AI applications in QFT
Part I

Gert Aarts

Swansea University
Prifysgol Abertawe

MITP, Mainz, July 2025

ML seems to be everywhere

what can ML do for theoretical physics?

what can theoretical physics do for ML?

AI4Science

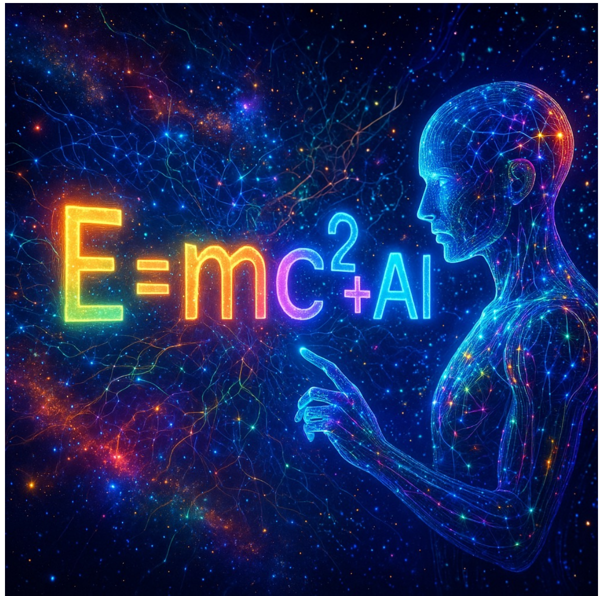
Science4AI

ML seems to be everywhere

what can ML do for theoretical physics?

what can theoretical physics do for ML?

AI4Science



Science4AI

and no, I don’t mean this!

Applications of machine learning to lattice quantum field theory
D Boyda, et al, Snowmass 2021, 2202.05838 [hep-lat]

ML and QFT/LFT

Physics-driven learning for inverse problems in QCD
G Aarts, K Fukushima, T Hatsuda, A Ipp, S Shi, L Wang, K Zhou
Nature Rev. Phys. **7** (2025) no.3, 154 [2501.05580 [hep-lat]]

- ML is explored in lattice field theory in many ways
  - generation of ensembles (generative AI)
  - parameter tuning
  - observable estimation
  - inverse problems
  - sign problem optimisation
  - …

- fast moving, exploratory → (quasi-)rigorous?
- first steps are easy, pursuing the standards we are used to in LFT is harder

ML and QFT: three lectures

choice is biased by my own ~~interests~~ ~~understanding~~ what I think I understand

- lecture I: generative AI: diffusion models for LFT
- lecture II: stochastic gradient descent, random matrix theory and phase diagrams
- lecture III: selected topics (detection of phase transitions, inverse RG, …)

Please ask questions and interrupt me!

Diffusion models and lattice field theory

Gert Aarts

Swansea University
Prifysgol Abertawe

MITP, Mainz, July 2025

Background and references

this lecture is based on what I learned about diffusion models with Lingxiao Wang, Kai Zhou and Diaa Habibi

and about stochastic dynamics with Nucu Stamatescu, Erhard Seiler and Denes Sexty in the more ancient past

- L Wang, GA, K Zhou, JHEP 05 (2024) 060 [2309.17082 [hep-lat]]
- GA, D Habibi, L Wang, K Zhou, Mach.Learn.Sci.Tech. **6** (2025) 2, 025004 [2410.21212 [hep-lat]]
  and PoS(Lattice 2024) [2412.01919 [hep-lat]]
- Q Zhu, W Wang, GA, K Zhou, L Wang, 2502.05504 [hep-lat]

PhD students: **Diaa Habibi**, **Qianteng Zhu**

Outline

- generative AI and diffusion models
- basics: stochastic differential equations (SDEs) and Fokker-Planck equations (FPEs)
- relation between diffusion models and stochastic quantisation in lattice field theory
- detailed study using tools of statistical field theory
- outlook

Generative AI using diffusion models

https://encord.com/blog/diffusion-models/

denoising

Generative Modeling by Estimating Gradients of the Data Distribution
Yang Song, Stefano Ermon
1907.05600 [cs.LG]

interpolation

Score-Based Generative Modeling through Stochastic Differential Equations
Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, Ben Poole, 2011.13456 [cs.LG]

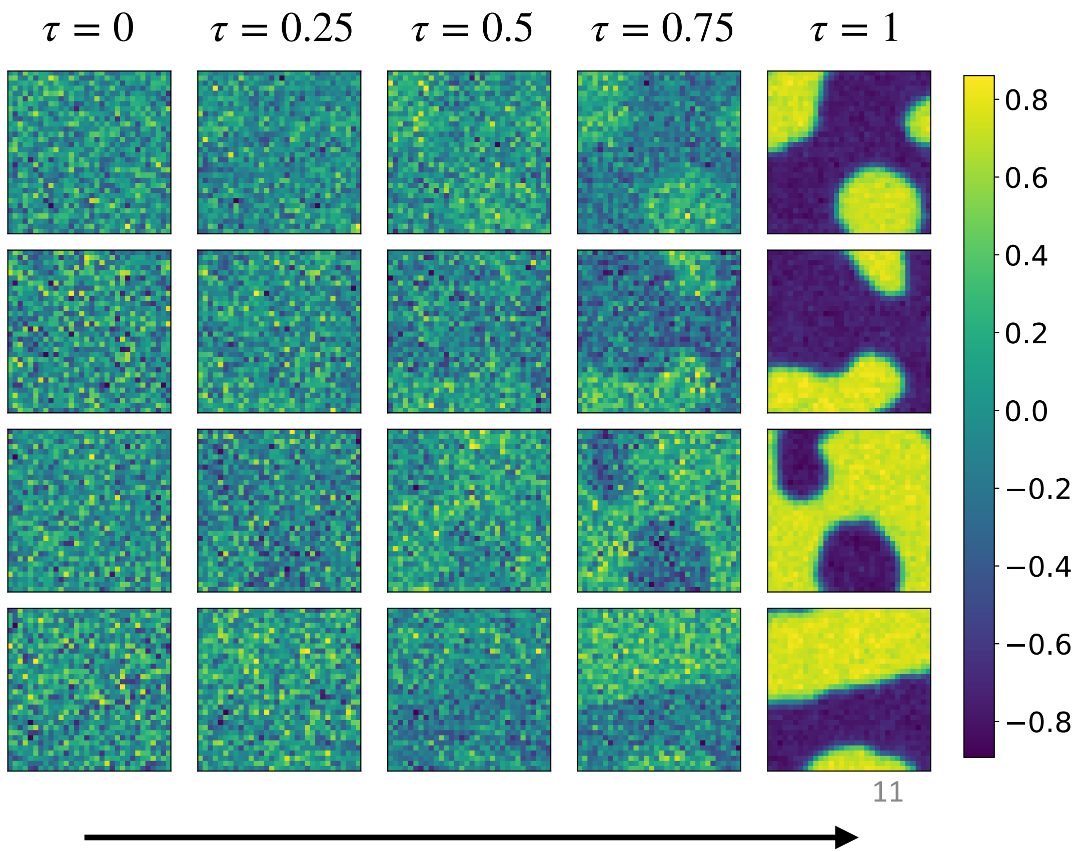
Diffusion model for 2d ϕ^4 lattice scalar theory

- 32^2 lattice, choice of action parameters in symmetric and broken phase
- training data set generated using Hybrid Monte Carlo (HMC)

- first application of diffusion models in lattice field theory

generating configurations:

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

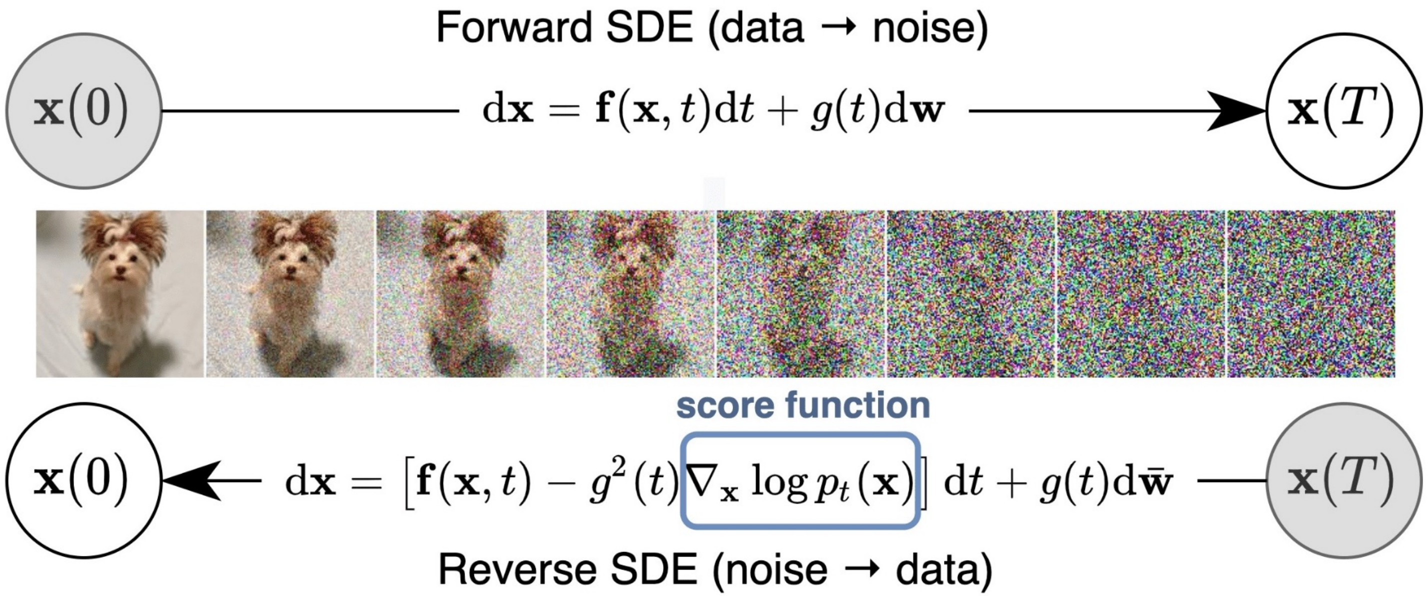


L Wang, GA, K Zhou, JHEP 05 (2024) 060 [2309.17082 [hep-lat]]

Diffusion models: stochastic dynamics

employ stochastic dynamics to generate images or configurations

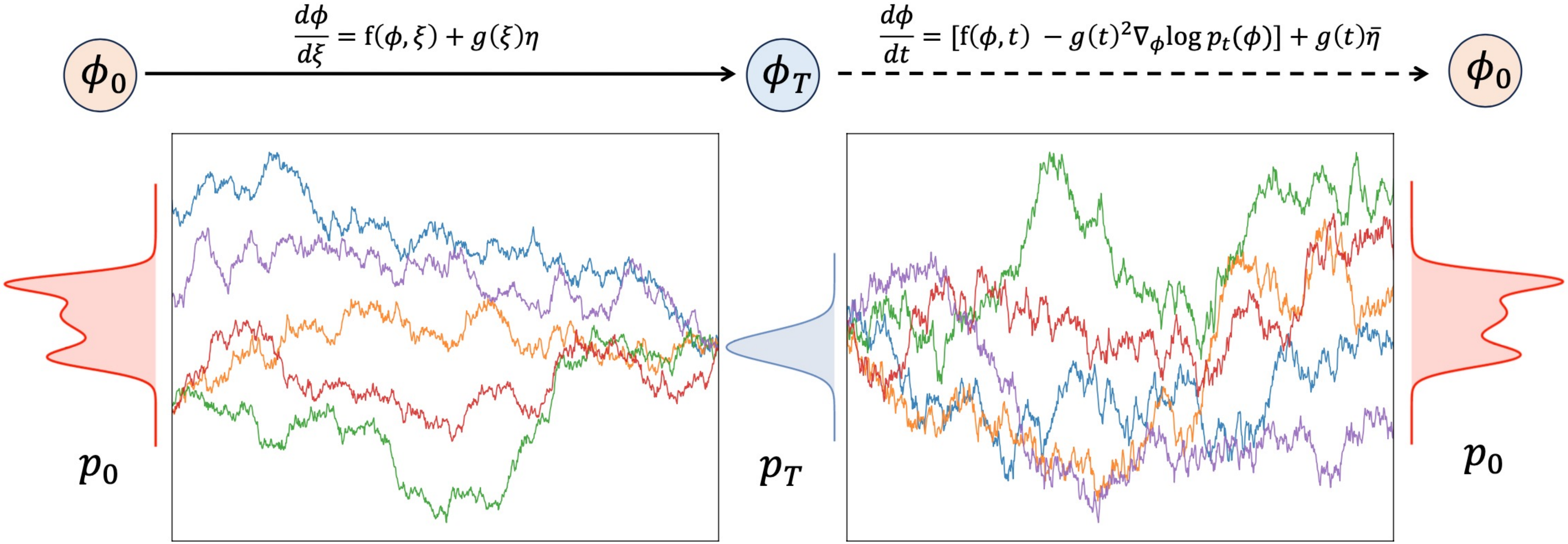
- start with data set of images or configurations
- make the images more blurred by applying noise (forward process)
- learn steps in this process
  … and then revert it
- create new images from noise



https://theaisummer.com/diffusion-models/

Prior and target distributions

- in terms of distributions: p_0 is target (non-trivial), p_T is the prior (easy)



forward process backward process

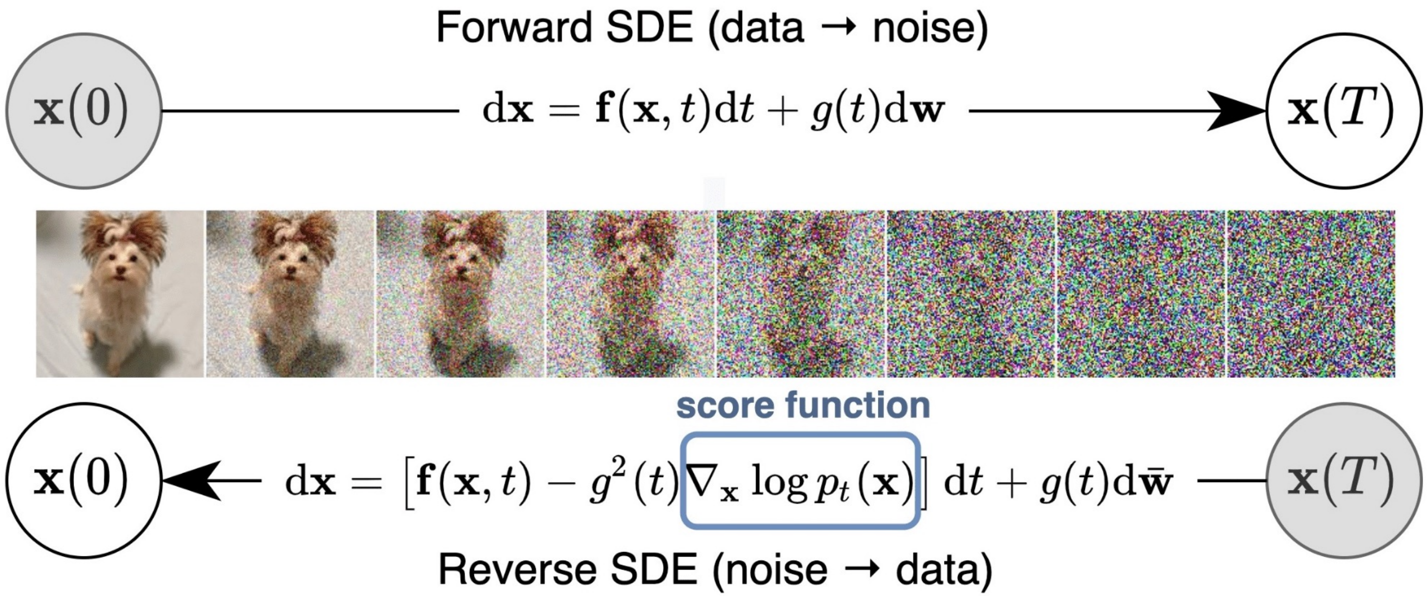
Outline

- generative AI and diffusion models
- **basics: stochastic differential equations (SDEs) and Fokker-Planck equations (FPE)**
- relation between diffusion models and stochastic quantisation in lattice field theory
- detailed study using tools of statistical field theory
- outlook

Diffusion models

three ingredients:

- target distribution, consisting of real-world data or from known distribution (in physics)
- forward stochastic process
- backward stochastic process



https://theaisummer.com/diffusion-models/

Stochastic different equations (SDEs)

- two main approaches:
  - denoising diffusion probabilistic models (DDPMs), variance preserving schemes
  - variance expanding schemes
- unified description using SDEs [Yang Song, et al, arXiv:2011.13456 [cs.LG]]

- here: basic intro to set the stage

- notation can differ (Brownian motion, Wiener process, continuous time, …)
- ➢ I'll be non-rigorous but hopefully (!) correct

- tutorial exercises will go through most steps in some detail

Stochastic different equations (SDEs)

- one degree of freedom $\dot{x}(t) = \frac{1}{2}K[x(t)] + \eta(t)$ describe distribution $P(x) \sim e^{-S(x)}$
- force/drift term $K(x) = \nabla \log P(x) = -S'(x)$ stochastic term $\langle \eta(t)\eta(t')\rangle = \delta(t-t')$
- simple discretisation $x_{n+1} = x_n + \frac{\epsilon}{2}K(x_n) + \sqrt{\epsilon}\eta_n$ $\langle \eta_n \eta_{n'}\rangle = \delta_{nn'}$
- express dynamics in terms of probability distribution

$$\langle O[x(t)]\rangle_\eta = \int dx\, P(x,t) O(x) = \langle O\rangle_{P(t)}$$

- replace noise average by average over distribution: $P(x,t)$ satisfies Fokker-Planck equation

Derivation of FPE from SDE

- SDE/Langevin equation

$$\dot{x}(t) = \frac{1}{2}K[x(t)] + \eta(t)$$

- replace sum over trajectories by time evolution of distribution

$$\langle O[x(t)] \rangle_\eta = \int dx\, P(x,t) O(x) = \langle O \rangle_{P(t)}$$

- Fokker-Planck equation

$$\partial_t P(x,t) = \frac{1}{2}\partial_x \left(\partial_x - K(x)\right) P(x,t)$$

- derivation in tutorial exercise

Fokker-Planck equation, stationary solution

- FPE: $\partial_t P(x,t) = \frac{1}{2}\partial_x\left(\partial_x - K(x)\right)P(x,t)$ $\quad K(x) = \nabla \log P(x) = -S'(x)$
- stationary solution: $\partial_t P(x,t) = 0$
- if force is derivative of action: $\left(\partial_x - K(x)\right)P(x) = 0 \quad \Leftrightarrow \quad \left(\partial_x + S'(x)\right)P(x) = 0$
- stationary distribution: $P(x) \sim e^{-S(x)}$ expected result

standard result for Brownian motion, add a few more ingredients:

- time dependent noise strength, or diffusion coefficient
- time dependent force

Time-dependent noise, diffusion coefficient

- to cover information at all scales in data set: time dependent noise

$$\dot{x} = \frac{1}{2} g^2(t) K[x(t)] + g(t)\eta(t)$$

- corresponding FPE: $$\partial_t P(x,t) = \frac{1}{2} g^2(t) \partial_x \left[\partial_x - K(x)\right] P(x,t)$$

- can hence also be seen as reparameterisation of 'time'

- in ML jargon: noise schedulers

Many degrees of freedom, field theory

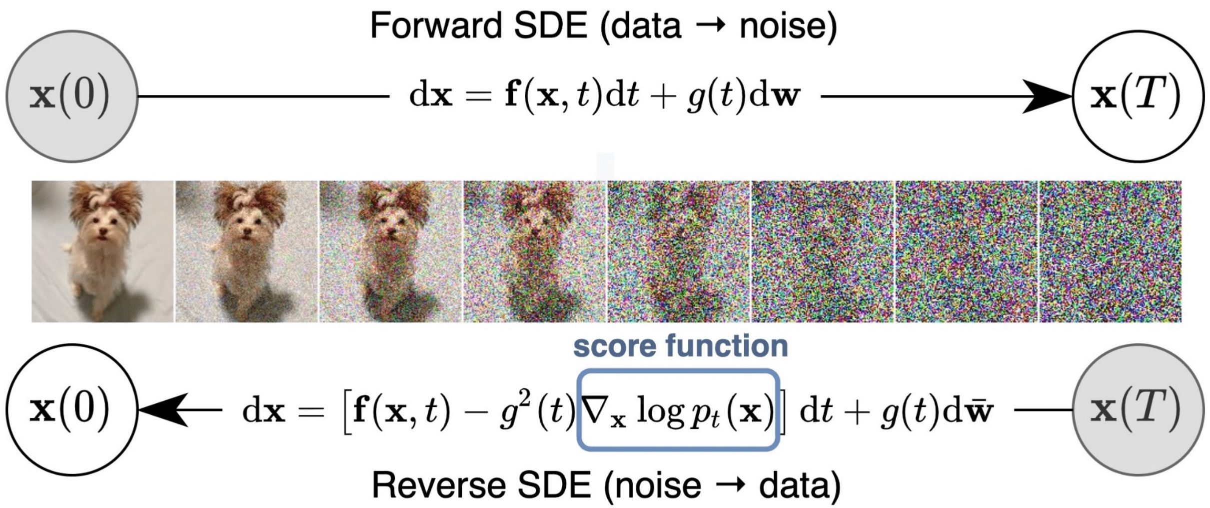
- distributions are functionals (path integrals) $P[\phi] = \frac{1}{Z} e^{-S[\phi]}$ $Z = \int D\phi \, e^{-S[\phi]}$

- SDE $$\frac{\partial \phi(x,t)}{\partial t} = \frac{1}{2} g^2(t) K[\phi(x), t] + g(t) \eta(x,t)$$

- FPE $$\partial_t P[\phi, t] = \frac{1}{2} g^2(t) \int d^n x \, \frac{\delta}{\delta \phi(x)} \left(\frac{\delta}{\delta \phi(x)} - K[\phi(x), t] \right) P[\phi, t]$$

Evolving distributions

- apply this framework to evolve distributions forward and backward



SDE/FPE evolves distribution in time

- forward evolution: start from data, erase information but learn along the way
- add increasing levels of noise, simplest case: no drift term $\dot{x}(t) = g(t)\eta(t)$ $\eta \sim \mathcal{N}(0,1)$
- time-dependent noise strength: $g(t) = \sigma^{t/T}$ $0 \leq t \leq T$
- solution: $x(t) = x_0 + \int_0^t ds\, g(s)\eta(s)$ $\Rightarrow$ $x(t) = x_0 + \sigma(t)\eta(t)$
- variance keeps increasing $\langle (x(t) - x_0)^2 \rangle = \sigma^2(t)$ $\sigma^2(t) = \int_0^t ds\, g^2(s)$
- 'erases' the information from the initial data set

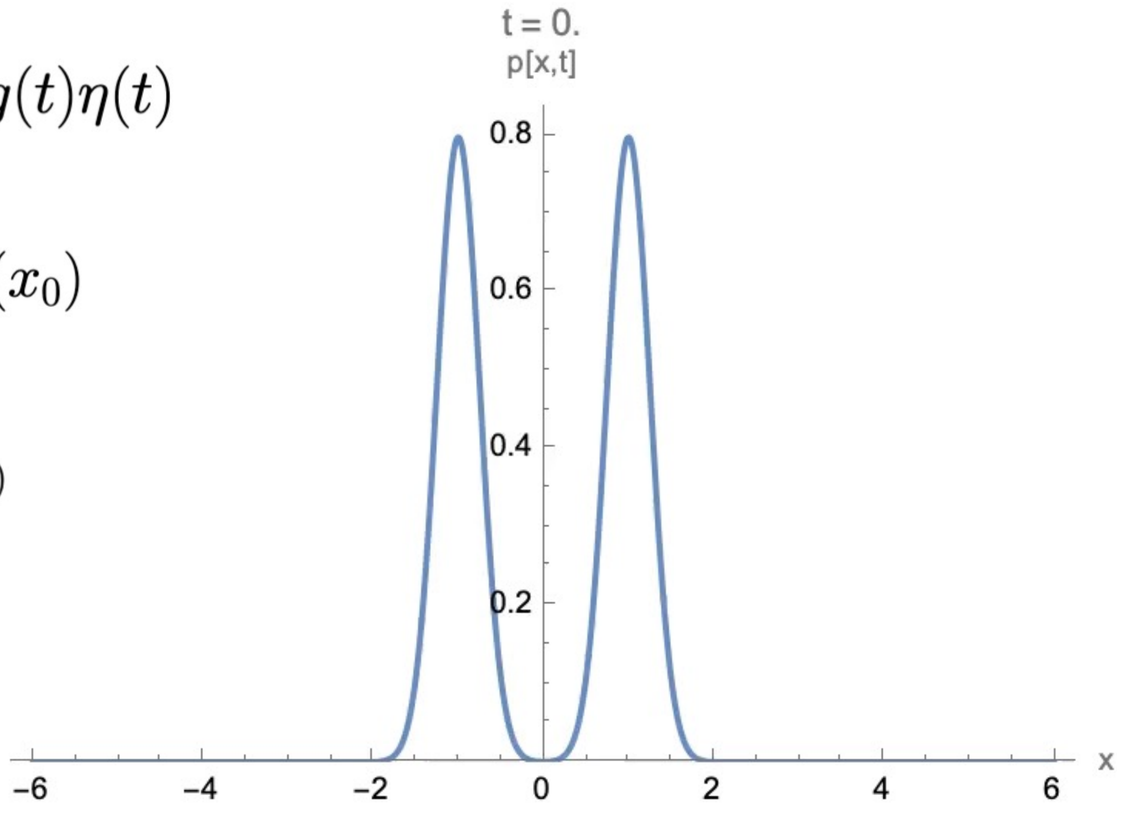
FPE: $\partial_t P_t(x) = \frac{1}{2} g^2(t) \partial_x^2 P_t(x)$

Example: forward evolution

- initial distribution $P_0(x_0)$: two Gaussian peaks (Gaussian mixture)
- add noise in variance-expanding scheme $\dot{x}(t) = g(t)\eta(t)$
- analytical description $P_t(x) = \int dx_0\, P_t(x|x_0) P_0(x_0)$

$$P_t(x|x_0) = \mathcal{N}(x; x_0, \sigma^2(t)) = \frac{1}{\sqrt{2\pi\sigma^2(t)}} e^{-(x-x_0)^2/(2\sigma^2(t))}$$

- peak structure erased



Manifold hypothesis

- logical separation between data – distribution $P_0(x_0)$ – and stochastic process

$$P_t(x) = \int dx_0 \, P_t(x|x_0) P_0(x_0)$$

$$P_t(x|x_0) = \mathcal{N}(x; x_0, \sigma^2(t)) = \frac{1}{\sqrt{2\pi\sigma^2(t)}} e^{-(x-x_0)^2/(2\sigma^2(t))}$$

- manifold hypothesis: real-world data concentrated on low-dimensional manifolds embedded in a high-dimensional space (the ambient space)

- at the end of the forward process, the entire high-dimensional space should be covered

- adding noise with increasing strength ensures all data structures are captured

Backward evolution: the score

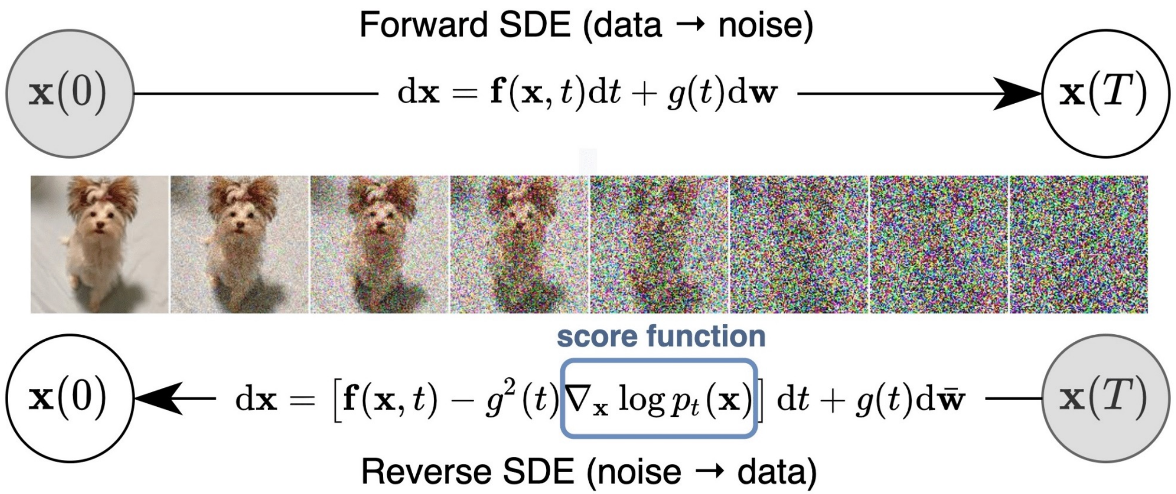
- structure emerges from noise: add a drift term, the score

- from structure of FPE: drift drives distribution to desired target distribution

- use Anderson equation [B.D.O. Anderson (1982)] $\tau = T - t$

$$x'(\tau) = -\frac{1}{2}K(x(\tau), T-\tau) + g^2(T-\tau)\partial_x \log(P(x, T-\tau) + g(T-\tau)\eta(\tau)$$

- SDE includes new term: score $\nabla \log P_t(x)$



Example: backward evolution

$0 \leq t \leq T$

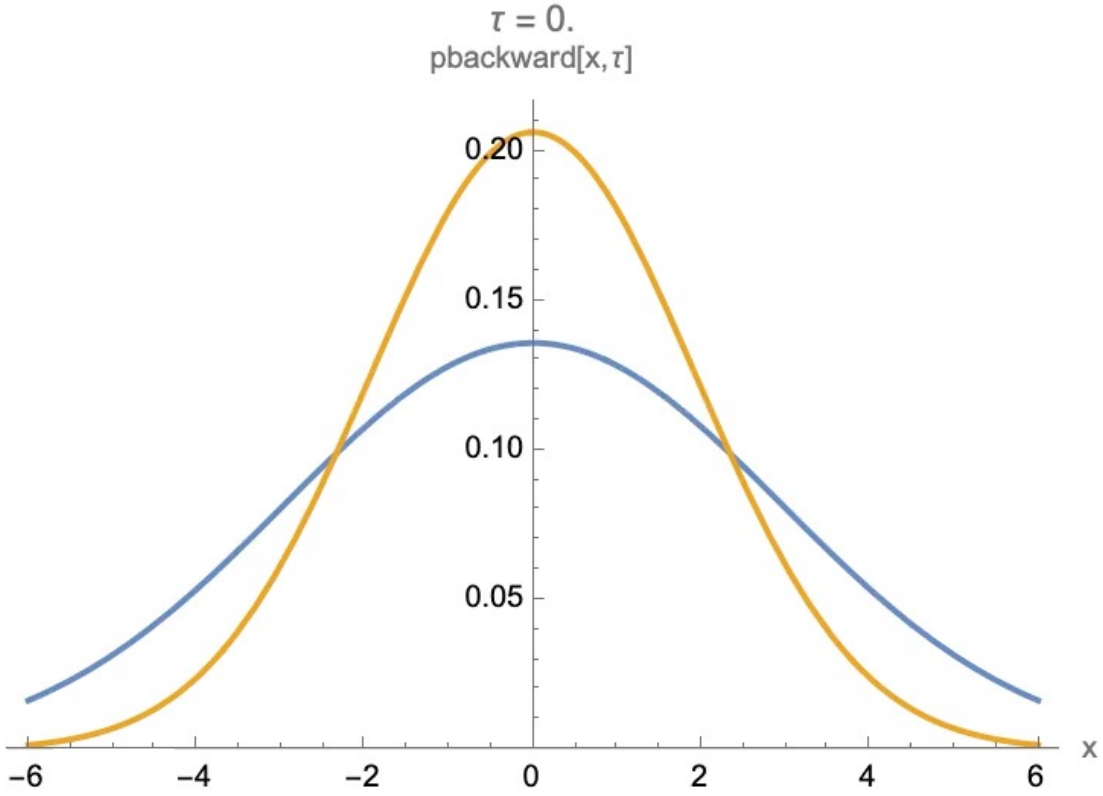
noise profile $g(t) = \sigma^{t/T}$

- target distribution: two Gaussian peaks
- forward process $\dot{x}(t) = K(x(t), t) + g(t)\eta(t)$
- corresponding backward process

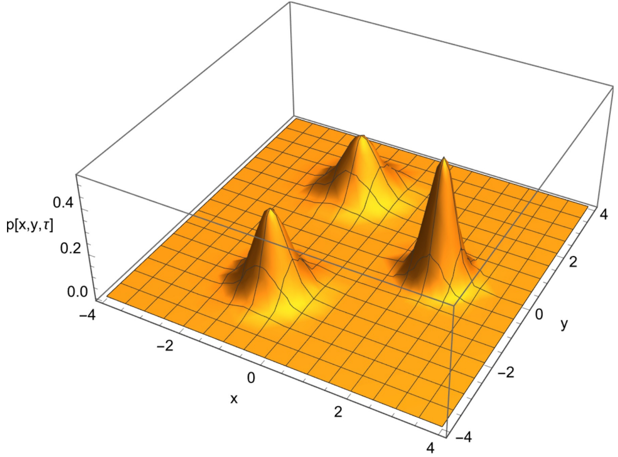
$$\begin{aligned} x'(\tau) = &- \frac{1}{2} K(x(\tau), T-\tau) \\ &+ g^2(T-\tau)\partial_x \log(P(x, T-\tau) \\ &+ g(T-\tau)\eta(\tau) \end{aligned}$$

with $\tau = T - t$

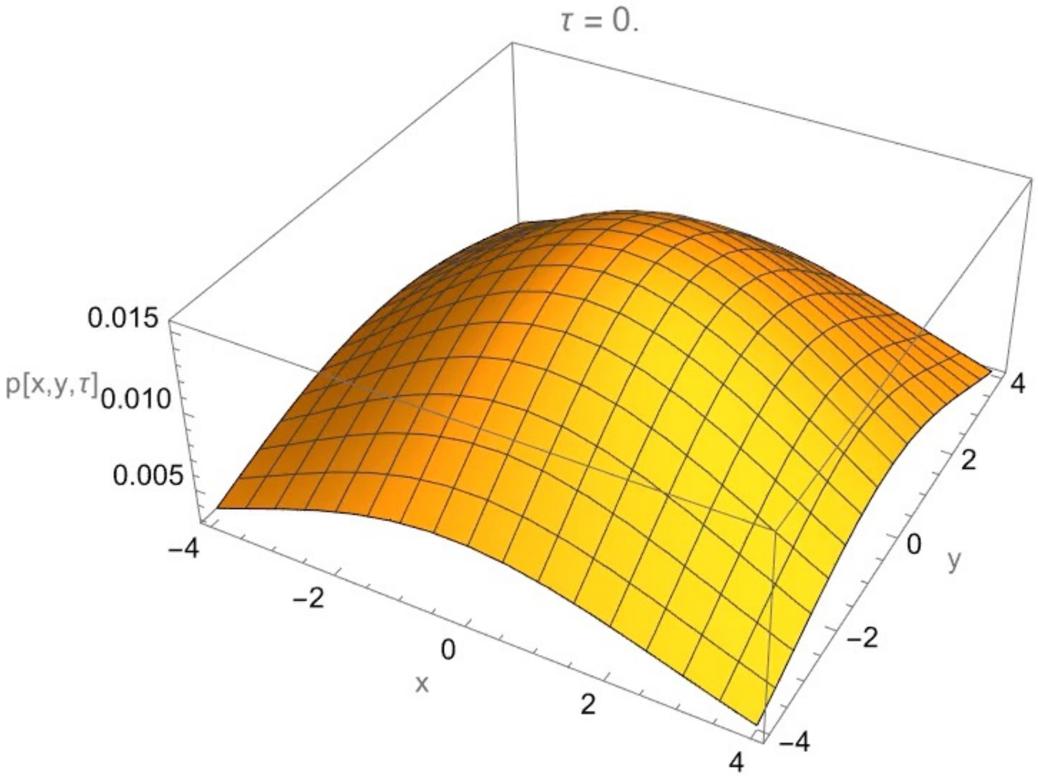
solve FPE for backward process using two initial distributions



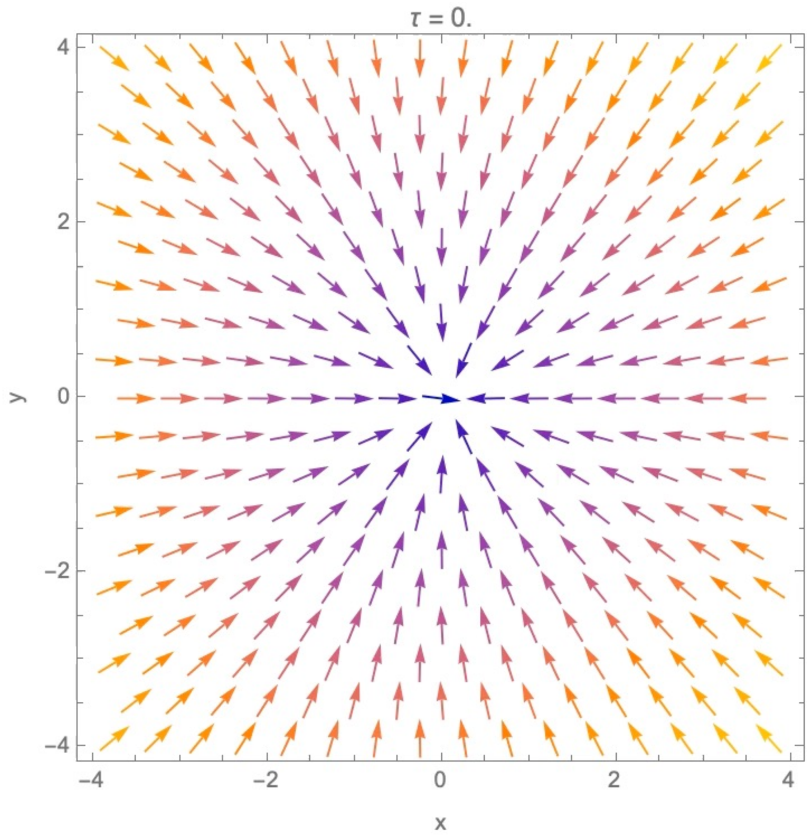
2D example: three Gaussian peaks



backward process, starting from wide normal distribution

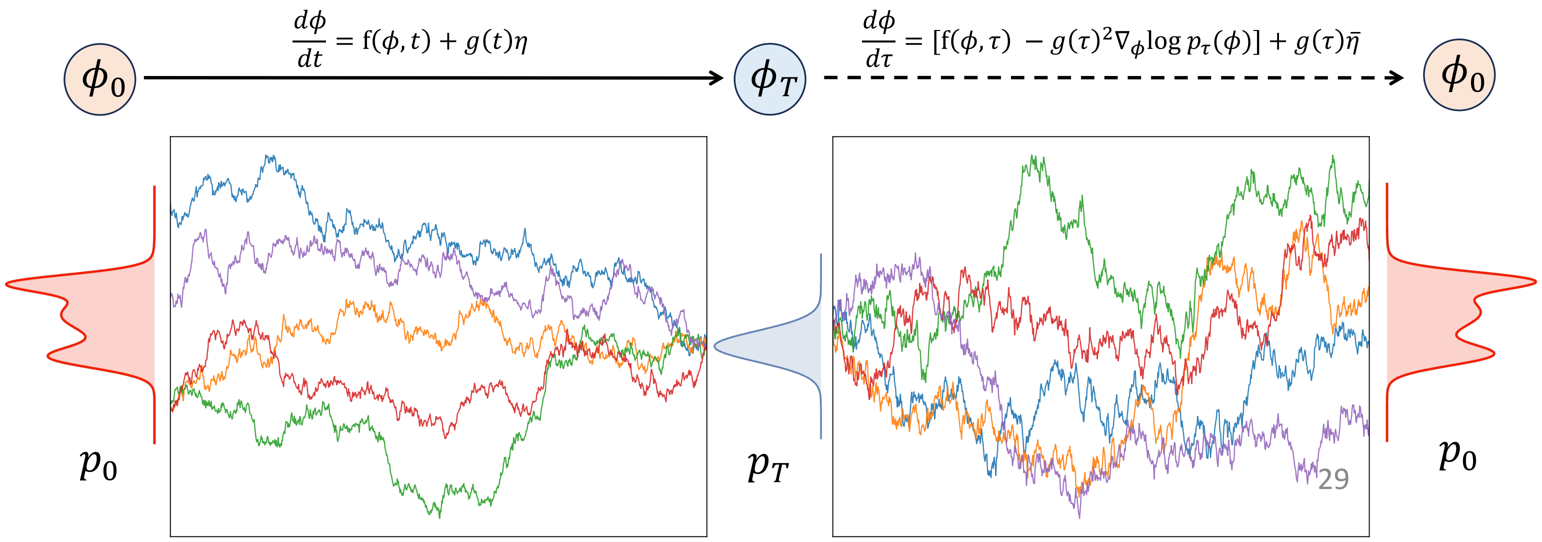


score $\nabla P_t(x, y)$ during backward process



Where does ML come in?

- so far, analysis of SDEs and FPEs
- time-dependent distribution $P(x,t) = P_t(x)$ describes forward and backward process
- in general **score** $\nabla \log P_t(x)$ is not known, needs to be "learnt" during forward process
- score matching
- minimise loss function



Score matching: learn the drift for backward process

- one degree of freedom, variance-expanding scheme: $\dot{x}(t) = g(t)\eta(t) \qquad \eta \sim \mathcal{N}(0,1)$
- time-dependent distribution $P(x,t) = P_t(x)$ describes forward and backward process
- so-called **score** $\nabla \log P_t(x)$ is not known, needs to be "learnt"
- loss function $\mathcal{L}(\theta) = \frac{1}{2}\int_0^T dt\, \mathbb{E}_{P_t(x)}\left[\sigma^2(t)\,\|s_\theta(x,t) - \nabla \log P_t(x)\|^2\right] \qquad \sigma^2(t) = \int_0^t ds\, g^2(s)$
- $s_\theta(x,t)$ approximates score, vector field learnt by some neural network
- introduce conditional distribution $P_t(x) = \int dx_0\, P_t(x|x_0)P_0(x_0)$ initial data $P_0(x_0)$

$$P_t(x) = \int dx_0\, P_t(x|x_0) P_0(x_0)$$

Score matching: learn the drift

- loss function $\mathcal{L}(\theta) = \frac{1}{2}\int_0^T dt\, \mathbb{E}_{P_t(x)}\left[\sigma^2(t)\, \|s_\theta(x,t) - \nabla \log P_t(x)\|^2\right]$
- diffusion process $\dot{x}(t) = g(t)\eta(t)$ easily solved $x(t) = x_0 + \sigma(t)\eta(t)$ $\quad \sigma^2(t) = \int_0^t ds\, g^2(s)$
- conditional distribution $P_t(x|x_0) = \mathcal{N}(x; x_0, \sigma^2(t)) = \frac{1}{\sqrt{2\pi\sigma^2(t)}} e^{-(x-x_0)^2/(2\sigma^2(t))}$

(use Jensen's inequality)

- and hence $\nabla \log P_t(x_t|x_0) = -(x_t - x_0)/\sigma^2(t)$
- loss function $$\mathcal{L}(\theta) = \frac{1}{2}\int_0^T dt\, \mathbb{E}_{P_t(x_t)}\left[\left\|\sigma(t)s_\theta(x_t,t) + \frac{x_t - x_0}{\sigma(t)}\right\|^2\right]$$ $$= \frac{1}{2}\int_0^T dt\, \mathbb{E}_{P_t(x_t)}\left[\|\sigma(t)s_\theta(x_t,t) + \eta(t)\|^2\right]$$

tractable, computable

ML applications

- two main approaches, depending on choice of drift in forward process:
  - denoising diffusion probabilistic models (DDPMs), variance preserving schemes

    linear drift term $$\dot{x}(t) = -\frac{1}{2}g^2(t)x(t) + g(t)\eta(t)$$

  - variance expanding schemes

    no drift term $$\dot{x}(t) = g(t)\eta(t)$$

- in both cases the transition amplitude $P_t(x|x_0)$ is known analytically and setup works

Outline

- generative AI and diffusion models
- basics: stochastic differential equations (SDEs) and Fokker-Planck equations (FPEs)
- **relation between diffusion models and stochastic quantisation in lattice field theory**
- detailed study using tools of statistical field theory
- outlook

Lattice field theory simulations

- create sequence of configurations to estimate observables
- statistically independent, satisfy detailed balance, ergodic
- based on Boltzmann weight $P(\phi) \sim e^{-S(\phi)}$ importance sampling
- hybrid Monte Carlo (HMC) widely used
- some issues: critical slowing down near phase transitions, topological freezing in the presence of topological sectors, …
- stochastic quantisation, early proposal for LFT simulations (Parisi & Wu 1980)

Diffusion models and stochastic quantisation

- configurations are generated during backward process

- stochastic process with time-dependent drift and noise strength

$$\frac{\partial \phi(x,\tau)}{\partial \tau} = g^2(\tau)\nabla_\phi \log P(\phi;\tau) + g(\tau)\eta(x,\tau)$$

- write $P(\phi;\tau) = \frac{e^{-S(\phi,\tau)}}{Z}$ such that $\nabla_\phi \log P(\phi,\tau) = -\nabla_\phi S(\phi,\tau)$

- then $$\frac{\partial \phi(x,\tau)}{\partial \tau} = -g^2(\tau)\nabla_\phi S(\phi,\tau) + g(\tau)\eta(x,\tau)$$

Diffusion models and stochastic quantisation

- then $$\frac{\partial \phi(x,\tau)}{\partial \tau} = -g^2(\tau)\nabla_\phi S(\phi,\tau) + g(\tau)\eta(x,\tau)$$
- very familiar to (lattice) field theorists
- stochastic quantisation (Parisi & Wu 1980)
- path integral quantisation via a stochastic process in fictitious time $$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\nabla_\phi S(\phi) + \sqrt{2}\eta(x,\tau)$$
- stationary solution of associated Fokker-Planck equation $P(\phi) \sim e^{-S(\phi)}$

Diffusion models and stochastic quantisation

$$\frac{\partial \phi(x,\tau)}{\partial \tau} = g^2(\tau)\nabla_\phi \log P(\phi;\tau) + g(\tau)\eta(x,\tau)$$

$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\nabla_\phi S(\phi) + \sqrt{2}\eta(x,\tau)$$

similarities and differences:

✓ SQ: fixed drift, determined from known action
constant noise variance (but can be generalised using kernels)
thermalisation followed by long-term evolution in equilibrium

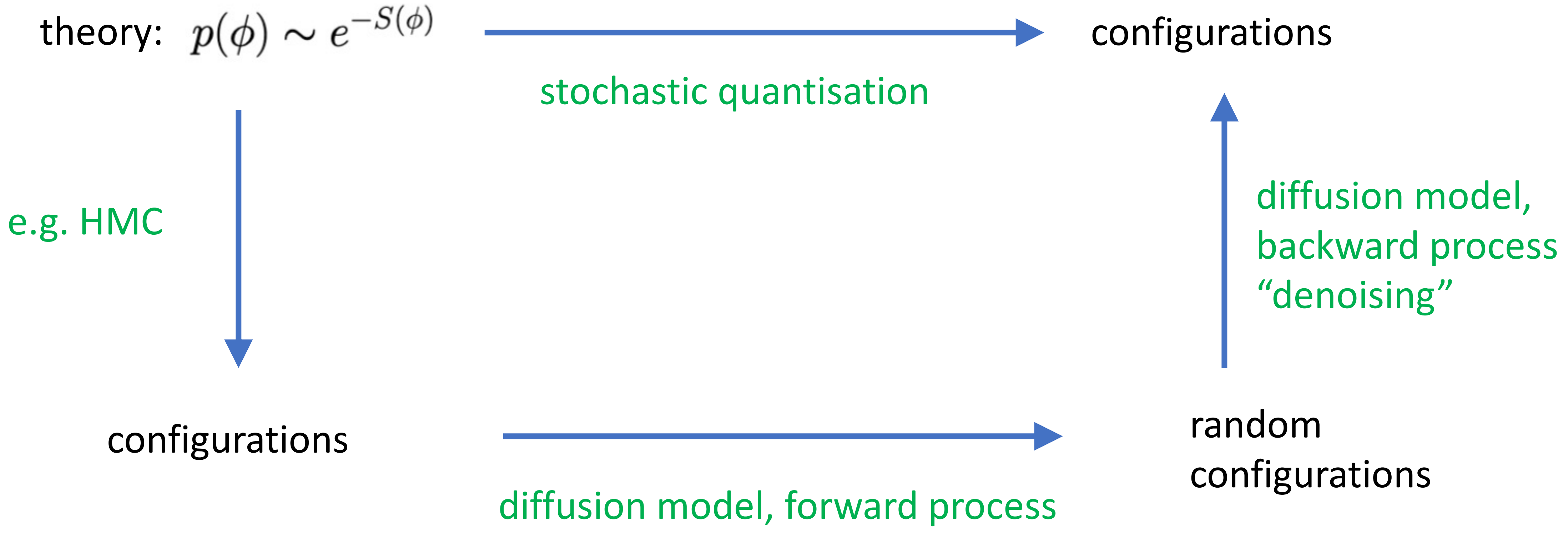
✓ DM: drift and noise variance time-dependent, learn from data
evolution between $0 \leq \tau \leq T = 1$, many short runs

side remark:
I worked on stochastic quantisation in QCD and theories with a sign problem during 2008-2015

GA and IO Stamatescu, Stochastic quantisation at finite chemical potential, JHEP 09 (2008) 018 [0807.1597 [hep-lat]]

Diffusion models and stochastic quantisation

- diffusion models as an alternative approach to stochastic quantisation



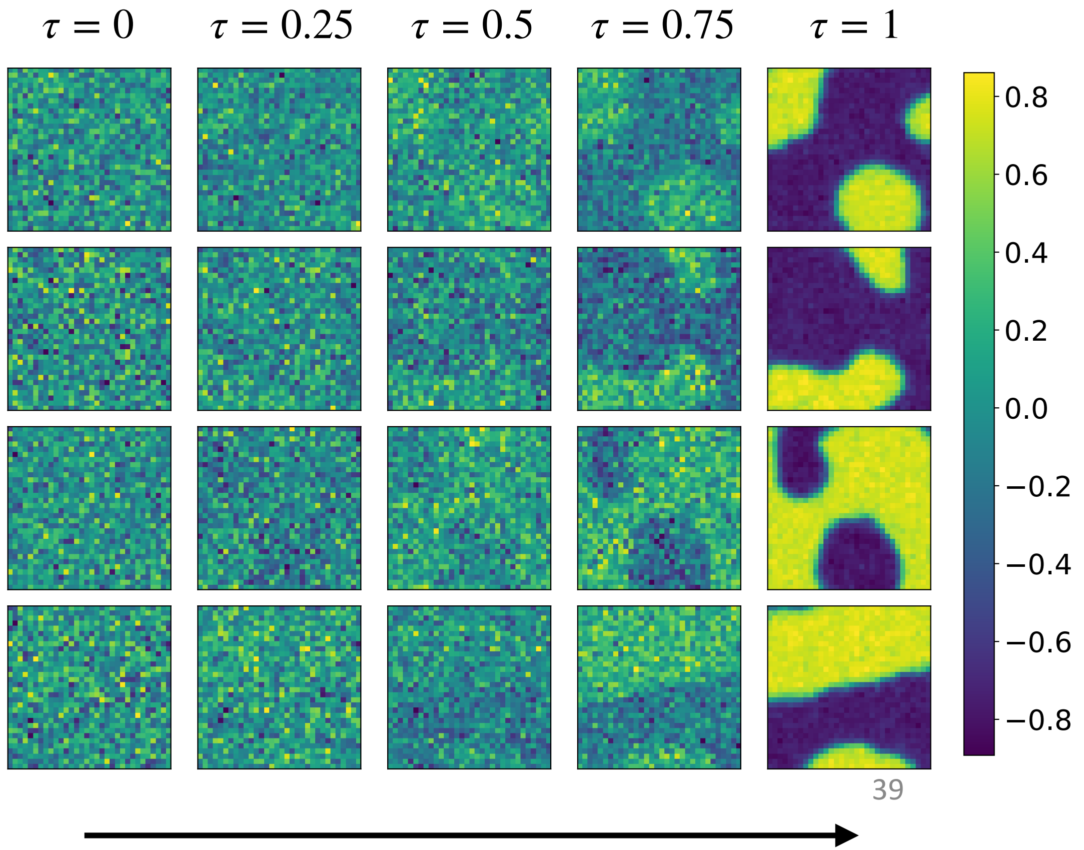
Diffusion model for 2d ϕ^4 lattice scalar theory

- 32^2 lattice, choice of action parameters in symmetric and broken phase
- training data set generated using Hybrid Monte Carlo (HMC)

- first application of diffusion models in lattice field theory

generating configurations:

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



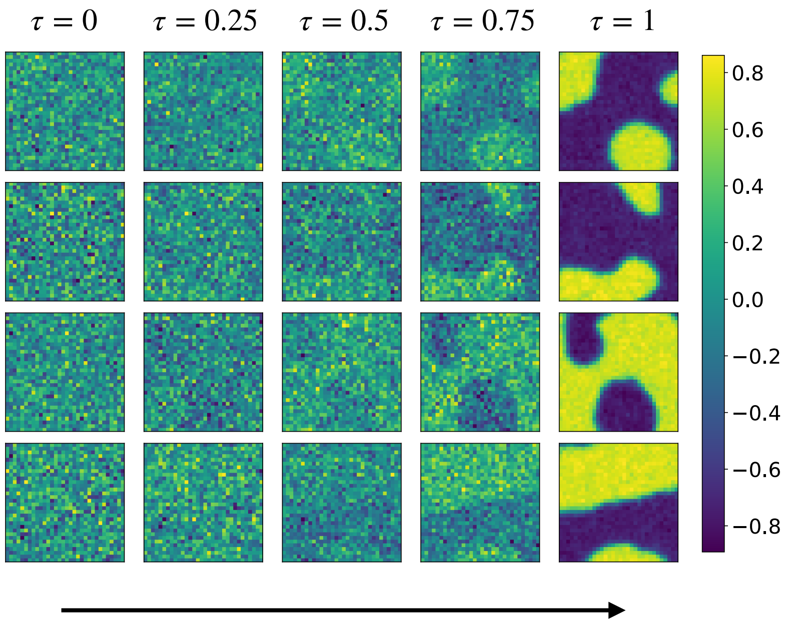
Diffusion models for LFT

- in “real-world” applications the target or data distribution is not known analytically
- only samples are available for learning or training
- in physics applications, we usually know the theory and and hence the distribution
- this allows for use of physical intuition in designing diffusion models
- physics-conditioned DMs for lattice gauge theory [2502.05504 [hep-lat]]
- inclusion of accept/reject step to make algorithm exact [2502.05504 [hep-lat]]

Outline

- generative AI and diffusion models
- basics: stochastic differential equations (SDEs) and Fokker-Planck equations (FPEs)
- relation between diffusion models and stochastic quantisation in lattice field theory
- **detailed study using tools of statistical field theory**
- outlook

Diffusion models



ok, so it seems to work: many questions

- correlations: how are they destroyed and rebuilt?
- usually attention is on two-point function or variance
- but higher n-point functions contain interactions in field theory
- essential for applications in field theory, correlations = interactions
- focus on moments and cumulants

discuss forward and backward process in more detail

GA, Diaa Habibi, L Wang, K Zhou, Mach.Learn.Sci.Tech. 6 (2025) 2, 025004 [2410.21212 [hep-lat]]

Diffusion models in more detail

- forward process $\dot{x}(t) = K(x(t), t) + g(t)\eta(t)$ $0 \le t \le T$

noise profile $g(t) = \sigma^{t/T}$

- backward process

$$x'(\tau) = -K(x(\tau), T-\tau) + g^2(T-\tau)\partial_x \log P(x, T-\tau) + g(T-\tau)\eta(\tau)$$

score

$\tau = T - t$

two main schemes

- variance-expanding (VE): no drift $K(x, t) = 0$
- variance-preserving (VP) or denoising diffusion probabilistic models (DDPMs):

linear drift $K(x(t), t) = -\frac{1}{2}k(t)x(t)$

assume first moment vanishes

$$x_0 \to x_0 - \mathbb{E}_{P_0}[x_0]$$

Solve forward process

- forward process $\dot{x}(t) = K(x(t),t) + g(t)\eta(t)$ $\quad K(x(t),t) = -\frac{1}{2}k(t)x(t)$
- initial data from target ensemble $x_0 \sim P_0(x_0)$
- solution $x(t) = x_0 f(t,0) + \int_0^t ds\, f(t,s)g(s)\eta(s)$ $\quad f(t,s) = e^{-\frac{1}{2}\int_s^t ds'\, k(s')}$
- second moment/cumulant/variance $\kappa_2(t) = \mu_2(t) = \mu_2(0) f^2(t,0) + \Xi(t)$

$$\Xi(t) = \int_0^t ds \int_0^t ds'\, f(t,s)f(t,s')g(s)g(s')\mathbb{E}_\eta[\eta(s)\eta(s')] = \int_0^t ds\, f^2(t,s)g^2(s)$$

$$f(t,s) = e^{-\frac{1}{2}\int_s^t ds'\, k(s')}$$

Higher-order moments and cumulants

- moments $\mu_n(t) = \mathbb{E}[x^n(t)]$ and cumulants $\kappa_n(t)$: straightforward algebra

$$\kappa_3(t) = \mu_3(t) = \kappa_3(0) f^3(t,0)$$

$$\mu_4(t) = \mu_4(0) f^4(t,0) + 6\mu_2(0) f^2(t,0)\Xi(t) + 3\Xi^2(t)$$

$$\kappa_4(t) = \mu_4(t) - 3\mu_2^2(t) = \left[\mu_4(0) - 3\mu_2^2(0)\right] f^4(t,0) = \kappa_4(0) f^4(t,0)$$

$$\kappa_5(t) = \left[\mu_5(0) - 10\mu_3(0)\mu_2(0)\right] f^5(t,0) = \kappa_5(0) f^5(t,0)$$

→ $$\kappa_{n>2}(t) = \kappa_n(0) f^n(t,0)$$

variance-expanding scheme: no drift

$$f(t,0) = 1$$

higher cumulants conserved!

$$\Xi(t) = \int_0^t ds\, f^2(t,s) g^2(s)$$

Proof to all orders

- generating functionals: average over both noise and target distributions

moments $Z[J] = \mathbb{E}[e^{J(t)x(t)}]$ cumulants $W[J] = \log Z[J]$

- noise average $$Z_\eta[J] = \mathbb{E}_\eta[e^{J(t)x(t)}] = \frac{\int D\eta\, e^{-\frac{1}{2}\int_0^t ds\, \eta^2(s) + J(t)\left[x_0 f(t,0) + \int_0^t ds\, f(t,s) g(s) \eta(s)\right]}}{\int D\eta\, e^{-\frac{1}{2}\int_0^t ds\, \eta^2(s)}}$$

- full average $$Z[J] = \mathbb{E}[e^{J(t)x(t)}] = e^{\frac{1}{2}J^2(t)\Xi(t)} \int dx_0\, P_0(x_0) e^{J(t)x_0 f(t,0)}$$

- cumulant generator $$W[J] = \log Z[J] = \frac{1}{2}J^2(t)\Xi(t) + \log \int dx_0\, P_0(x_0) e^{J(t)x_0 f(t,0)}$$

$$f(t,s) = e^{-\frac{1}{2}\int_s^t ds'\, k(s')}$$

$$\Xi(t) = \int_0^t ds\, f^2(t,s) g^2(s)$$

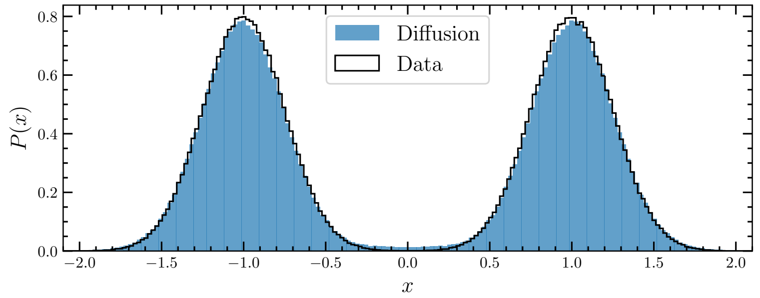
Proof to all orders: cumulants

- cumulant generator $\quad W[J] = \log Z[J] = \frac{1}{2} J^2(t)\Xi(t) + \log \int dx_0\, P_0(x_0) e^{J(t) x_0 f(t,0)}$

- 2nd cumulant $\quad \kappa_2(t) = \frac{d^2 W[J]}{dJ(t)^2}\Big|_{J=0} = \Xi(t) + \mathbb{E}_{P_0}[x_0^2] f^2(t,0)$ ✓

- higher-order cumulants $\quad \kappa_{n>2}(t) = \frac{d^n W[J]}{dJ(t)^n}\Big|_{J=0} = \frac{d^n}{dJ(t)^n} \log \mathbb{E}_{P_0}[e^{J(t) x_0 f(t,0)}]\Big|_{J=0} = \kappa_n(0) f^n(t,0)$ ✓

Toy model: two-peak distribution

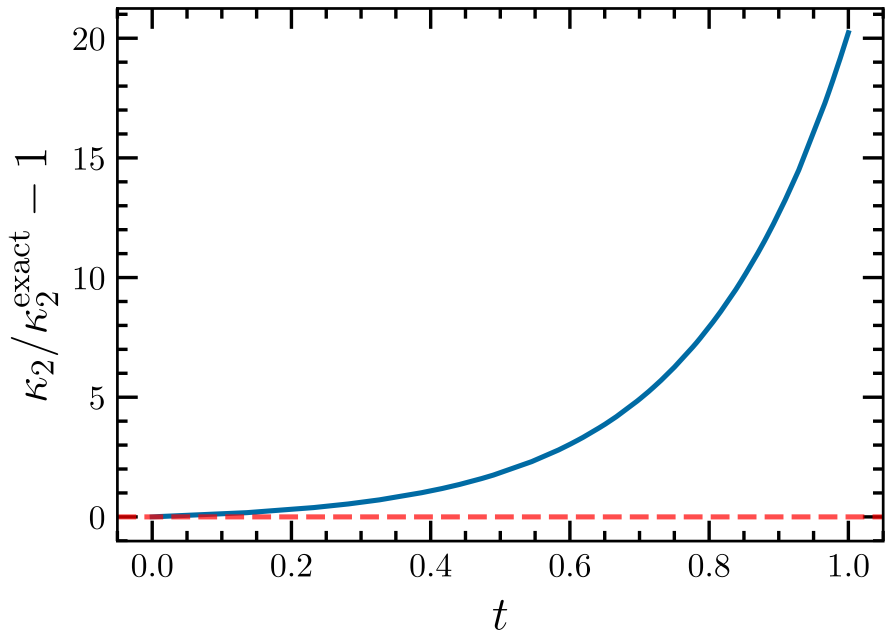


- test the predictions in simple zero-dimensional model
- sum of two Gaussians $P_0(x) = \frac{1}{2}\left[\mathcal{N}(x;\mu_0,\sigma_0^2) + \mathcal{N}(x;-\mu_0,\sigma_0^2)\right]$
- exactly solvable, all even cumulants non-zero, time-dependent score is known analytically
- present second moment and higher-order cumulants

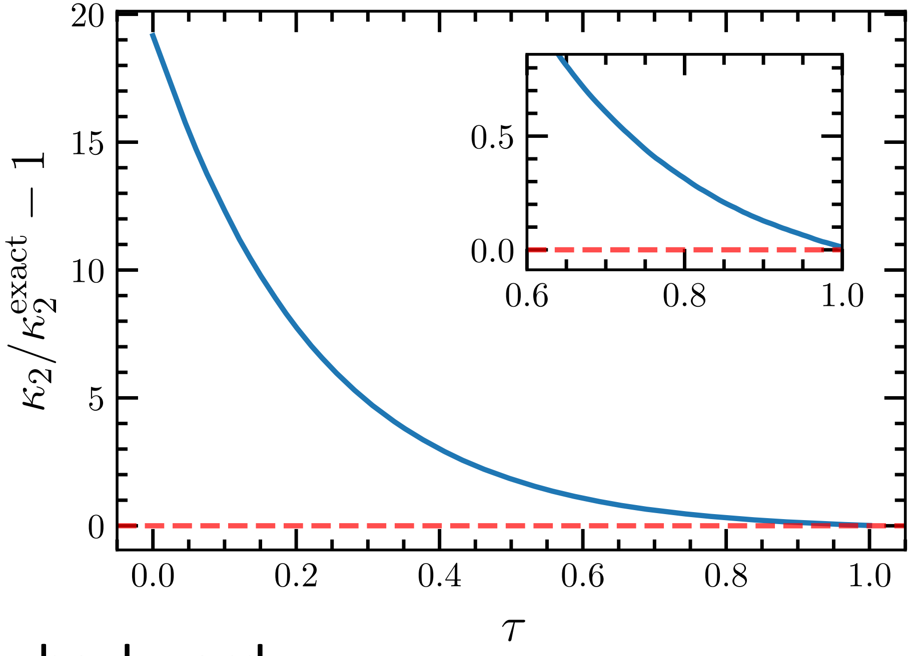
$f(t, s) = 1$

2^{nd} cumulant without drift

- variance-expanding scheme $\kappa_2(t) = \kappa_2(0) + \Xi(t)$ $\Xi(t) = \int_0^t ds\, g^2(s) \sim \sigma^{2t/T}$



forward



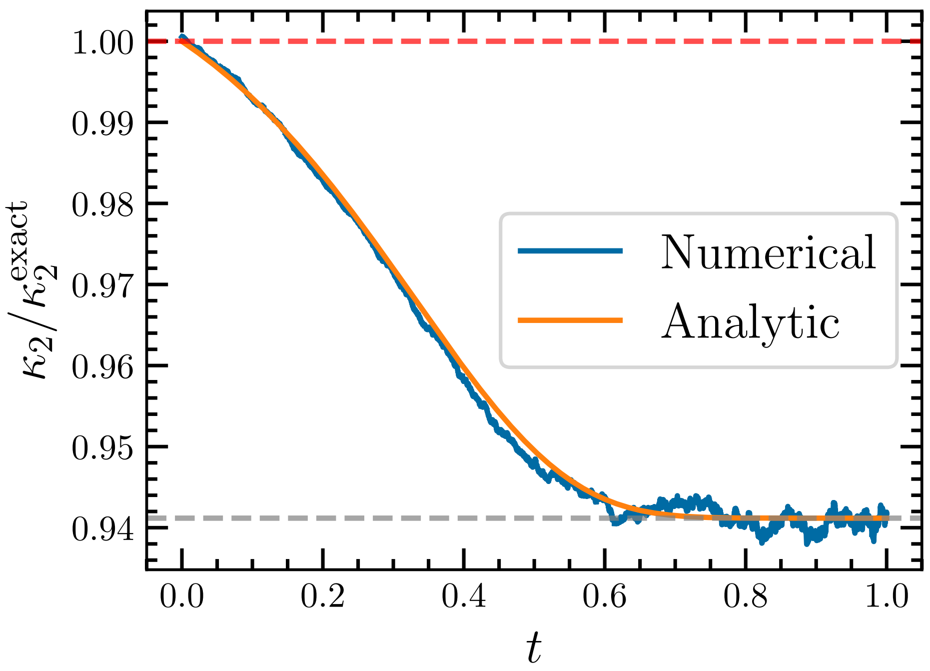
backward

$$f(t,s) = e^{-\frac{1}{2}u(t)+\frac{1}{2}u(s)}$$

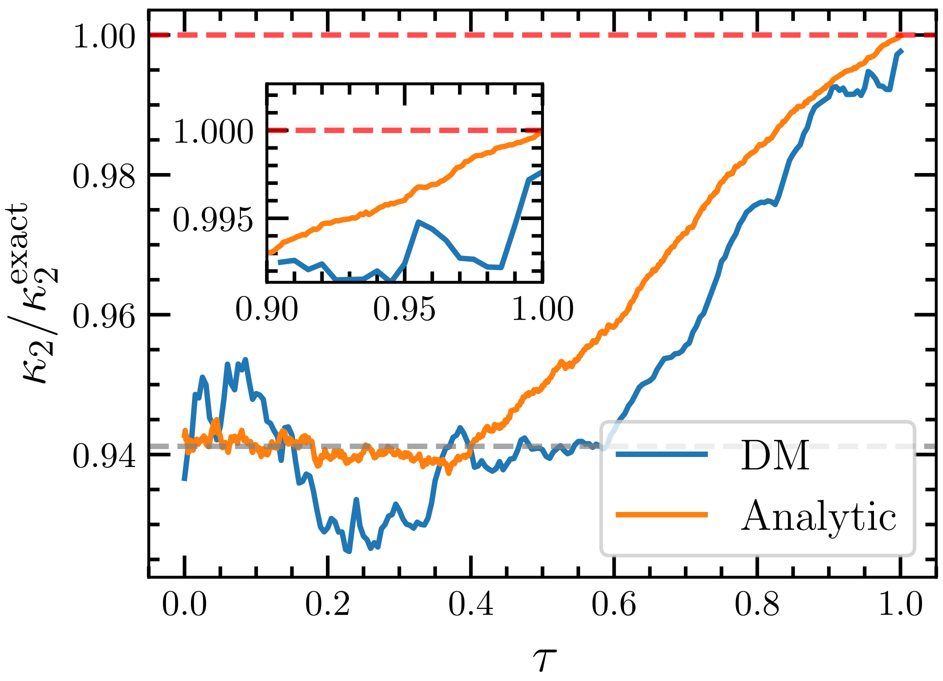
$$u(t) = \int_0^t ds\, g^2(s)$$

2nd cumulant with drift (DDPM)

- variance-preserving scheme $\kappa_2(t) = \mu^2(t) + \sigma^2(t) = \left(\mu_0^2 + \sigma_0^2 - 1\right) f^2(t,0) + 1$



forward



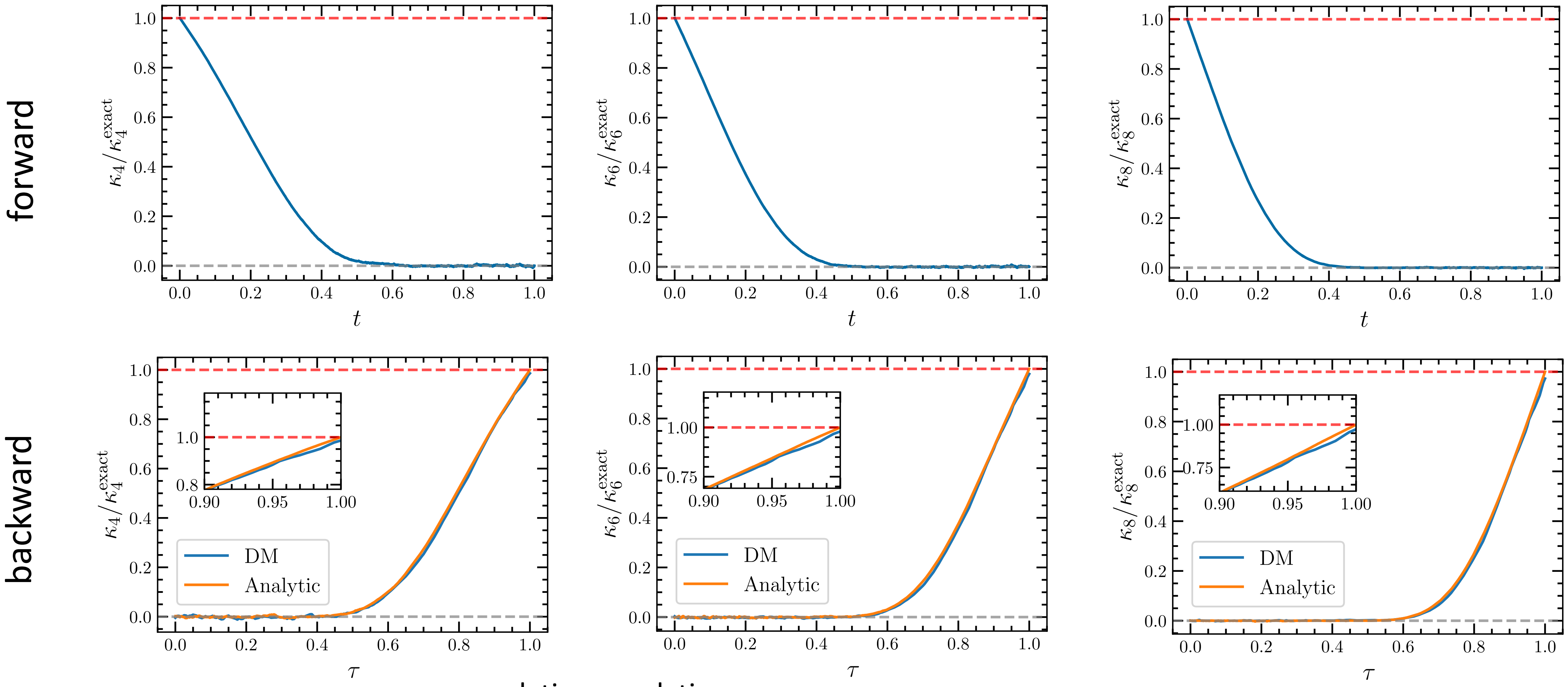
backward

analytic = analytic score

$$\kappa_{n>2}(t) = \kappa_n(0) f^n(t,0)$$

$$f(t,0) \to 0$$

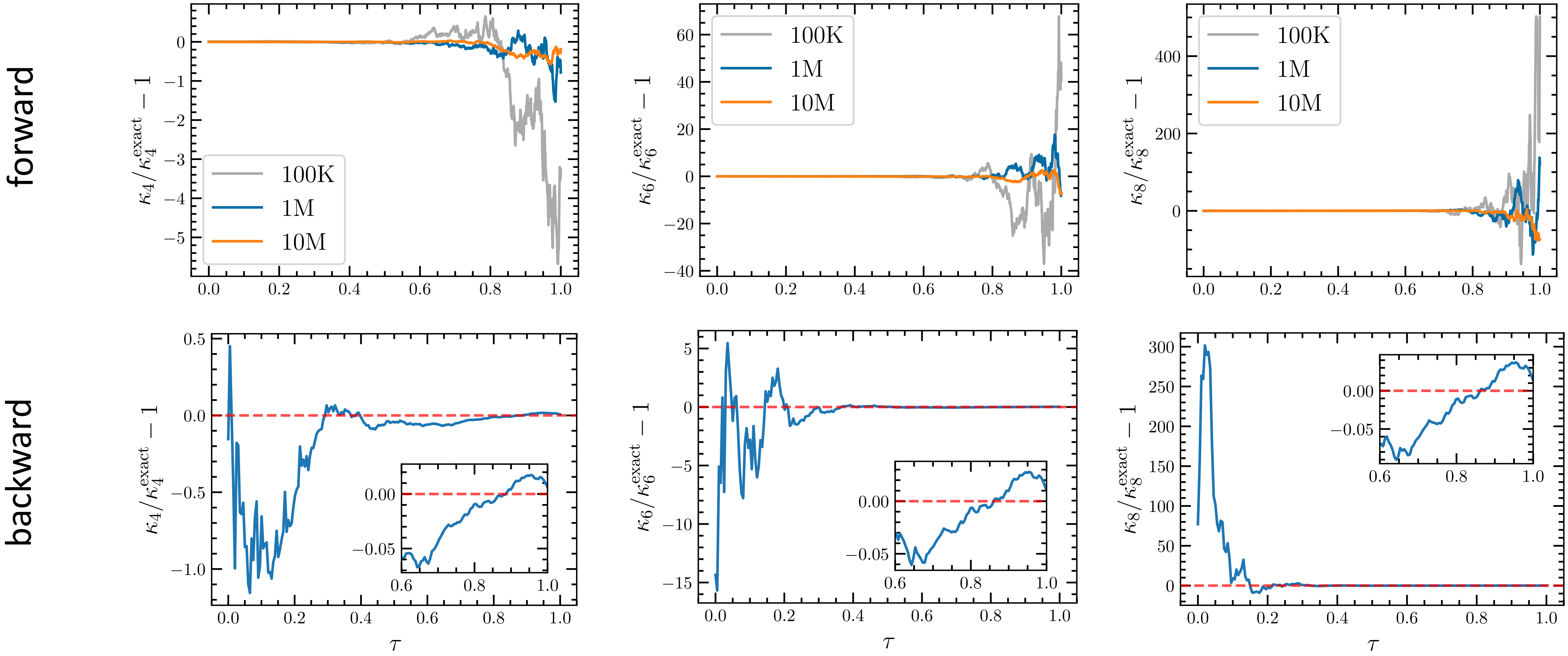
4th, 6th, 8th cumulant with drift (DDPM)



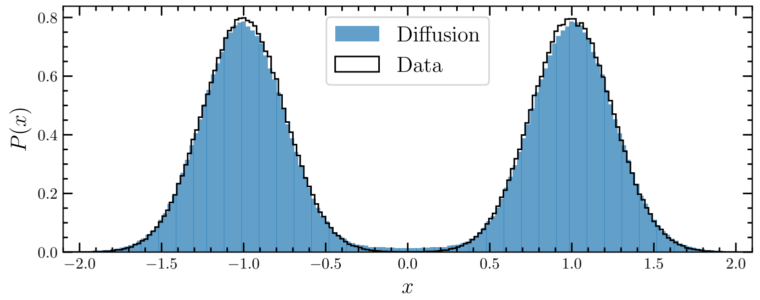
analytic = analytic score

$$\kappa_{n>2}(t) = \kappa_n(0)$$

4th, 6th, 8th cumulant without drift



Comparison between schemes



	κ_2	κ_4	κ_6	κ_8
Exact	1.0625	−2	16	−272
Data	1.0624(5)	−2.000(2)	16.00(2)	−272.0(6)
Variance expanding	1.0692(6)	−2.001(2)	16.03(3)	−272.7(6)
Variance preserving (DDPM)	1.0609(5)	−1.976(2)	15.72(2)	−265.6(6)

expectation values at the end of the backward process

✓ variance-expanding scheme slightly outperforms variance-preserving scheme
✓ can be improved by adapting the noise schedule

Higher-order cumulants

- with drift (DDPM): cumulants go to zero, distribution becomes normal
- without drift (variance-expanding): higher-order cumulants are conserved,
  up to numerical cancellations, required between moments which increase in time
- initial conditions for backward process taken from normal distribution
- score has higher-order cumulants encoded: cumulants are reconstructed

Two-dimensional scalar fields

extension to scalar fields trivial: each lattice point is treated separately

- forward $$\partial_t \phi(x,t) = K[\phi(x,t),t] + g(t)\eta(x,t)$$
- backward $$\partial_\tau \phi(x,\tau) = -K[\phi(x,\tau),T-\tau] + g^2(T-\tau)\nabla_\phi \log P(\phi,T-\tau) + g(T-\tau)\eta(x,\tau)$$
- two-point function $$G(x,y;t) \equiv \mathbb{E}[\phi(x,t)\phi(y,t)] = \mathbb{E}_{P_0}[\phi_0(x)\phi_0(y)]f^2(t,0) + \Xi(t)\delta(x-y)$$
- moments $\mu_n(x,t) = \mathbb{E}[\phi^n(x,t)]$ independent of x

$$\Xi(t) = \int_0^t ds\, f^2(t,s) g^2(s)$$

Generating functionals

full path integral
with sources

- moment generating

$$Z[J] = \mathbb{E}[e^{J(x,t)\phi(x,t)}] = e^{\frac{1}{2}J^2(x,t)\Xi(t)} \int D\phi_0\, P_0[\phi_0] e^{J(x,t)\phi_0(x)f(t,0)}$$

- cumulant generating

$$W[J] = \log Z[J] = \frac{1}{2} J^2(x,t)\Xi(t) + \log \int D\phi_0\, P_0[\phi_0] e^{J(x,t)\phi_0(x)f(t,0)}$$

- higher-order cumulants

$$\kappa_{n>2}(t) = \frac{\delta^n W[J]}{\delta J(x,t)^n}\bigg|_{J=0} = \frac{\delta^n}{\delta J(x,t)^n} \log \mathbb{E}_{P_0}[e^{J(x,t)\phi_0(x)f(t,0)}]\bigg|_{J=0}$$

variance preserving

$$f(t,0) \to 0$$

variance expanding

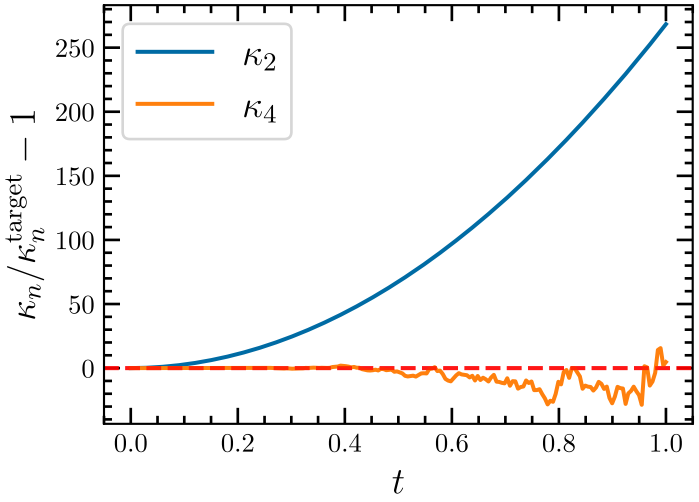
$$f(t,0)) = 1$$

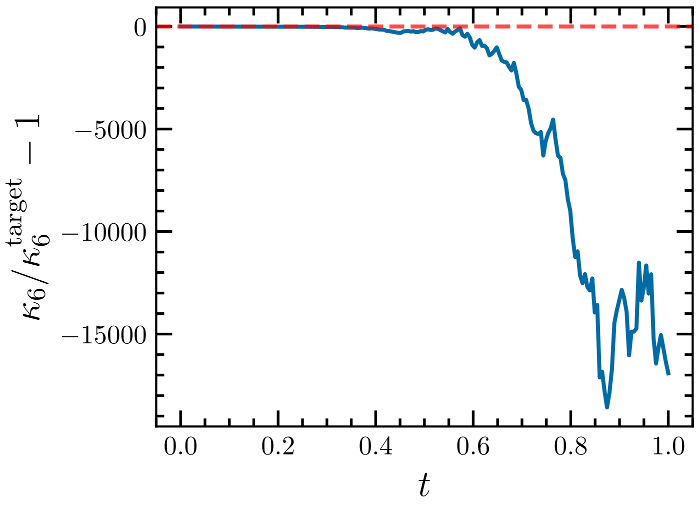
ϕ^4 theory: 2nd, 4th, 6th cumulant without drift

κ_2, κ_4

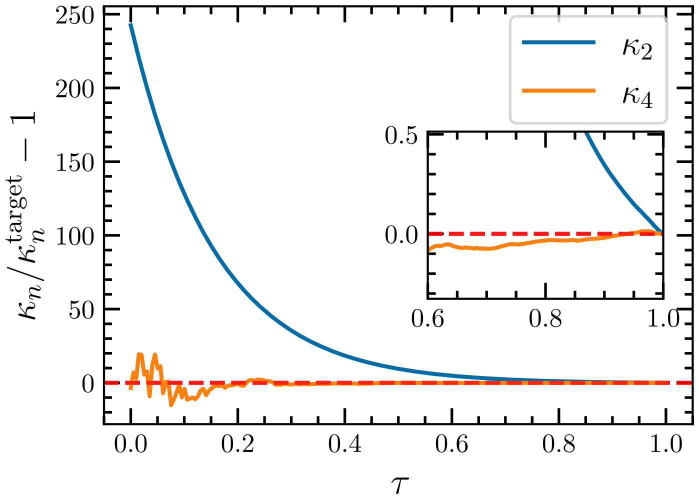
κ_6

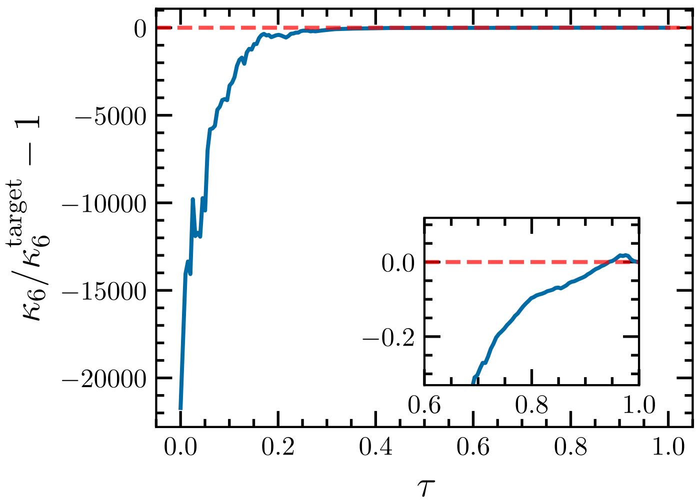
forward





backward





Extensions

- U(1) gauge theory in two dimensions, exactness of algorithm, include accept/reject step 2502.05504 [hep-lat]

- complex actions → first results at *Lattice* conference, in progress, 2412.01919 [hep-lat]

- fermionic models (Gross-Neveu model, Schwinger model) → in progress

Incorporate (new/old) ideas in diffusion models

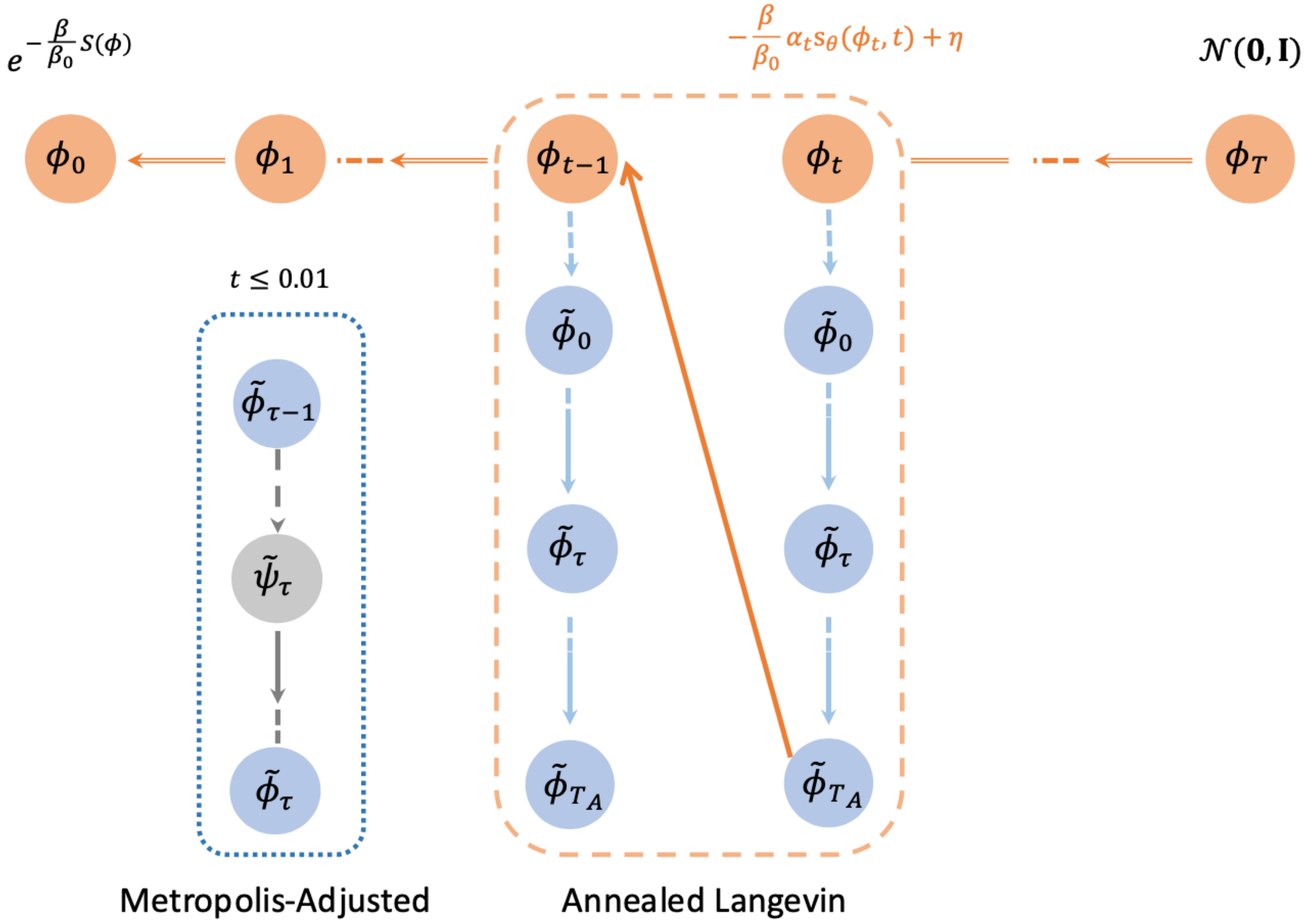
- exactness → include an accept/reject step
- thermalisation: score is time dependent, system never thermalises → annealing
- train at one set of parameters, apply trained model at different set → conditioning
- apply to 2D U(1) gauge theory

Qianteng Zhu, W Wang, GA, K Zhou, L Wang, 2502.05504 [hep-lat]

Incorporate (new/old) ideas in DM dynamics

backward process
(after model has been trained)

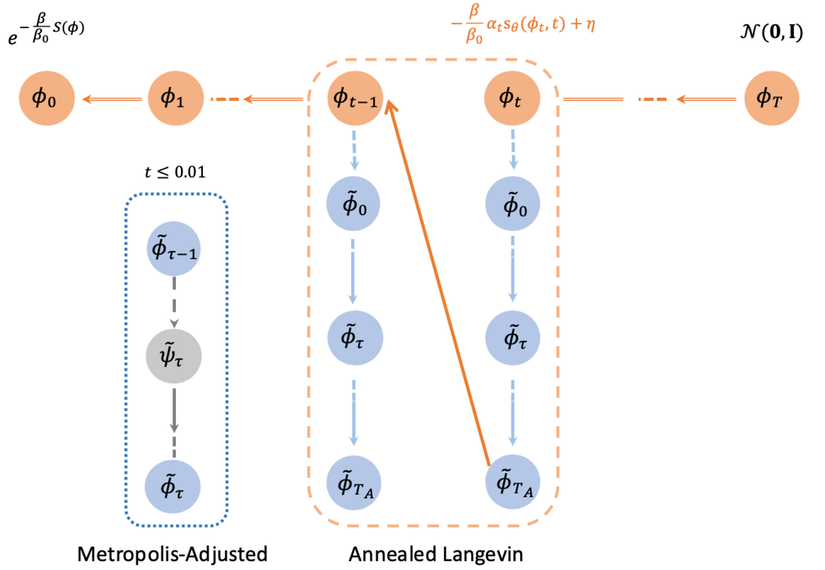
- Metropolis-adjusted Langevin algorithm (MALA)
- annealing stage: thermalisation
- reweighting from β_0 to β



Metropolis-adjusted Langevin algorithm (MALA)

- include an accept/reject step: well-known in Langevin dynamics *

$$\phi_{\tau+1} = \begin{cases} \psi_{\tau+1} & \text{with probability } \min\left\{1, \frac{p(\psi_{\tau+1})q(\phi_\tau|\psi_{\tau+1})}{p(\phi_\tau)q(\psi_{\tau+1}|\phi_\tau)}\right\} \\ \phi_\tau & \text{with the remaining probability,} \end{cases}$$



- include ratio of target distributions $p(\phi) \sim e^{-S(\phi)}$
- and ratios of transition amplitudes

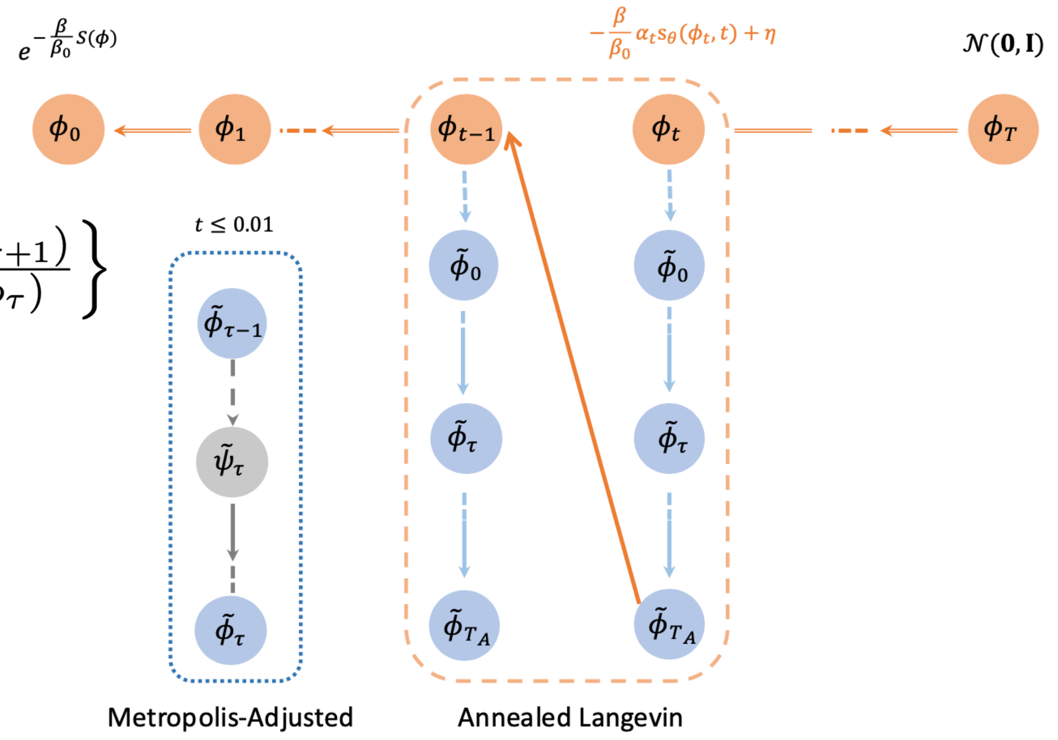
$$q(\phi_\tau|\psi_{\tau+1}) = \frac{1}{(4\pi\alpha_i)^{n/2}} \exp\left(-\frac{1}{4\alpha_i}\|\phi_\tau - (\psi_{\tau+1} + \alpha_i f(\psi_{\tau+1}, \tau+1))\|_2^2\right)$$

* G.O. Roberts and J.S. Rosenthal, Optimal scaling of discrete approximations to Langevin diffusions, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60 (1998) 255

Metropolis-adjusted Langevin algorithm (MALA)

- include an accept/reject step

$$\phi_{\tau+1} = \begin{cases} \psi_{\tau+1} & \text{with probability } \min\left\{1, \dfrac{p(\psi_{\tau+1})q(\phi_\tau|\psi_{\tau+1})}{p(\phi_\tau)q(\psi_{\tau+1}|\phi_\tau)}\right\} \\ \phi_\tau & \text{with the remaining probability,} \end{cases}$$



- only done towards end of backward process

- learned score should be fairly close to “exact” score

$$\nabla \log p(\phi) \qquad\qquad p(\phi) \sim e^{-S(\phi)}$$

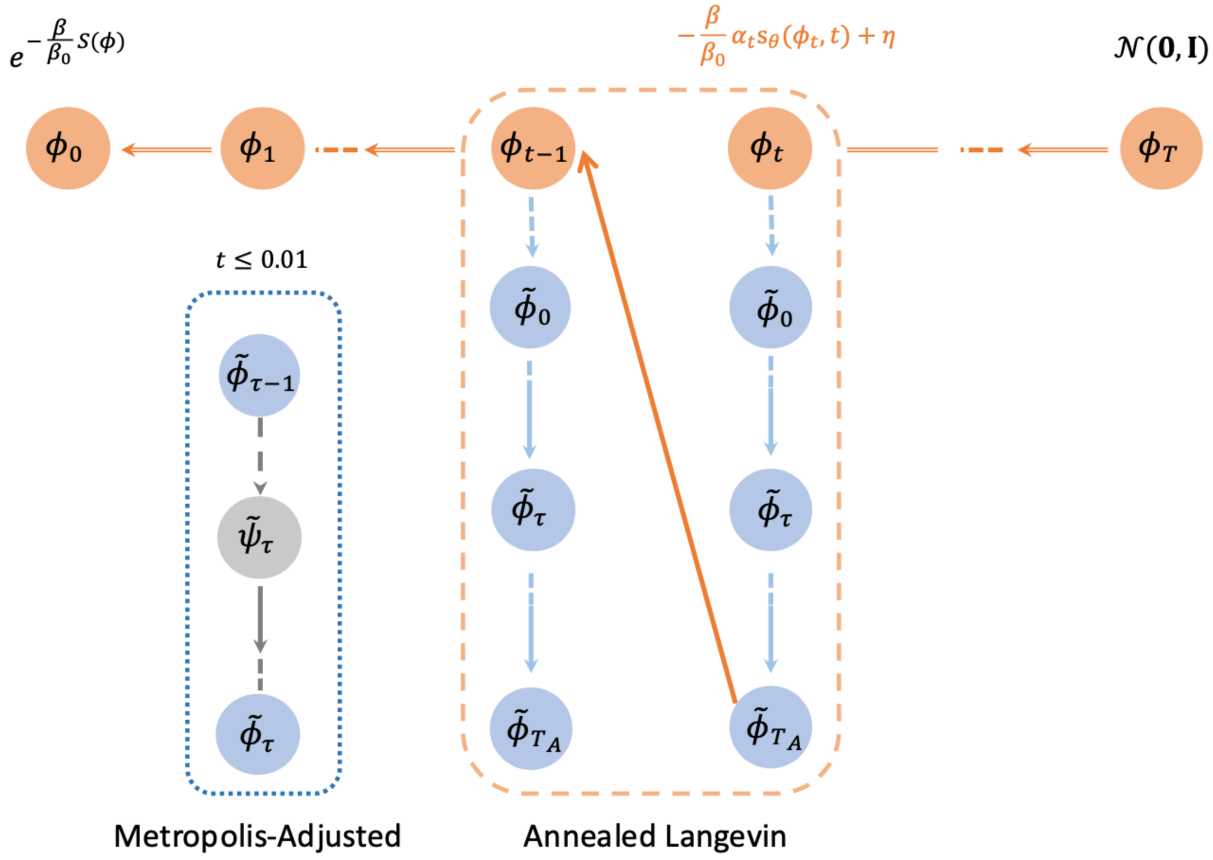
- Markov chain starting from each configuration towards end of backward process

Annealing

- score (drift or force in Langevin equation) is time dependent
- system never thermalises
- allow for additional steps at fixed score

→ annealing

- strictly speaking not needed, but seems useful

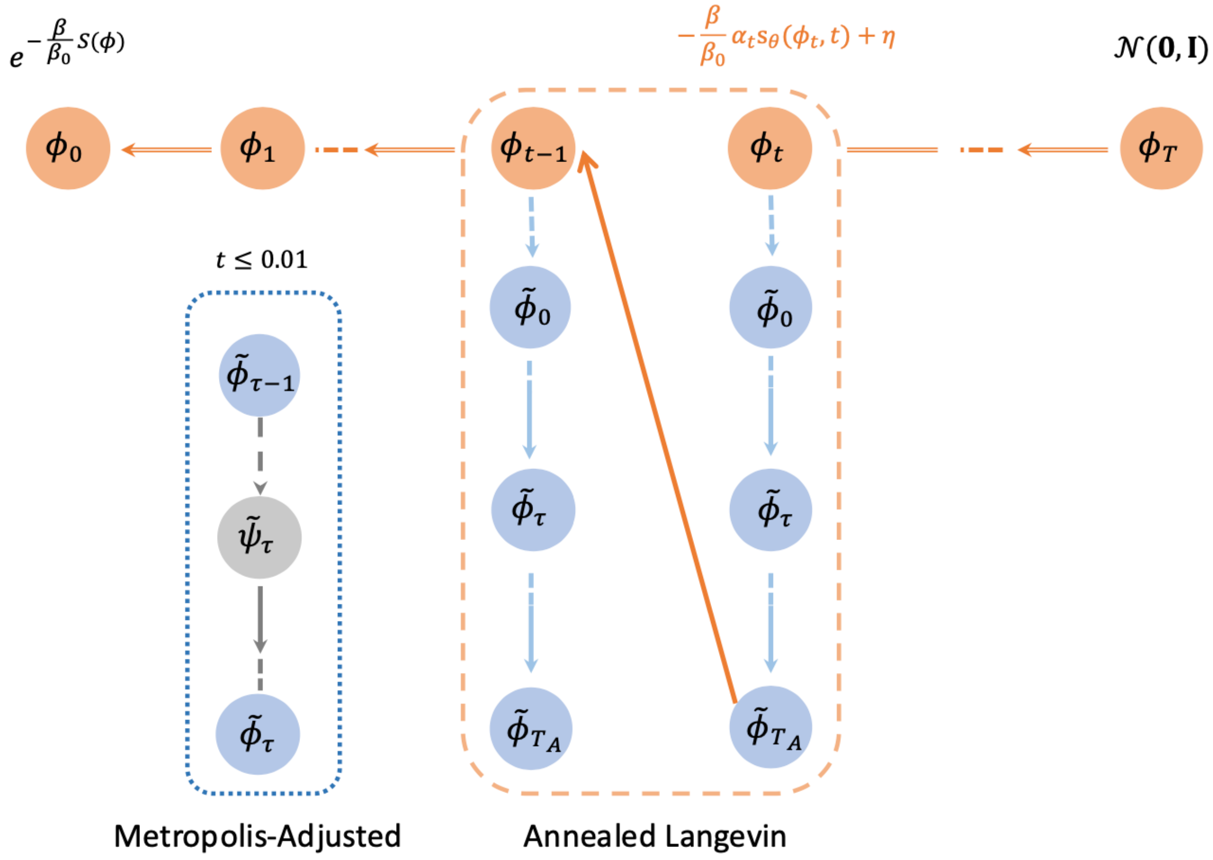


Physics conditioning (gauge theory)

- train using data generated at β_0
- employ at different β values
- applied to U(1) gauge theory:
  action scales with β

motivated by stochastic quantisation:

- drift is proportional to β



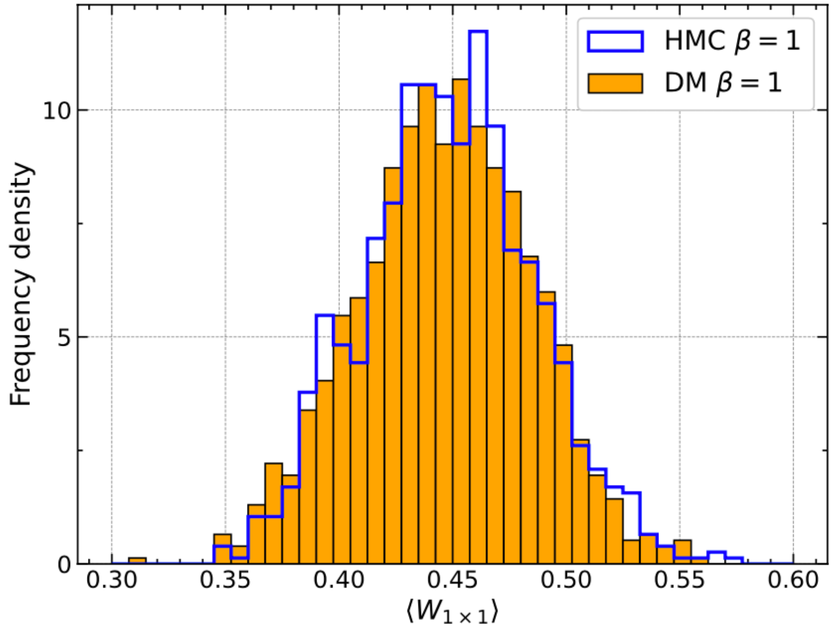
Two-dimensional U(1) gauge theory

- training: 30k configurations at $\beta = 1$ on 16^2 obtained using HMC
- generating: 1024 configs at $\beta = 1, 3, 5, 7, 9, 11$ on $8^2, 16^2, 32^2$

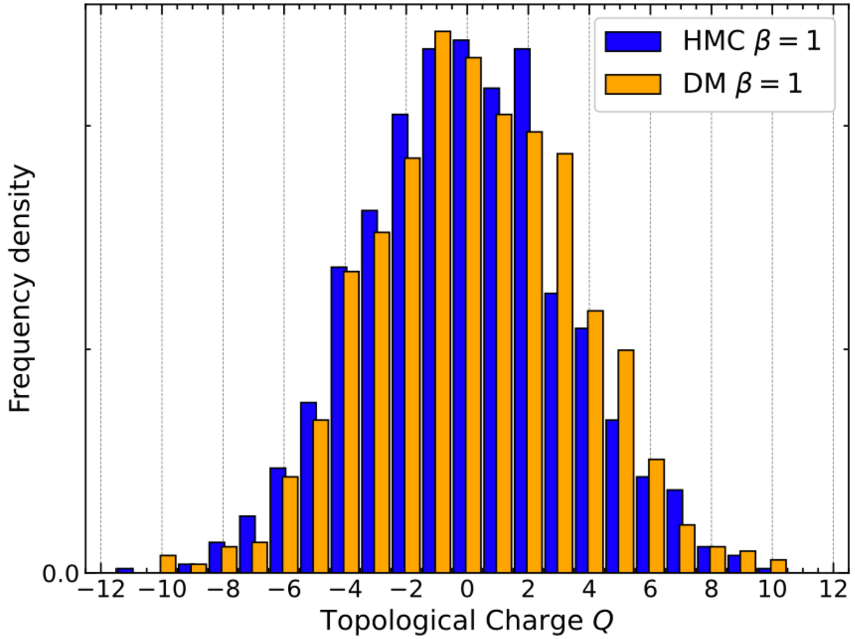
Two-dimensional U(1) gauge theory

- training: 30k configurations at $\beta = 1$ on 16^2 obtained using HMC
- generating: 1024 configs at $\beta = 1, 3, 5, 7, 9, 11$ on $8^2, 16^2, 32^2$

$\beta = 1$
$L = 16$



1x1 Wilson loop



topological charge

see paper for many more details and open questions

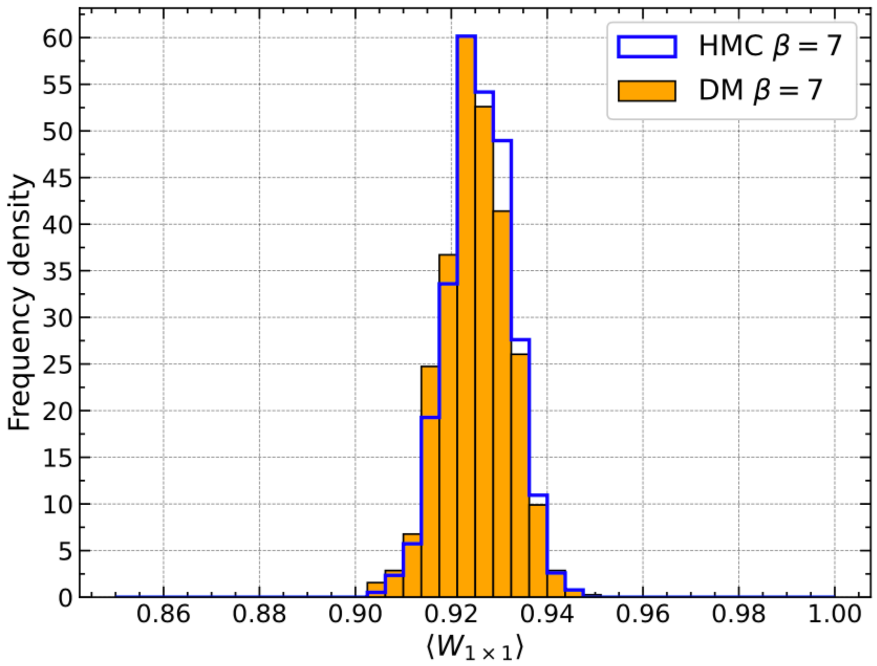
Two-dimensional U(1) gauge theory

- training: 30k configurations at $\beta = 1$ on 16^2 obtained using HMC
- generating: 1024 configs at at $\beta = 1, 3, 5, 7, 9, 11$ on $8^2, 16^2, 32^2$

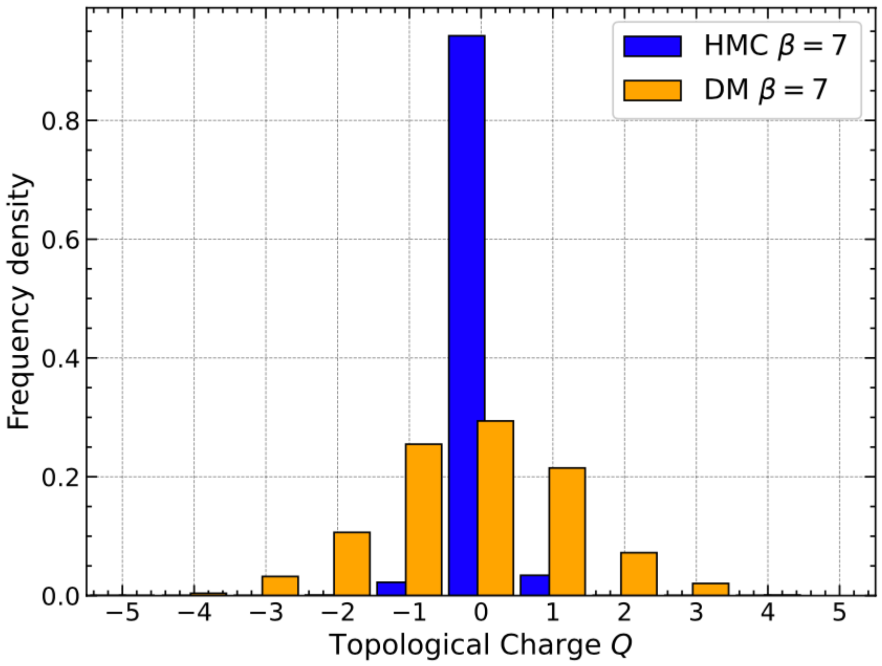
$\beta = 7$
$L = 16$

diffusion model trained at $\beta = 1$ but employed at $\beta = 7$

HMC suffers from topological freezing



1x1 Wilson loop



topological charge

Summary lecture I: diffusion models

- diffusion models offer a new approach for ensemble generation to explore in LFT
- learn from data: requires high-quality ensembles

- closely related to stochastic quantisation
- moment- and cumulant-generating functionals
- higher n-point functions important in LFT applications

- include accept/reject step, annealing
- physics conditioning: train on one ensemble, apply to different couplings, lattice volumes, …

- in progress: application to theories with fermions, gauge theories, complex actions, …