Generative sampling with physics-informed kernels

arXiv：2510.26678

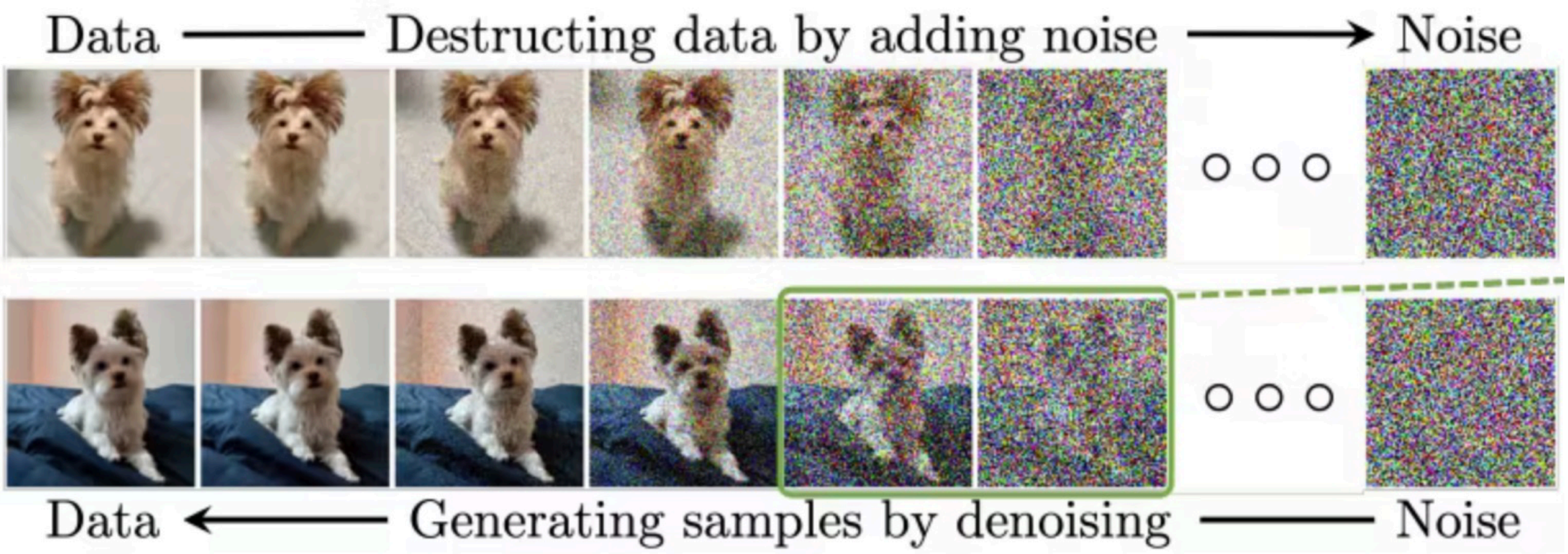
Friederike Ihssen,[1] Renzo Kapust,[1,*] and Jan M. Pawlowski[1,2]

Journal Club - 2025.11.6

Generative model for sampling

- Traditional sampling method is usually based on MCMC
- Several practical problems might occur: critical slowing-down, parallelization

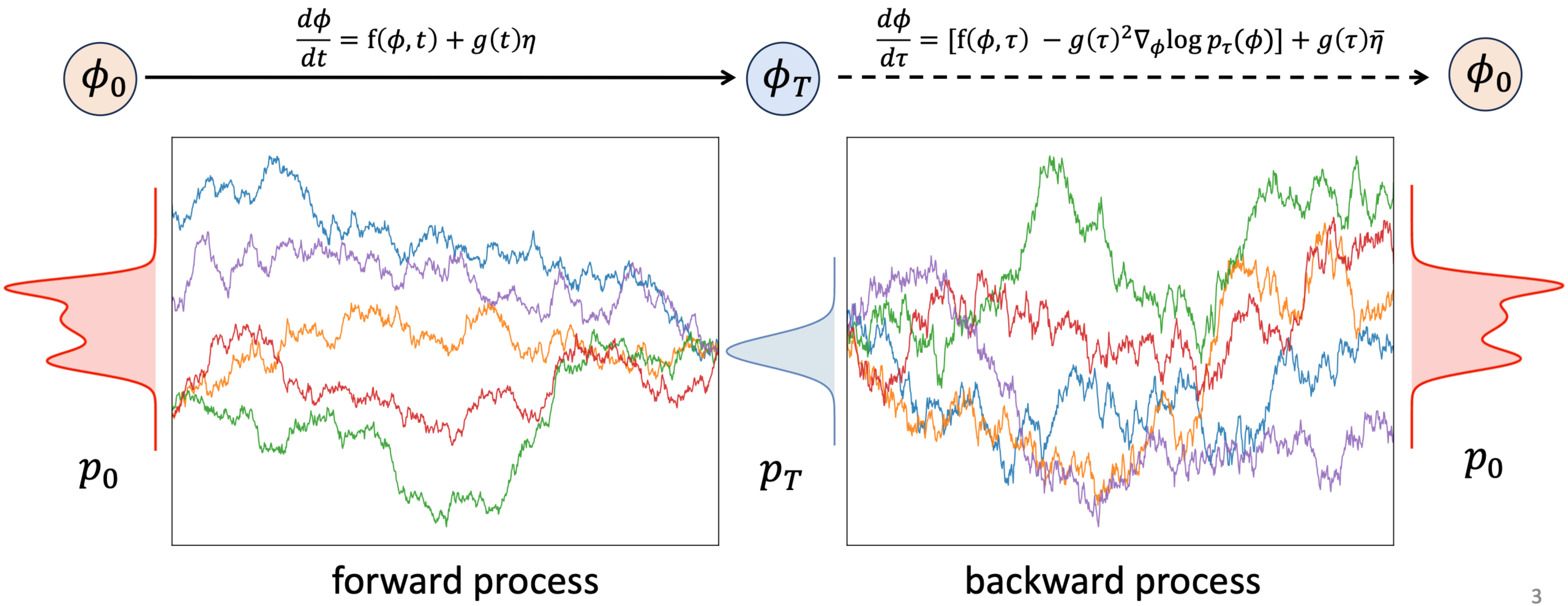
Generative model for sampling



create new images (or configurations) by learning from existing data sets (or ensembles)

Generative model for sampling

- in pictures: p_0 is target (non-trivial), p_T is the prior (easy)

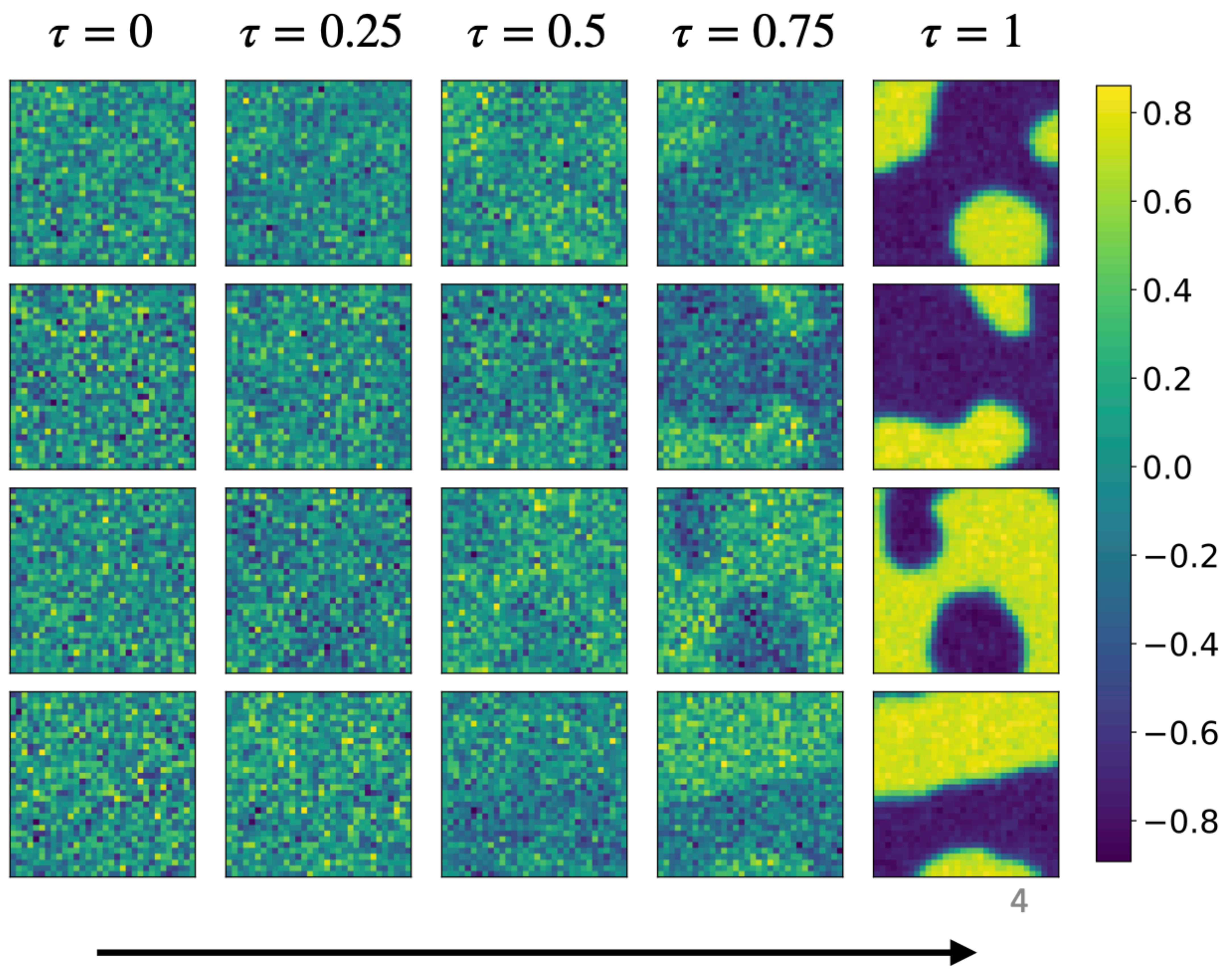


Generative model for sampling

- apply approach to lattice field theory, ambition is to improve upon standard (MC) methods
- 32^2 lattice, parameters in symmetric and broken phase
- training data set generated using Hybrid Monte Carlo (HMC)

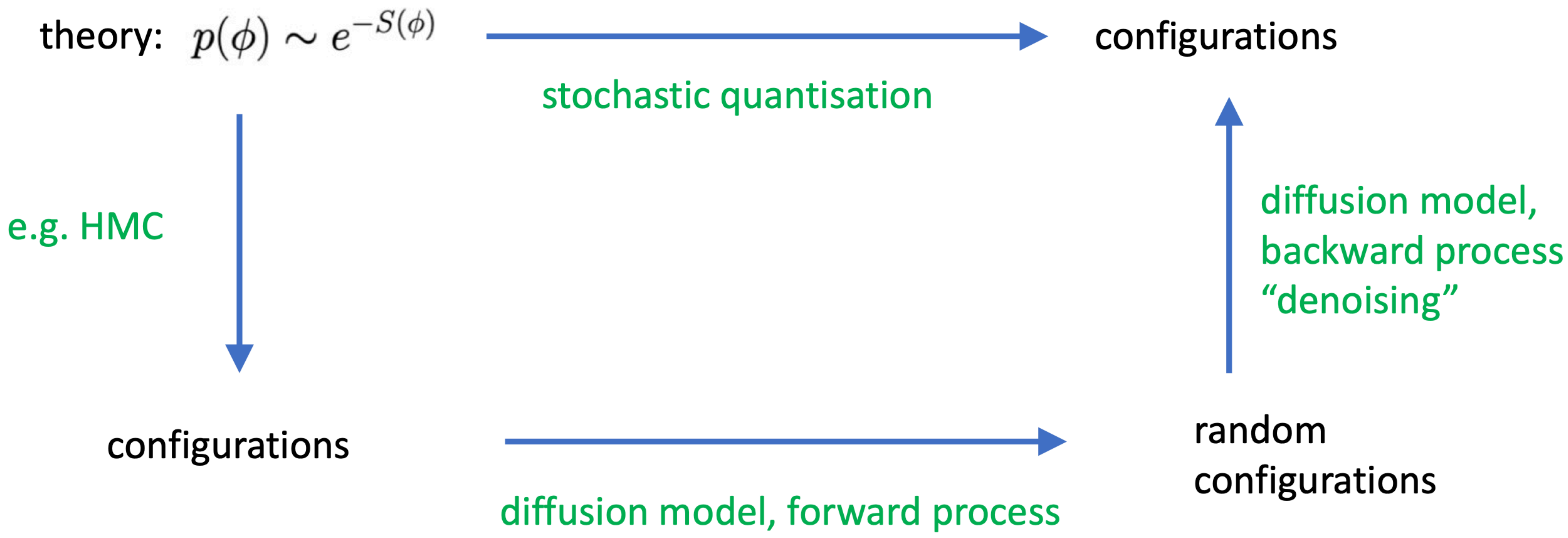
generating configurations:

- broken phase
- “denoising” (backward process)
- large-scale clusters emerge, as expected



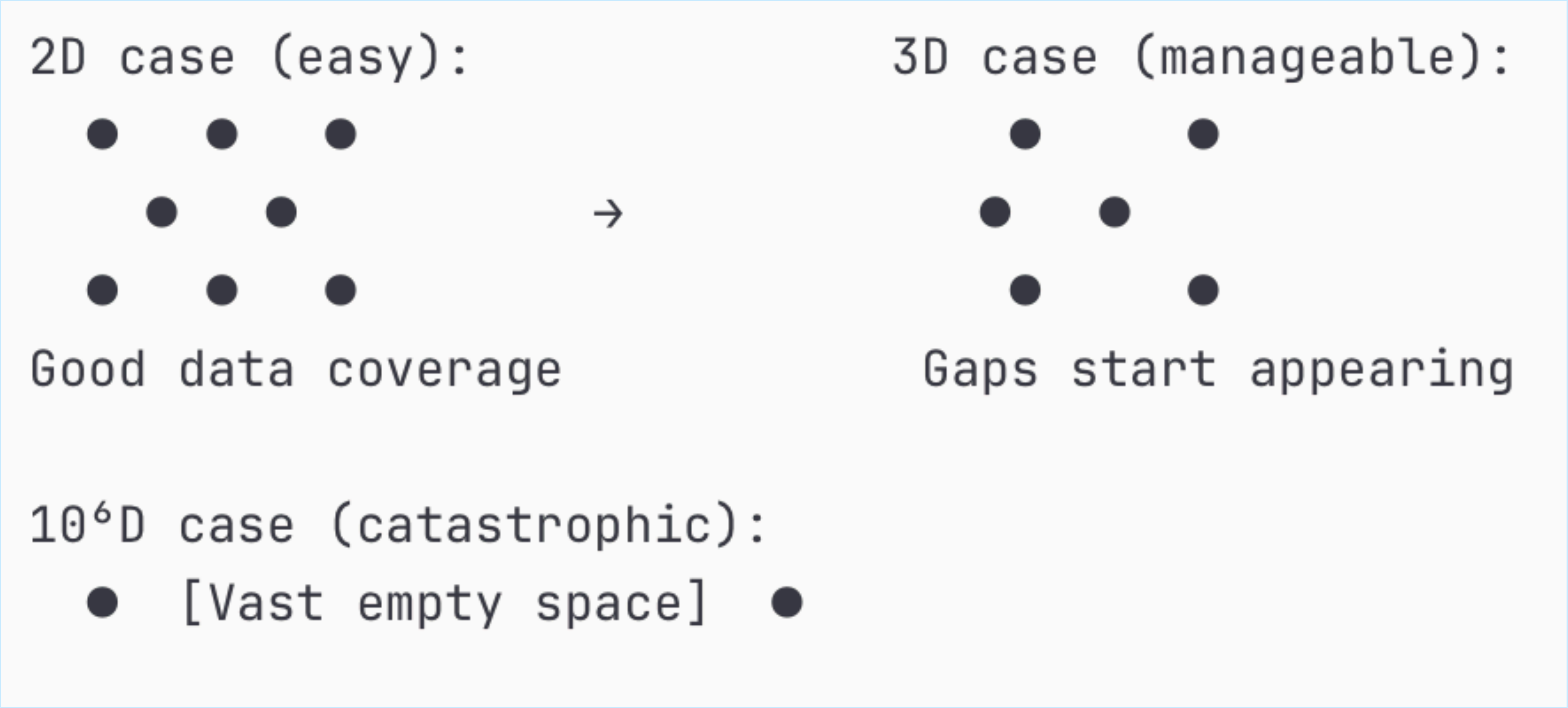
JHEP 05 (2024) 060 [2309.17082 [hep-lat]]

Generative model for sampling



Motivation

- However, this generative model approach faces two out-of-domain (OOD) problems



Interpolation in high-dimensional space

- With given number of data points, only finite number of cumulants can be accurately learned
- Extrapolation to higher order cumulant would be costing

Extrapolation with limited data

Motivation

- The above problems are absent for distributions which have a finite number of independent moments. E.g.

$$Y = e^X, X \sim \mathcal{N}(0, \sigma^2) \qquad\qquad Y \to \log Y$$

Infinite number of cumulants　　　　　　Only 1 cumulant

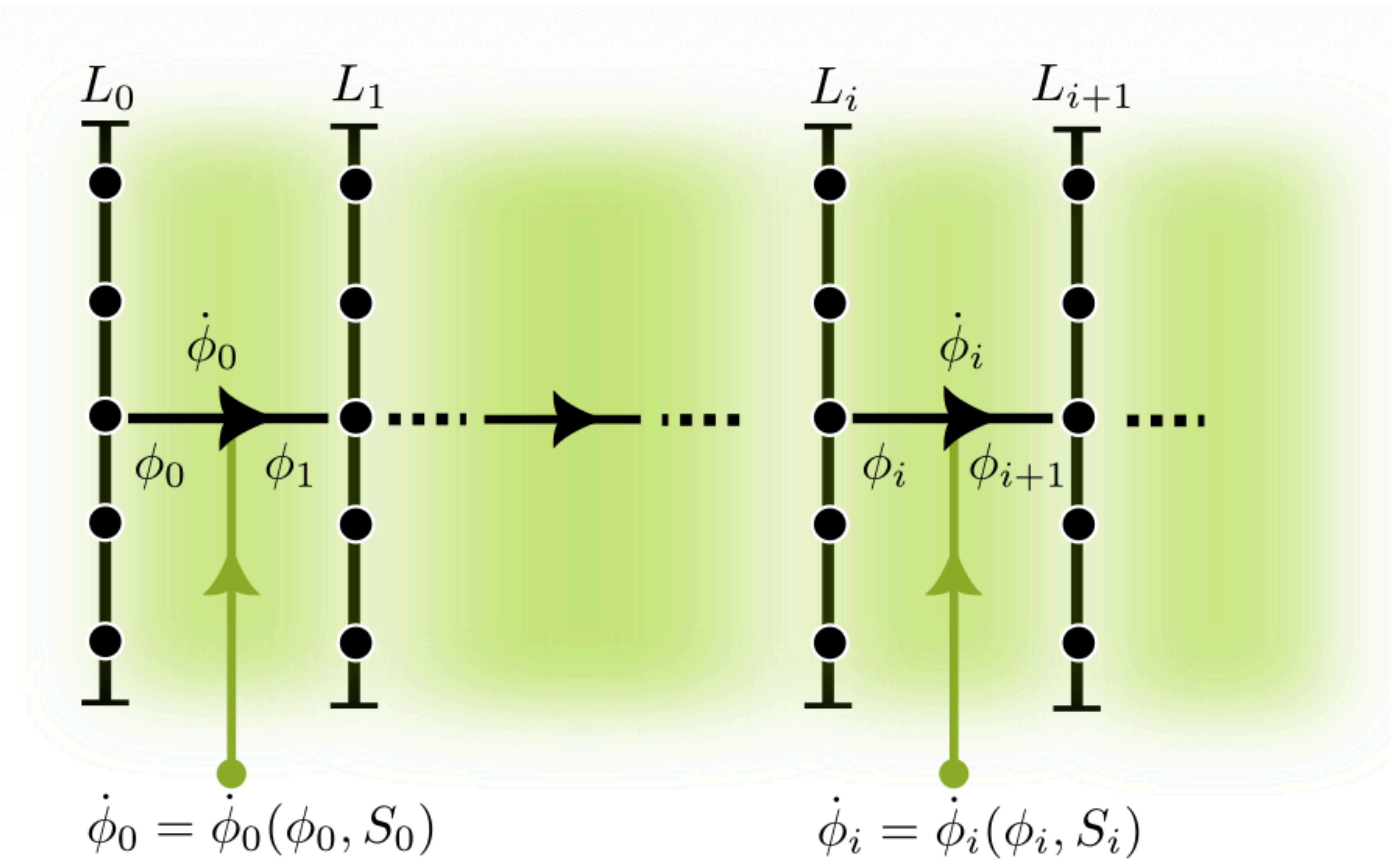
- If a generative network can constructively unravel such a typically highly non-linear transformation within a rather finite data or training set, the generative task is practically solved. — Physics-informed renormalisation group flows (PIRGs)

PIRGs in a nutshell

- The aim of the method is to sample with distribution

$$p(\varphi) = \frac{1}{\mathcal{N}} e^{-\hat{S}(\varphi)} \to e^{-S(\varphi)}$$

- PIRGs introduces RG time t
- Pair $\{S_t(\phi), \phi_t\}$ runs with t
- $t = 0$, distribution from simple model
- $t = 1$, target distribution



PIRGs in a nutshell

- We first replace φ with ϕ by a non-linear transformation. As t runs from 0 to 1, it undergoes several infinitesimal transformations.

- The transformation comes with a change of distribution $p_t(\phi)$ or its Laplace transform

$$Z(J) = \int \mathscr{D}\phi p_t(\phi) e^{\sum_{\hat{n}} \phi_{\hat{n}} J_{\hat{n}}}$$

- The most general scale and reparametrisation RG transformation is accommodated by the flow of the measure

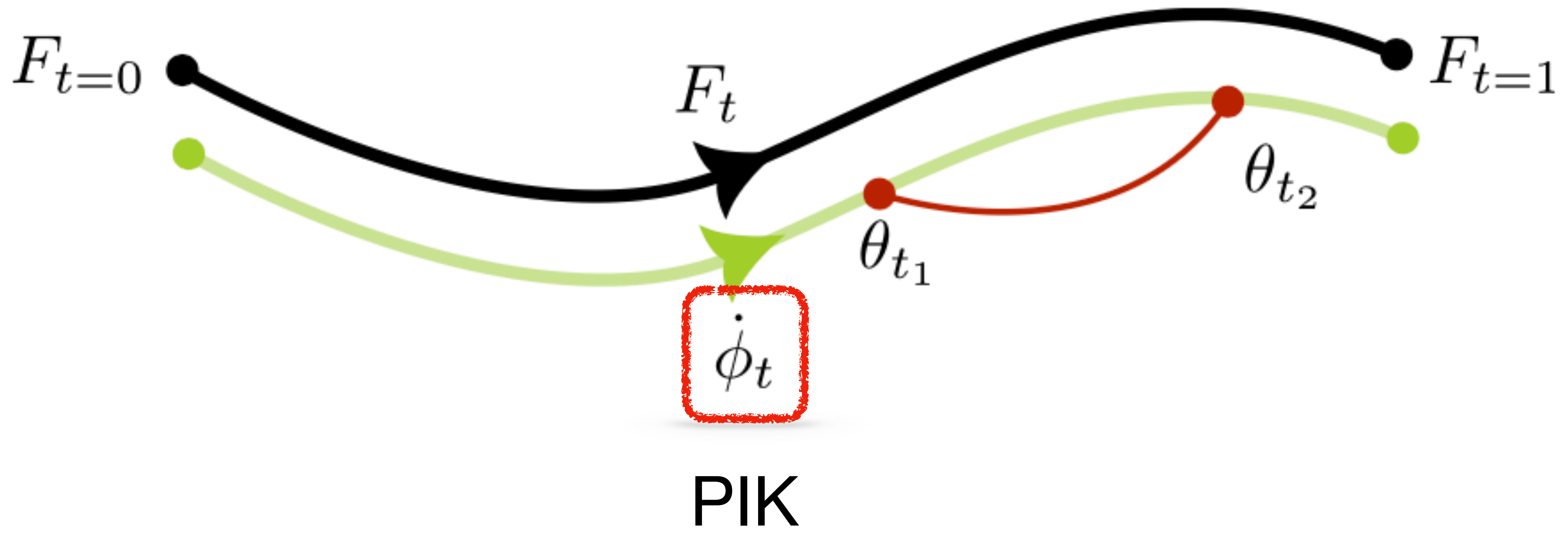
$$\frac{d\,p_t(\phi)}{dt} = \frac{\partial}{\partial \phi}\Big[\Psi_t(\phi)\,p_t(\phi)\Big]$$

PIRGs in a nutshell

- By choosing $\Psi \propto \dot{\phi}$, the total change of field and action or negative log likelihood is covered by

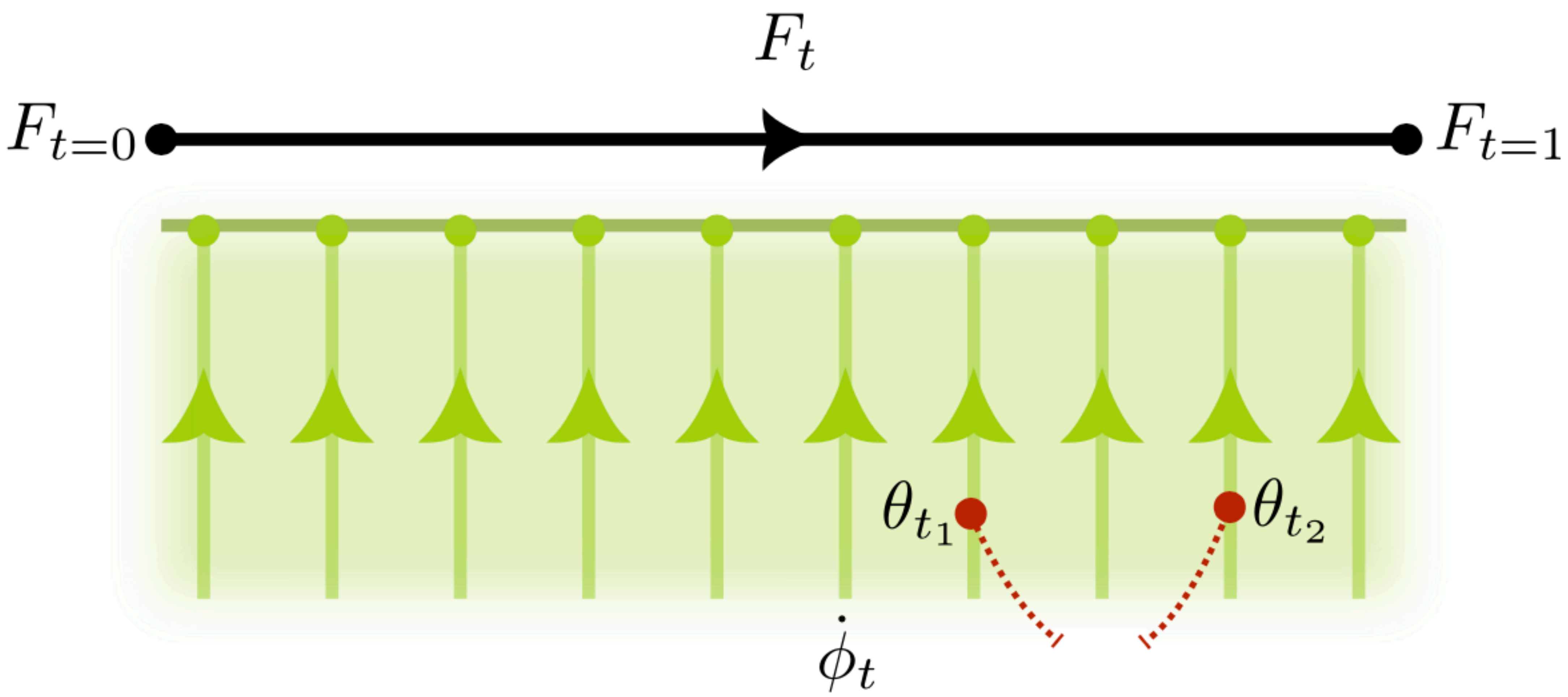
$$\frac{dS_t(\phi)}{dt} + \dot{\phi}_t(\phi)\,\frac{\partial}{\partial\phi} S_t(\phi) = \frac{\partial}{\partial\phi}\,\dot{\phi}_t(\phi)$$

- $S_t(\phi)$ at $t = 0$ and $t = 1$ are fixed, but the path is accessible
- Linear PDE for $\dot{\phi}_t$ (using NN to solve)



PIRG in a nutshell

- With $\dot{\phi}_t$ at each t, ϕ_t can be easily determined
- Key properties of PIKs
  - Independent kernels
  - OOD resolution
  - Optimization



Parametrisation of $S_t(\phi)$

$$S_t(\phi) = \hat{S}_t(\phi) + \log \mathcal{N}_t .$$

$$\hat{S}_t(\phi) = \sum_i c_{i,t}\, \mathcal{O}_i(\phi)$$

$$\frac{d \log \mathcal{N}_t}{dt} = - \int \mathcal{D}\phi\, p_t(\phi) \frac{d\hat{S}_t(\phi)}{dt}$$

Hard to calculate

- $\mathcal{O}_i(\phi)$ is a set of basis functions, and have many choices

Parametrisation of $S_t(\phi)$

- Since $d \log \mathcal{N}_t / dt$ is field independent, we can introduce a simple reference configuration χ and calculate the difference

$$
\begin{aligned}
0 = & \frac{d\hat{S}_t(\phi)}{dt} - \frac{d\hat{S}_t(\chi)}{dt} \\
& - \left[\frac{\partial}{\partial\phi} - \frac{\partial\hat{S}_t(\phi)}{\partial\phi} \right] \dot{\phi}_t(\phi) \\
& + \left[\frac{\partial}{\partial\phi} - \frac{\partial\hat{S}_t(\phi)}{\partial\phi} \right] \dot{\phi}_t(\phi) \Big|_{\phi=\chi}
\end{aligned}
$$

Parametrisation of $\dot{\phi}_t(\phi)$

- The above parametrisation of S_t fixes all S_t-dependent terms in the Wegner equation, and leaves us with the task of solving it for the physics-informed kernels $\dot{\phi}_t(\phi)$

$$\dot{\phi}_t(\phi) = \sum_j k_{j,t}\, K_{j,t}(\phi)$$

$$\rightarrow \qquad A_t\, k_t = b_t$$

PIKs at work — zero dimensional ϕ^4

$$\hat{S}(\varphi) = \frac{1}{2} m^2 \varphi^2 + \frac{\lambda}{4!}\varphi^4 , \qquad \varphi \in \mathbb{R}$$

- The PIRG pair $\{S_t(\phi), \phi_t\}$ is defined by

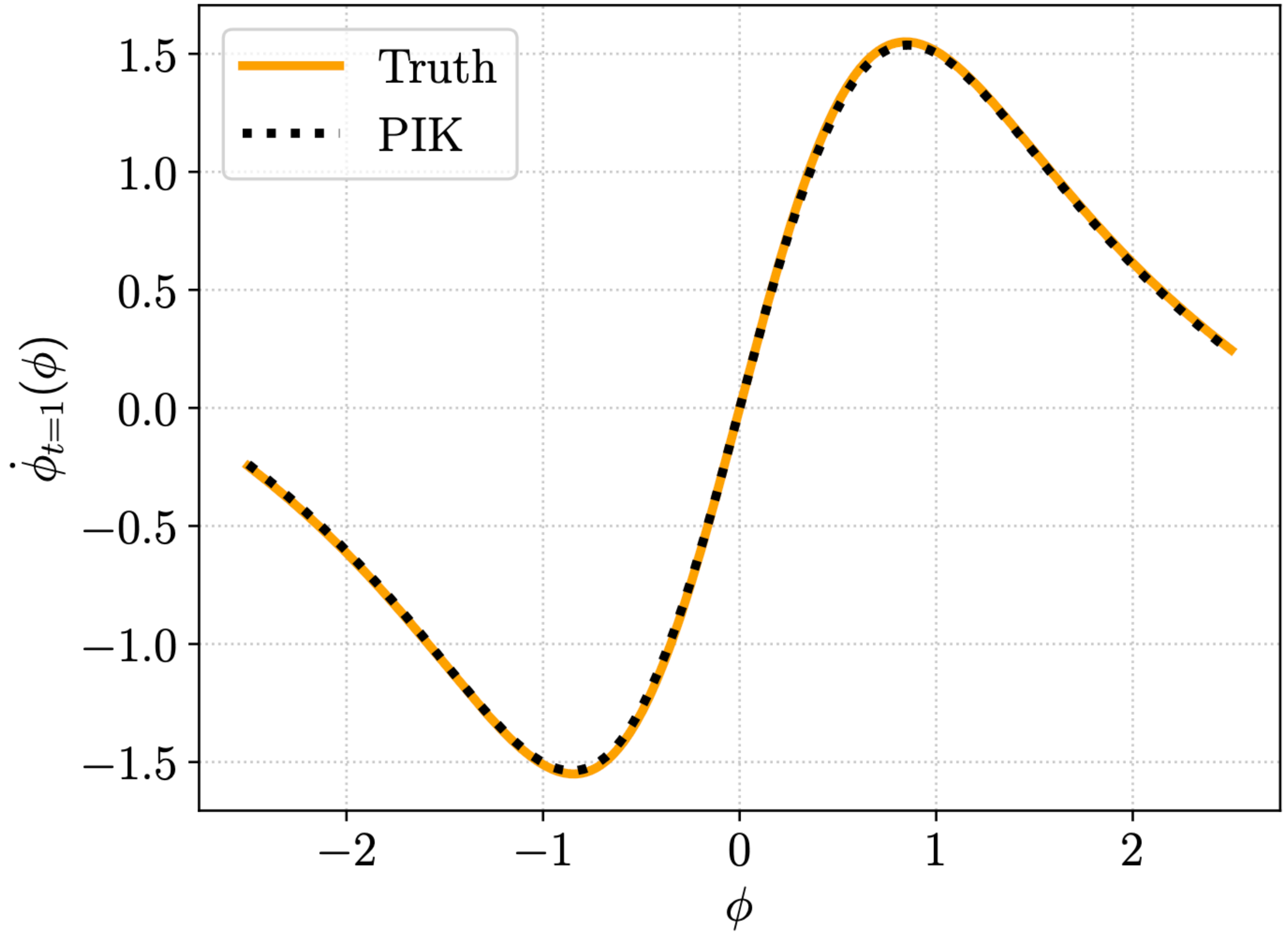
$$\hat{S}_t(\phi) = \frac{1}{2} m^2(t)\, \phi^2 + \frac{\lambda(t)}{4}\, \phi^4 , \qquad \phi \in \mathbb{R} ,$$

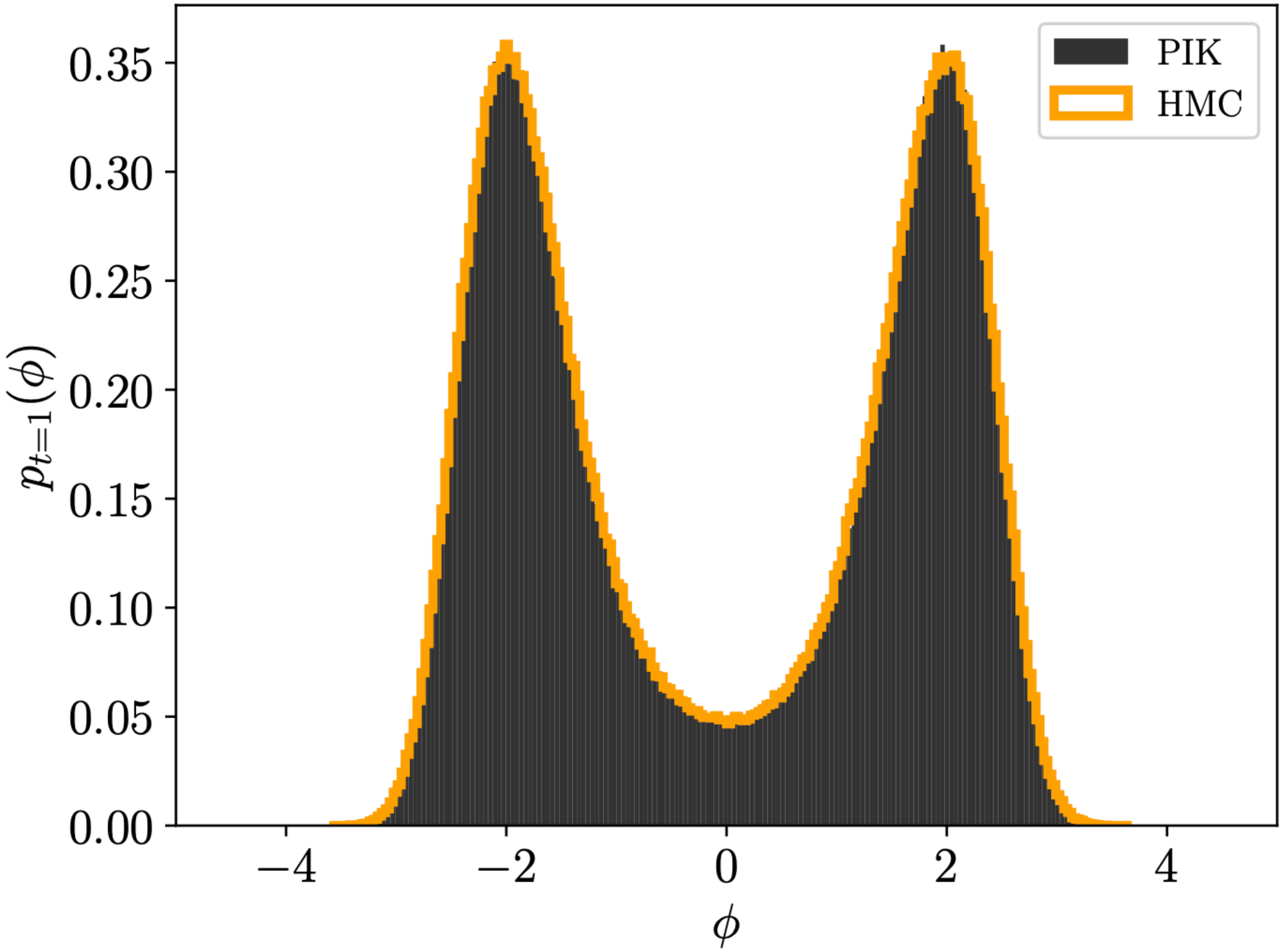
with

$$m^2(t) = m_0^2 + t \left(m_1^2 - m_0^2\right)$$
$$\lambda(t) = \lambda_0 + t \left(\lambda_1 - \lambda_0\right) .$$

$$m_0^2 = 1 , \quad \lambda_0 = 0 \quad \rightarrow \quad m_1^2 = -2 , \quad \lambda_1 = \frac{1}{2}$$

PIKs at work — zero dimensional ϕ^4





Optimisation with parameter-conditional PIKs

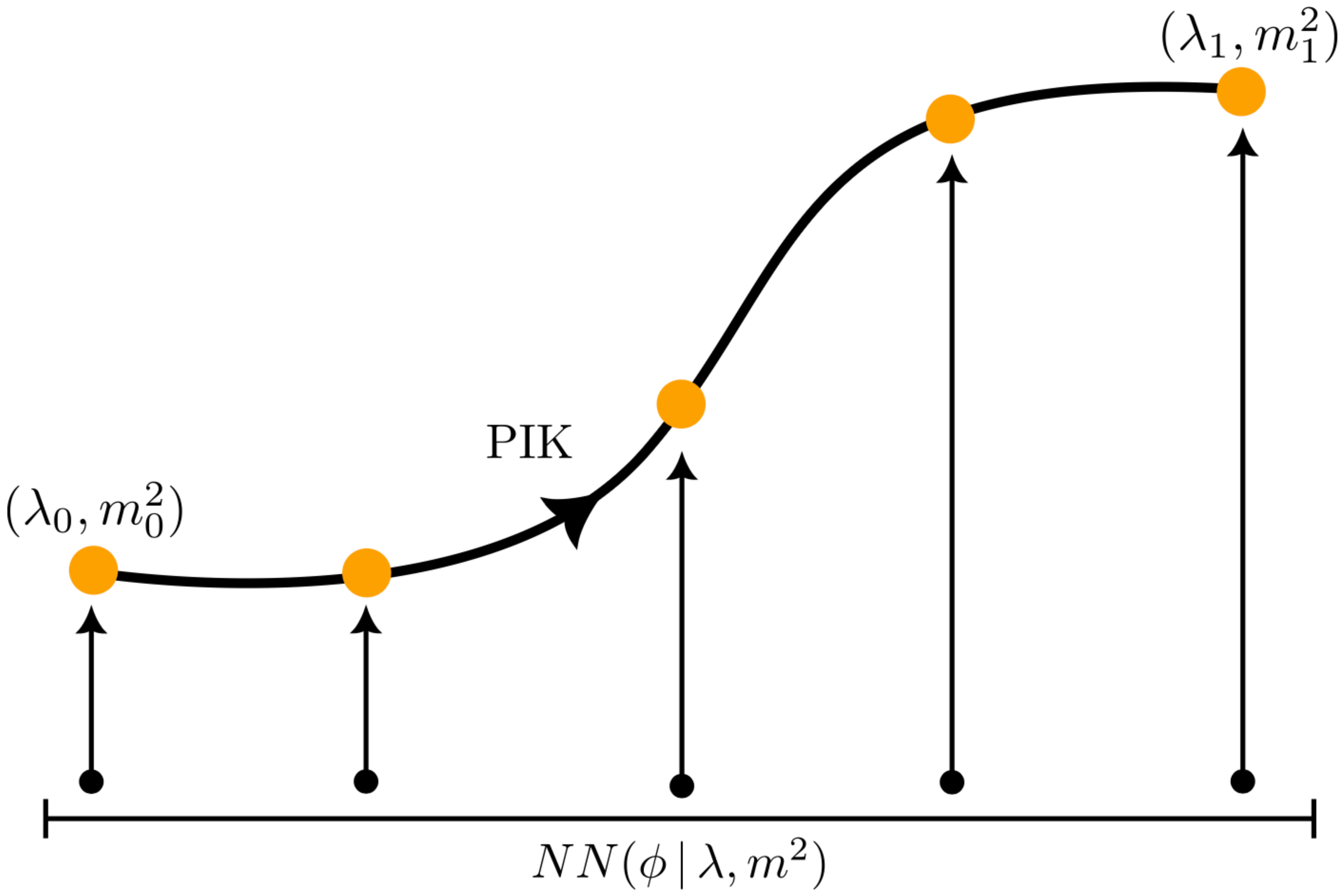


Figure 7. Difference in the kind of parameter conditionality for PIKs differs and traditional generative models $NN(\phi \mid \lambda, m^2)$. While PIKs move along the path determined by the parameters, traditional models need to sample each set of configurations individually.