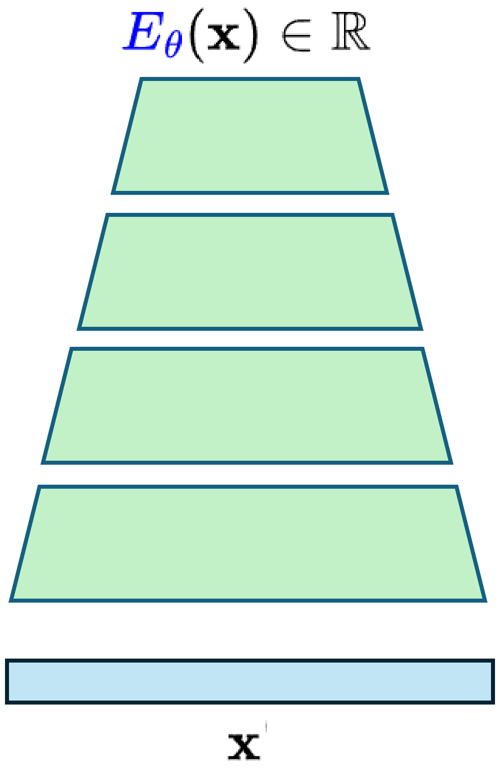
Optimizing the energy surface

$$\min_{\theta} \mathbb{E}_{\mathbf{x} \sim p_D}[-\ln p_\theta(\mathbf{x})]$$

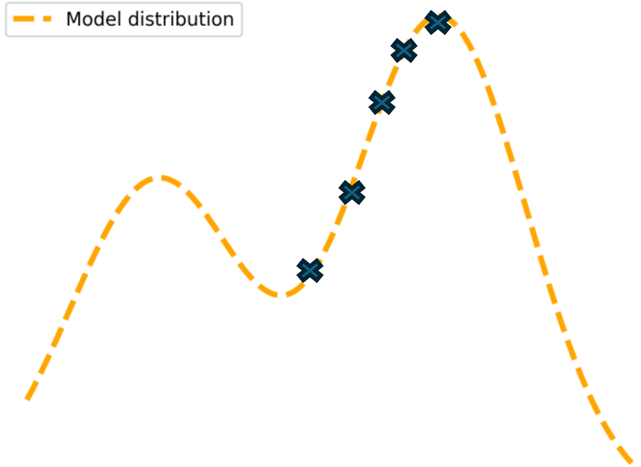
Optimized by **gradient** descent

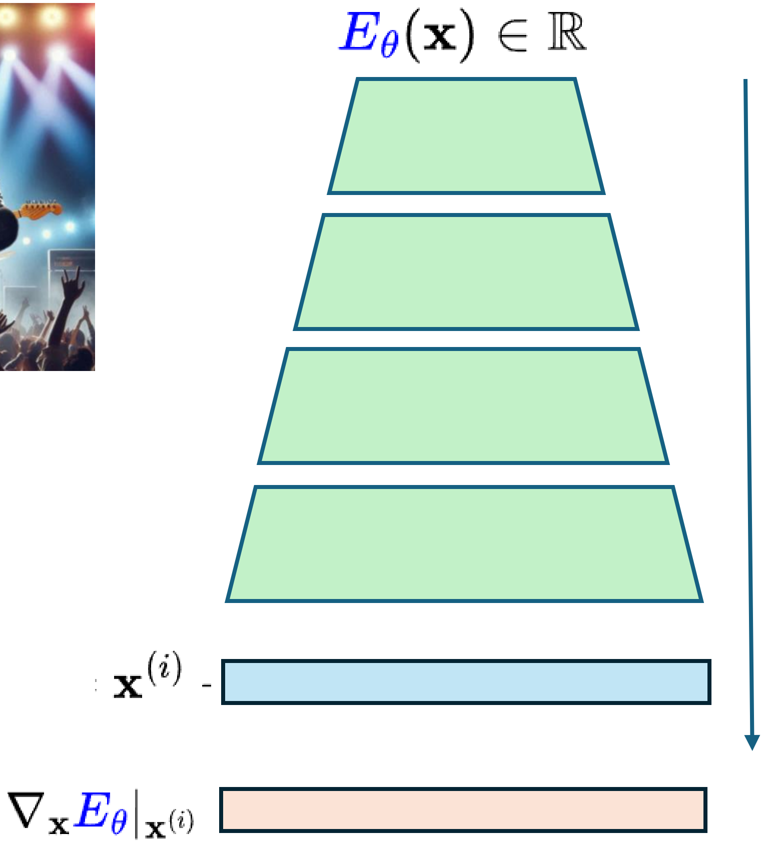
$$\nabla_\theta \mathbb{E}_{\mathbf{x} \sim p_D}[-\ln p_\theta(\mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim p_D}[\nabla_\theta E_\theta(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_\theta}[\nabla_\theta E_\theta(\mathbf{x})]$$





Iterative data generation





$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \frac{\alpha_i}{2} \cdot \nabla_{\mathbf{x}} \ln p_\theta(\mathbf{x})|_{\mathbf{x}^{(i)}} + \alpha_i \cdot \epsilon, \qquad \epsilon \sim \mathcal{N}(0,1), \; \lim_{i \to \infty} \alpha_i = 0$$

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \frac{\alpha_i}{2} \cdot \nabla_{\mathbf{x}} E_\theta|_{\mathbf{x}^{(i)}} + \alpha_i \cdot \epsilon$$

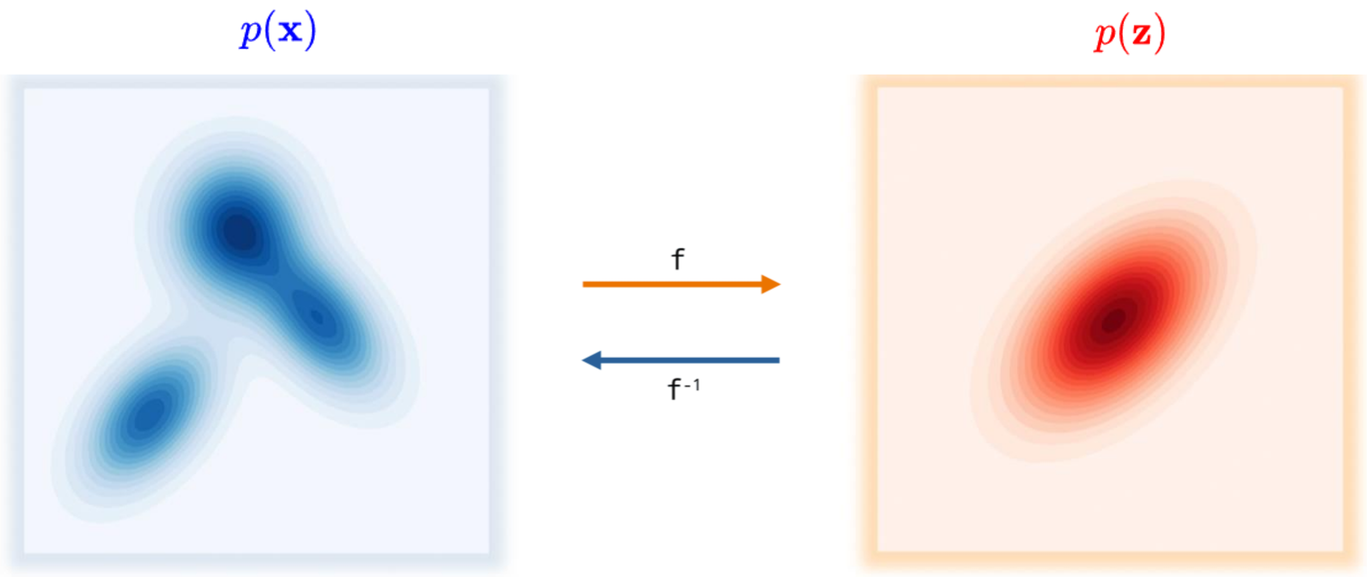
$$\nabla_{\mathbf{x}} \ln p_\theta(\mathbf{x})|_{\mathbf{x}^{(i)}} = f_\theta(\mathbf{x}^{(i)})$$

Score matching

Normalizing flows

Let's get rid of normalization constant Z

Change of random variables



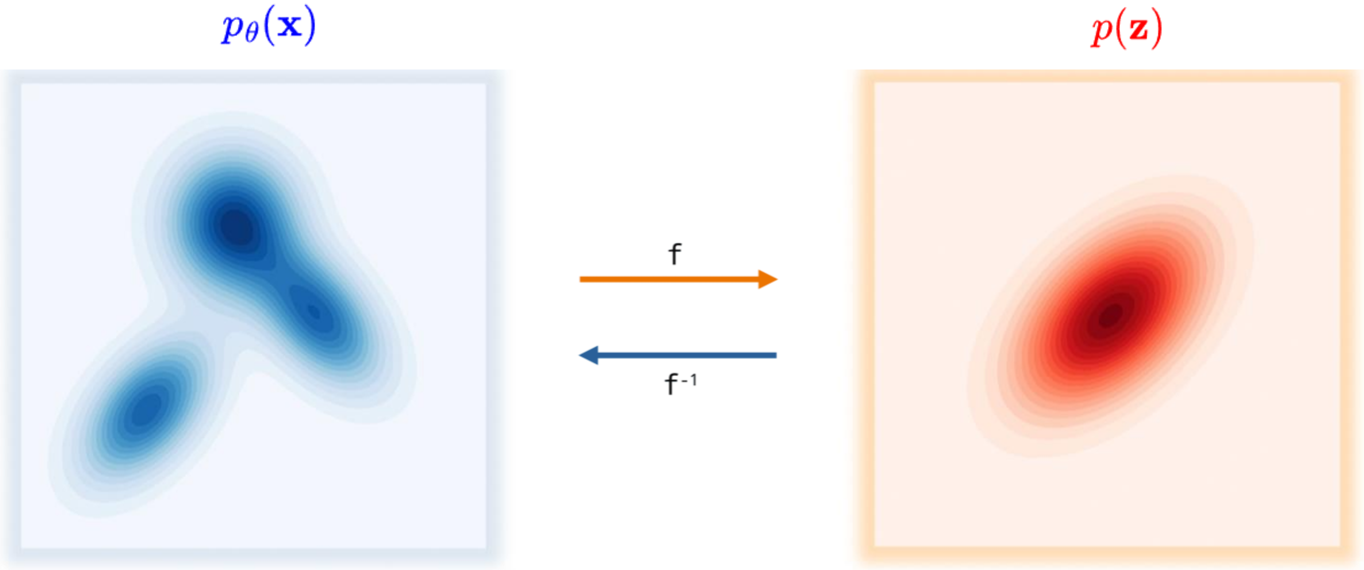
$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|, \qquad \mathbf{z} = f(\mathbf{x}) \qquad f : \mathbb{R}^d \to \mathbb{R}^d$$

The determinant
of Jacobian of *f*

Diffeomorphism
(function and its inverse are differentiable)

Generative modeling by change of variables

Some complex model distribution



Known prior distribution e.g **Gaussian**

$$p_\theta(\mathbf{x}) = p(\mathbf{z}) \left| \det \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right|, \qquad \mathbf{z} = f_\theta(\mathbf{x}) \qquad f : \mathbb{R}^d \to \mathbb{R}^d$$

The determinant of Jacobian of *f*

Parameterized diffeomorphism (Deep model)

Normalizing flows

$$p_\theta(\mathbf{x}) = p(\mathbf{z}) \left| \det \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right|, \quad \mathbf{z} = f_\theta(\mathbf{x})$$

Diffeomorphism

$$\mathbf{x} = \mathbf{z}_0 \overset{f_1}{\longleftrightarrow} \mathbf{z}_1 \overset{f_2}{\longleftrightarrow} \mathbf{z}_2 \overset{f_3}{\longleftrightarrow} \cdots \overset{f_{i-1}}{\longleftrightarrow} \mathbf{z}_i \overset{f_i}{\longleftrightarrow} \cdots \overset{f_T}{\longleftrightarrow} \mathbf{z}_T, \quad \mathbf{z}_T \sim \mathcal{N}(0, \mathrm{I}).$$

Diffeomorphism

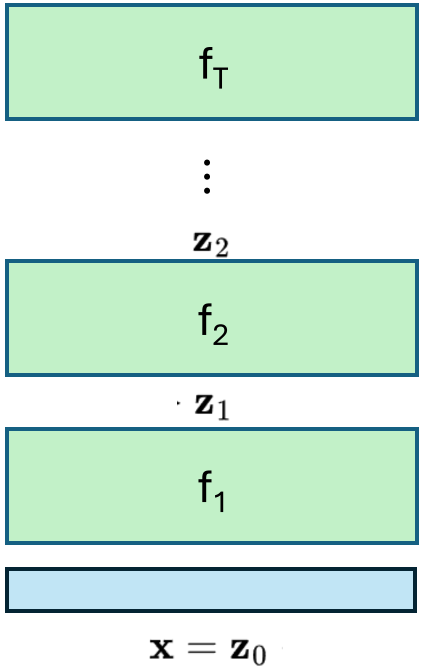
$$\ln p_\theta(\mathbf{x}) = \ln p(\mathbf{z}_T) + \sum_{t=1}^{T} \ln \left| \det \frac{\partial \mathbf{z}_t}{\partial \mathbf{z}_{t-1}} \right|, \quad \mathbf{z}_t = f_{\theta_t}(\mathbf{z}_{t-1})$$

Invertible?

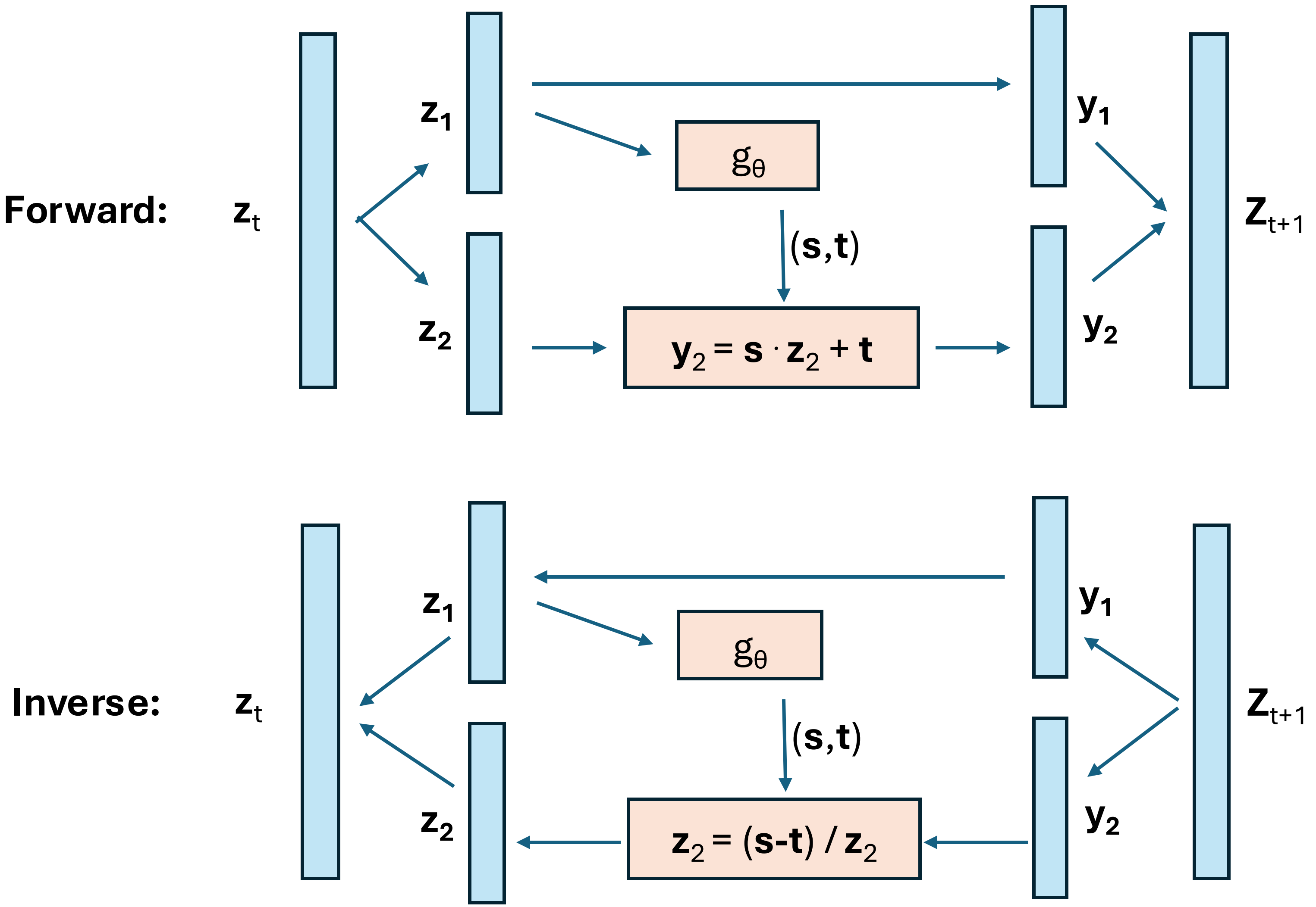
O(d³) complexity?

$$\dim(\mathbf{z}_T) = \dim(\mathbf{x})$$

$$\mathbf{z}_T \sim \mathcal{N}(0, \mathrm{I}).$$

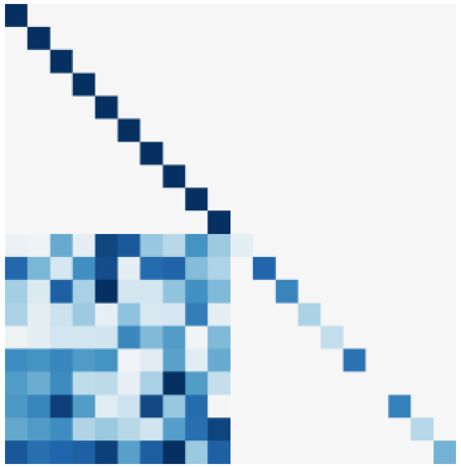


Normalizing flows with affine coupling layers



Jacobian of coupling layer

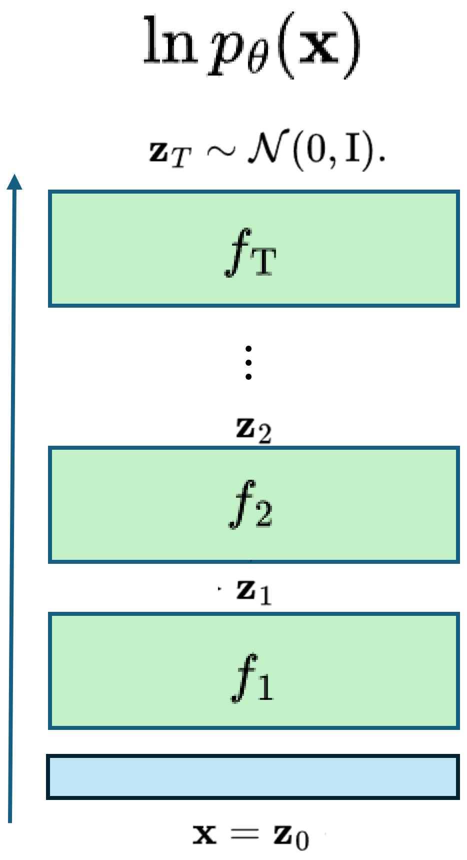
$$\frac{\partial \mathbf{z}_{t+1}}{\partial \mathbf{z}_t} = ?$$

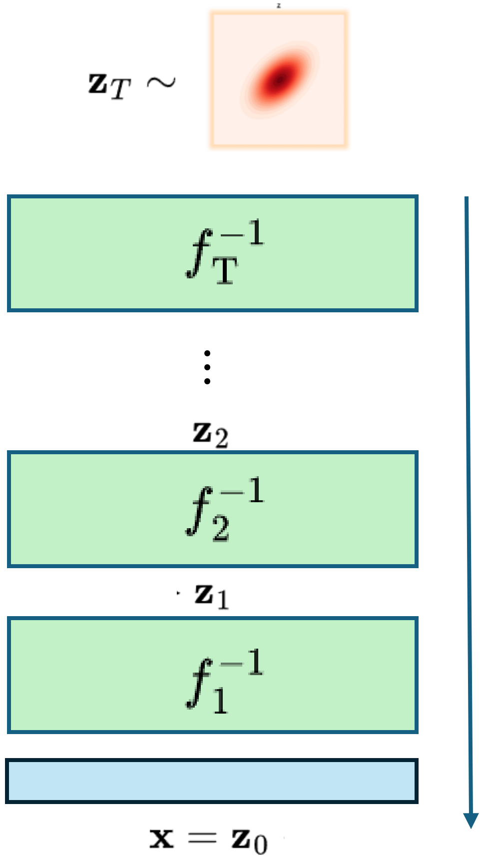


Lower triangular matrix

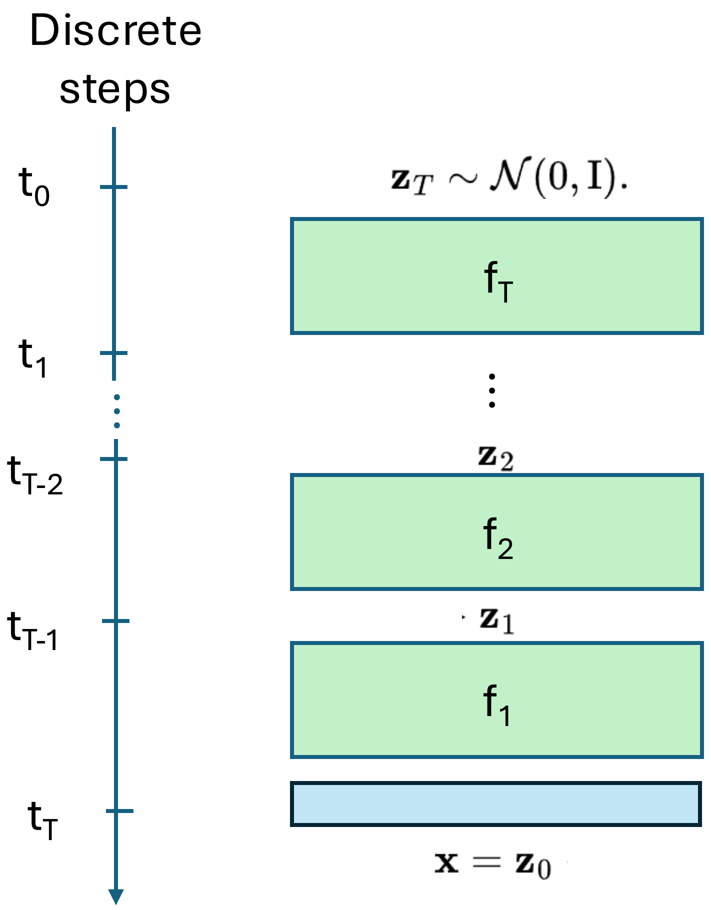
$$\left| \det \frac{\partial \mathbf{z}_{t+1}}{\partial \mathbf{z}_t} \right|$$ has O(d) complexity

Forward/inverse with normalizing flows





Discrete normalizing flows



Continuous normalizing flows

Ordinary differential equation (IVP)

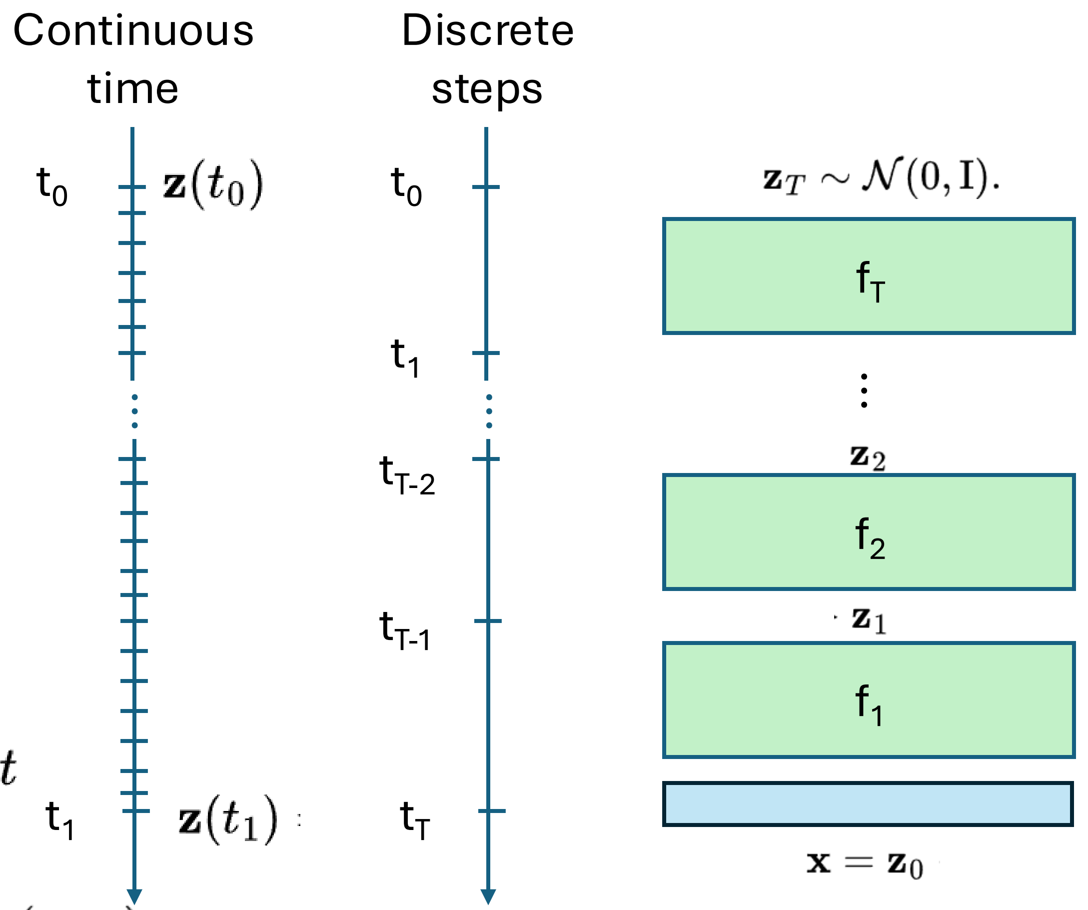
$$\frac{d\mathbf{z}(t)}{dt} = f(\mathbf{z}(t), t; \theta), \quad \mathbf{z}(t_1) = \mathbf{x}$$

$$\mathbf{z}(t_0) = \mathbf{z}(t_1) + \int_{t_1}^{t_0} f(\mathbf{z}(t), t; \theta)\ dt$$

Instantaneous change of variables

$$\frac{d \ln p(\mathbf{z}(t))}{dt} = -\mathrm{tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right)$$

$$\ln p(\mathbf{z}(t_1)) = \ln p(\mathbf{z}(t_0)) - \int_{t_0}^{t_1} \mathrm{tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right)\ dt$$



$$\ln p_\theta(\mathbf{x}) = \ln p(\mathbf{z}_T) + \sum_{t=1}^{T} \ln \left| \det \frac{\partial \mathbf{z}_t}{\partial \mathbf{z}_{t-1}} \right|, \quad \mathbf{z}_t = f_{\theta_t}(\mathbf{z}_{t-1})$$

Backpropagation trough ODE solver!

Flow matching

Go with the flow

Reinterpreting continuous normalizing flows

Continuous NF ODE:

$$\frac{d\mathbf{z}(t)}{dt} = f(\mathbf{z}(t), t; \theta), \quad \mathbf{z}(t_1) = \mathbf{x}$$

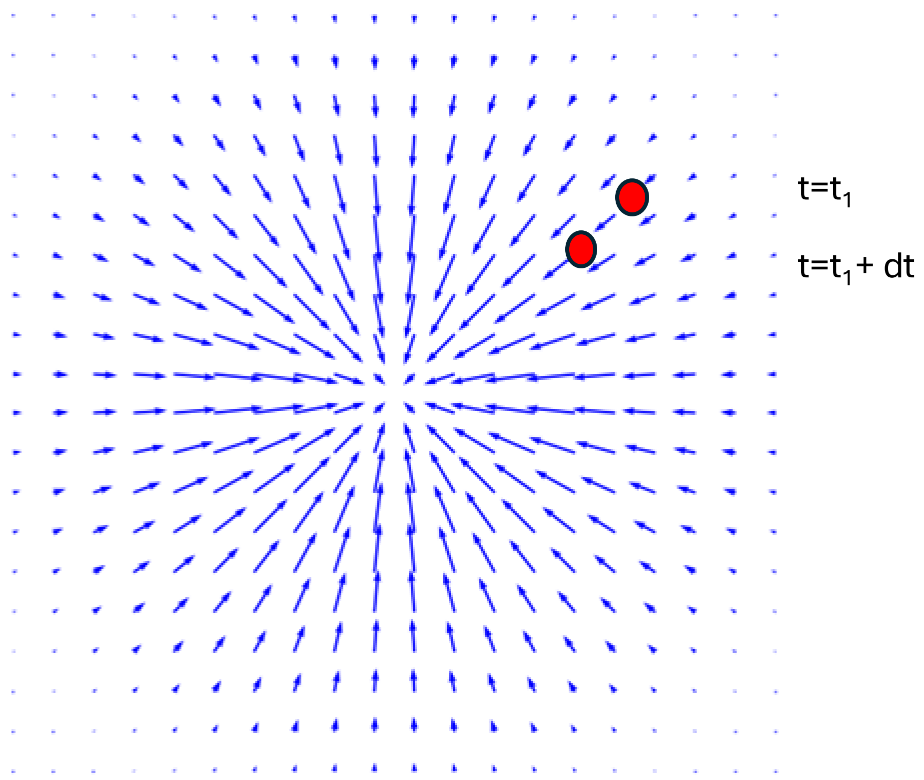
Flow matching ODE:

$$\frac{d}{dt}\phi_t(\mathbf{x}) = v_t(\phi_t(\mathbf{x})) \quad \phi_0(\mathbf{x}) = \mathbf{x}$$

Diffeomorphic map (flow): $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$

Vector field: $v_t : \mathbb{R}^d \to \mathbb{R}^d$

Probability path: $p_t : \mathbb{R}^d \to \mathbb{R}_+$



Example of vector field

Reinterpreting continuous normalizing flows

Diffeomorphic map (flow): $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$

Vector field: $v_t : \mathbb{R}^d \to \mathbb{R}^d$

Probability path: $p_t : \mathbb{R}^d \to \mathbb{R}_+$

Flow matching ODE:

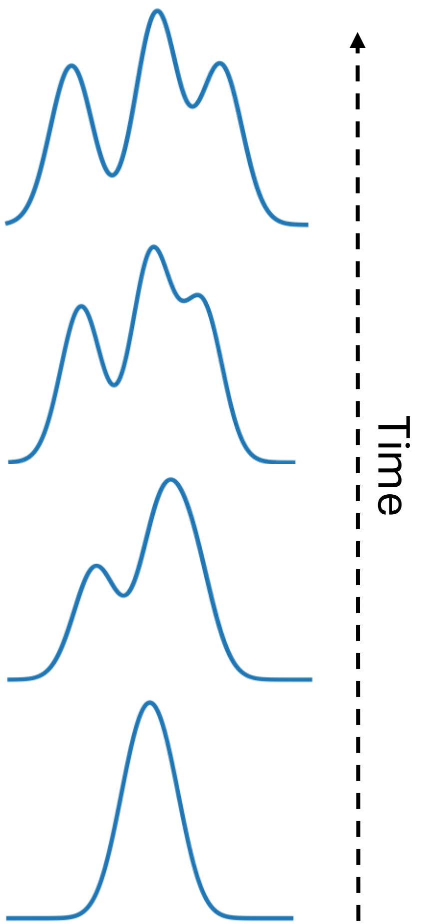
$$\frac{d}{dt}\phi_t(\mathbf{x}) = v_t(\phi_t(\mathbf{x})) \qquad \phi_0(\mathbf{x}) = \mathbf{x}$$

Continuity equation (in log form)**:**

$$\frac{d}{dt}\ln p_t(\phi_t(\mathbf{x})) - \mathrm{div}(v_t(\phi_t(\mathbf{x}))) = 0$$

Implicitly defines

$p_{t=1}(\mathbf{x}) = q(\mathbf{x}) = p_D(\mathbf{x})$



$p_{t=0}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, \mathbf{I})$

Flow matching objective

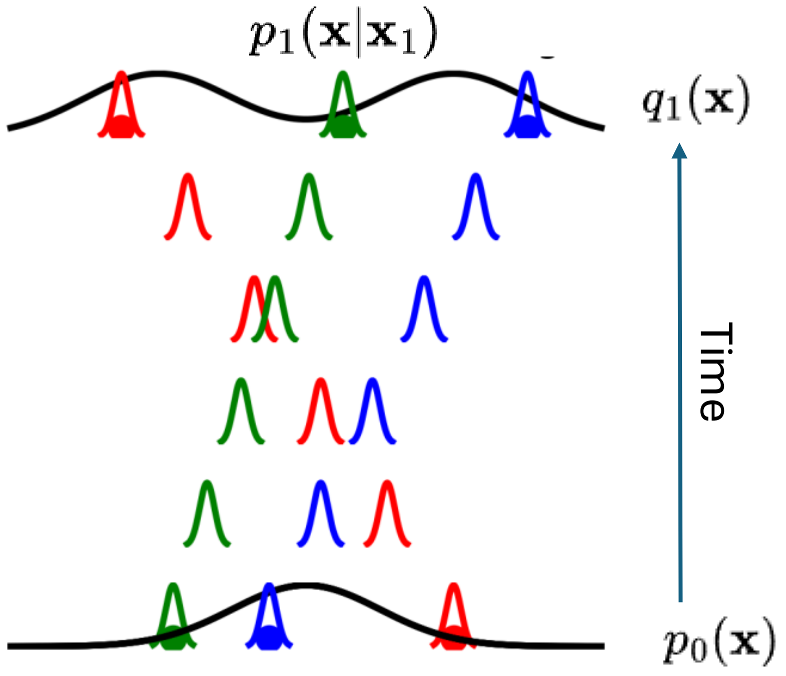
$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\,\mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[||v_t(\mathbf{x};\theta) - u_t(\mathbf{x})||_2^2\right]$$

Vector field of distribution *q* **(unknown)**

Conditional probability path

Marginal probability path

$$p_t(\mathbf{x}) = \int p_t(\mathbf{x}|\mathbf{x}_1)\,q(\mathbf{x}_1)\,d\mathbf{x}_1$$



Conditional vector field

$$u_t(\mathbf{x};\mathbf{x}_1)$$

Conditional flow

$$\frac{d}{dt}\psi_t(\mathbf{x}) = u_t(\psi_t(\mathbf{x});\mathbf{x}_1)$$

Conditional flow matching (**tractable**):

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\,\mathbb{E}_{\mathbf{x}_1\sim q(\mathbf{x}_1)}\,\mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x}|\mathbf{x}_1)}\left[||v_t(\mathbf{x};\theta) - u_t(\mathbf{x};\mathbf{x}_1)||_2^2\right]$$

$$\nabla_\theta\,\mathcal{L}_{\mathrm{FM}}(\theta) = \nabla_\theta\,\mathcal{L}_{\mathrm{CFM}}(\theta)$$

Rectified flow matching – Stable Diffusion 3

Goal: Learn the shortest paths between conditional distribuiton and the prior.

Conditional flow: $\psi_t(\mathbf{x}) = t \cdot \mathbf{x}_1 + (1 - (1 - \sigma_{\min})t) \cdot \mathbf{x}$

t=0: $\psi_0(\mathbf{x}) = \mathbf{x}_0$

t=1: $\psi_1(\mathbf{x}) = \mathbf{x}_1 + \sigma_{\min} \cdot \mathbf{x}$

Diffusion models

Noising is easy, denoising is hard.

Denoising diffusion

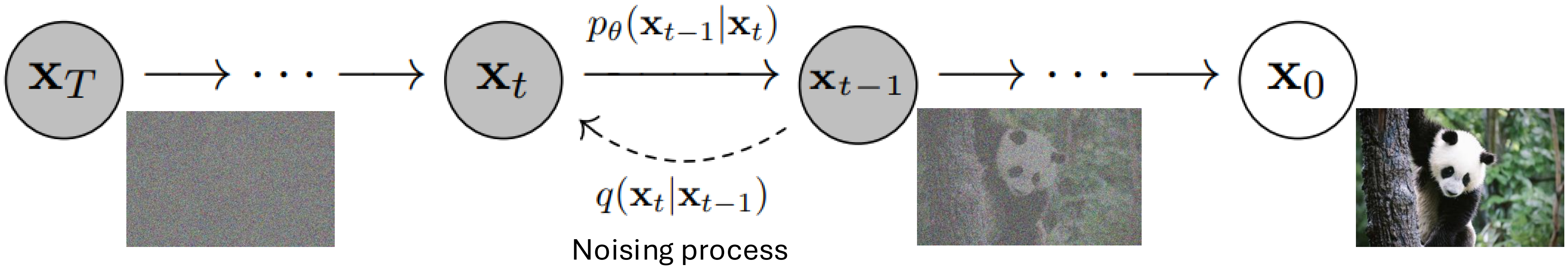
$$\mathbf{x}_0 \sim p_D \qquad p_\theta(\mathbf{x}_0) = \int p_\theta(\overbrace{\mathbf{x}_T, \ldots, \mathbf{x}_1}^{\text{Latents}}, \mathbf{x}_0)\; d\mathbf{x}_T \ldots d\mathbf{x}_0 = \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$$

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T})\; d\mathbf{x}_{1:T} = \int p_\theta(\mathbf{x}_{0:T}) \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\, d\mathbf{x}_{1:T} \quad \Longrightarrow \quad \ln p_\theta(\mathbf{x}_0) \geq \boxed{\mathbb{E}_q \left[\ln \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]}$$

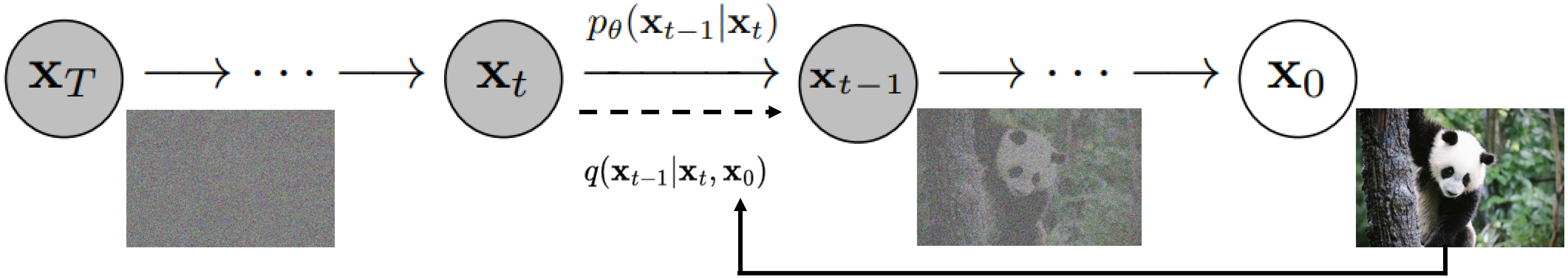
Markov chain factorization:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t|\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$



Denoising diffusion in practice



$$\ln p_\theta(\mathbf{x}_T) \geq \mathbb{E}_q\left[\ln \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \approx -\sum_{t>1} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\mathrm{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$$

$$\Downarrow$$

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$$

$$\sum_{t>1} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[c(t) \cdot ||\boldsymbol{\mu}_q(\mathbf{x}_t, t) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)||_2^2]$$

$$\Downarrow$$

$$\min_\theta \mathbb{E}_{t\sim\mathcal{U}\{1,T\}} \mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\tilde{c}(t) \cdot ||\boldsymbol{\epsilon}_0 - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)||_2^2]$$

Generating samples by denoising

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ $\mathbf{x}_0 \sim p_D$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \sigma_t \mathbf{z} \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

Conditional generation

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ $\mathbf{x}_0 \sim p_D$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \sigma_t \mathbf{z} \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{y}, t) + \sigma_t \mathbf{z} \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

Conditional generation

Rockstar panda being a hunter in Pieter Bruegel the Elder's Hunters in the Snow painting.

The Hunters in the Snow
Pieter Bruegel the Elder, 1565.

Rockstar Panda in the Snow
Some nonlinear function, 2024.

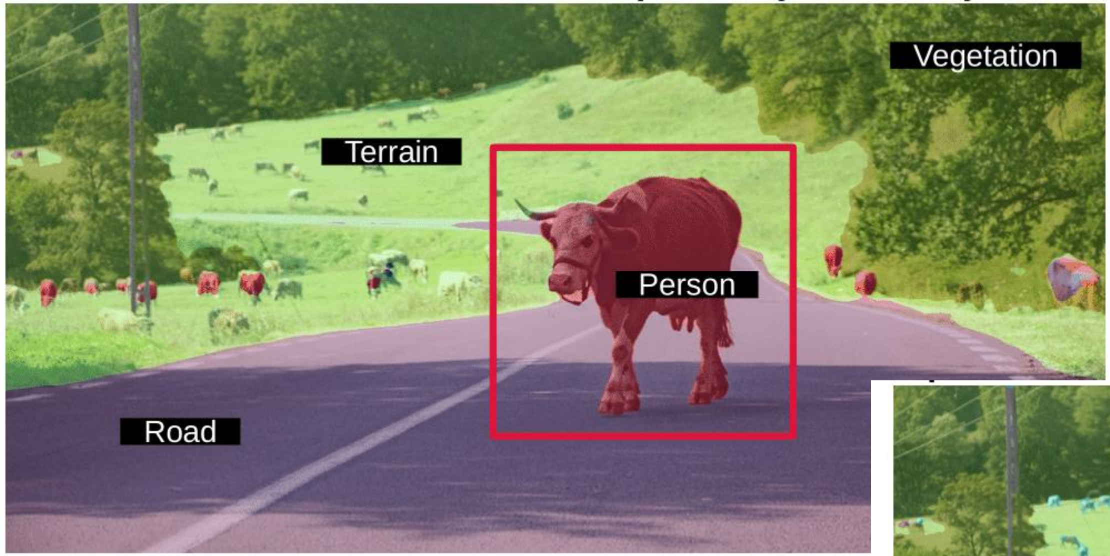
Applications

Transportation, biology, climate, ...

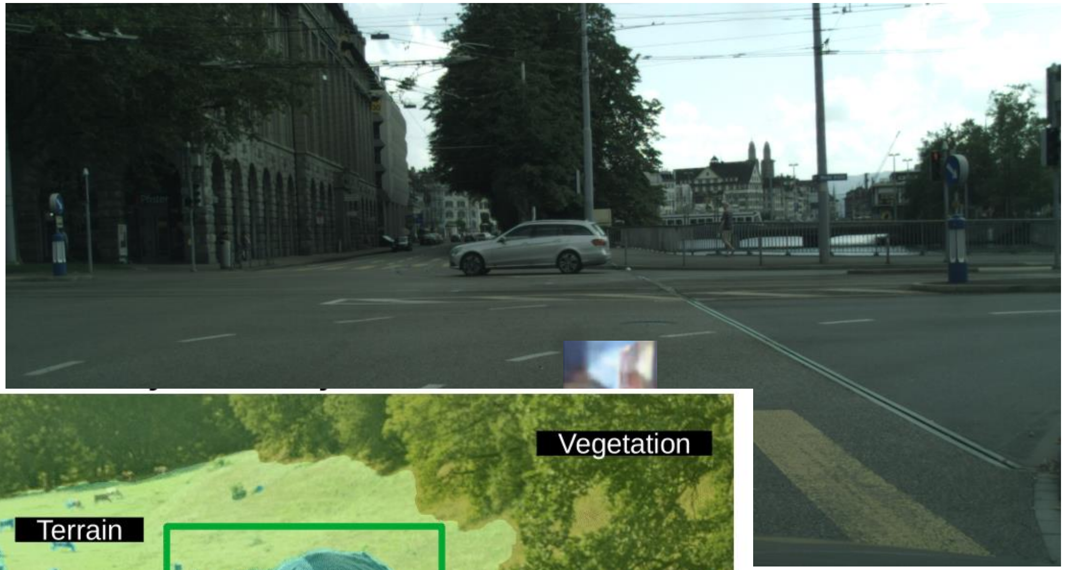
Unexpected objects in traffic

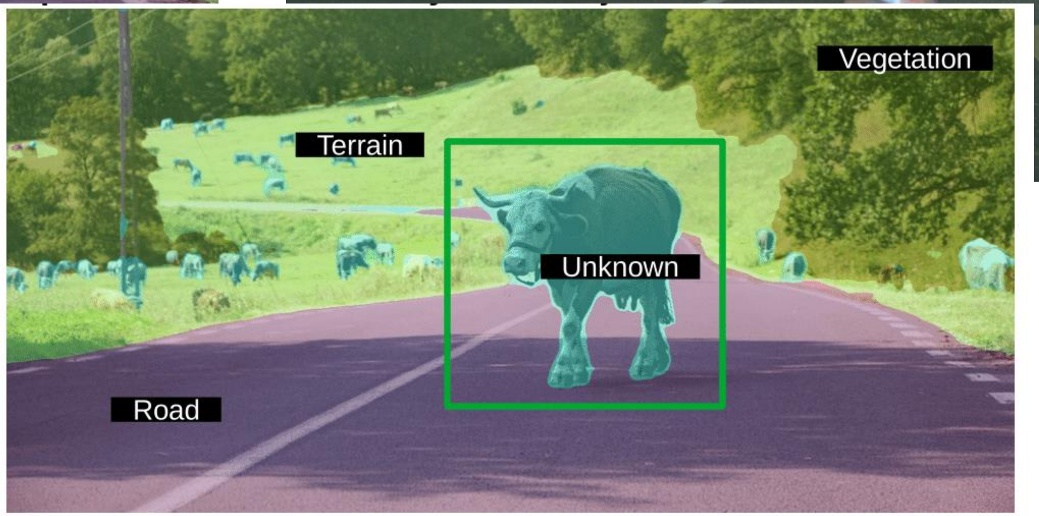
Training on simulated anomalies helps!

Unknown animal recognized as something known.



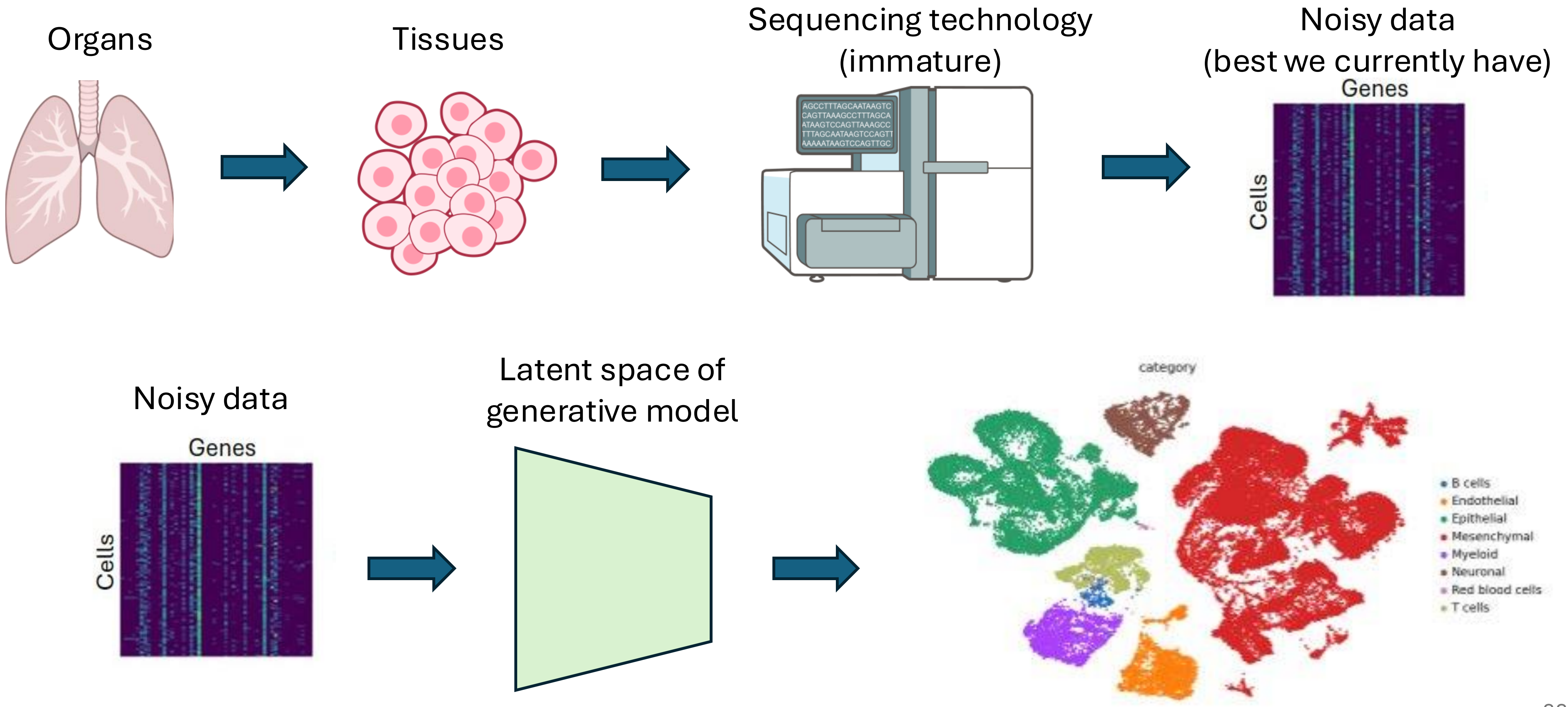
Simulate anomalies with generative models.





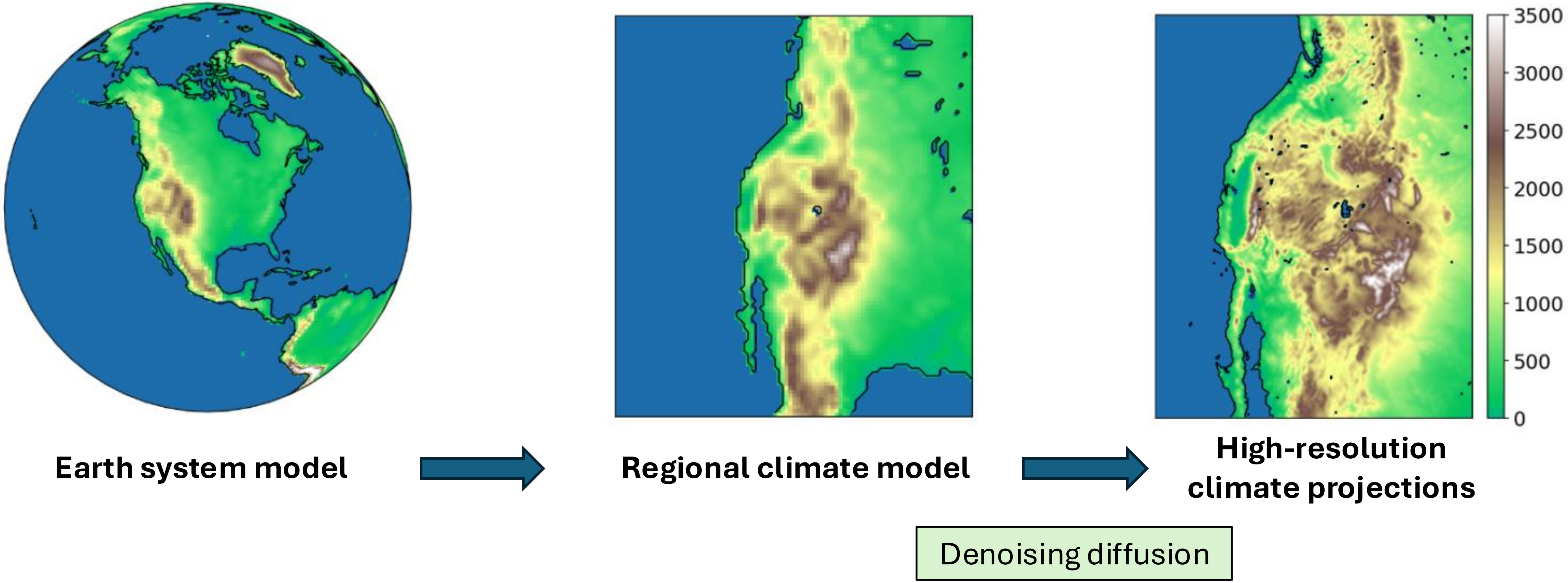
Hybrid Open-set Segmentation with Synthetic Negative Data. Grcić & Šegvić, 2024.

Uncovering biological signals in noisy data



Deep generative modeling for single-cell transcriptomics, Lopez et al. 2018.

Improved climate projections



Dynamical-generative downscaling of climate model ensembles. Lopey-Gomez et al. 2024.

Conclusion & outlook

What's next?

Conclusion

Latent variable models

Diffusion models

Flow matching

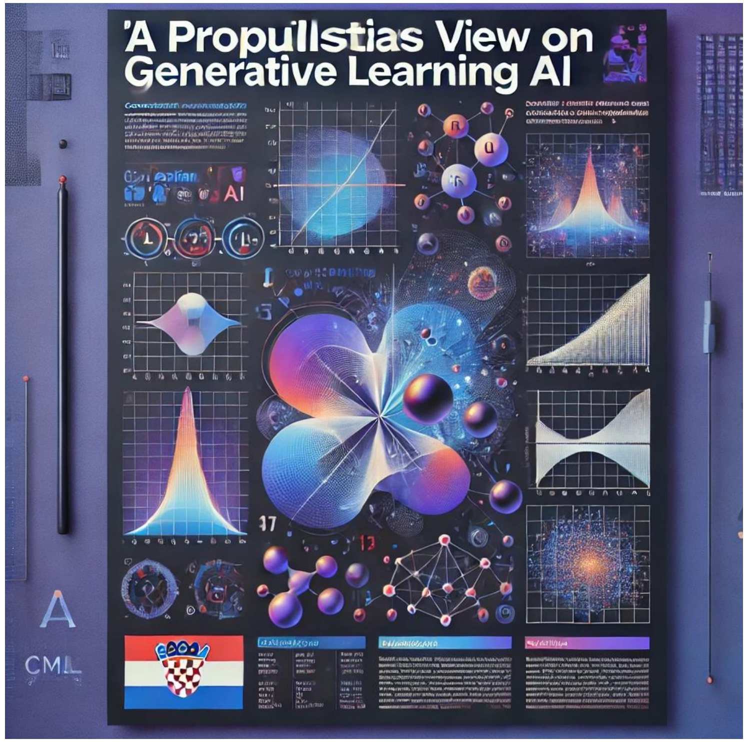
Continuous NFs

Normalizing flows

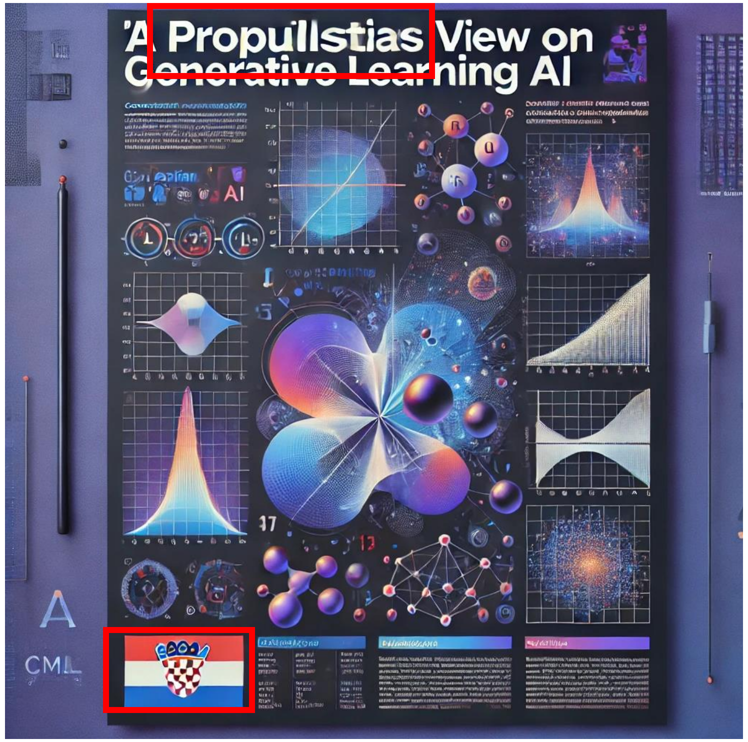
Score matching

Energy-based models

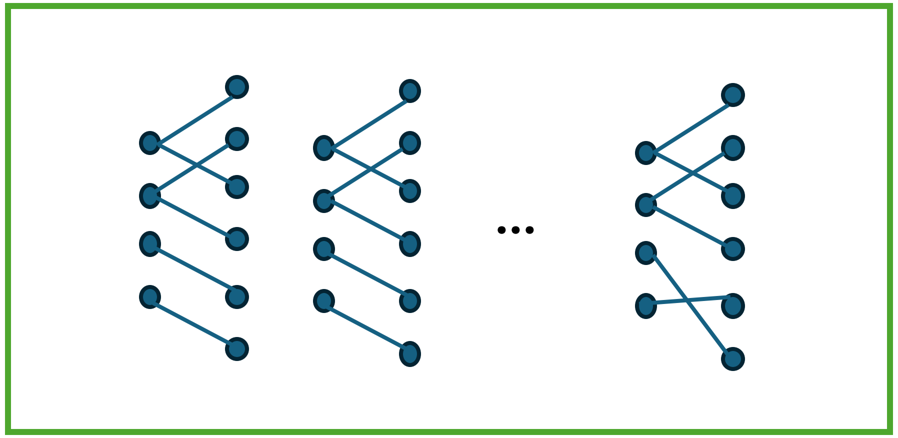
Current issues



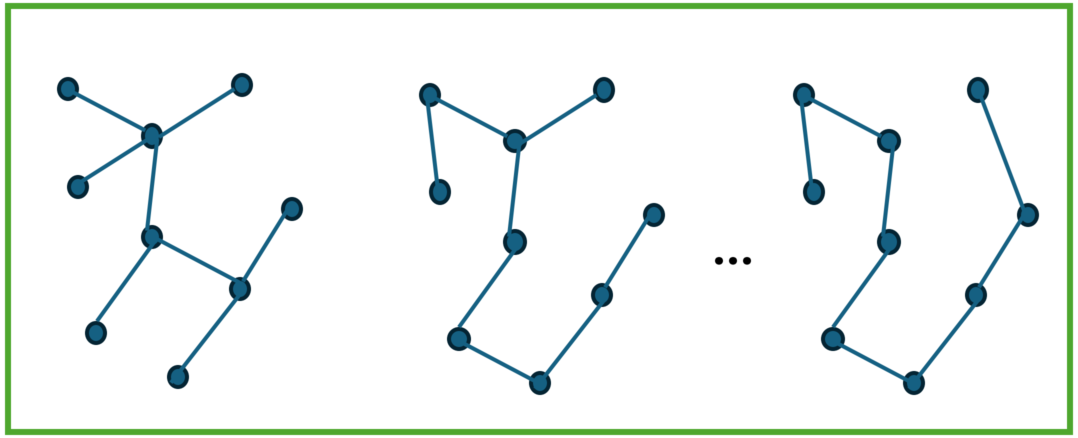
Current issues



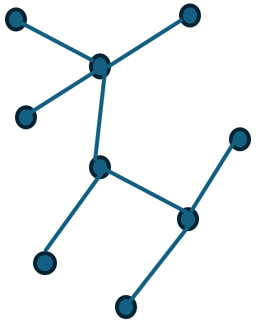
Modeling structured random variables



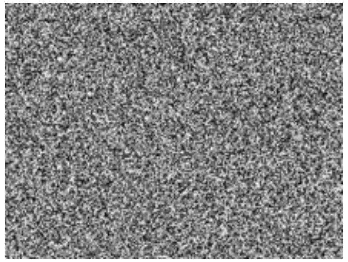
What if realizations of **x** are **bipartite graphs**?



What if realizations of **x** are **spanning trees**?



+



Noise

=

References & useful reads

- "How to Train Your Energy-Based Models" Song & Kingma, 2021.
- "Flow matching for generative modeling" Lipman et al., 2023.
- "Understanding Diffusion Models: A Unified Perspective", Calvin Luo, 2022.
- "Glow: Generative Flow with Invertible 1×1 Convolutions" Kingma & Dhariwal, 2018.
- "Neural Ordinary Differential Equations" T.Q. Chan et al. 2018.
- "Classifier-Free Diffusion Guidance" Ho & Salimans, 2021.
- Blogposts, Jakub M. Tomczak, https://jmtomczak.github.io