Neural Differential Equations with applications in generative models

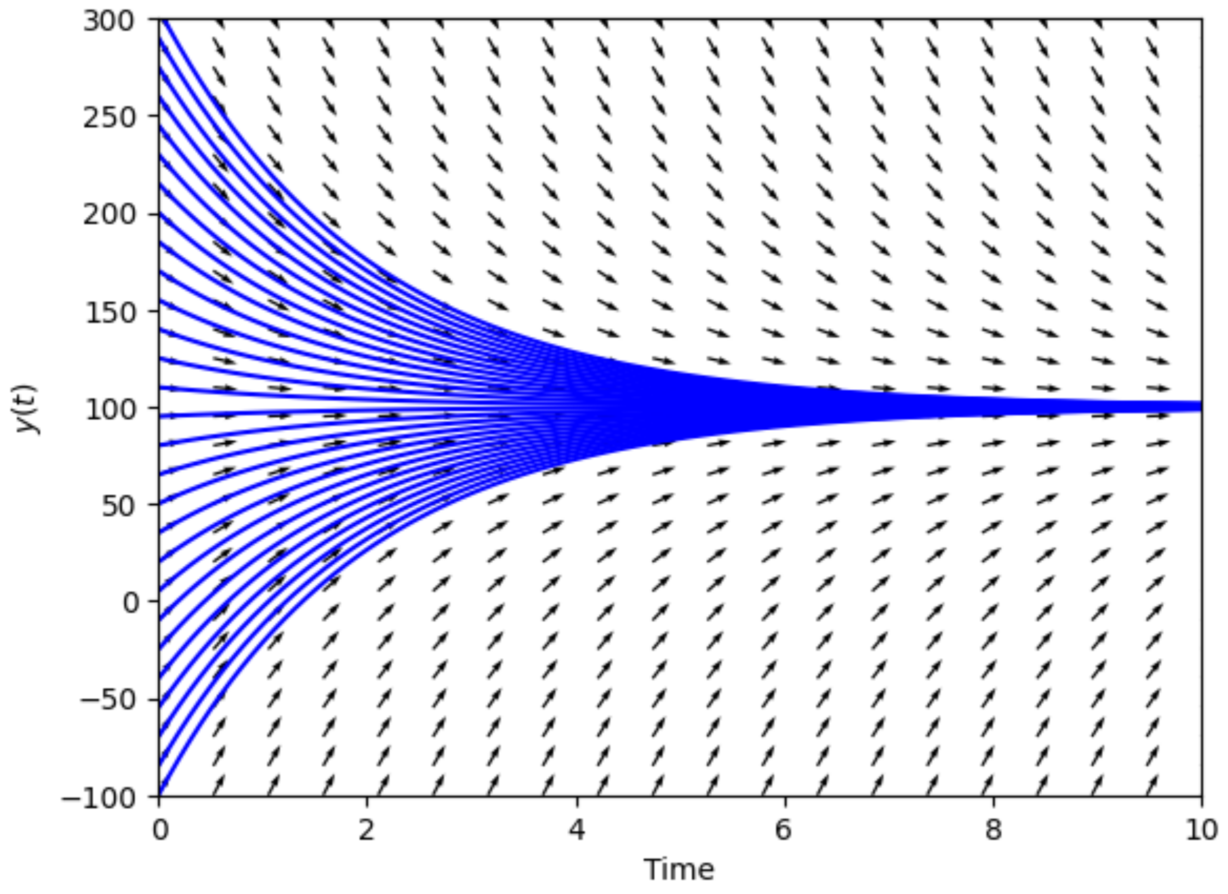
EaGR workshop @ AIM

Nicole Yang, Emory University

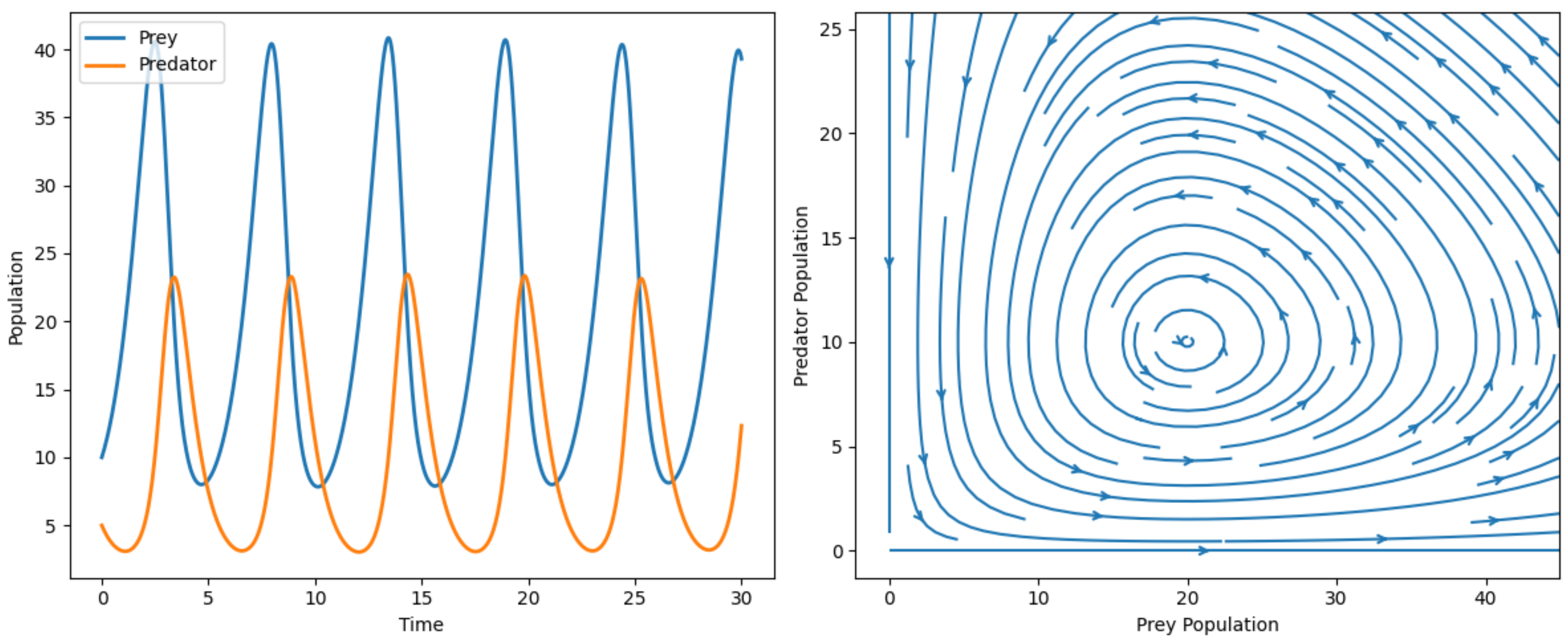
June 2025

Dynamical systems

- Describe how the state of a system evolves over time



$y'(t) = 50 - 0.5y(t), t \in [0,10]$



Lotka–Volterra predator-prey model

Dynamical systems

Modern Perspective

- Multi-layer Perceptron (MLP) with L layers

$$x_\ell = F_{\theta_\ell}(x_{\ell-1}), \text{ where } F_{\theta_\ell}(x) = \sigma(W_\ell x + b_\ell),\ \ell = 1,\ldots,L$$

- Residual neural networks (ResNets)

$$x_\ell = x_{\ell-1} + F_{\theta_\ell}(x_{\ell-1})$$

Today's roadmap

- Neural ODEs (NODEs)
  - Supervised learning examples via NODEs
  - Fundamentals in probability
  - Generative modeling example - Continuous Normalizing Flow
- Neural differential equations in latent space
  - KL divergence, ELBO
  - Latent ODEs
- Diffusion models
- References

Afternoon’s hands-on session

- Experiment examples
- How to read a paper
- Pick a paper you are interested in
- Ideas in action!

  Team up to discuss and brainstorm

Neural Ordinary Differential Equations

- Consider an initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = f_\theta(t, y(t)), \text{ for } t \in [0,T], y(0) = y_0,$$

where $f_\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ is parametrized by neural networks.

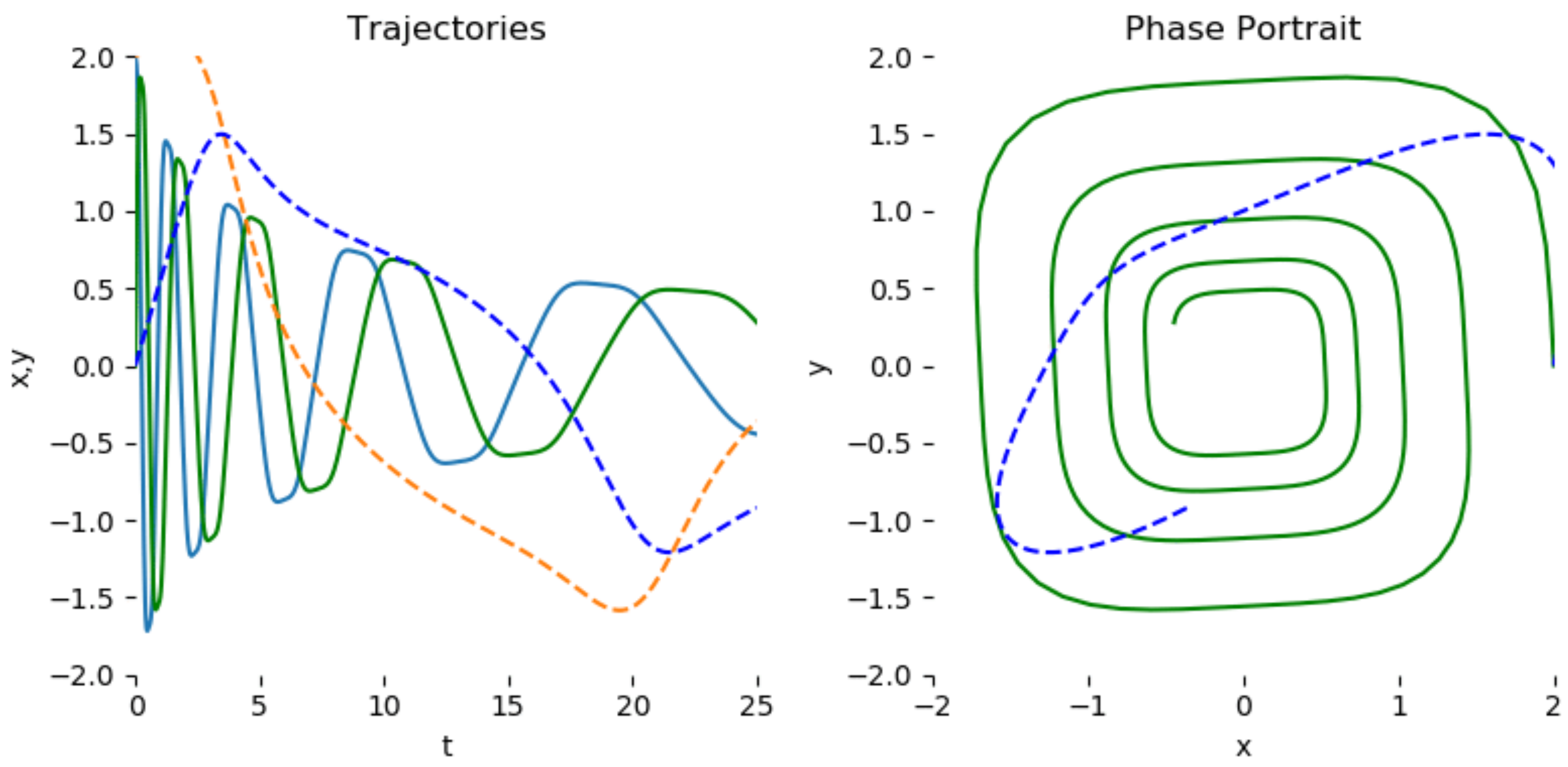
- Existence and Uniqueness

By Picard's existence theorem, if f_θ is continuous in time and uniformly Lipschitz in y, then there exists a unique function $t \to y(t)$ that satisfies the above IVP over $[0,T]$.

Neural Ordinary Differential Equations

A supervised learning example

- Consider $\frac{\mathrm{d}}{\mathrm{d}t} y(t) = f_\theta(t, y(t))$, for $t \in [0, T]$, $y(0) = y_0$,
  - $f_\theta(y) = W_2 \tanh(W_1 (y^3) + b_1) + b_2$
  - Train θ by minimizing $\frac{1}{N} \sum_{i=1}^{N} \|y_{\text{true}}(t_i) - y_\theta(t_i)\|_1$



Source link

Discussion

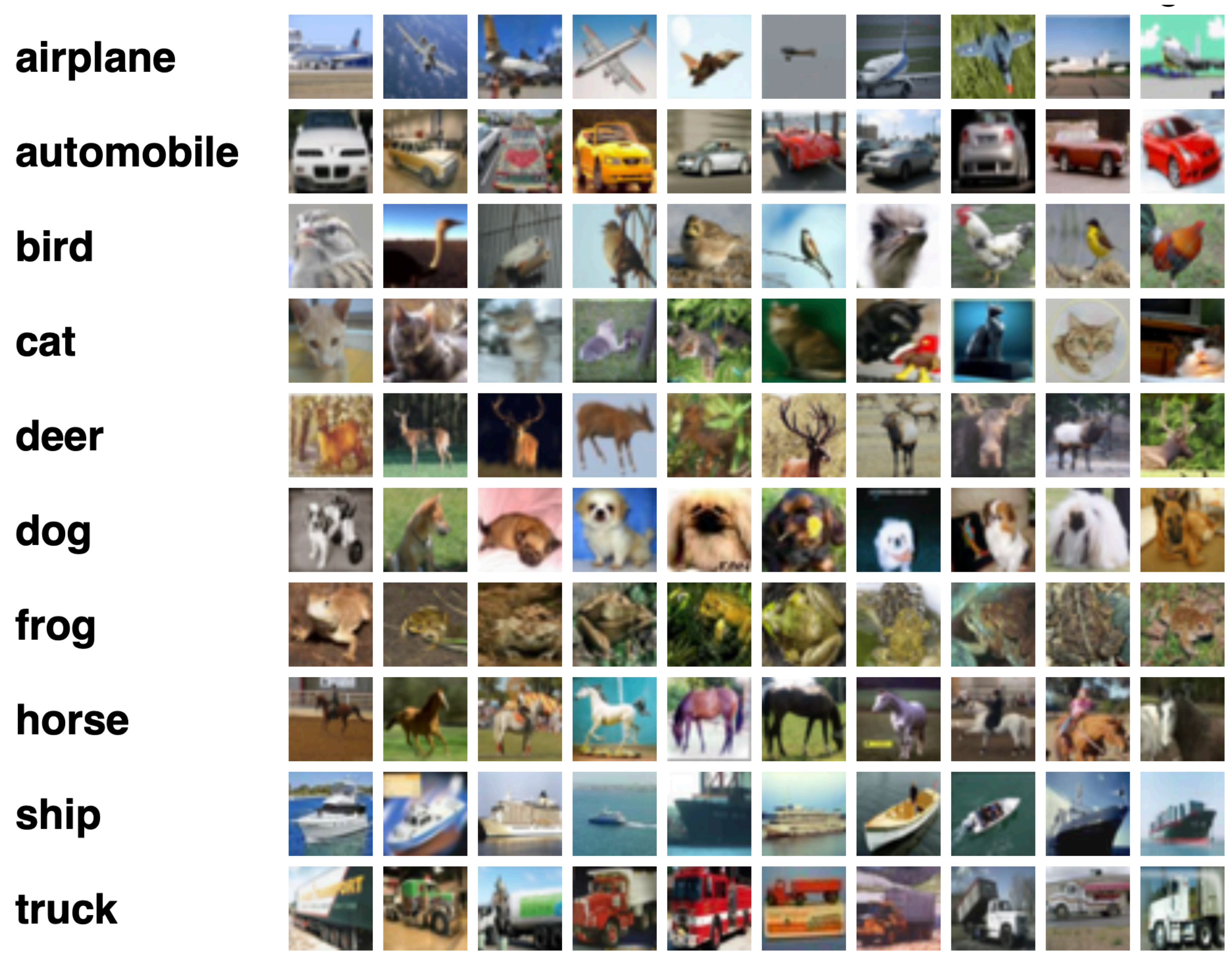
Any questions?

- A continuous-time deep learning framework

  Haber&Ruthotto '17, E '17, Chen et al., '18

- How do you update θ? Back propagation

- Applications



Neural Ordinary Differential Equations

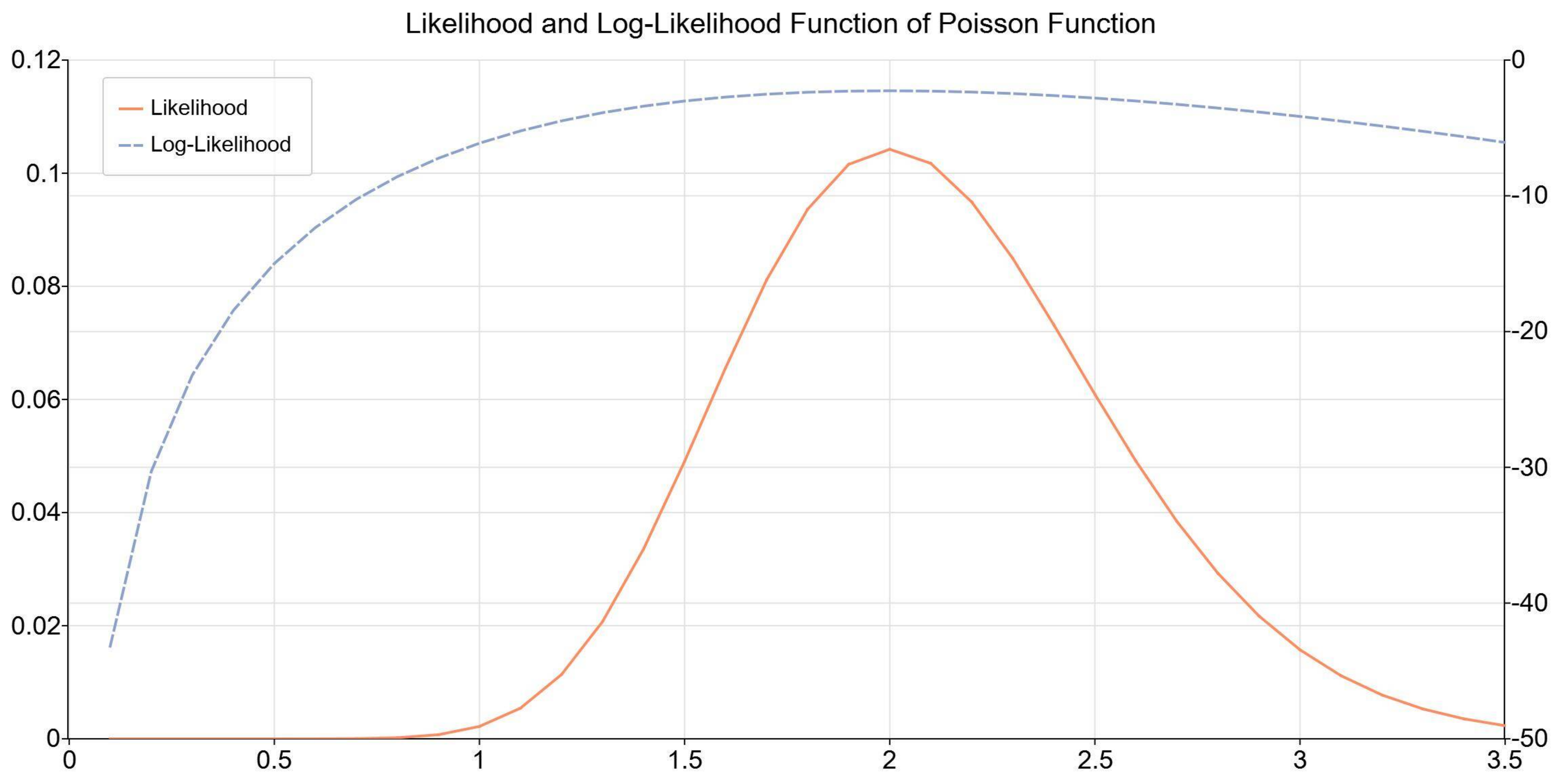
Unsupervised?

- Consider $\frac{\mathrm{d}}{\mathrm{d}t}y(t) = f_\theta(t, y(t))$, for $t \in [0,T]$, $y(0) = y_0$,
  - Randomness

    We have access to some data (samples from a distribution)

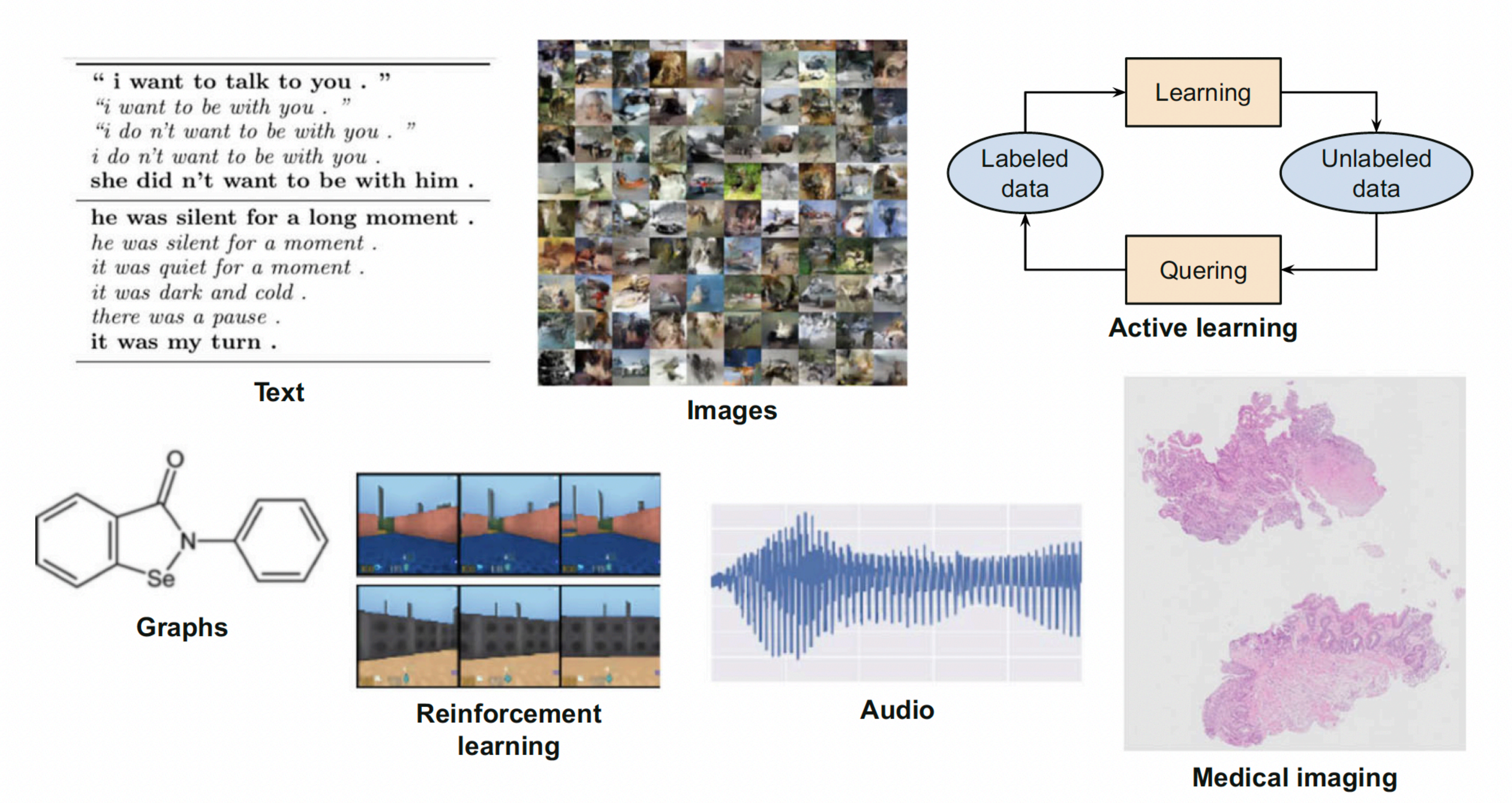
    Use $y(T)$ to approximate the data distribution

  - Evaluation



Source link

Generative Models

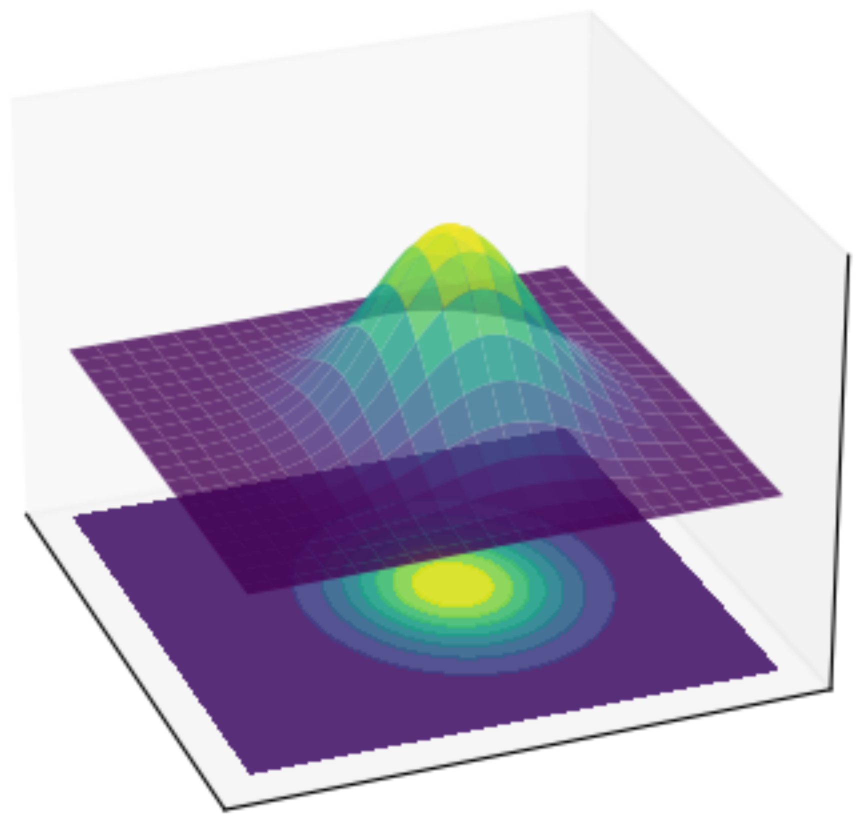


Source link

Generative Models

A mathematical formulation

- Given some observations, sample from a data distribution (unknown!)



Fundamentals

Probability space and probability distributions

Fundamentals

Pushforward

- Pushforward map φ

- Recall the change of variables formula: Let φ be injective, continuously differentiable function such that $(v_1, \ldots, v_n) = \varphi(u_1, \ldots, u_n)$. Then, $dv_1 \cdots dv_n = |\det(J_\varphi)| du_1 \cdots du_n$.

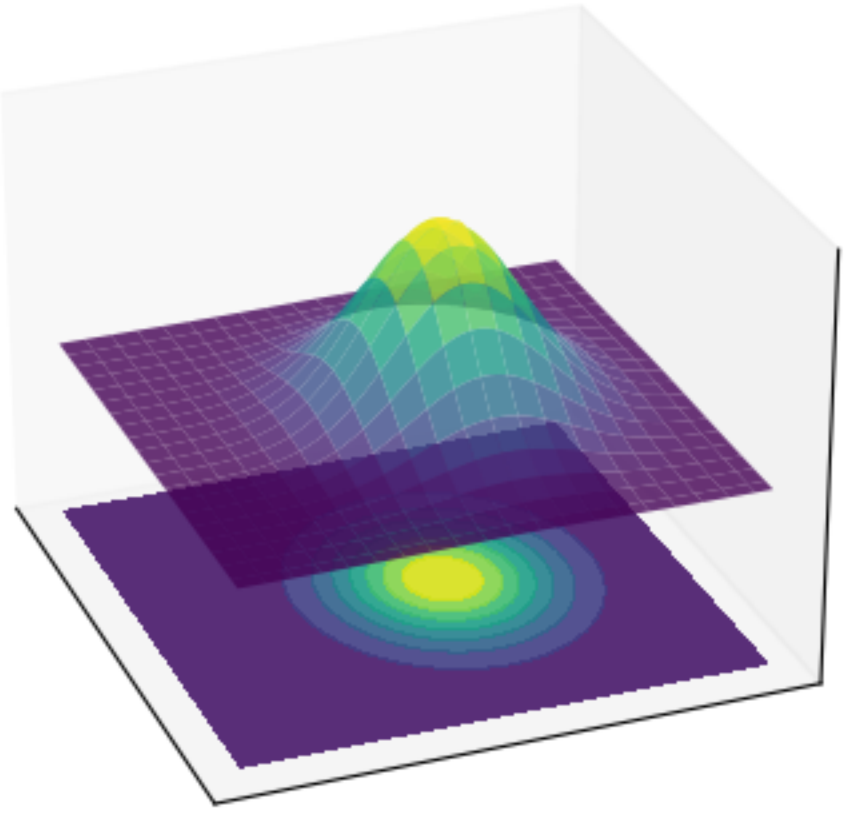
Fundamentals

Continuity equation

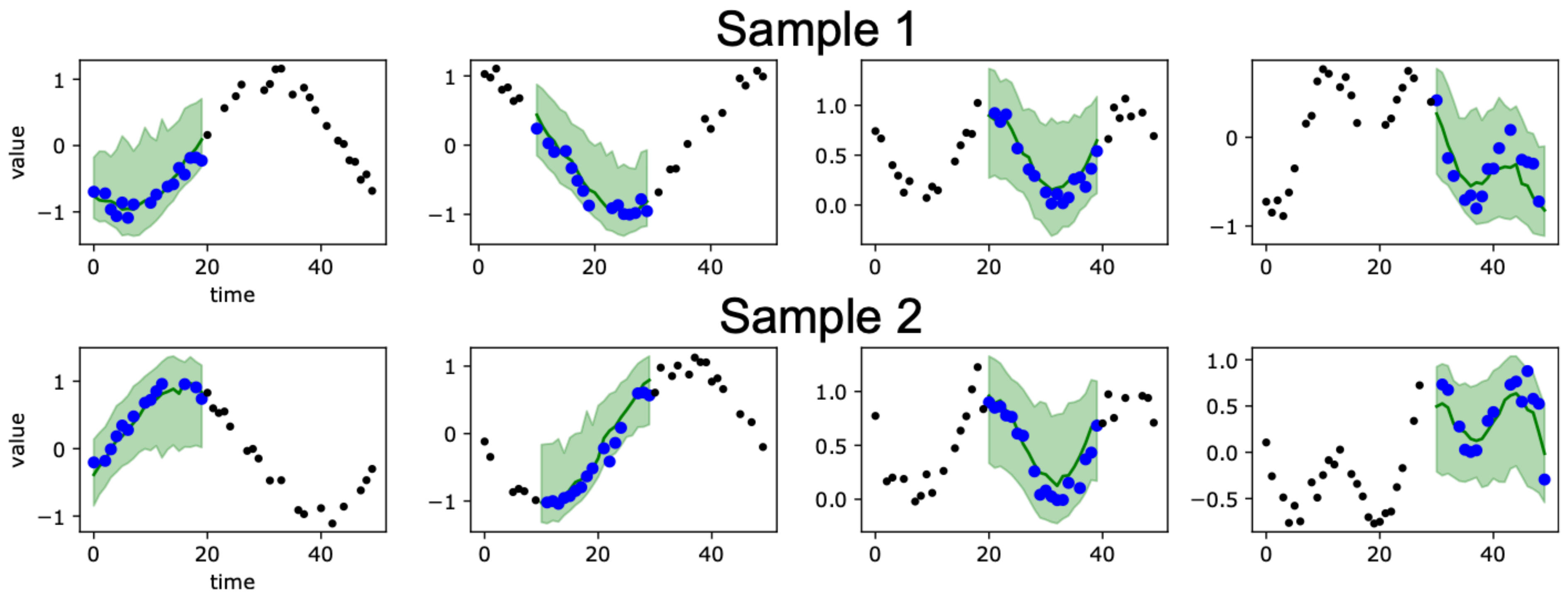
Generative Models

A mathematical formulation

- Given some observations, sample from a data distribution (unknown!)



Source link



Source link

Continuous Normalizing Flows

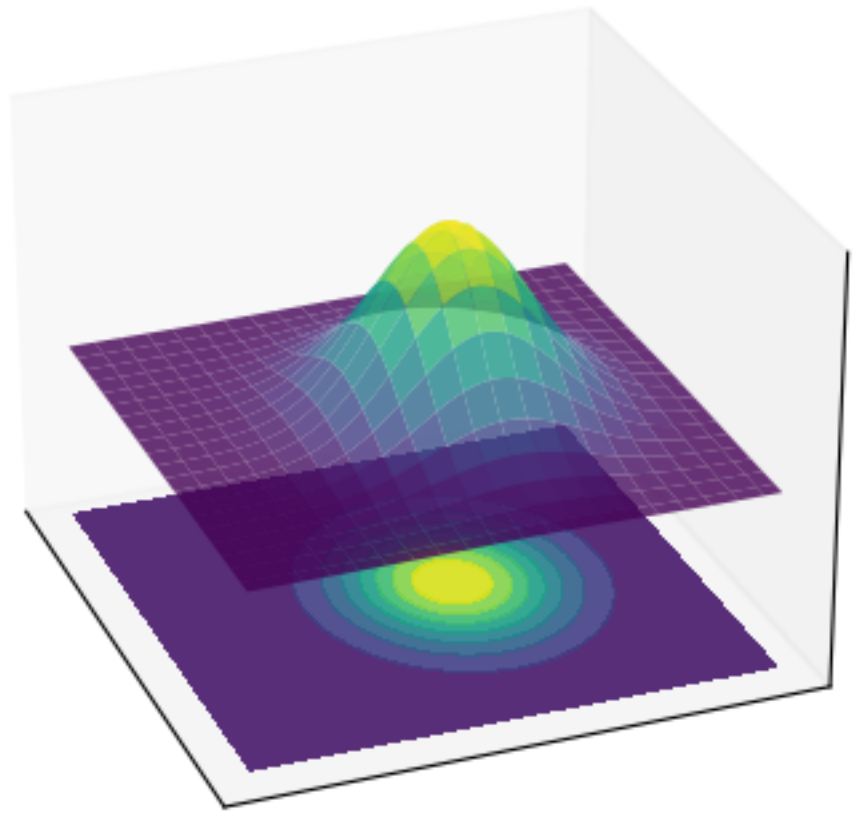
A generative model via Neural Ordinary Differential Equations

- Flow map

  Given a initial condition y_0 and a time t, flow map $\Phi_t(y_0) = y(t)$

- Pushforward

  $\mu_t = \Phi_{t\#}\mu_0$



Continuous Normalizing Flows

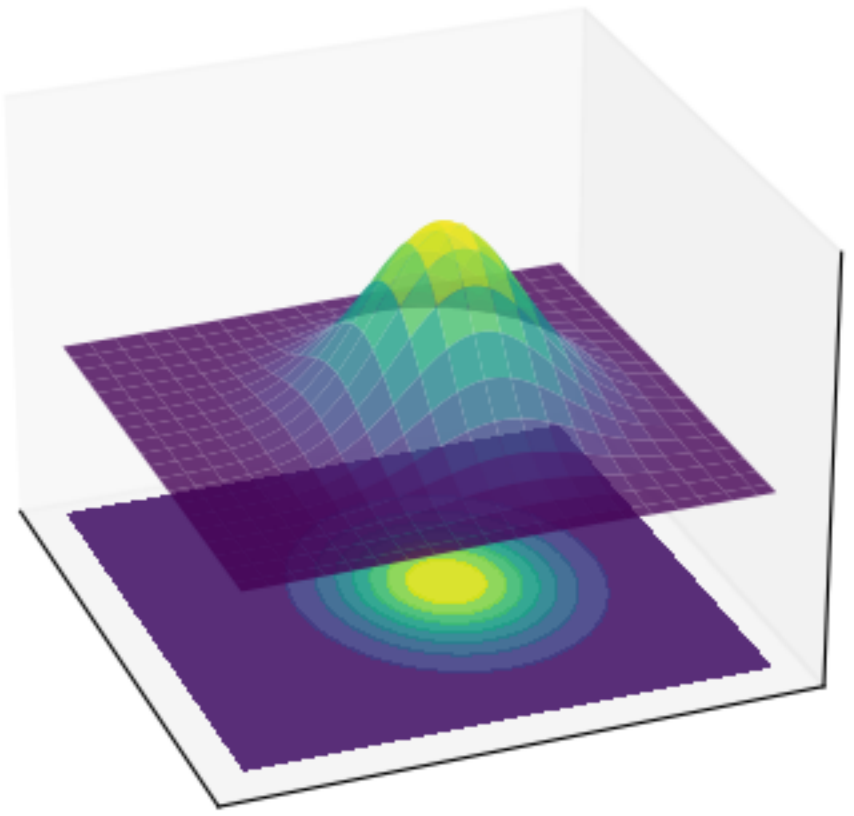
A generative model via Neural Ordinary Differential Equations

- Flow map

  Given a initial condition y_0 and a time t, flow map $\Phi_t(y_0) = y(t)$

- Pushforward

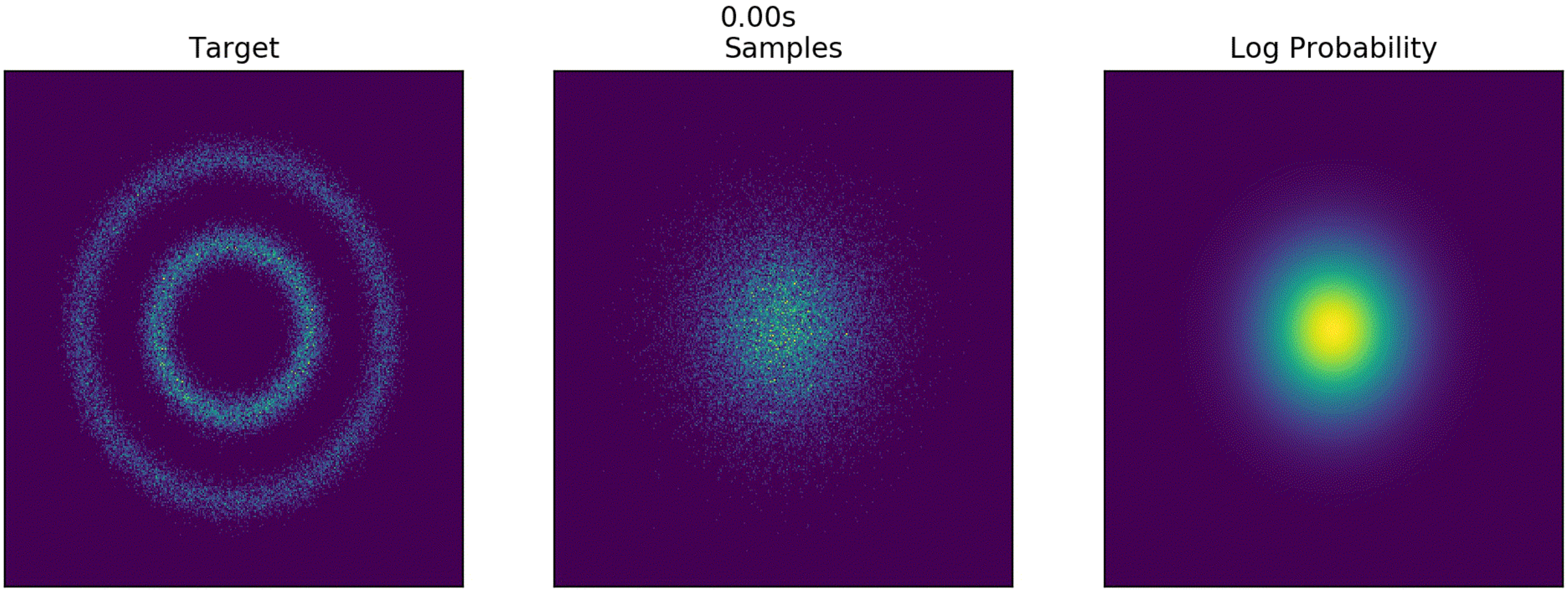
  $\mu_t = \Phi_{t\#}\mu_0$



$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} z_\theta(t) \\ \ell_\theta(t) \end{bmatrix} = \begin{bmatrix} f_\theta(z_\theta(t), t) \\ -\mathrm{tr}\left(\frac{\partial f_\theta}{\partial z_\theta}\right) \end{bmatrix}, \quad \begin{bmatrix} z_\theta(T) \\ \ell_\theta(T) \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}.$$

Continuous Normalizing Flows

A generative model via Neural Ordinary Differential Equations



Continuous Normalizing Flows

Discussion

- Exact likelihood evaluation
- Density Estimation
- Generation - text, audio, image, video etc.

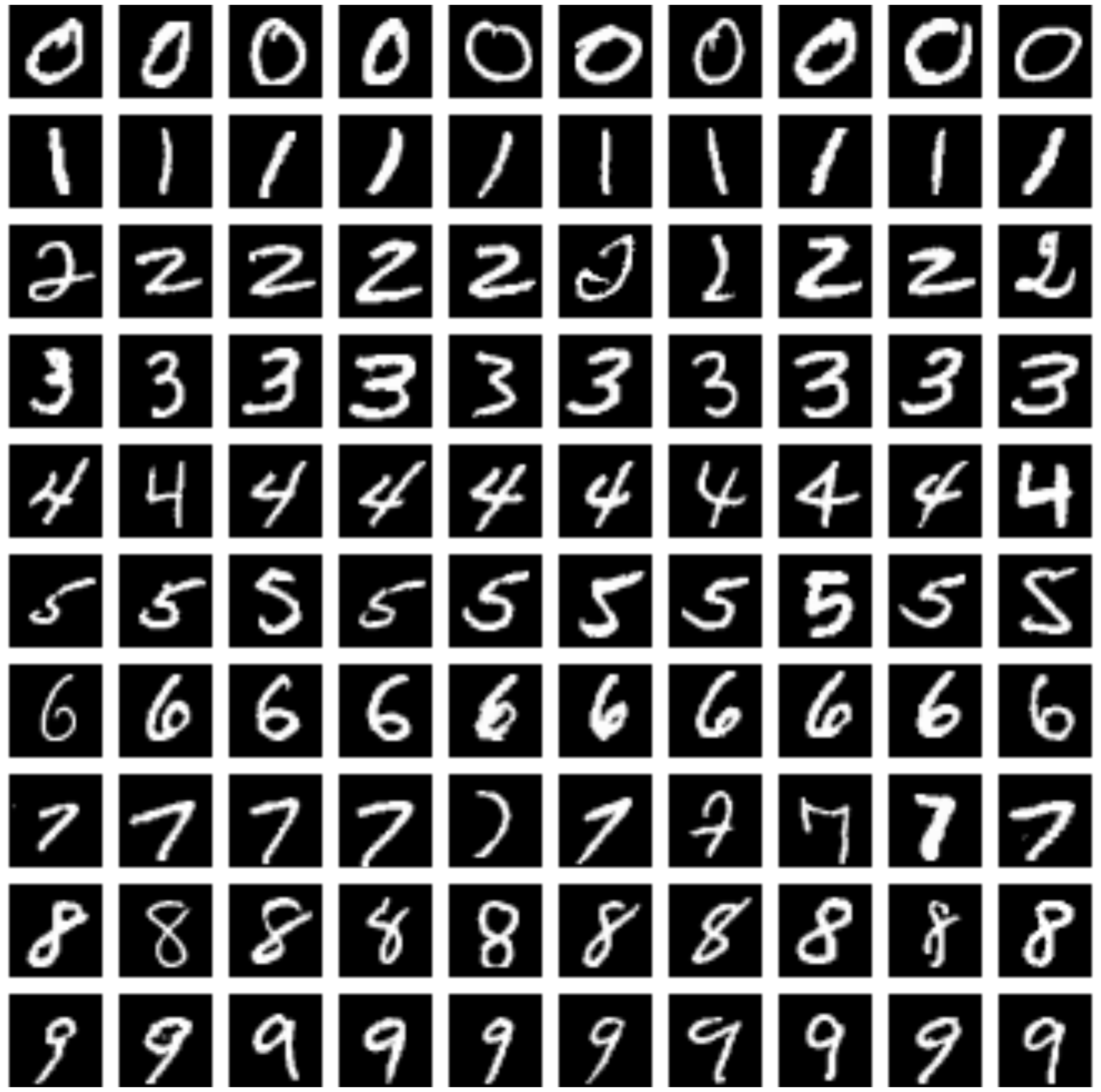
Today's roadmap

- Neural ODEs (NODEs)
  - Supervised learning examples via NODEs
  - Fundamentals in probability
  - Generative modeling example - Continuous Normalizing Flow
- Neural differential equations in latent space
  - KL divergence, ELBO
  - Latent ODEs
- Diffusion models
- References

Continuous Normalizing Flows

Discussion

- Exact likelihood evaluation
- Density Estimation
- Generation - text, audio, image, video etc.



- Limitations in CNF
  - Bijection:

    need the input output dimension to be the same
  - Expressivity:

    data space dimension and latent space dimension need to be the same

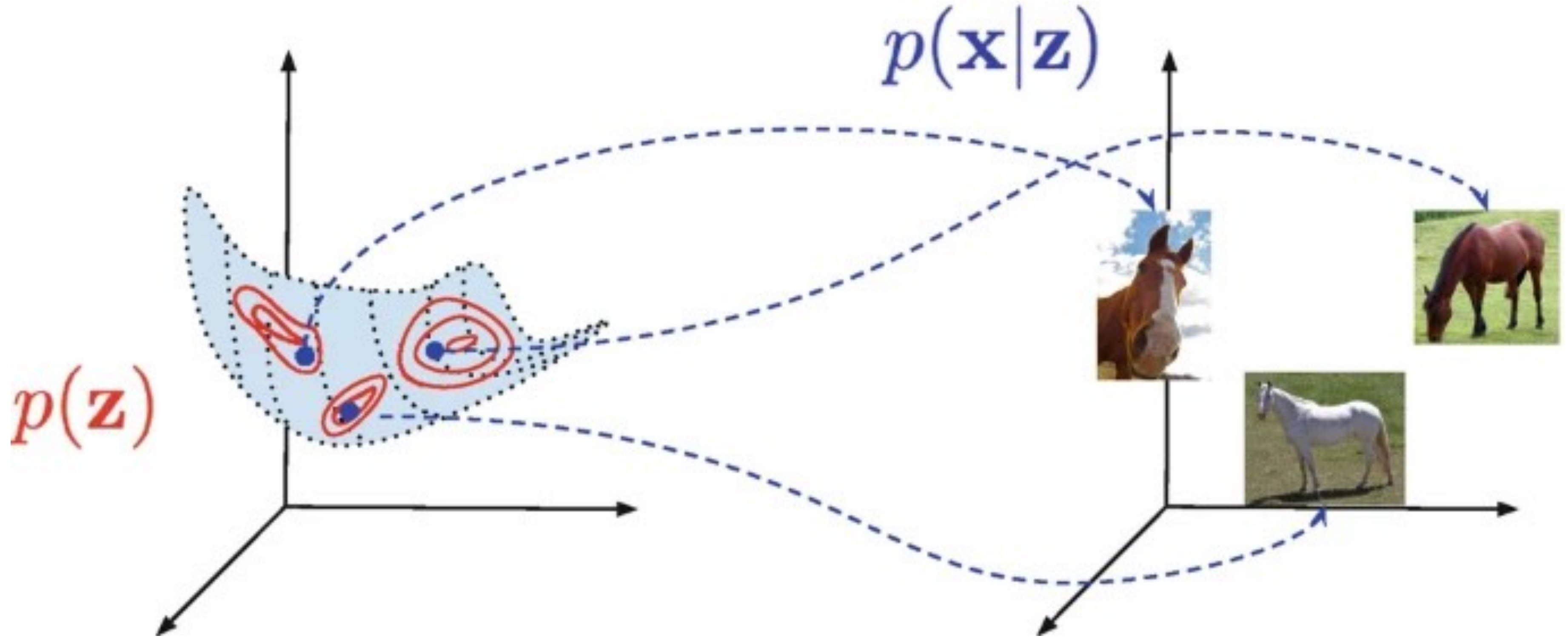
Latent Variables

- Principle Component

  Data x (high dimensional)

  Latent variables z (lower dimensional,

  hidden factors in data)



Source link

- Evaluation of likelihood
  - Generative process: first sample z, we learn the relation as $p(x\,|\,z)$
  - Note that, we only have access to x during training. Hence, the likelihood is computed as

$$p(x) = \int_{\mathcal{Z}} p(x\,|\,z)p(z)dz$$

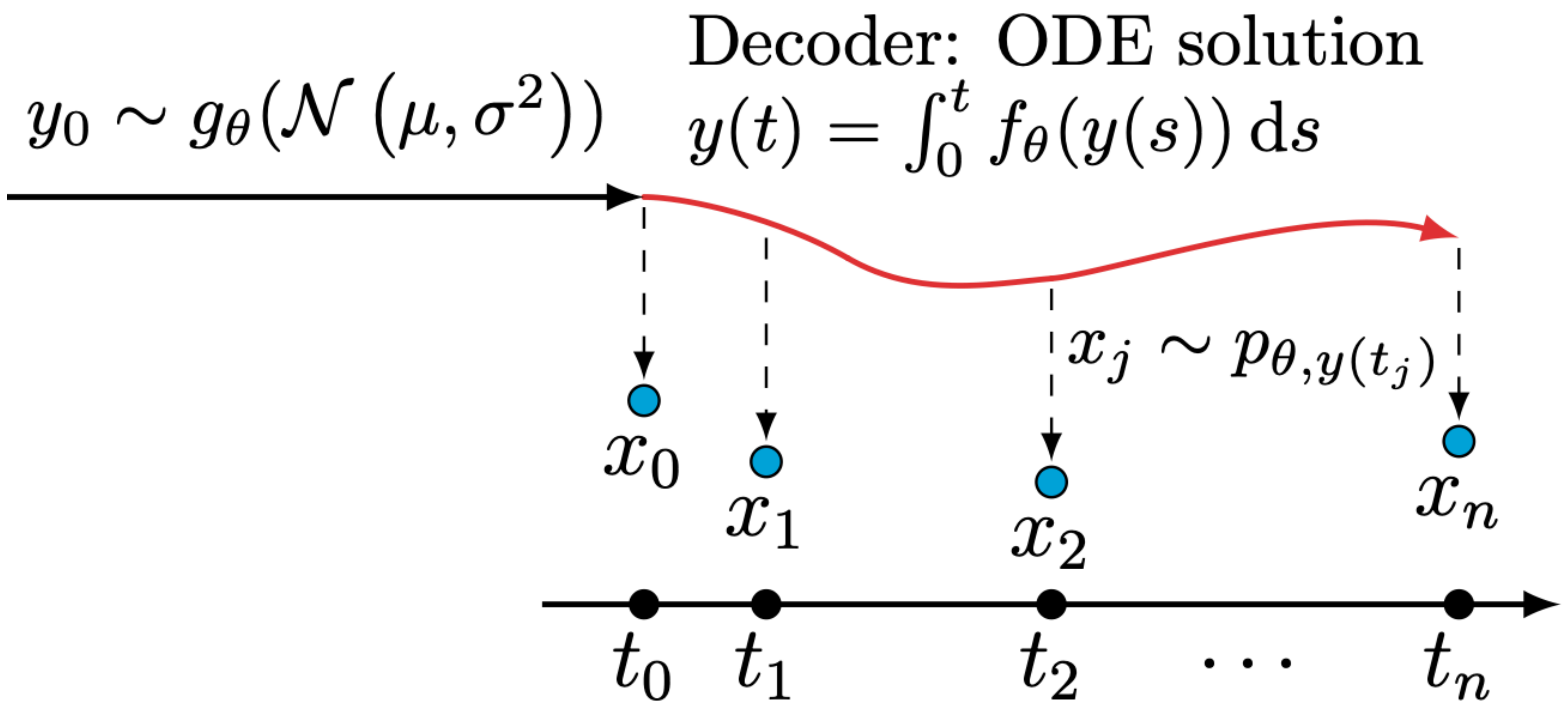
Evidence Lower Bound

- Jensen's inequality : For any convex function f, it holds that

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f\left(x_i\right), \text{ where } \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1.$$

- Let's compute $\log p_\theta(x)$!

A latent structure for NODEs

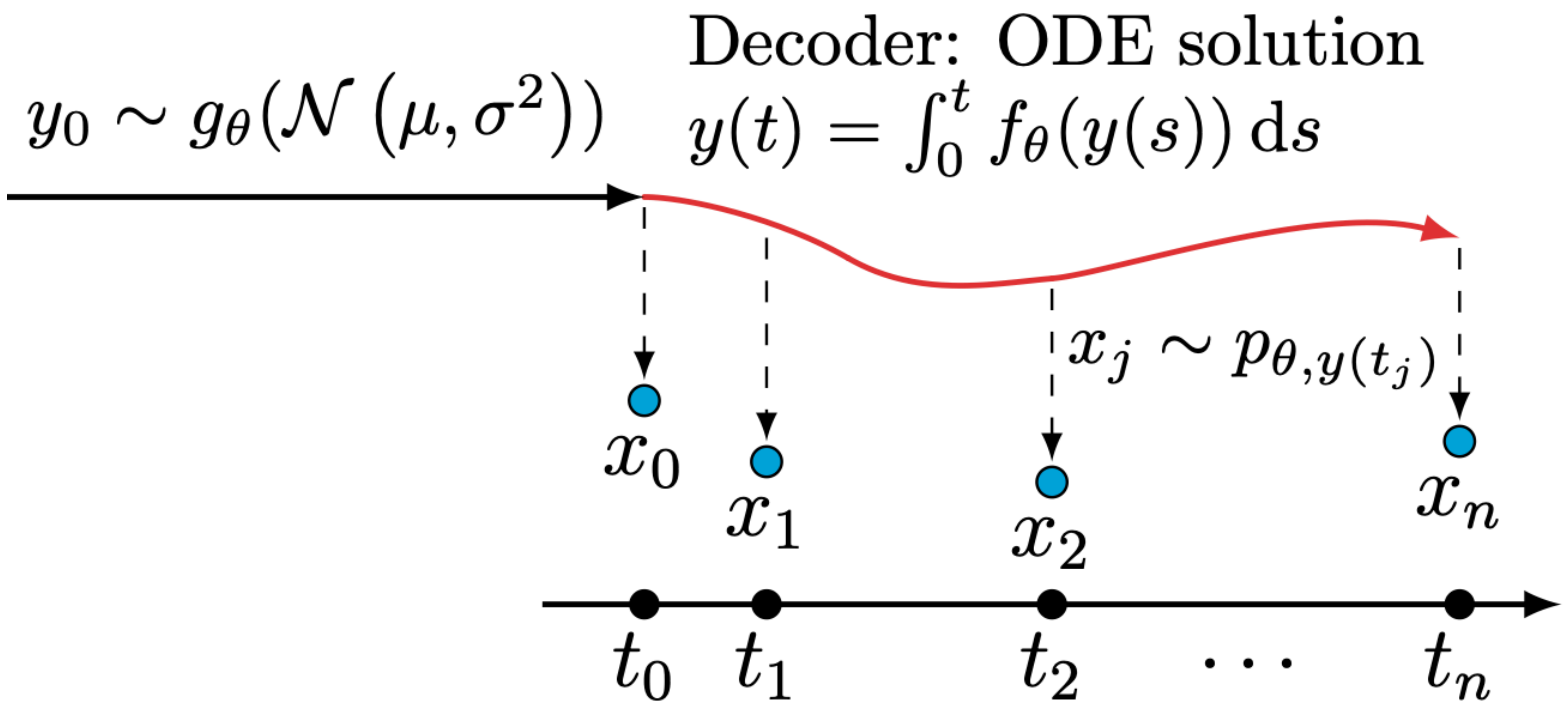


Source link

Architecture:

- Encode - Decoder
- NODEs evolve in latent space

A latent structure for NODEs



Source link

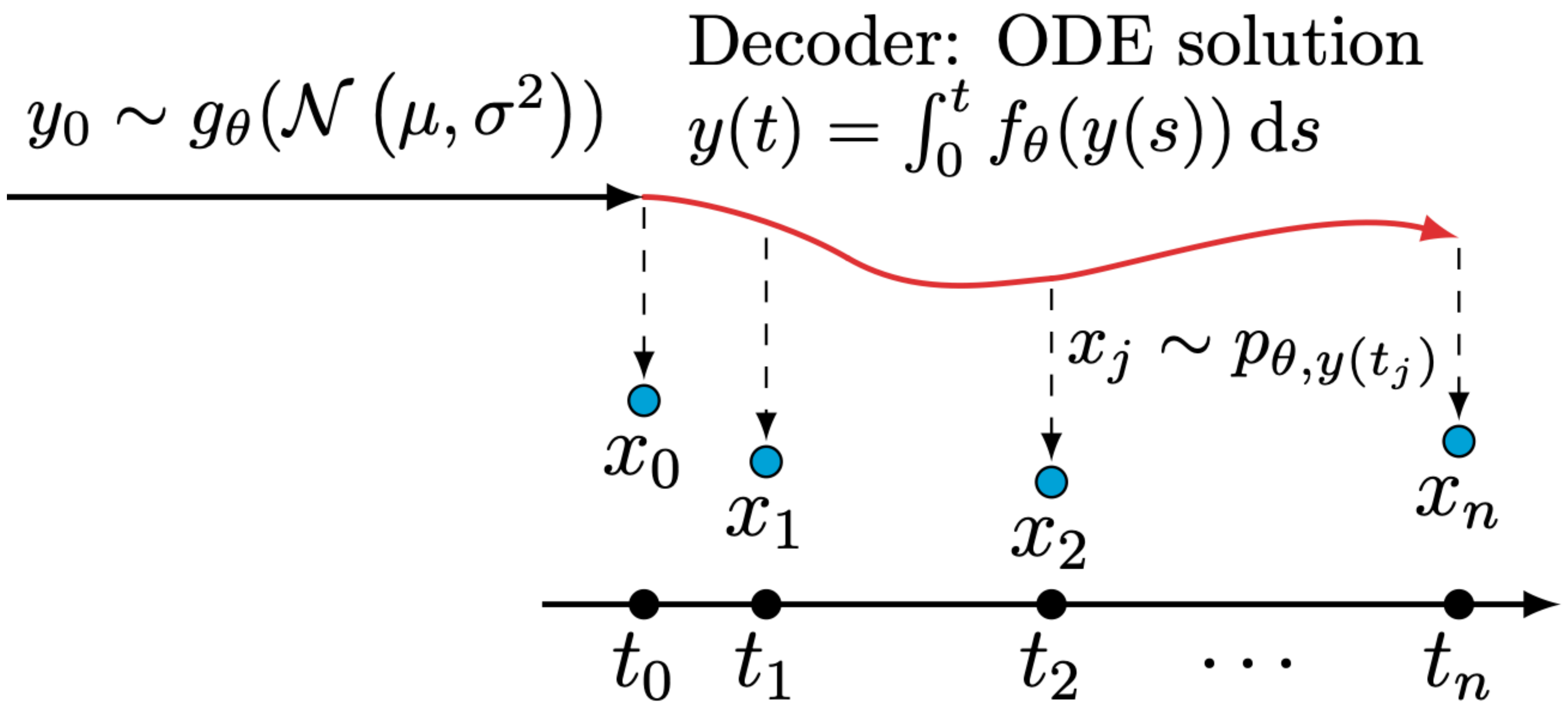
Architecture:

- Encode - Decoder
- NODEs evolve in latent space

Difference from previous methods:

- Dynamic
- Can take irregular observations

A latent structure for NODEs



- Exercise: ELBO for the sequential likelihood

A latent structure for NODEs

Discussion

Today's roadmap

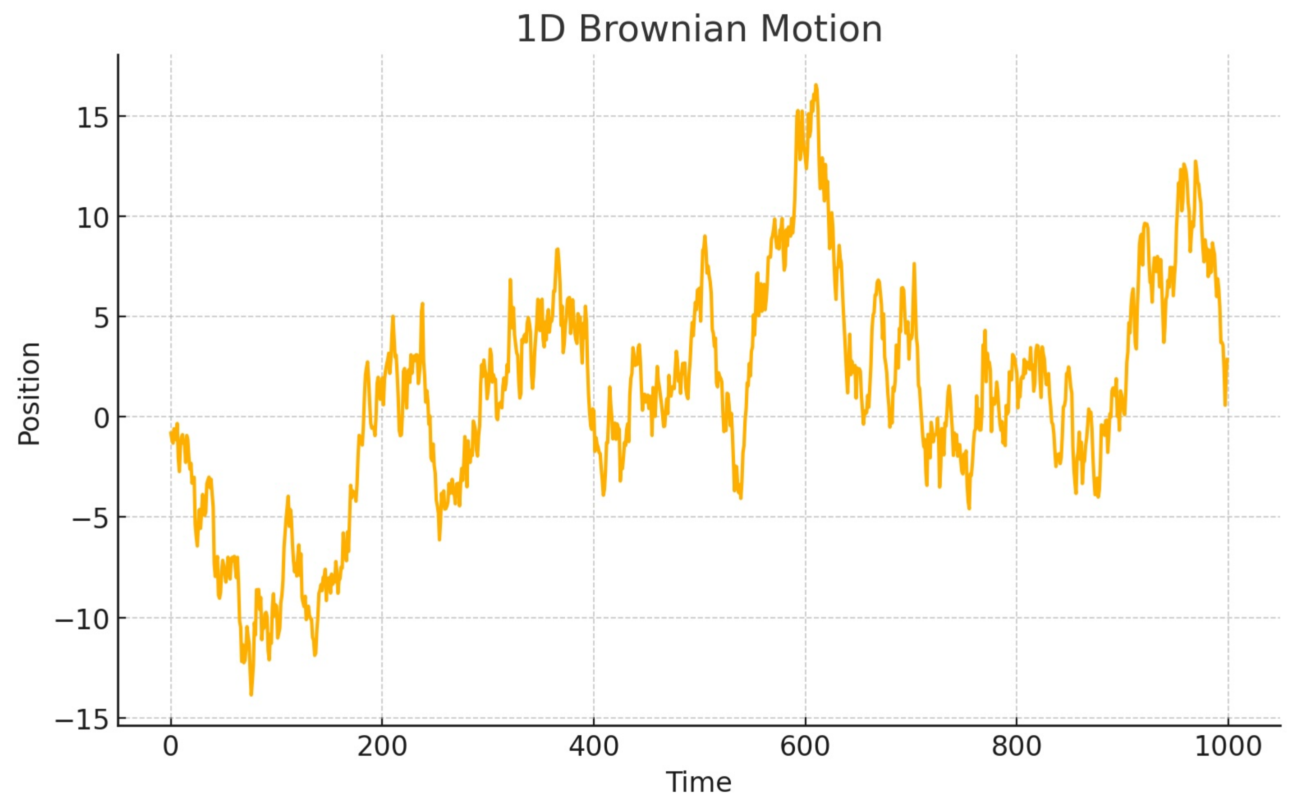
- Neural ODEs (NODEs)
  - Supervised learning examples via NODEs
  - Fundamentals in probability
  - Generative modeling example - Continuous Normalizing Flow
- Neural differential equations in latent space
  - KL divergence, ELBO
  - Latent ODEs
- Diffusion models
- References

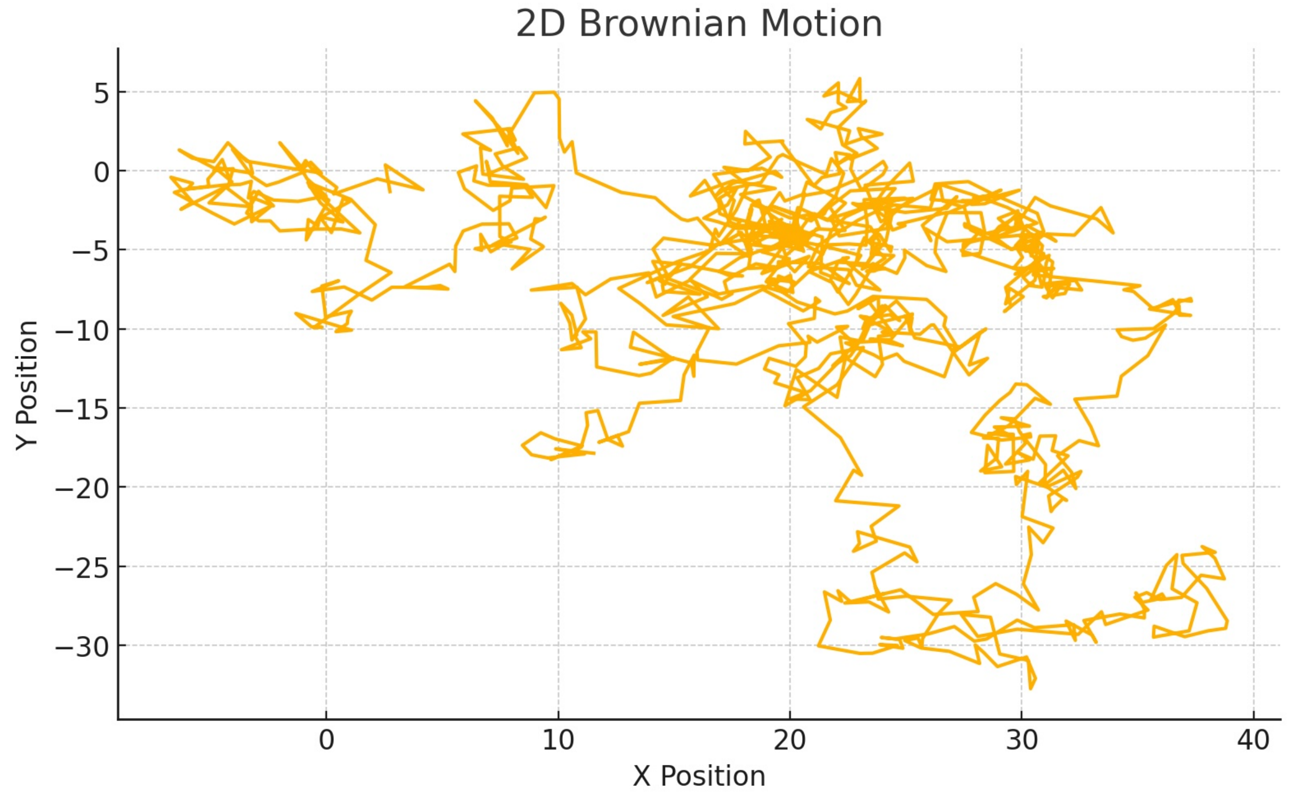
Stochastic Differential Equations

Random trajectories

$$\mathrm{d}X_t = b(t, X_t)\mathrm{d}t + \sigma(t, X_t)\mathrm{d}W_t\,,$$

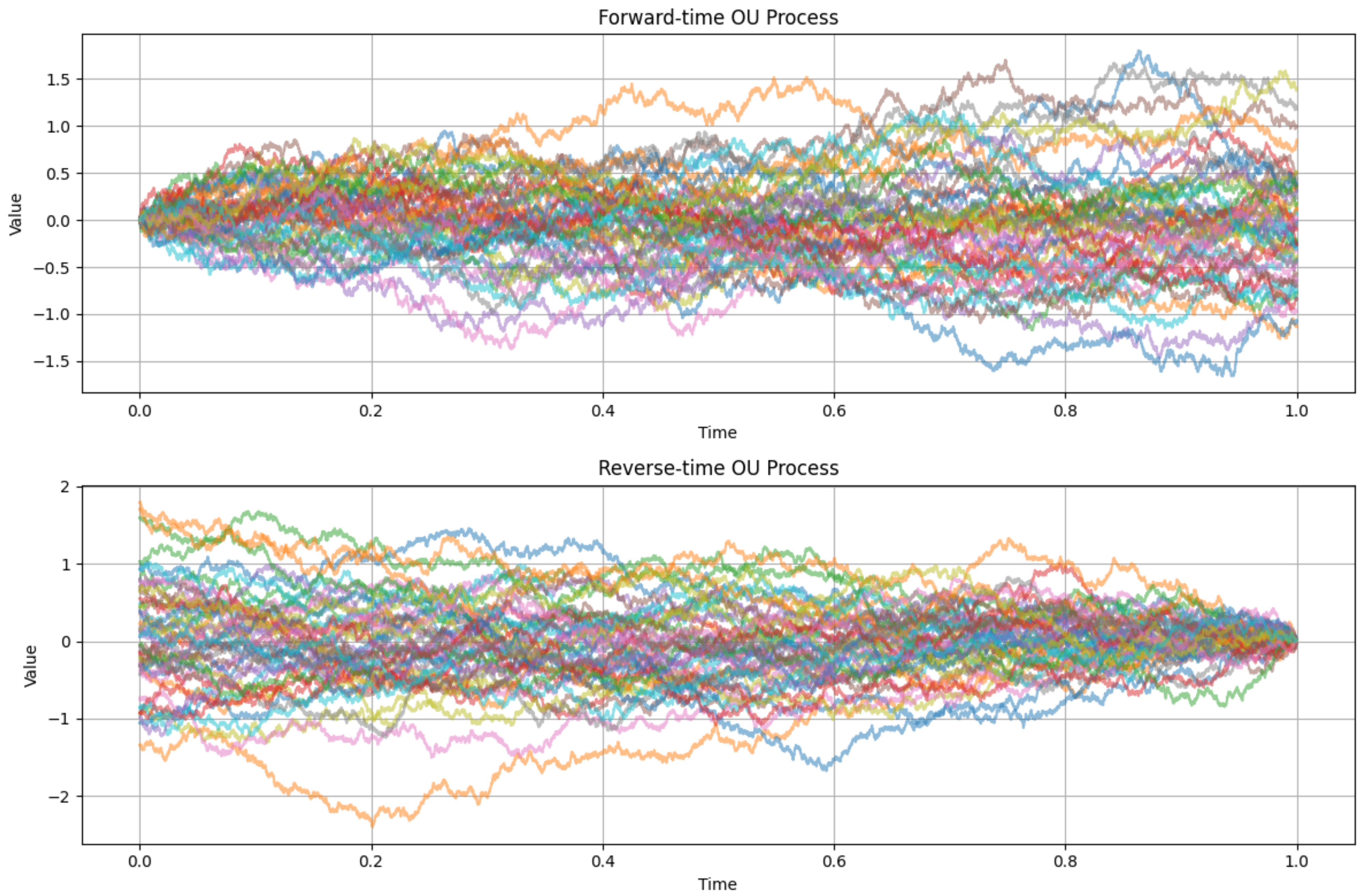
$$W_{t+\Delta t} - W_t \sim N(0,\Delta t).$$

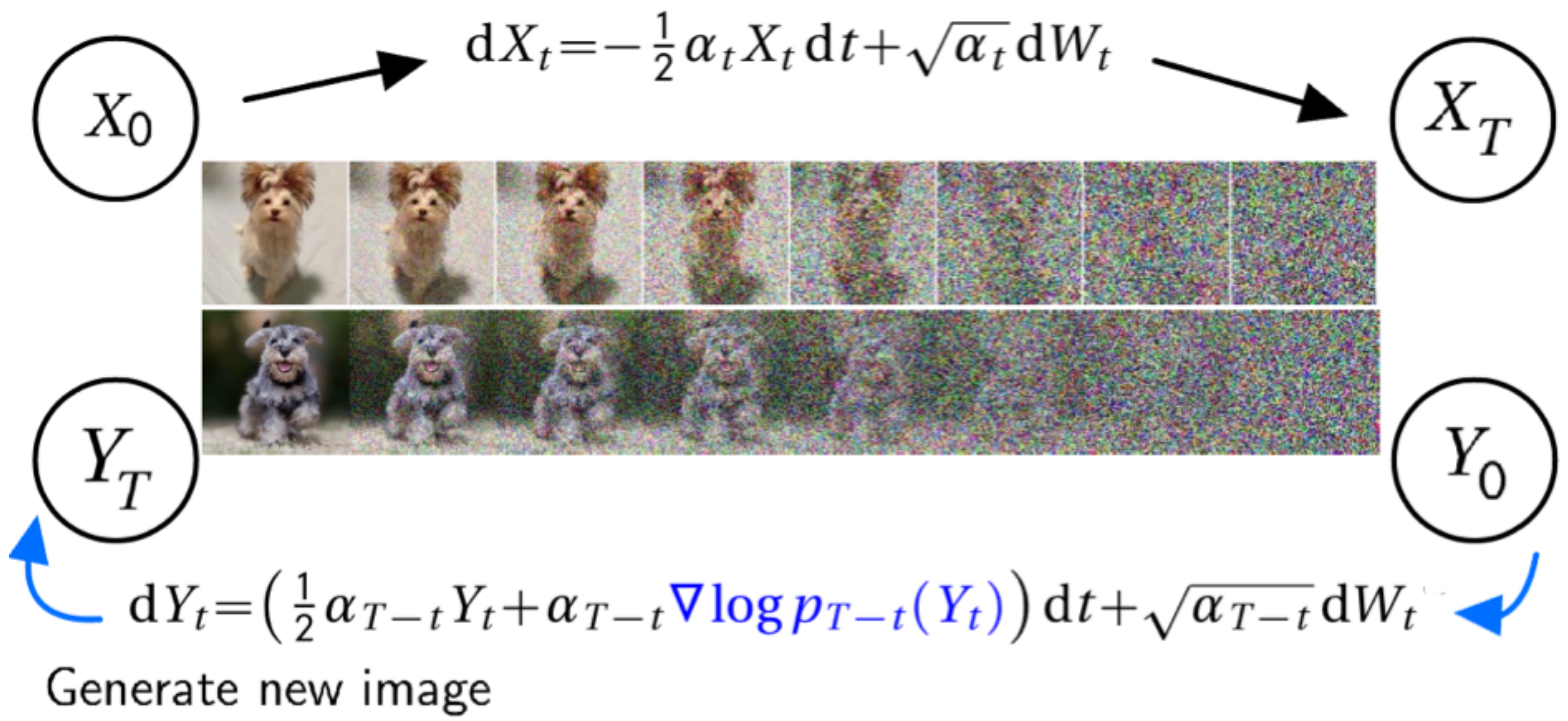




Diffusion Models

- $dX_t = -X_t dt + \sqrt{2} dW_t, \quad X_0 = x\,.$





- $d\widetilde{X}_t = \left(\widetilde{X}_t - 2\frac{\widetilde{X}_t - e^{-(T-t)}x}{1 - e^{-2(T-t)}} \right) dt + \sqrt{2} dW_t$

Diffusion Models

Discussion

- Score-matching loss
- Static or Dynamics?

References

Resources

- Durrett, R. (2019). *Probability: theory and examples* (Vol. 49). Cambridge university press.
- Tomczak, J. M. (2024). *Deep Generative Modeling*. Cham: Springer International Publishing.
- Haber, E., & Ruthotto, L. (2017). Stable architectures for deep neural networks. *Inverse problems, 34*(1), 014004.
- Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems, 31*.
- Rubanova, Y., Chen, R. T., & Duvenaud, D. K. (2019). Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems, 32*.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

References

For afternoon discussion

Onken, D., Fung, S. W., Li, X., & Ruthotto, L. (2021). **Ot-flow: Fast and accurate continuous normalizing flows via optimal transport**. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 10, pp. 9223-9232).

Orozco, R., Herrmann, F. J., & Chen, P. (2024). **Probabilistic Bayesian optimal experimental design using conditional normalizing flows**. arXiv preprint arXiv:2402.18337.

Hagemann, P., Mildenberger, S., Ruthotto, L., Steidl, G., & Yang, N. (2025). **Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation**. arXiv preprint arXiv:2303.04772. SIAM Journal of Mathematics in Data Science.

Park, J., Yang, N., & Chandramoorthy, N. (2024). **When are dynamical systems learned from time series data statistically accurate?**. arXiv preprint arXiv:2411.06311. NeuRIPS 2024

Li, X., Wong, T. K. L., Chen, R. T., & Duvenaud, D. (2020). **Scalable gradients for stochastic differential equations**. In International Conference on Artificial Intelligence and Statistics (pp. 3870-3882). PMLR.