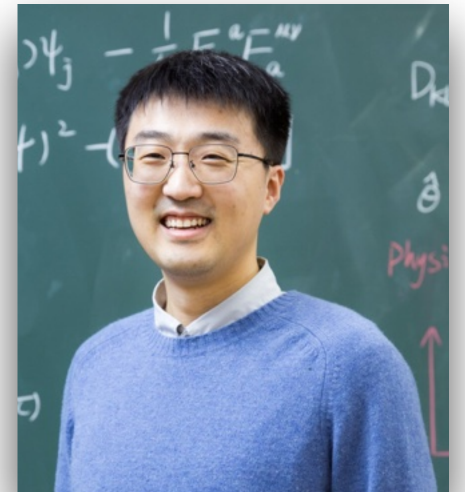
Diffusion Models as Stochastic Quantization on the Lattice

Lingxiao Wang(王凌霄)
AI as Science team, RIKEN-iTHEMS
Institute for Physics of Intelligence(iπ), UTokyo

On behalf of DM-QFT collaboration

DLAP Seminar, **2026-06-11**

Physics for AI, AI for Physics



Lingxiao Wang

Deputy Director of AI as Science Team
RIKEN-iTHEMS

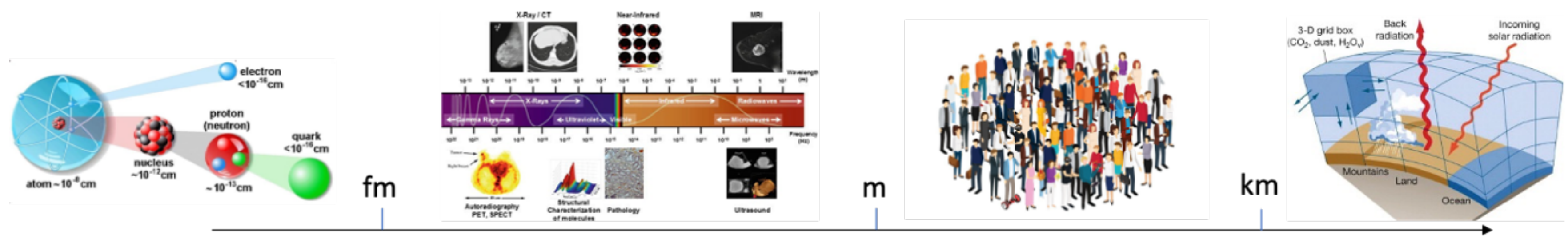
Assistant Professor
UTokyo

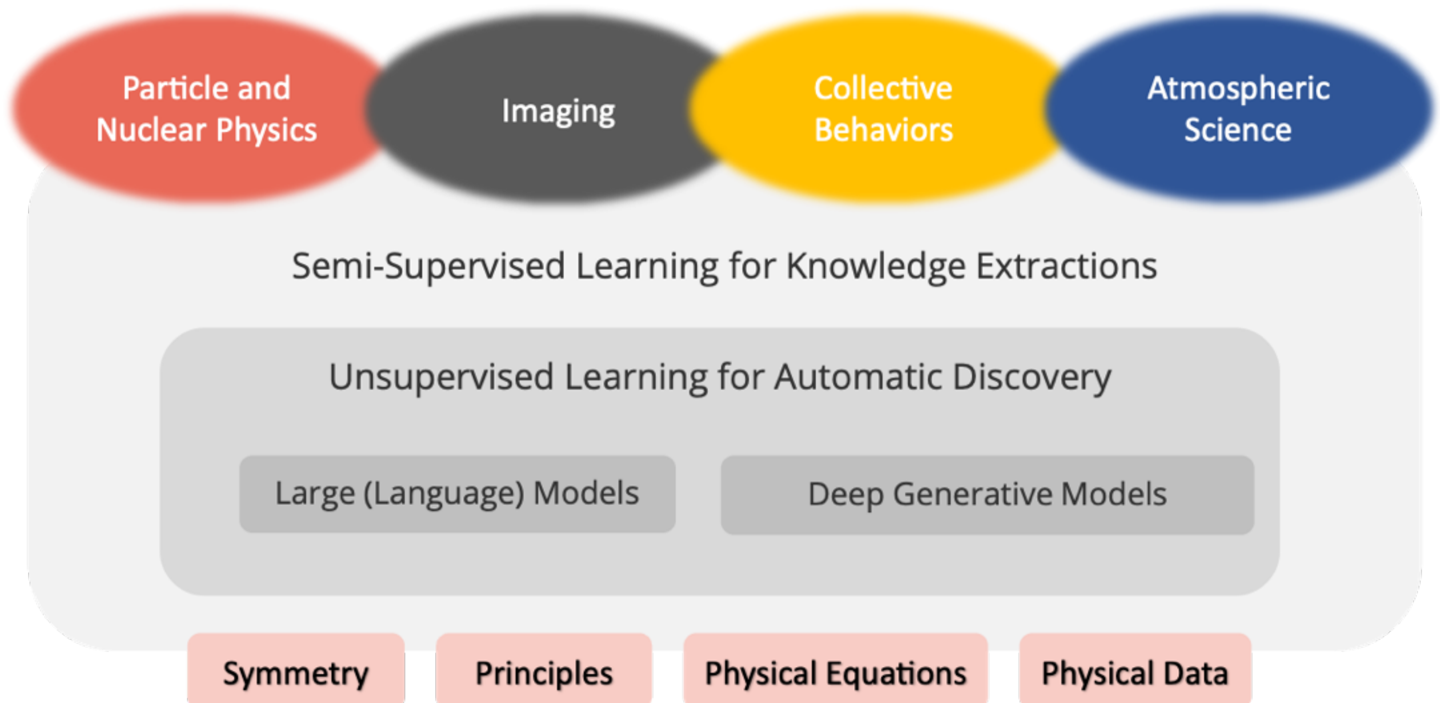
Background: Particle and Nuclear Physics, Machine Learning, AI for Science

PhD in Physics, 2020
Tsinghua University

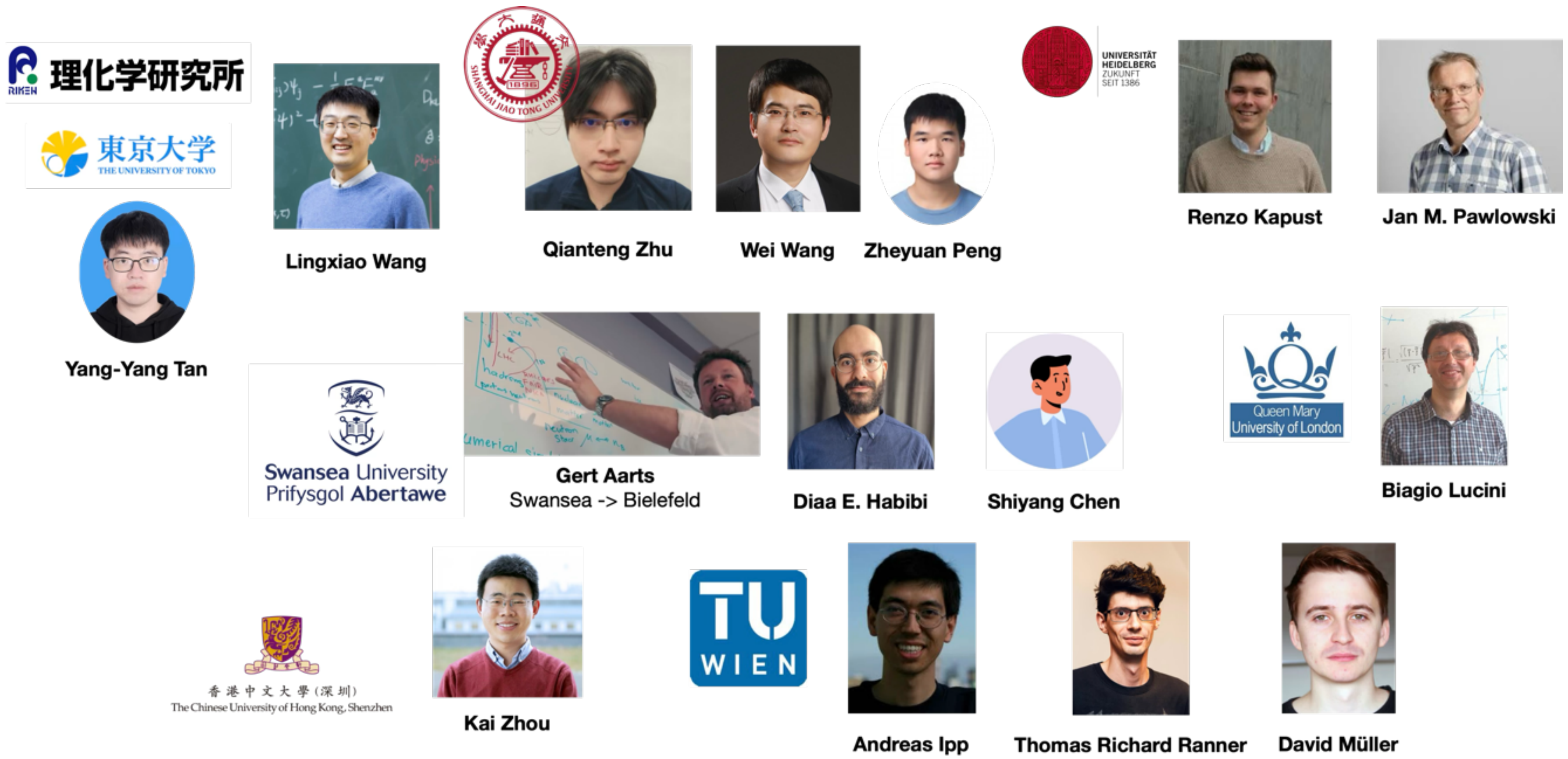
Research Interests

- Inverse Problems in Quantum Chromodynamics(QCD) Physics
- **Stochastic Qauntization** and **Generative Models**
- Generative Models for **Lattice Field Theory**
- Phase Transitions and Critical Behaviors





Background and References



DM-QFT

- **LW**, G Aarts, K Zhou, *JHEP 05 (2024) 060* [2309.17082 [hep-lat]]
- G Aarts, D Habibi, **LW***, K Zhou, *Mach.Learn.Sci.Tech. 6 (2025) 2, 025004* and NeurIPS 2024 ML4PS [2410.21212 [hep-lat]] (Best ‘Physics for AI’ Paper)
- G Aarts, K Fukushima, T Hatsuda, A Ipp, S Shi, **LW***, K Zhou, *Nature Rev.Phys. 7 (2025) 3, 154-163*, [2501.05580 [hep-lat]]
- Q Zhu, G Aarts, W Wang, K Zhou, **LW***, *JHEP03(2026)111* [2502.05504 [hep-lat]] and NeurIPS 2024 ML4PS [2410.19602 [hep-lat]]
- G Aarts, D Habibi, A Ipp, D, Müller, T Ranner, **LW***, W Wang, Q Zhu [2601.19552 [hep-lat]]

Outline

What ‣ **What is Diffusion Model?**

‣ **Physics of Diffusion Models**

‣ Stochastic Quantization

‣ Effective Actions

Why ‣ **DM on the Lattice**

‣ **Generation of Lattice Fields**

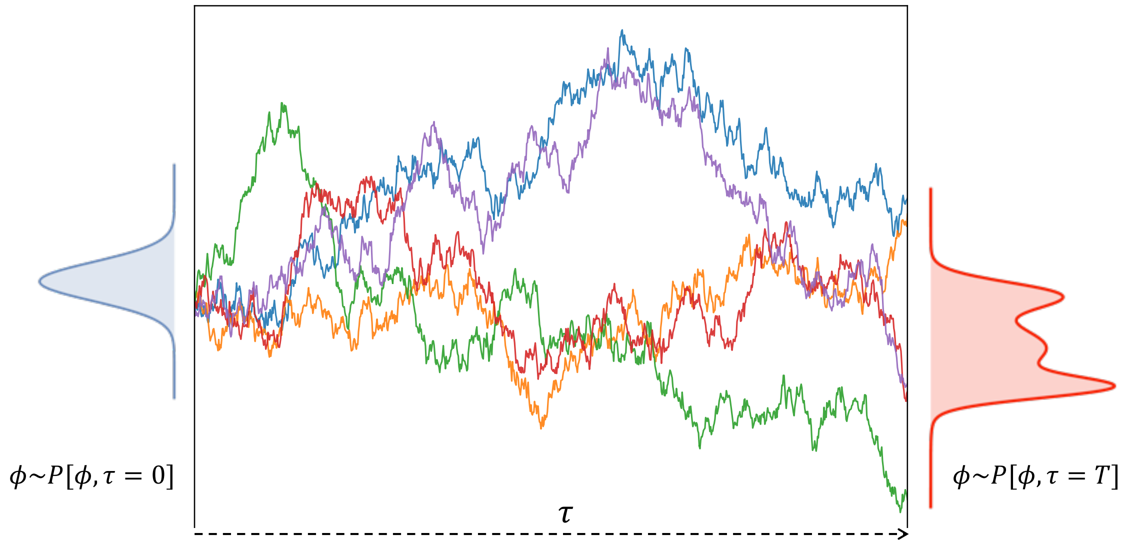
How ‣ **Expandability, Exactness, Effciency**

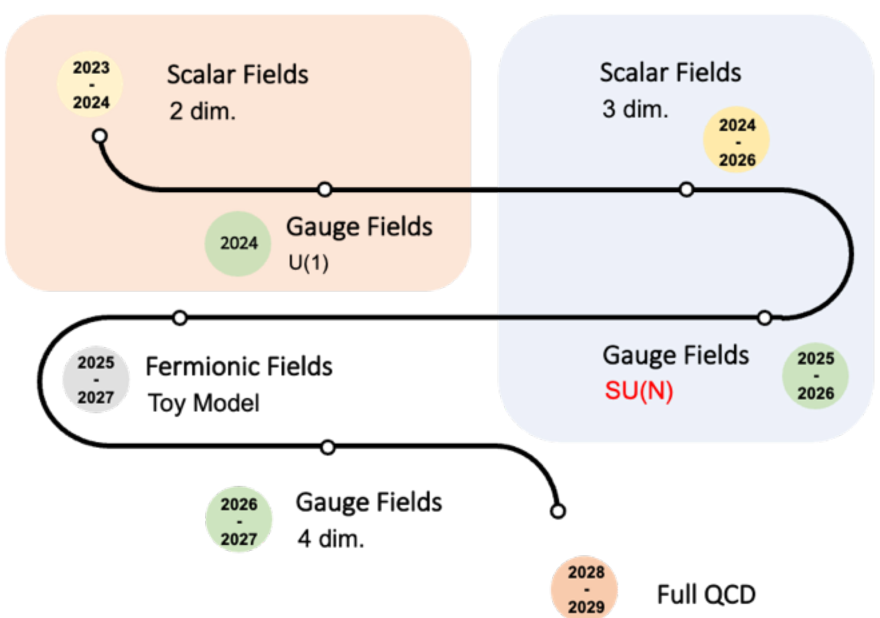
‣ Scalar Fields

‣ Gauge Fields

New ‣ **Optimal Stochastic Quantization**

‣ **Summary and Outlook**

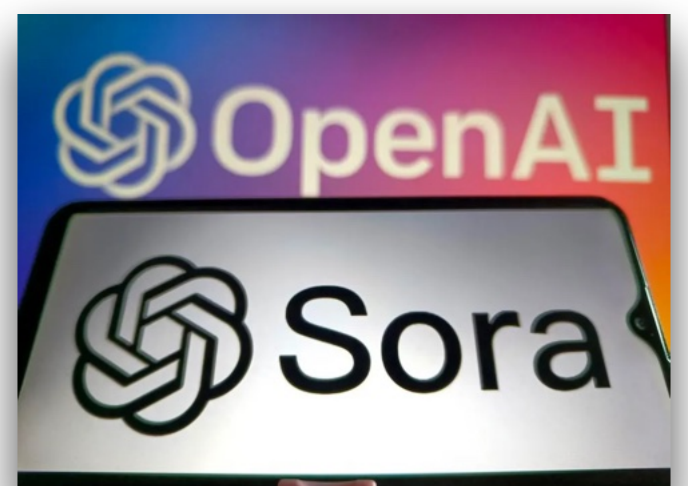




What is Diffusion Model?

A Diffusion Model is a type of **generative model** used in machine learning to **create (or generate) data** — such as images, audio, or text — by **gradually transforming random noise into structured data**.

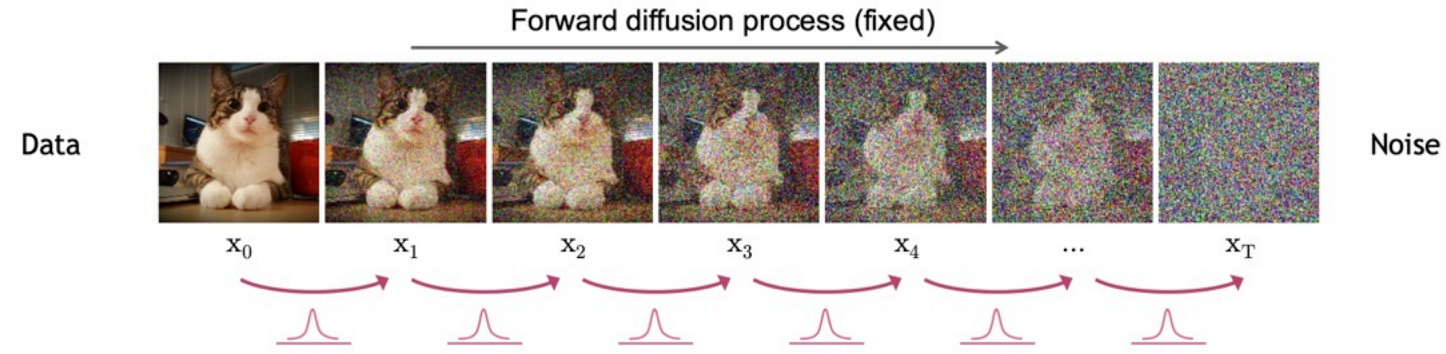
— GPT-5



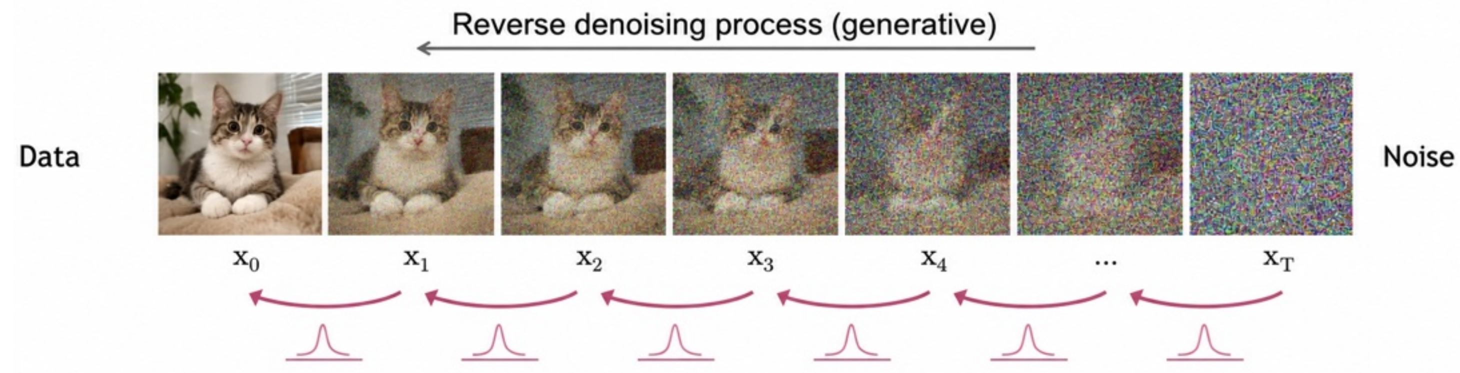
Diffusion Models

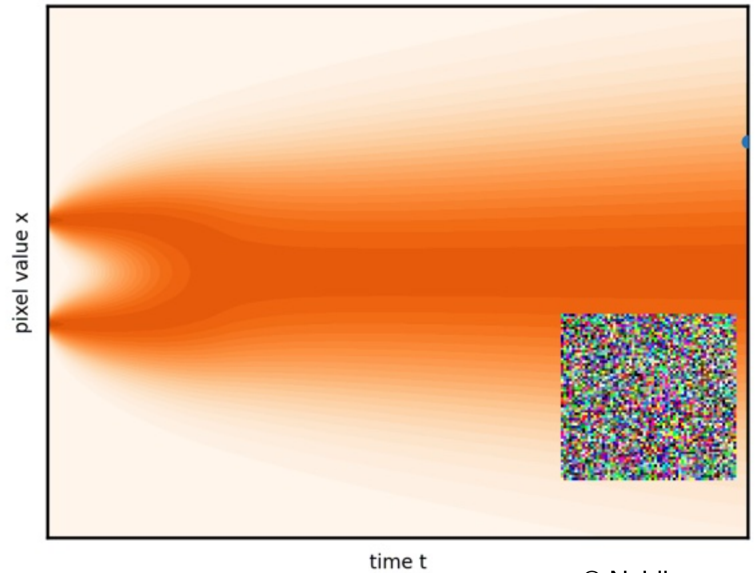
Practical Level

- Forward diffusion process **gradually adds noise** to input
- Reverse denoising process learns to **generate data by denoising**



Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020





© Nvidia

Probabilistic models
To learn how to **denoise**
from a simple distribution
to a target distribution

Diffusion Models

Reverse Diffusion

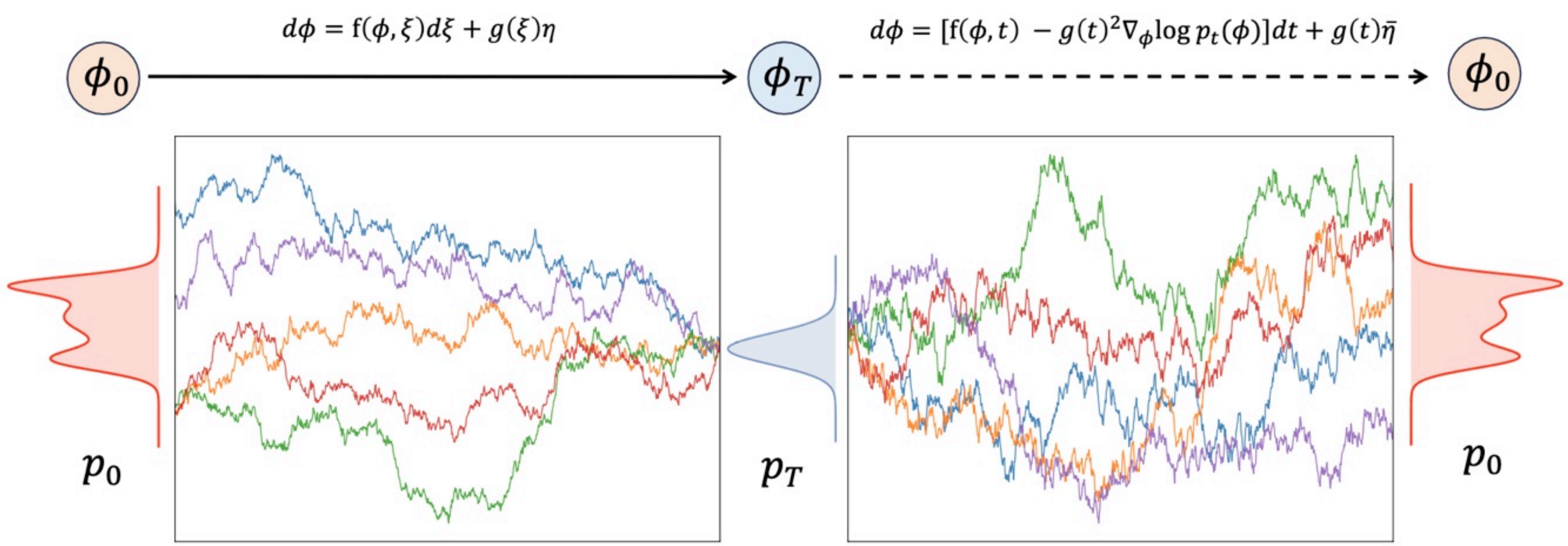
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Anderson, in Stochastic Processes and their Applications, 1982

- **Forward Diffusion process is driven by a stochastic differential equation(SDE)**
  - **Drift term**: pulls towards mode
  - **Diffusion term**: injects noise
- **Reverse Diffusion SDE**
  - Drift term is adjusted with a "**Score Function**"
  - But how to get the score function ?

$$\frac{d\phi}{d\xi} = f(\phi, \xi) + g(\xi)\eta(\xi)$$

$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\nabla_\phi \log p_t(\phi)\right] + g(t)\bar{\eta}(t)$$



Diffusion Models

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Anderson, in Stochastic Processes and their Applications, 1982

Reverse Diffusion

- **Reverse Generative Diffusion SDE**
  - Drift term is adjusted with a "**Score Function**"
  - But how to get the score function ?

$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\,\nabla_{\phi}\log p_t(\phi)\right] + g(t)\bar{\eta}(t)$$

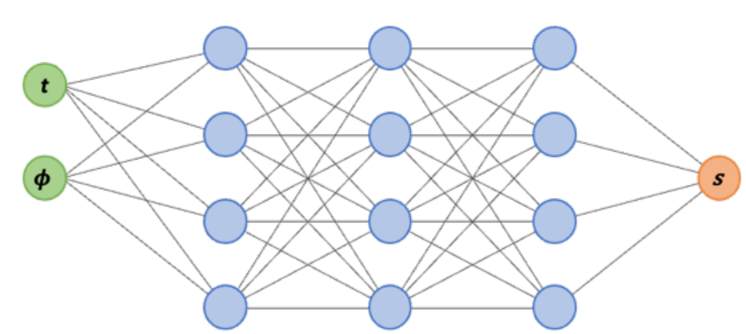
$$\phi_0 \sim p_{\text{data}} \xrightarrow{\phi_t \sim p_{0t}(\phi_t|\phi_0)} \phi_t \xrightarrow{\textbf{match conditional score}} s_\theta(\phi_t, t) \approx \nabla_{\phi_t}\log p_{0t}(\phi_t\,|\,\phi_0)$$

Vincent, Pascal. Neural computation 23.7 (2011): 1661-1674.

$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\mathbf{s}_{\hat{\theta}}(\phi, t)\right] + g(t)\bar{\eta}(t)$$

Universal Approximation Theorem (1989,1991)

A feed-forward network with a single hidden layer containing a finite number of neurons can approximate arbitrary continuous functions.



Stochastic Quantization

$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)} + \eta(x,\tau)$$

$$\langle \eta(x,\tau) \rangle = 0, \quad \langle \eta(x,\tau)\eta(x',\tau') \rangle = 2\alpha\delta(x-x')\delta(\tau-\tau')$$

τ: **fictitious** time, α: diffusion constant

Parisi & Wu (1980); Damgaard & Hüffel (1987); Namiki (1993)

- **Fokker-Planck equation**

$$\frac{\partial P[\phi,\tau]}{\partial \tau} = \alpha \int d^n x \left\{ \frac{\delta}{\delta \phi} \left(\frac{\delta}{\delta \phi} + \frac{\delta S_E}{\delta \phi} \right) \right\} P[\phi,\tau]$$

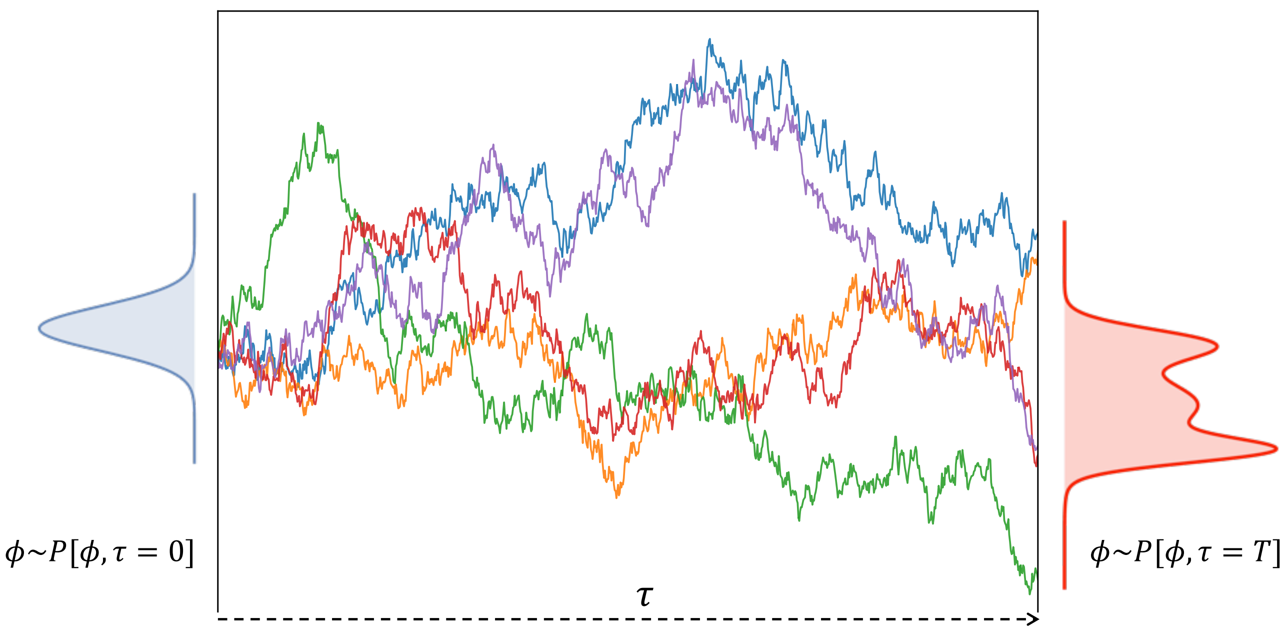
Equilibrium solution (long-time limit),

$$P_{\text{eq}}[\phi] \propto e^{-\frac{1}{\alpha} S_E[\phi]}$$

- **Quantum Diffusion with** $\alpha = \hbar$

$$P_{\text{eq}}[\phi] \sim e^{-\frac{1}{\hbar} S_E[\phi]} = P_{\text{quantum}}[\phi]$$

Thermal Equilibrium Limit
→ Quantum Distribution



Trajectories

1. No need gauge-fixing!
2. Can handle Fermion fields naturally
→(Complex Langevin Method)

Recent review: Gert Aarts, Dénes Sexty, 2604.24290 [hep-lat]

Stochastic Quantization

Reverse Diffusion

- **Diffusion Models (w/o drift term)**

$$\frac{d\phi}{dt} = -g(t)^2 \nabla_\phi \log p_t(\phi) + g(t)\bar{\eta}$$

  - **Define:** $\tau \equiv T - t (d\tau \equiv -dt)$

$$\frac{d\phi}{d\tau} = g_\tau^2 \nabla_\phi \log q_\tau(\phi) + g_\tau \bar{\eta}$$

introducing **noise scale:** $\langle \bar{\eta}^2 \rangle \equiv 2\bar{\alpha}$, **time scale:** $g_\tau^2 \Delta\tau$

- **FP equation**

$$\frac{\partial p_\tau(\phi)}{\partial \tau} = \int d^n x \left\{ g_\tau^2 \bar{\alpha} \frac{\delta}{\delta\phi} \left(\frac{\delta}{\delta\phi} + \frac{1}{\bar{\alpha}} \nabla_\phi S_{\mathbf{DM}} \right) \right\} p_\tau(\phi)$$

$$\nabla_\phi S_{\mathbf{DM}} \equiv -\bar{\alpha} \nabla_\phi \log q_\tau(\phi)$$

Reverse Diffusion		Stochastic Quantization
$\tau = T - t$	time variable	fictitious time τ
$q_\tau(\phi) = p_{T-\tau}(\phi)$	distribution	$P[\phi, \tau]$
$S_{\mathrm{DM}}(\phi, \tau) = -\bar{\alpha} \log q_\tau(\phi)$	effective action	$S_E[\phi]$
$\frac{d\phi}{d\tau} = g_\tau^2 \nabla_\phi \log q_\tau(\phi) + g_\tau \bar{\eta}(\tau)$	Langevin form	$\frac{d\phi}{d\tau} = -\frac{g_\tau^2}{\bar{\alpha}} \nabla_\phi S_{\mathbf{DM}} + g_\tau \bar{\eta}(\tau)$
$q_T(\phi) \simeq p_{\mathrm{data}}(\phi)$	equilibrium	$P_{\mathbf{eq}}[\phi] \propto e^{-S_{\mathrm{DM}}/\bar{\alpha}}$

$$p_{\tau=T}(\phi) \to P[\phi, T]$$

$$p_{eq}(\phi) \propto e^{-\frac{S_{DM}}{\bar{\alpha}}}$$

$$O(\bar{\alpha}) \sim O(\hbar) \sim 1$$

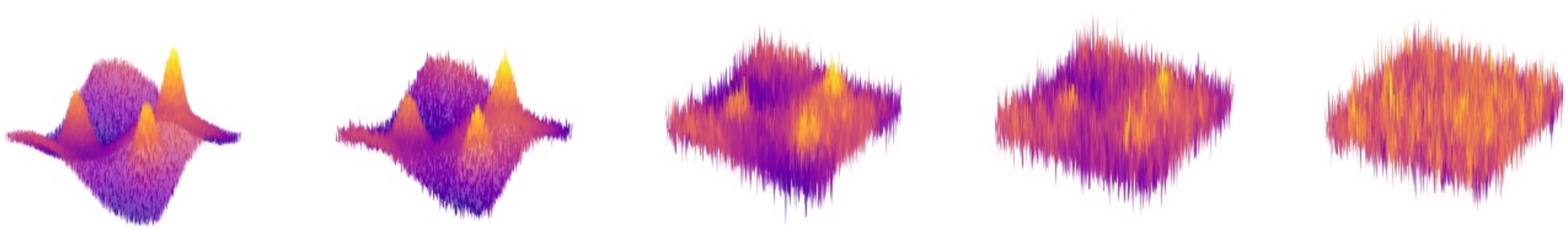
Reverse diffusion realizes **stochastic quantization** toward the data distribution.

LW, G Aarts, K Zhou, JHEP(2024)

DM as SQ

Aspect	Stochastic Quantization (SQ)	Diffusion Models (DM)
Drift term	Fixed, from known action	Time-dependent, learned from data
Noise variance	Constant (or kernel-generalized)	Time-dependent
Dynamics	Thermalization + long-time equilibrium	Finite-time evolution, $0 \leq \tau \leq T$, many short runs
Learning objective	Based on known physical action $S(\phi)$	Learn probability evolution $P(\phi;\tau)$ from data
Equation	$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\nabla_\phi S(\phi) + \sqrt{2}\,\eta(x,\tau)$	$\frac{\partial \phi(x,\tau)}{\partial \tau} = g^2(\tau)\nabla_\phi \log P(\phi;\tau) + g(\tau)\eta(x,\tau)$

Takeaway: The reverse process of a **well-trained diffusion model,** near $\tau \to T$, can be viewed as **stochastic quantization** toward the input/data distribution.



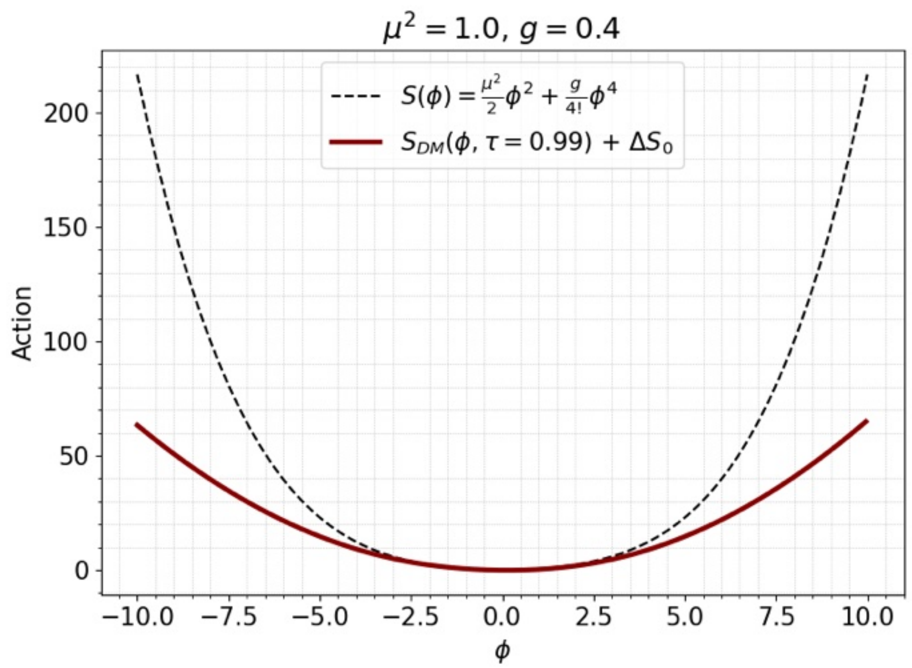
Effective Action

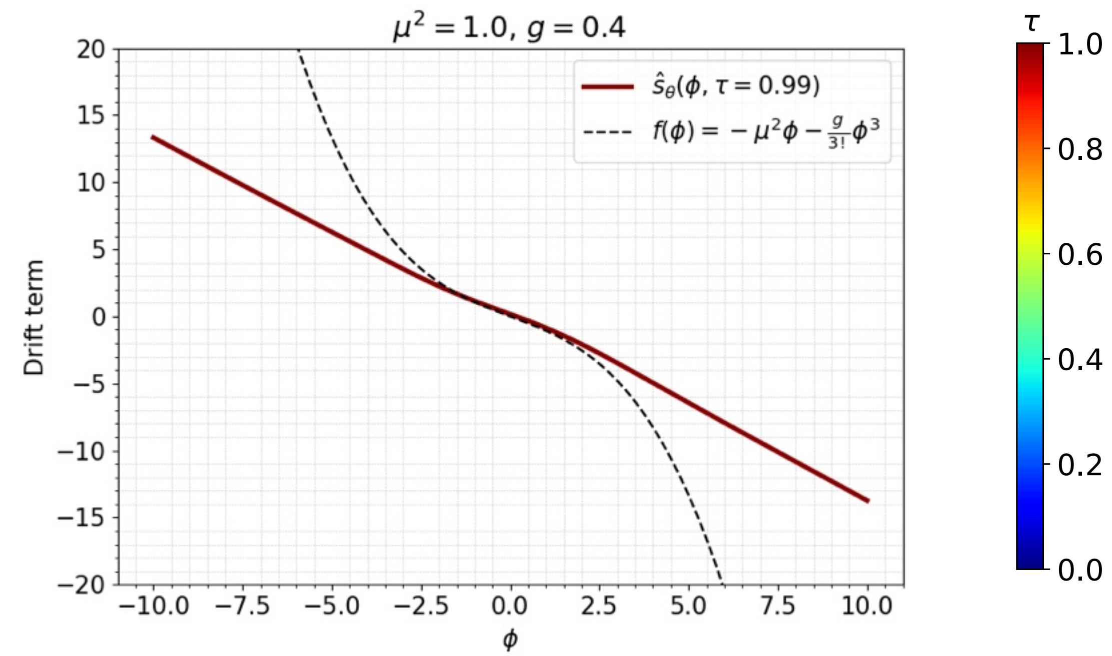
Single-Well

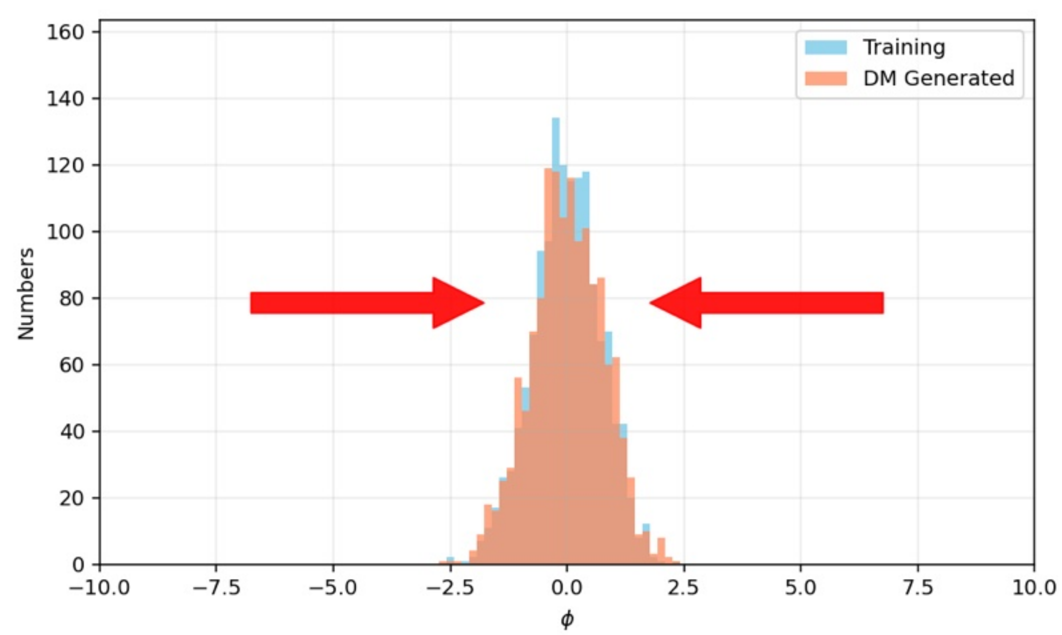
$$S(\phi) = \frac{\mu^2}{2}\phi^2 + \frac{g}{4!}\phi^4,$$

$$f(\phi) = -\mu^2\phi - \frac{g}{3!}\phi^3$$

0+0 dimensional FT (1 variable)





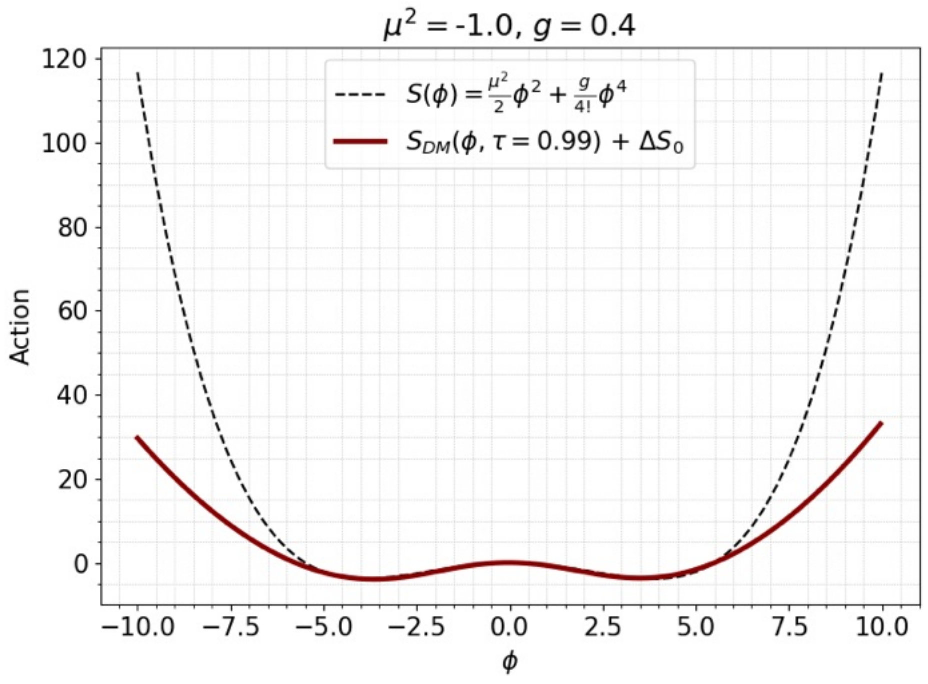


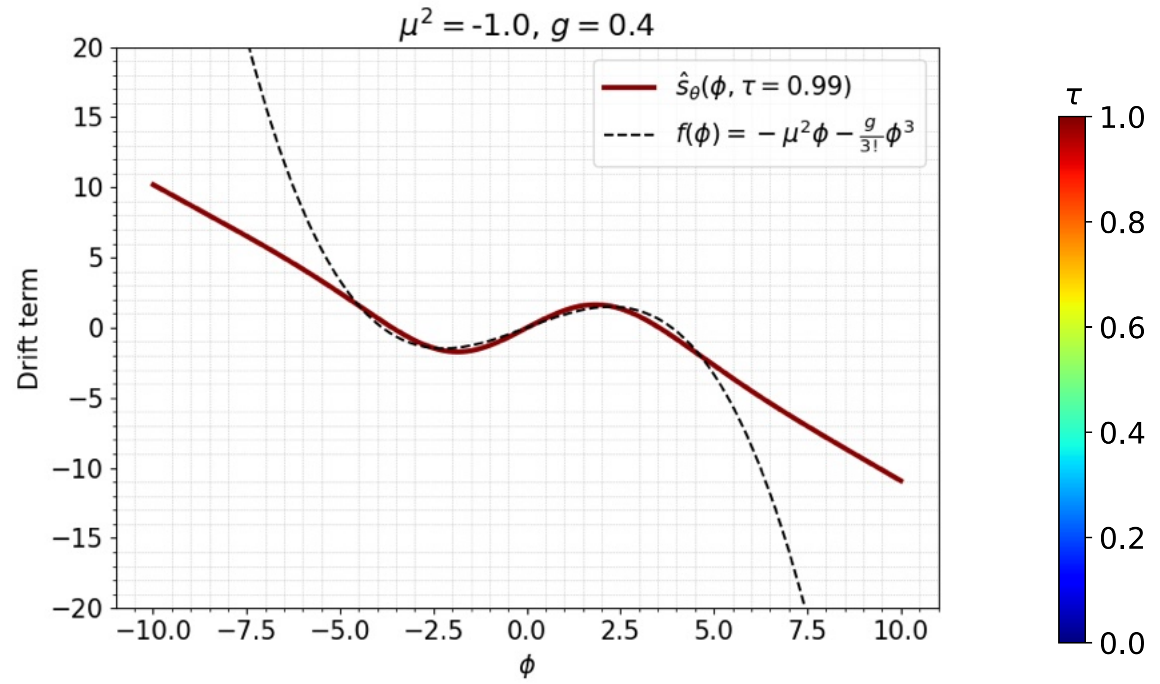
Effective Action

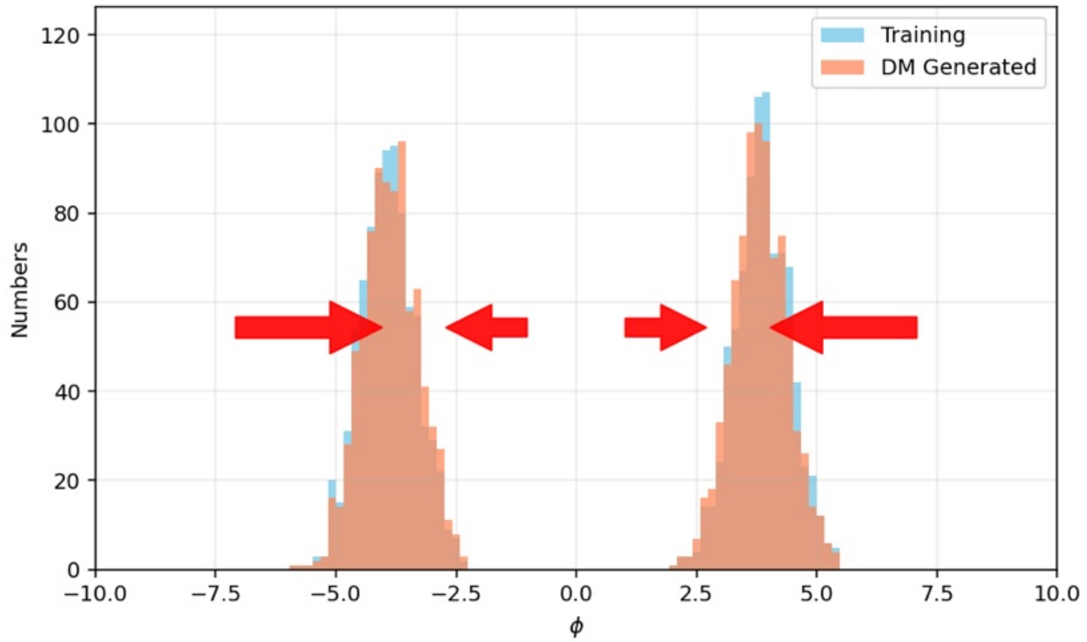
Double-Well

$$S(\phi) = \frac{\mu^2}{2}\phi^2 + \frac{g}{4!}\phi^4,$$
$$f(\phi) = -\mu^2\phi - \frac{g}{3!}\phi^3$$

0+0 dimensional FT (1 variable)



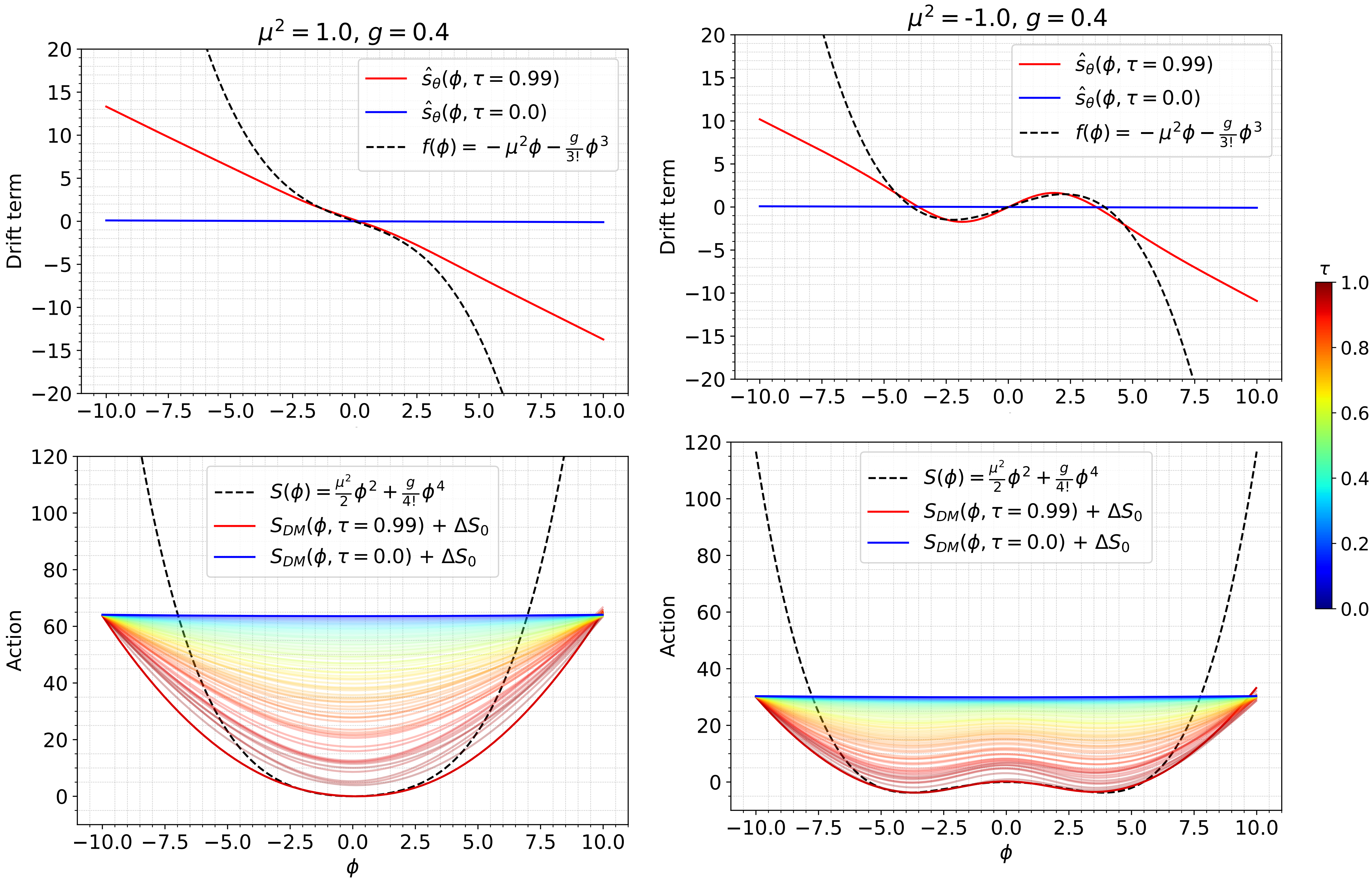




Effective Action

Flow

$$\frac{\partial S_{\mathbf{DM}}(\phi,\tau)}{\partial\tau} = \frac{\partial}{\partial\tau}\int^{\phi}\hat{\mathbf{s}}_\theta(\tilde{\phi},\tau)d\tilde{\phi}$$



Why Diffusion Models?

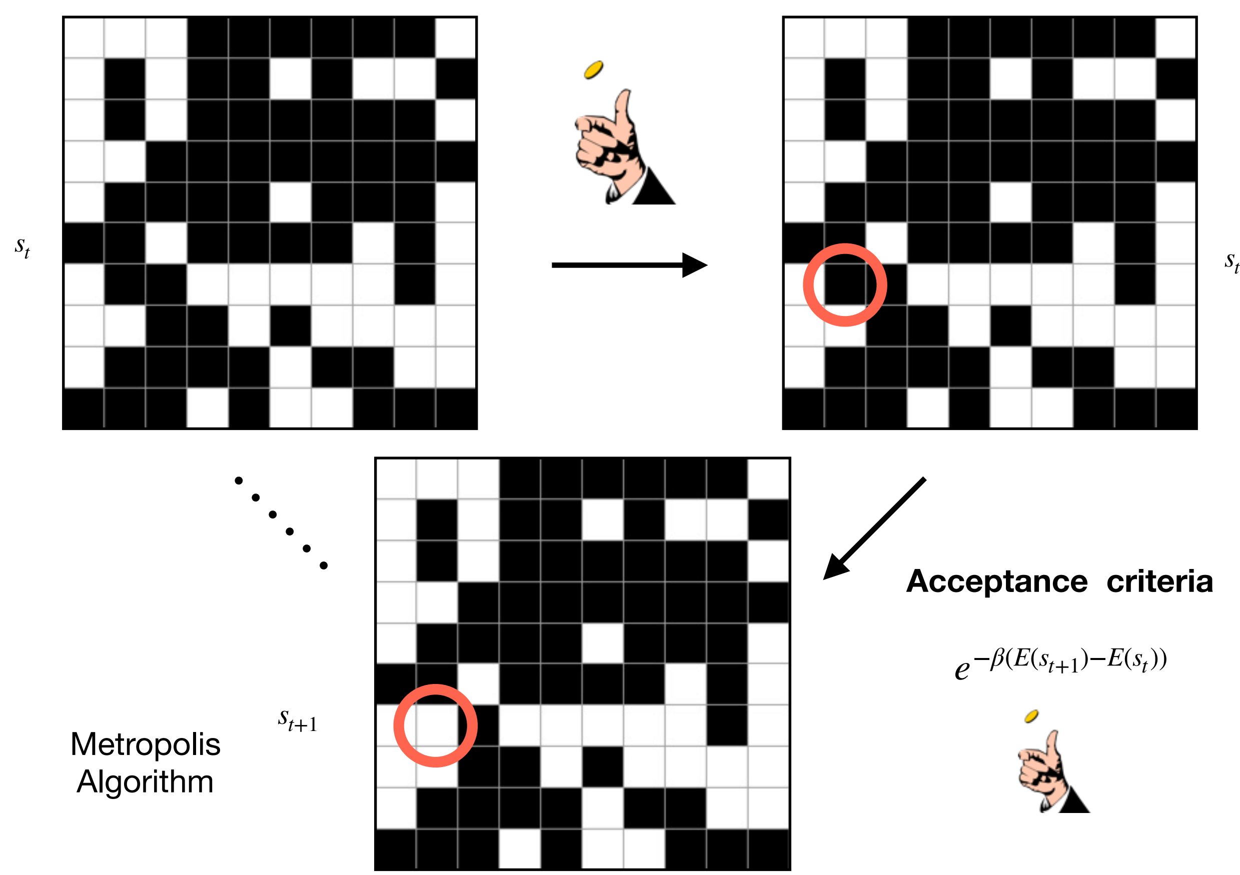
for Lattice simulations

Generative models

→ **underlying distributions in data**

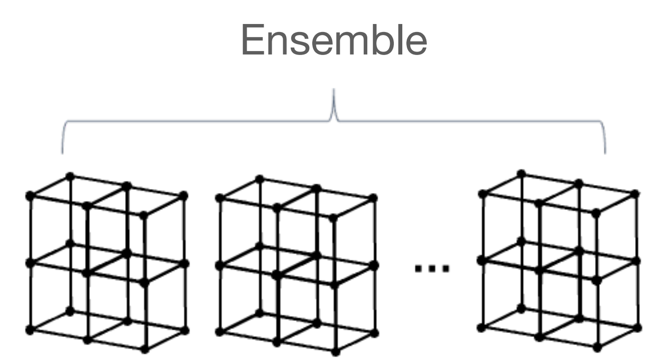
Lattice simulations

→ **physical distributions, sampling**



$$p(\phi) = e^{-S(\phi)}/Z$$

$$\langle O \rangle \approx \frac{1}{N}\sum_i O_i$$



Universal and Accurate, but

- Local update, low-efficiency
- **Critical Slowing Down**
  [U. Wolff, Nucl. Phys. B 17, 93 (1990)]

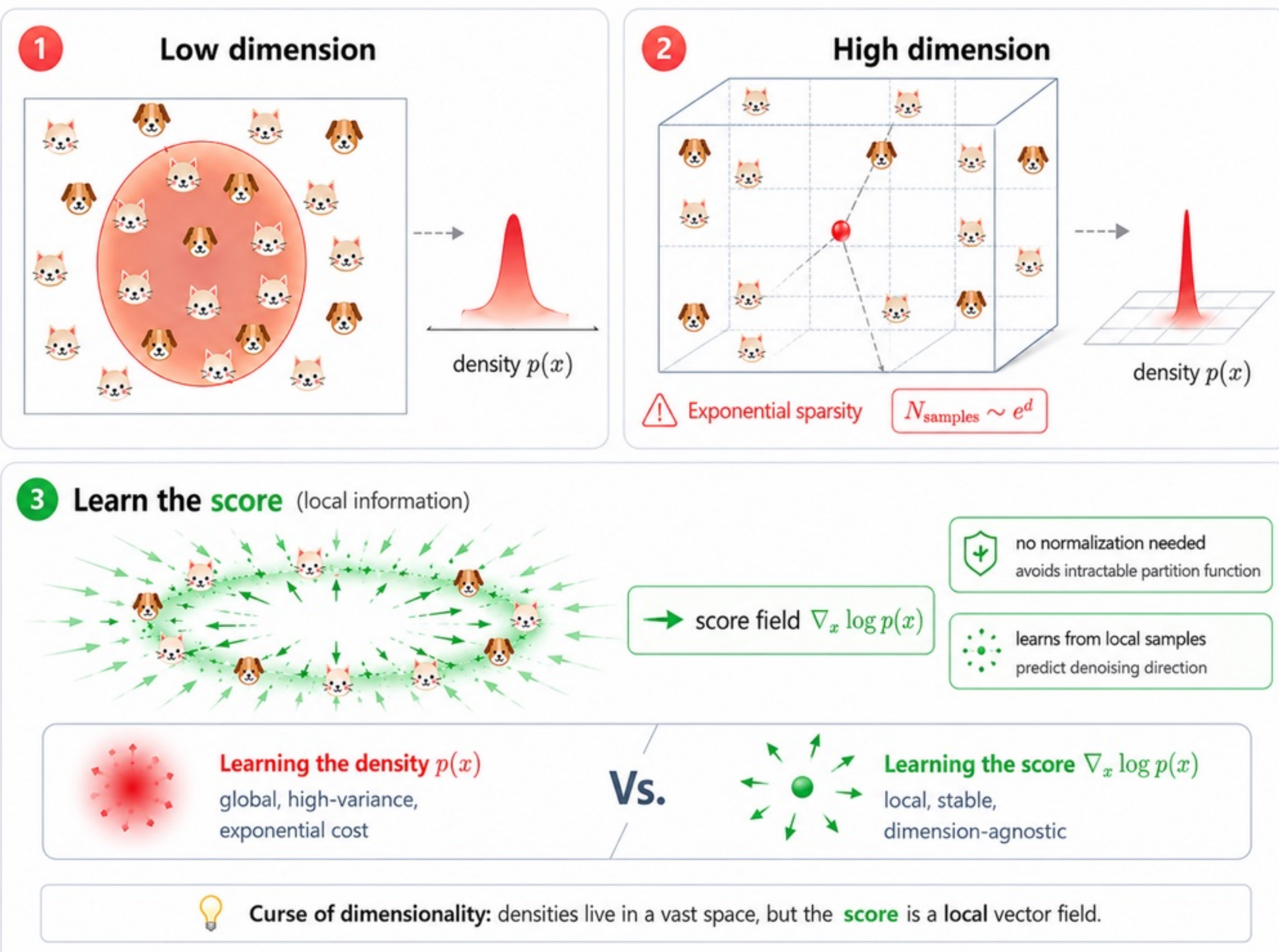
Why Diffusion Models?

for Lattice simulations

Generative models

$q_\theta(\phi) \rightarrow p(\phi)$

Physical distributions, sampling



Curse of Dimensionality

- **Implicit** Likelihood Models
  flexible generation, need data
  - VAEs and GANs

    D. Giataganas, et al., New J. Phys. 24, 043040 (2022).
    K. Zhou, et al., Phys. Rev. D 100, 011501 (2019).
    J. M. Pawlowski and J. M. Urban, MLST 1, 045011 (2020).
    J. Singh, et al., SciPost Phys. 11, 043 (2021).
- **Explicit** Likelihood Models
  tractable, but hard to learn
  - Autoregressive models

    D. Wu, et al., Phys. Rev. Lett. 122, 080602 (2019).
    L. Wang, et al., CPL 39, 120502 (2022).
    P. Białas, P. Korcyl, and T. Stebel, CPC 281, 108502 (2022).
  - Flow-based models

    M. Albergo, G.Kanwar, and P. Shanahan, Phys. Rev. D100,034515 (2019).
    G. Kanwar, et al., Phys. Rev. Lett. 125, 121601 (2020).
    K. A. Nicoli, et al., Phys. Rev. Lett. 126, 032001 (2021).
    L. Del Debbio, et al., Phys. Rev. D 104, 094507 (2021).
    M. Caselle, et al., J. High Energ. Phys. 2022, 15 (2022).
    S. Chen, et al., Phys. Rev. D 107, 056001(2023).
    K. Cranmer, G. Kanwar, S. Racanière, D. J. Rezende, and P. E. Shanahan, Nat Rev Phys 1 (2023).
    ....
- **Score-based models [This Talk]**
  implicit density, explicit score

Lattice Field Generations

What I will (or not) introduce

https://github.com/lingxiao-mlphys/DMasSQ
https://github.com/zzzqt/DM4U1
https://dm-qft.github.io/homepage

What I will introduce

- Numerical results for lattice scalar field in 2-dim and 3-dim, and gauge field simulations in 2-dim and 4-dim
- Expandability of diffusion models for different lattice sizes and couplings
- Exactness of generations → Metropolis algorithms

What I will not introduce, but can be found in Refs and GitHub pages

- Efficiency evaluations on acceptance rates and autocorrelation times
- Predetermined (Forward) Diffusion Schemes

  Variance Expanding (e.g., Score-Based DM), Variance Preserving (e.g., DDPM), etc.
- Training Objective (Loss Function) and Set-Ups

  Score-Matching, $\mathscr{L}_\theta = \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_0(\phi_0)} \mathbb{E}_{p_i(\phi_i|\phi_0)} \left[\left\| \mathbf{s}_\theta(\phi_i, \xi) - \nabla_{\phi_i} \log p_i(\phi_i | \phi_0) \right\|_2^2 \right]$
- Details of Neural Network for Score Functions

  CNNs, but modified for different field theories, e.g., L-CNN for gauge fields.

Lattice Scalar Fields

Scalar ϕ^4 field

- **Euclidean action** of bare fields

$$S_E = \int \mathrm{d}^d x \mathrm{d}\tau \left(\frac{1}{2}(\partial^2 \phi_0^2 + m^2 \phi_0^2) - \frac{\lambda_0}{4!}\phi_0^4 \right)$$

- Action on discrete lattice

$$S_E = \sum_x a^d [\sum_{\mu=1}^{d} \frac{(\phi_0(x+a\hat{\mu}) - \phi_0(x))^2}{a^2} + \frac{m_0^2}{2}\phi_0^2 + \frac{\lambda_0}{4!}\phi_0^4]$$

- Dimensionless form

$$S_E = \sum_x [-2\kappa \sum_{\mu=1}^{d} \phi(x)\phi(x+\hat{\mu}) + (1-2\lambda)\phi(x)^2 + \lambda\phi(x)^4]$$

$$a^{\frac{d-2}{2}}\phi_0 = (2\kappa)^{1/2}\phi$$

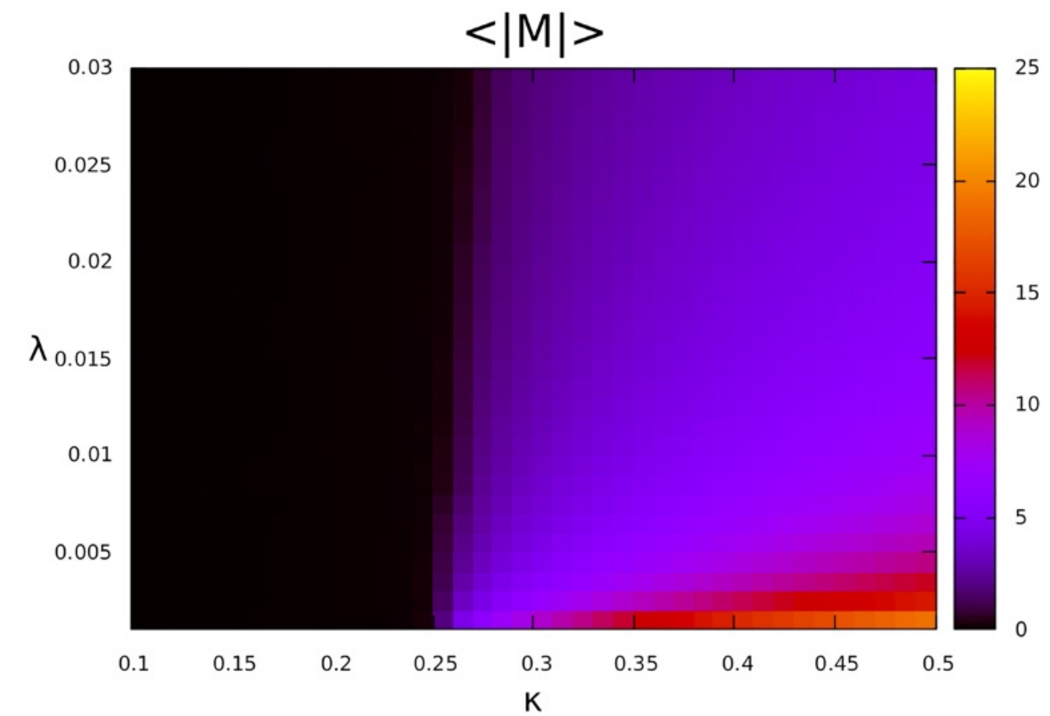
$$(am_0)^2 = \frac{1-2\lambda}{\kappa} - 2d, \quad a^{-d+4}\lambda_0 = \frac{6\lambda}{\kappa^2}$$

Hopping parameter κ, Coupling constant λ

Phase Transition

$\mathbb{Z}_2$ symmetry spontaneously broken above the critical point

Order parameter: magnetization

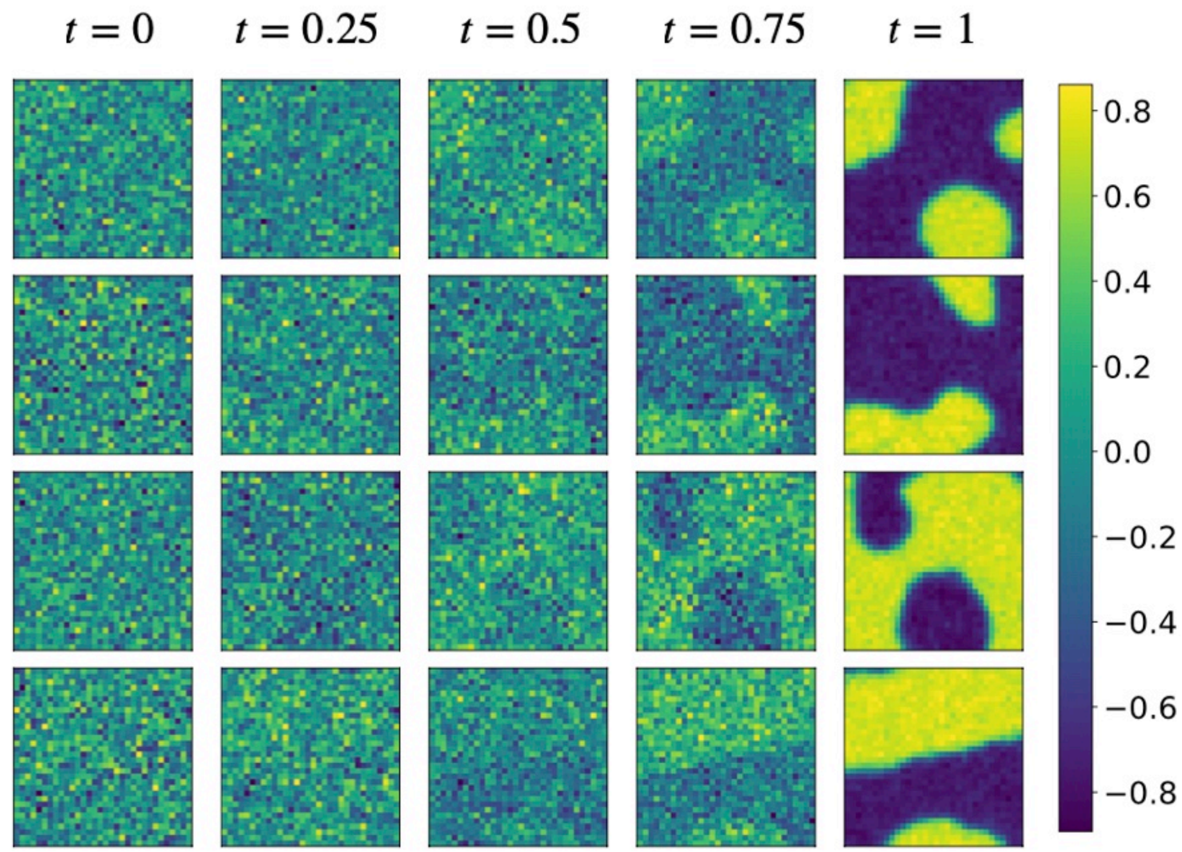


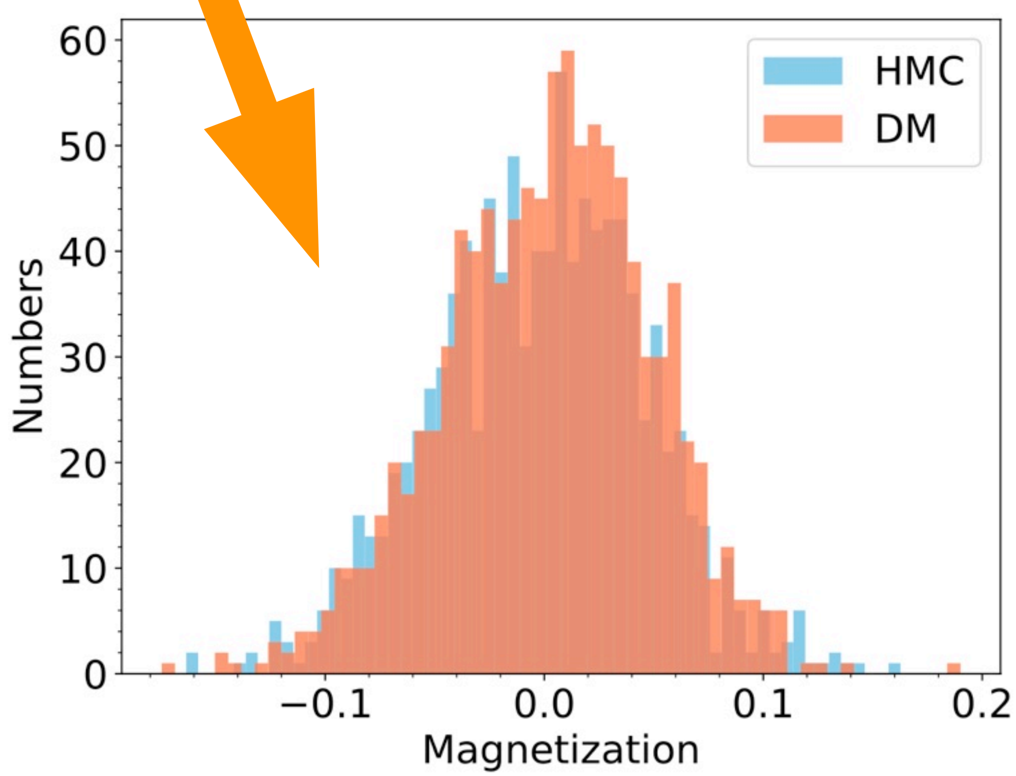
Phase diagram at $d = 2$ @Julian Urban

Lattice Scalar Fields

Different Phases

- **Diffusion Models**
  - Variance Expanding with $T = 1.0, \sigma = 25$
- **Data Generation**
  - 2D, 32×32 lattice; Hamiltonian Monte Carlo(HMC); 5120 configurations for training.
  - Broken phase: $\kappa = 1.0, \lambda = 0.022$
  - Symmetric phase: $\kappa = 0.21, \lambda = 0.022$





data-set	$\langle M \rangle$	χ_2	U_L
Training(HMC)	0.0012± 0.0007	2.5160 ± 0.0457	0.1042 ± 0.0367
Testing(HMC)	0.0018 ± 0.0015	2.4463 ± 0.1099	-0.0198 ± 0.1035
Generated(DM)	0.0017± 0.0015	2.4227 ± 0.1035	0.0484 ± 0.0959

LW, G Aarts, K Zhou, JHEP(2024)

Y Tang, **LW**, etc., arXiv:2606.xxx

Lattice Scalar Fields

Critical Regime

- Data-Preparation
  - Wolff cluster flips: $\phi_x \to -\phi_x$
  - Fourier-accelerated(**FA**)-**HMC**

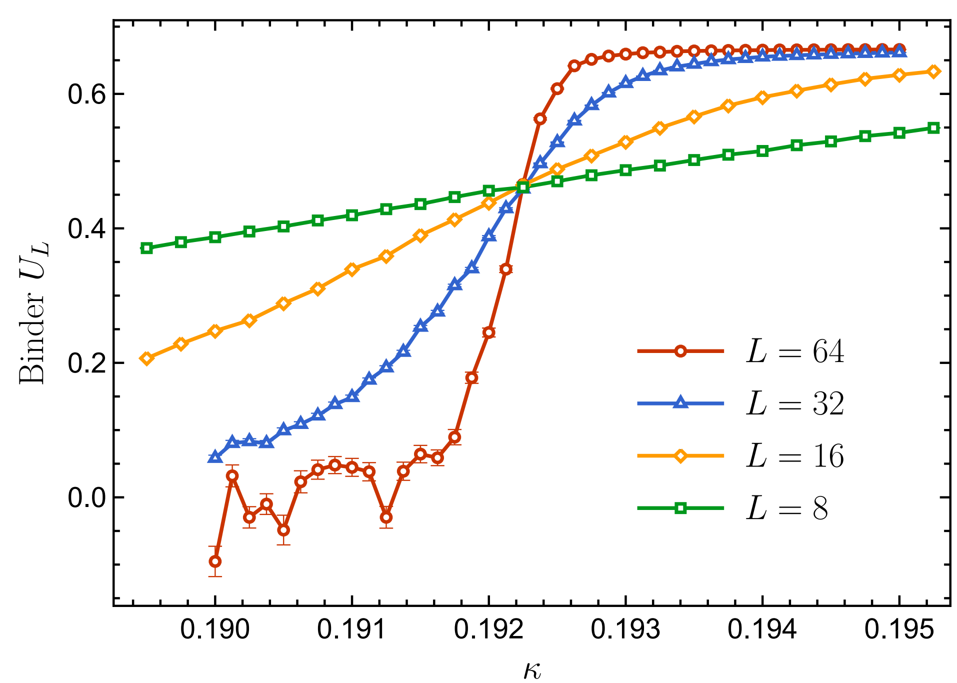
$$M(\hat{p}) = \hat{p}^2 + m_{\text{eff}}^2, \quad \hat{p}^2 = \sum_{\mu=1}^{D} 4\sin^2\!\left(\frac{\pi k_\mu}{L}\right)$$

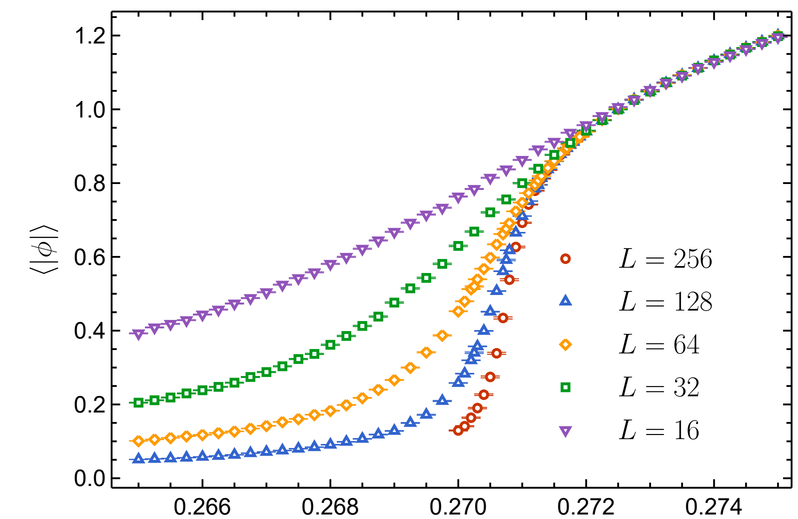
- **2D Critical Point**

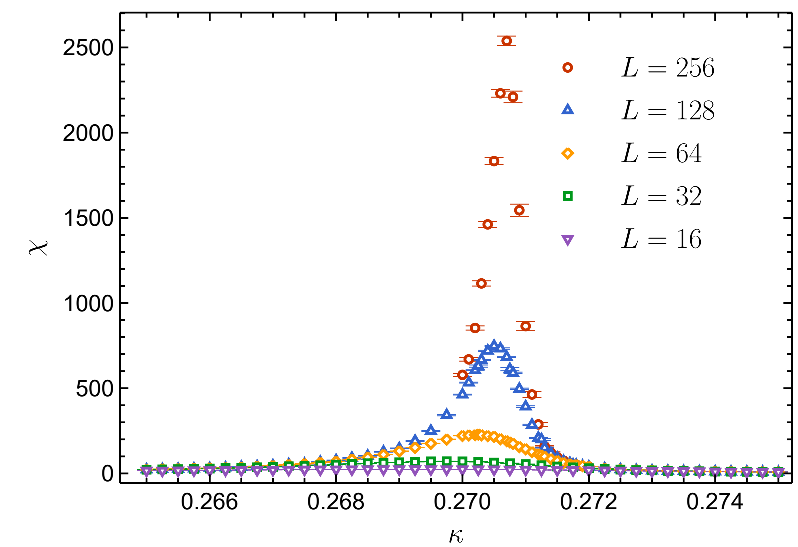
$$\kappa_c(\lambda = 0.022) \approx 0.27088$$

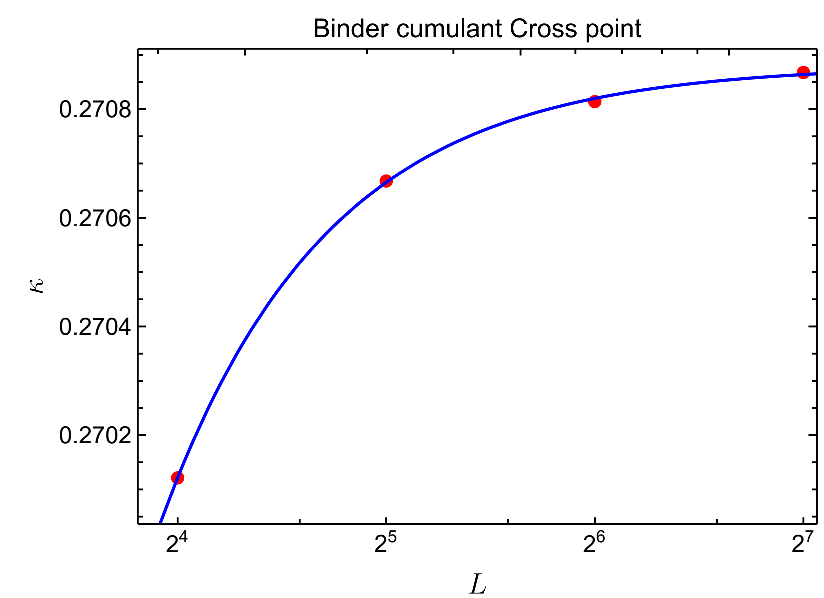
- **3D Critical Point**

$$\kappa_c(\lambda = 0.9) \approx 0.19225$$









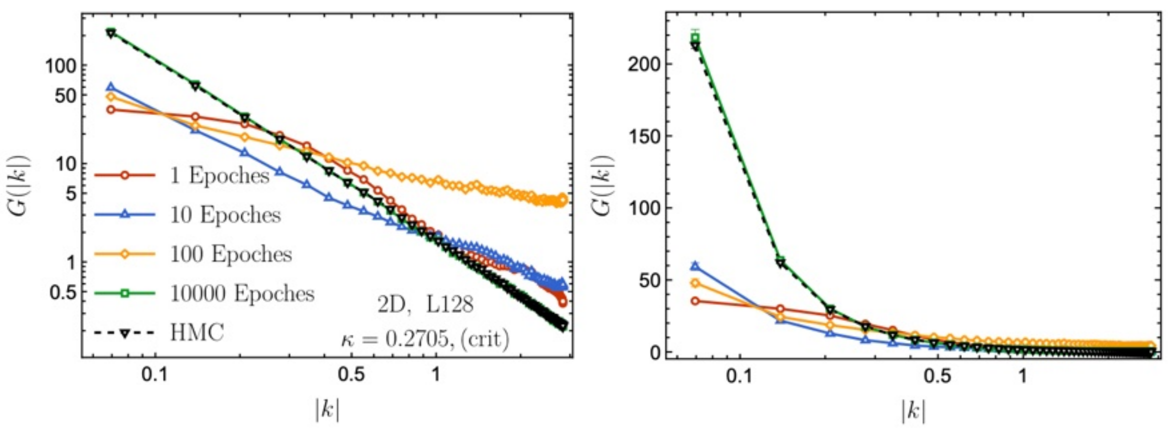
Y Tang, **LW**, etc., arXiv:2606.xxx

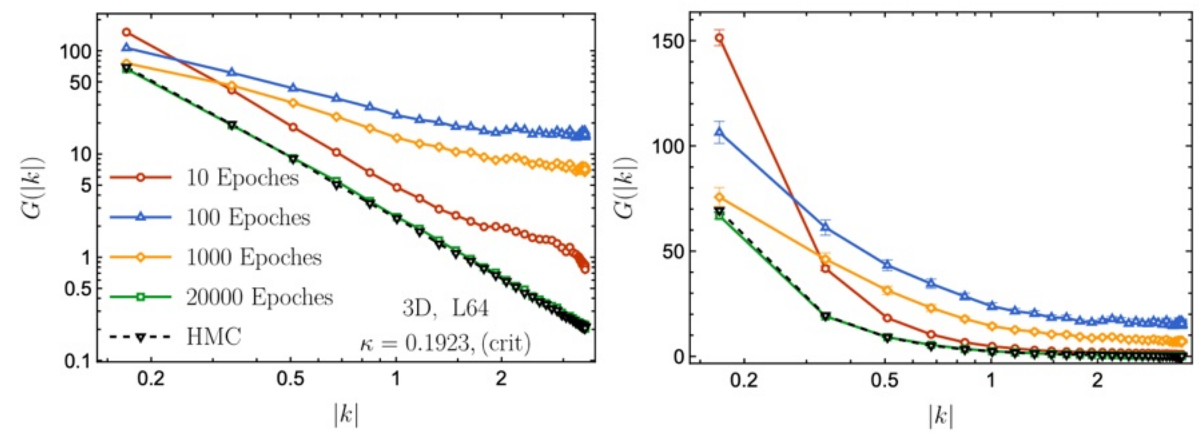
Lattice Scalar Fields

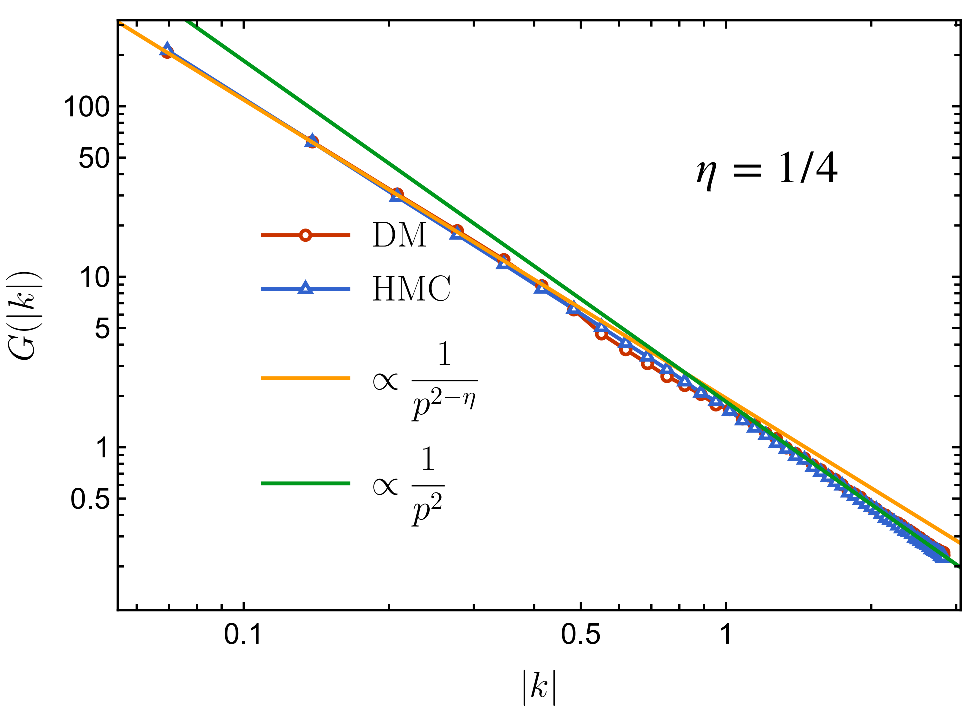
Criticality

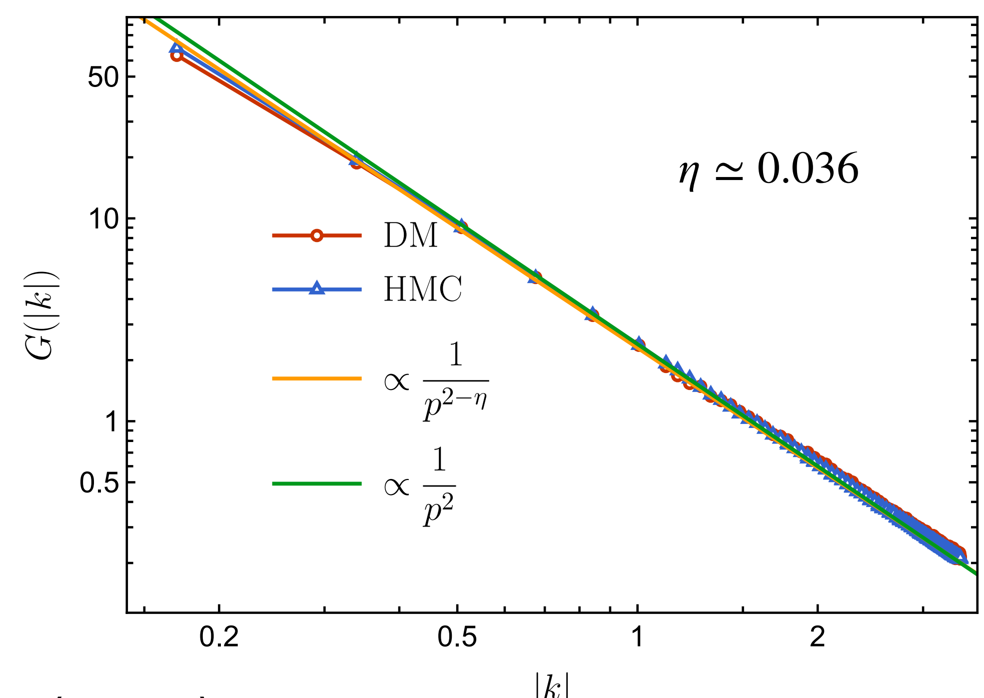
2D $L = 128, \lambda = 0.022, \kappa \approx \kappa_c = 0.2705$

3D $L = 64, \lambda = 0.9, \kappa \approx \kappa_c \approx 0.1923$





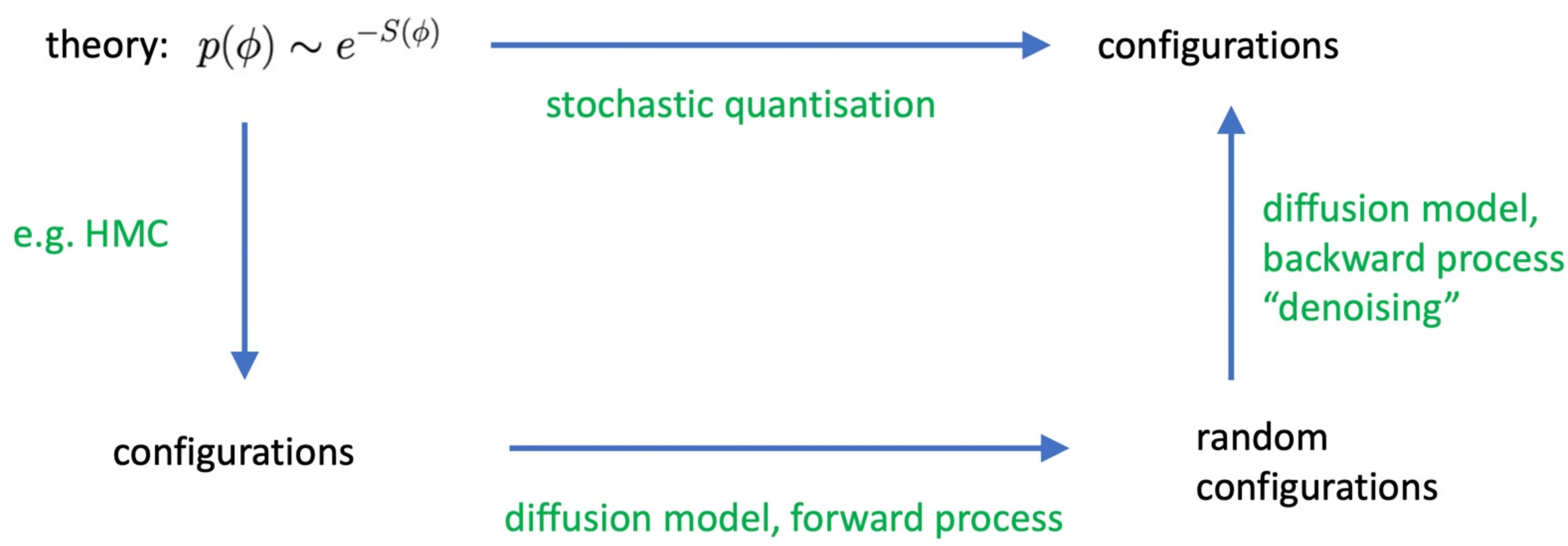




$$G(|k(q)|) = \left\langle \tilde{\phi}(q)\tilde{\phi}(-q) \right\rangle = \left\langle |\tilde{\phi}(q)|^2 \right\rangle$$

Expandability

Learn the Sampler, Not Just Samples



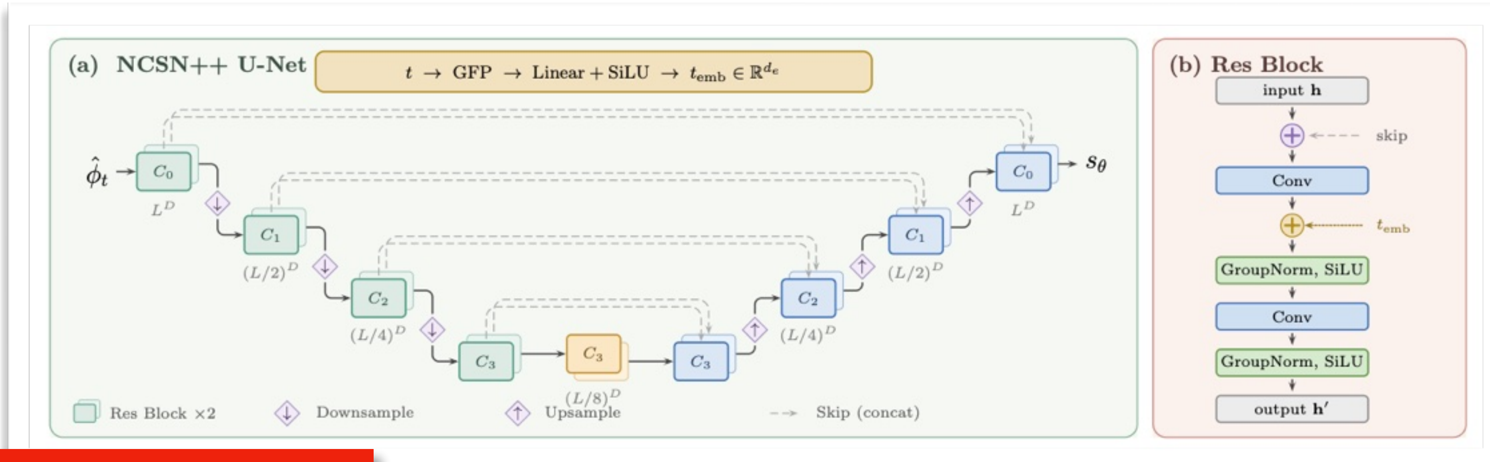
Did we take a detour?
No! We learned a reusable sampling map.

Train once, generate many configurations.
The learned score field can be reused or adapted across related lattice systems.

Near Critical Sampling

Y Tang, **LW**, etc., arXiv:2606.xxx

Yang-Yang Tan



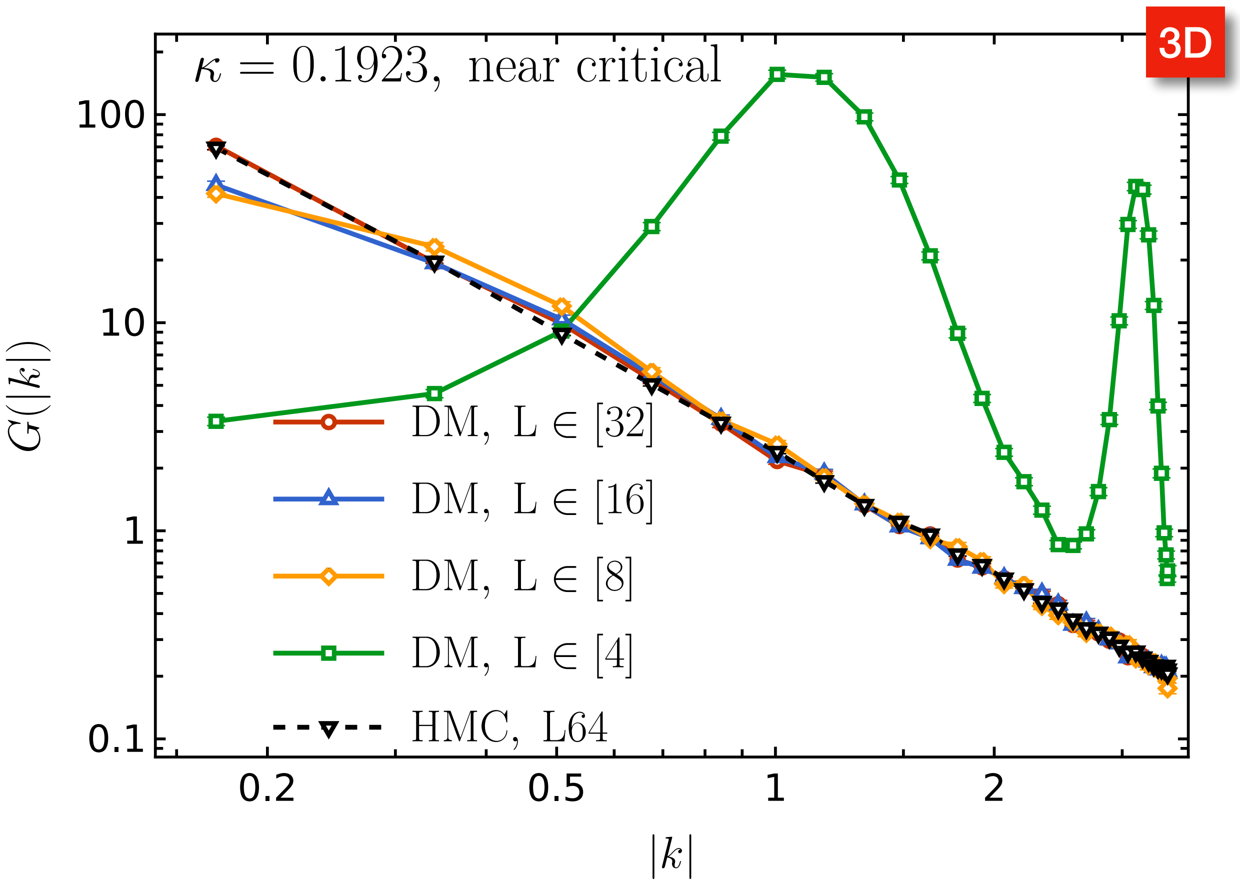
Local Operator

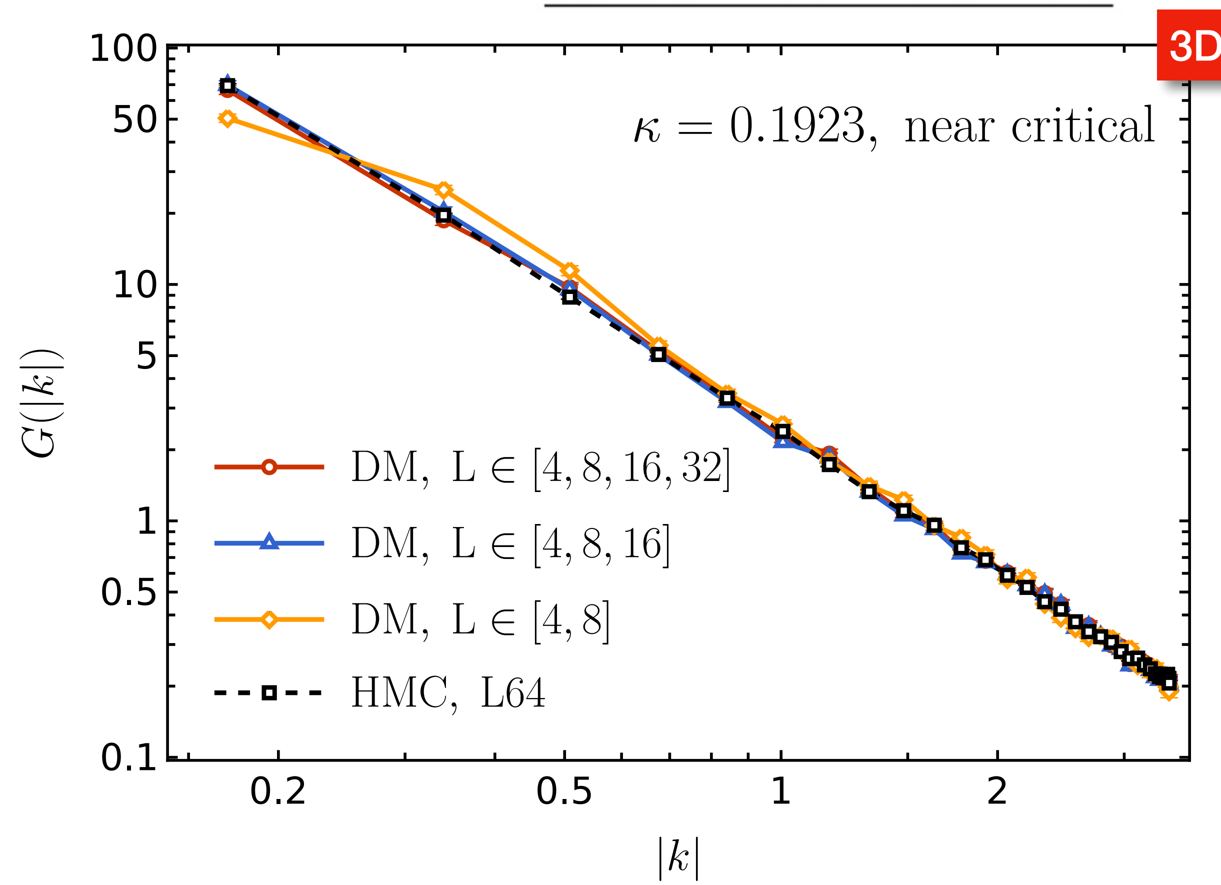
$$(f_\theta u)(x) = F_\theta\big(\{\, u(x+\delta) \mid \delta \in \Omega \,\}\big)$$

Train on small volumes and generate large ones

Acceptance Rate

training set	$\kappa = 0.18$	$\kappa \approx 0.1923$	$\kappa = 0.2$
{4, 8}	0.358(84)	0.326(130)	0.334(76)
{4, 8, 16}	0.548(135)	0.562(47)	0.531(130)
{4, 8, 16, 32}	0.745(38)	0.733(161)	0.618(114)





Expandability

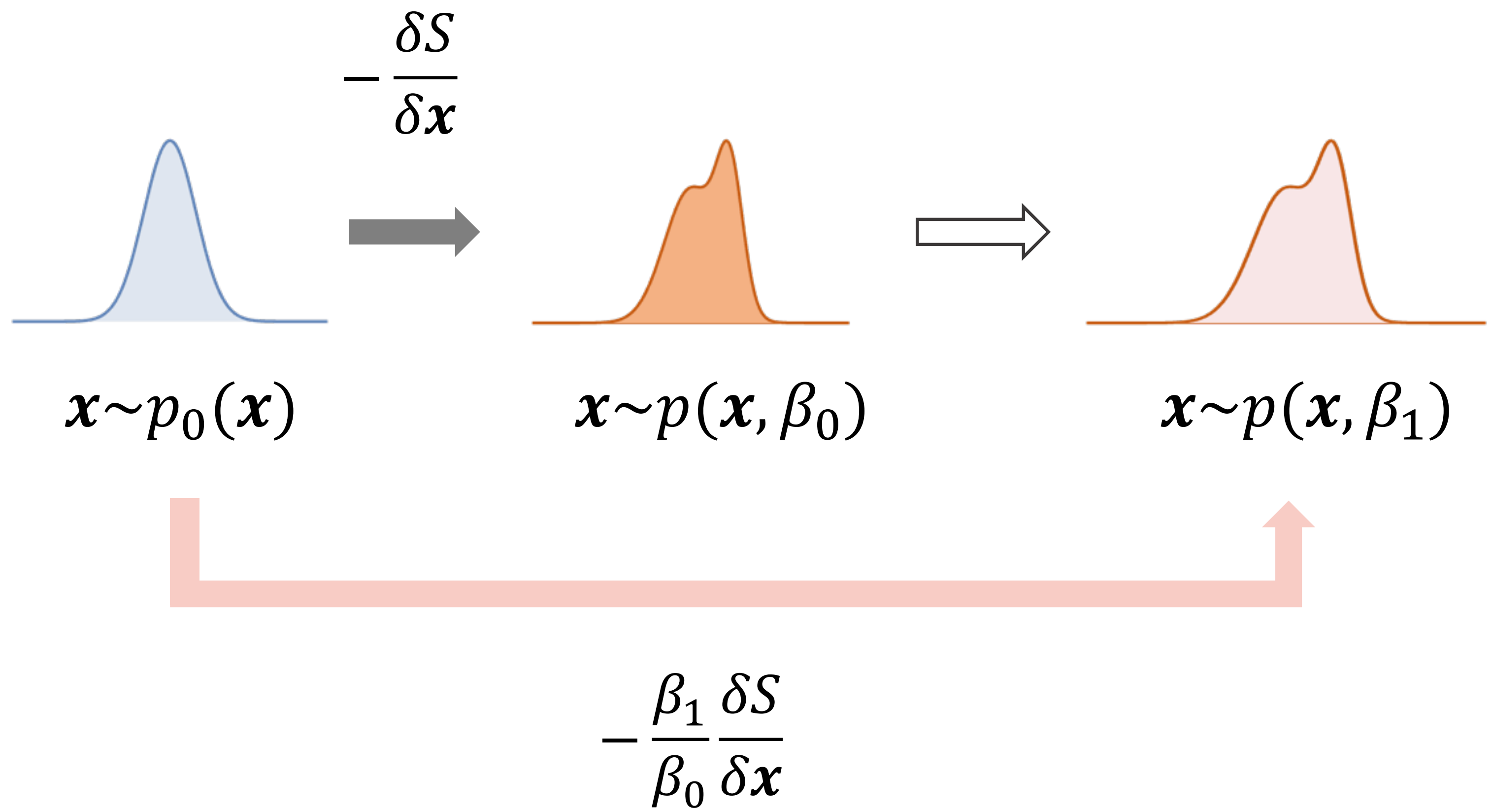
Learn the Sampler, Not Just Samples

$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\boxed{\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)}} + \eta(x,\tau)$$

Drift Term

$$\frac{d\phi}{dt} = \boxed{-\, g^2(t)\mathbf{s}_{\hat{\theta}}(\phi,t)} + g(t)\bar{\eta}(t)$$

Score Fucntion



e.g., pure gauge

$$S = \beta \sum_{\square} \left(1 - \mathbf{Re}(U_{\square})\right)$$

$$\tilde{\mathbf{s}}_{\hat{\theta}}(\phi,t) \equiv \beta \mathbf{s}_{\hat{\theta}}(\phi,t)$$

Expandability

Physics-Conditioned Sampler

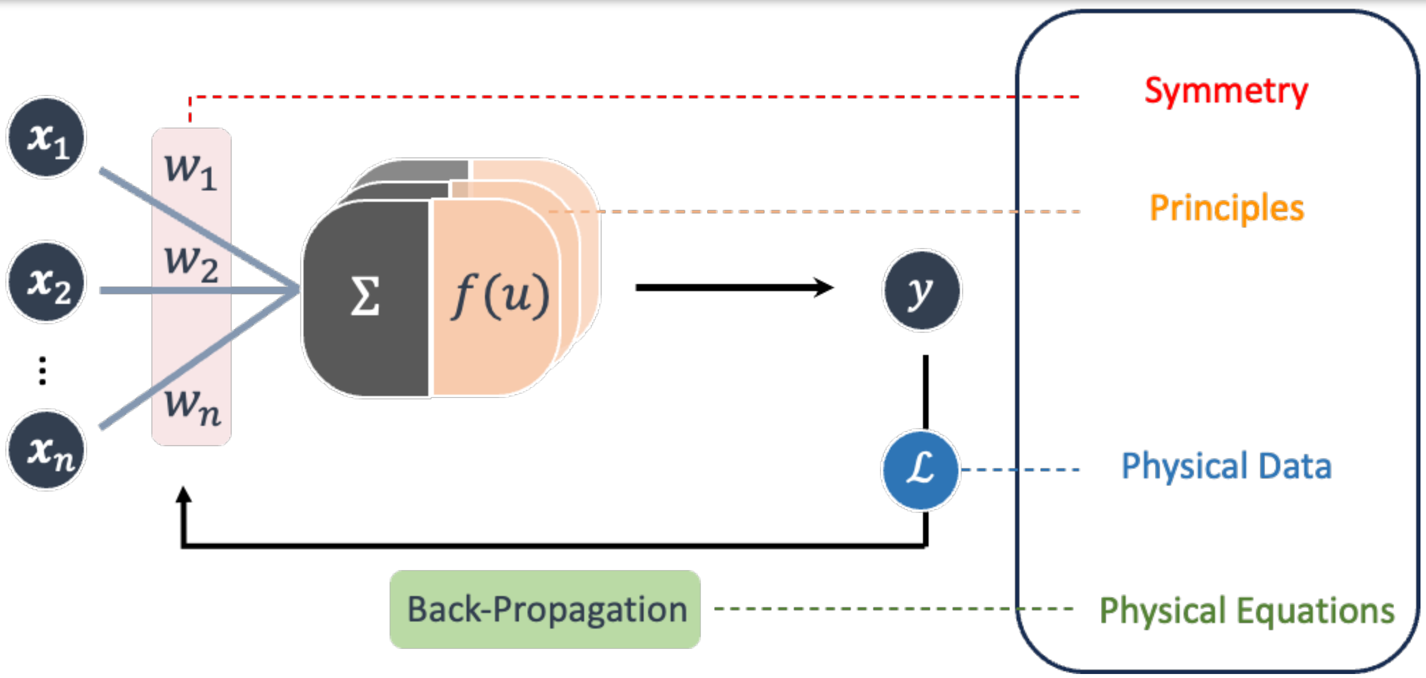
$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\boxed{\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)}} + \eta(x,\tau)$$

Drift Term

$$\frac{d\phi}{dt} = \boxed{- g^2(t)\mathbf{s}_{\hat{\theta}}(\phi, t)} + g(t)\bar{\eta}(t)$$

Score Fucntion

Different Lattice Sizes, Inverse Coupling Constants



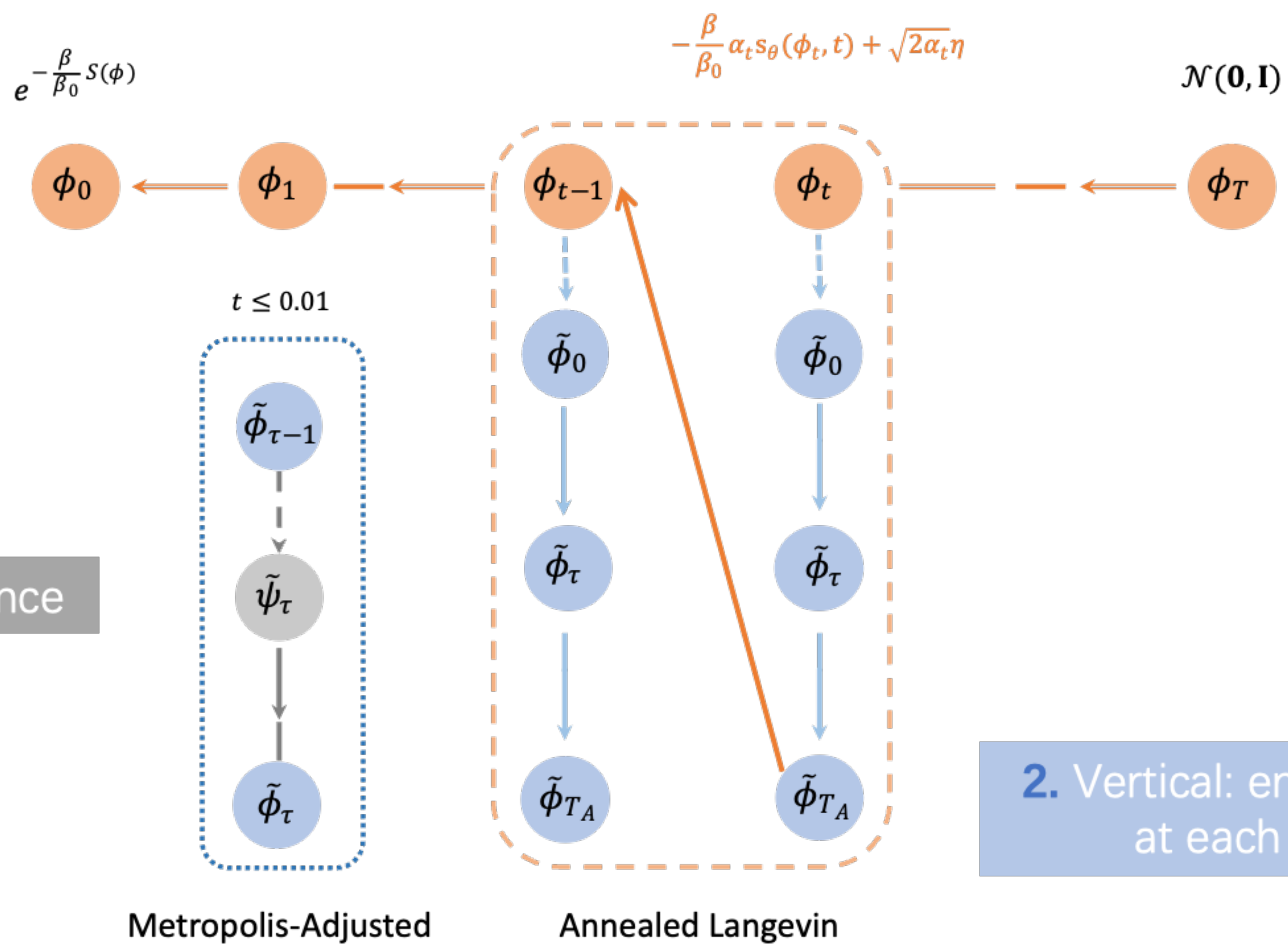
G. Aarts, K. Fukushima, T. Hatsuda, A. Ipp, S. Shi, **L. Wang***, and K. Zhou,
Physics-Driven Learning for Inverse Problems in Quantum Chromodynamics,
Nature Reviews Physics volume 7, pages154–163 (2025)

Exactness

Q Zhu, G Aarts, W Wang, K Zhou, **LW**, JHEP(2026)

How can one trust DM generation?

1. Horizontal : Euler-Maruyama scheme gradually decrease noise



Qianteng Zhu

3. Detailed balance

2. Vertical: ensure equilibrium at each noise scale

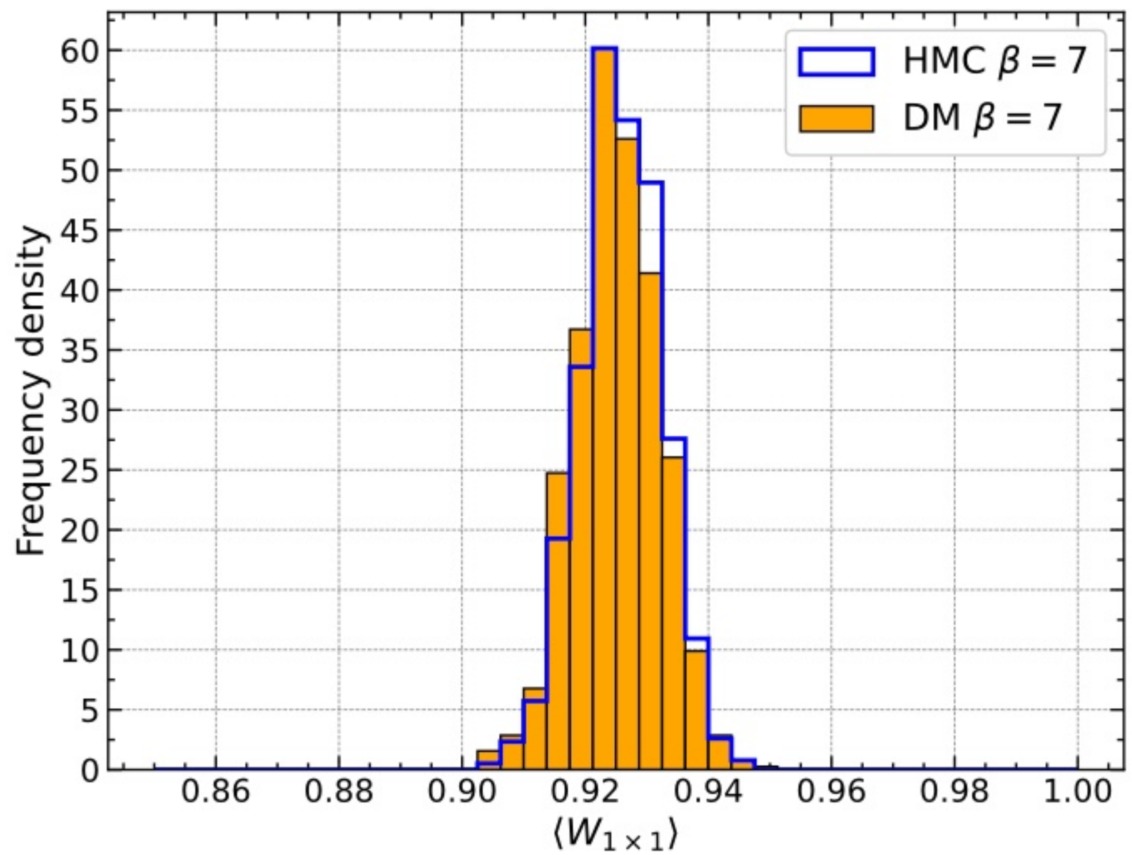
Metropolis-Adjusted Annealed Langevin Algorithm(MAALA)

DM for U(1) Gauge Field

Q Zhu, G Aarts, W Wang, K Zhou, **LW**, JHEP(2026)

Comprehensive Comparison for 2D U(1)

Learned at $\beta = 1$ with **30k** cofigurations, L = 16



No Topological Freezing!

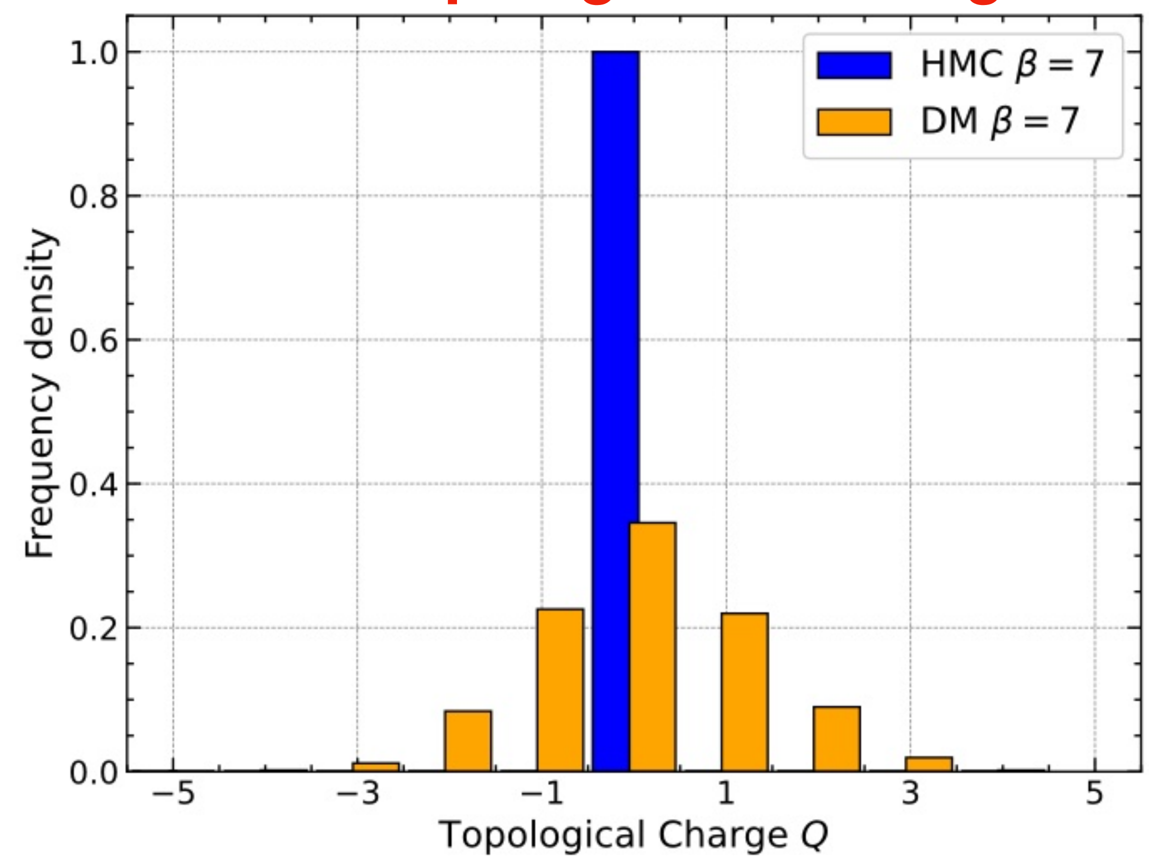


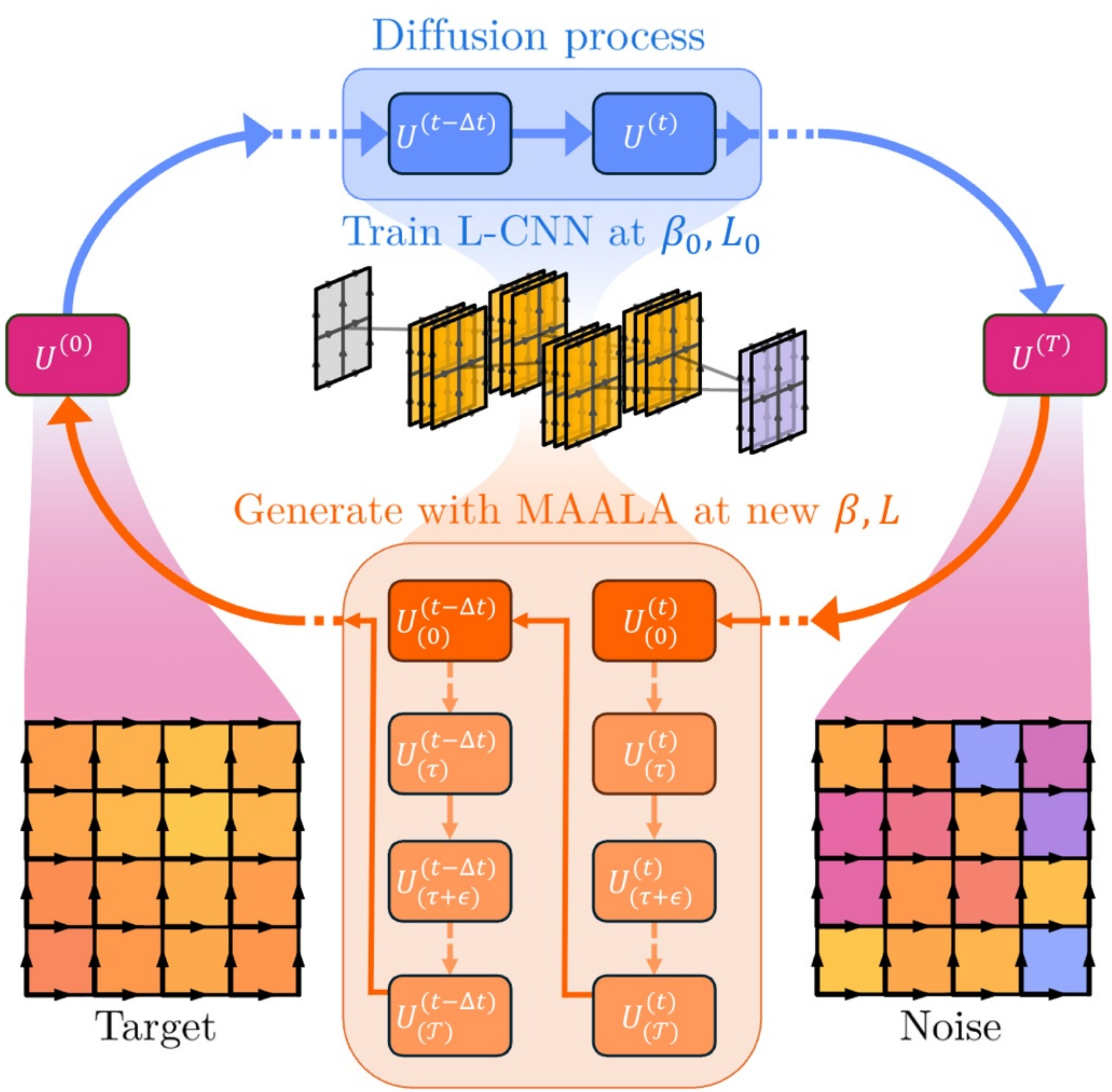
Table 2. Comparison of the $l \times l$ Wilson Loops for $L = 16, \beta = 7$

Loop size (l)	HMC	DM	Langevin	Exact
1	0.926(7)	0.926(7)	0.924(6)	0.926
2	0.737(31)	0.737(32)	0.730(34)	0.734
3	0.510(67)	0.496(72)	0.489(73)	0.498
4	0.311(97)	0.283(96)	0.283(106)	0.290

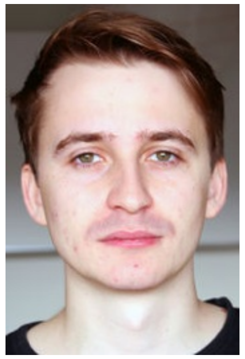
Table 3. Comparison of observables for $\beta = 7$ at different lattice sizes

Lattice Size (L)	1×1 Wilson Loop				Topological Susceptibility			
	HMC	DM	Langevin	Exact	HMC	DM	Langevin	Exact
8	0.927(13)	0.926(13)	0.921(13)	0.926	0.00006(3)	0.0040(12)	0.0143(5)	0.0040
16	0.926(7)	0.926(7)	0.924(6)	0.926	0.00013(2)	0.0045(5)	0.0131(5)	0.0039
32	0.926(3)	0.925(4)	0.924(4)	0.926	0.00013(2)	0.0040(4)	0.0137(6)	0.0039

Lattice U(N)/SU(N) Gauge Fields



Thomas R. Ranner



David Müller

Qianteng Zhu

G Aarts, D Habibi, A Ipp, D, Müller, T Ranner, **LW**, W Wang, Q Zhu, arXiv:2601.19552 [hep-lat]

DM for U(N)/SU(N) Gauge Fields

Group-Preserving Forward Process

T^a group generators, a color indices

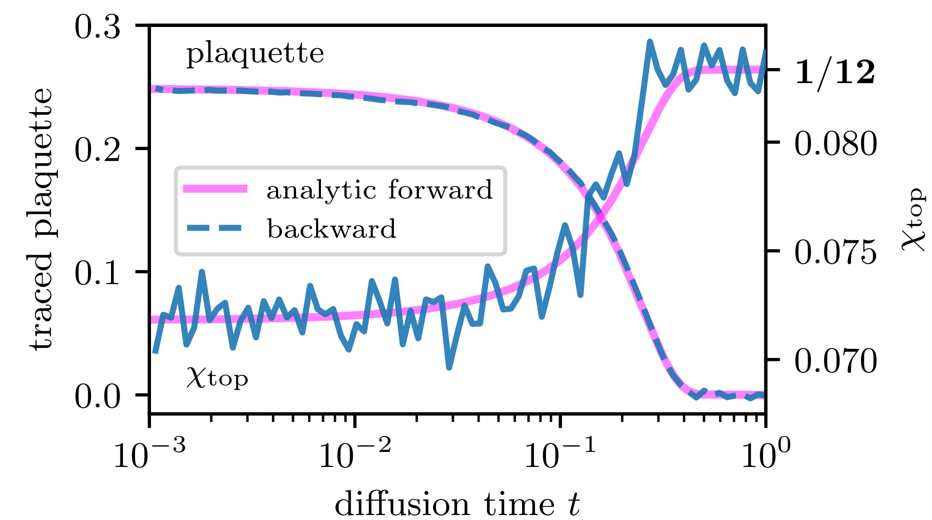
$$\mathrm{d}U^{(t)}_{x,\mu} = i\sum_a T^a\Big[K^a_{x,\mu}(\mathbf{U}^{(t)})\mathrm{d}t + g(t)\mathrm{d}W^{(t),a}_{x,\mu}\Big]\circ U^{(t)}_{x,\mu}$$

$$K^a_{x,\mu} = 0 \quad\Big\downarrow\quad \eta^a_{x,\mu} \sim \mathcal{N}(0,1), \quad \sigma(t) = \sqrt{(s^{2t}-1)/(2\ln s)}$$

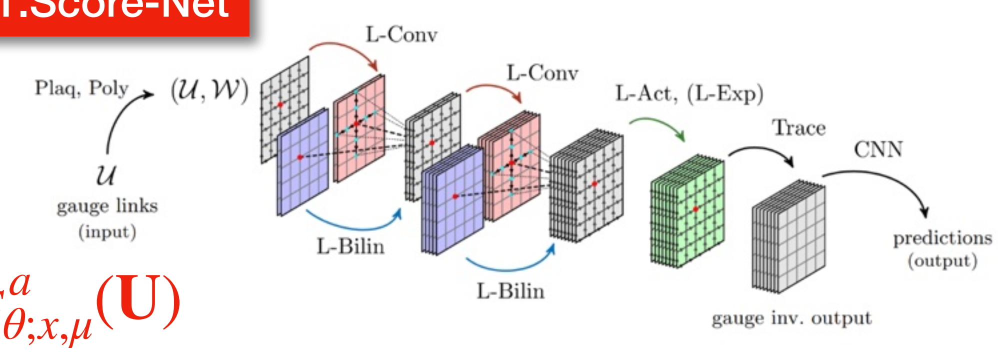
$$U^{(t)}_{x,\mu} = \exp\Big[i\sigma(t)\sum_a T^a \eta^a_{x,\mu}\Big]U^{(0)}_{x,\mu}$$

Loss Function

$$\int_0^T \mathrm{d}t \underset{\substack{p_0(\mathbf{U}^{(0)})\\ p_{0t}(\mathbf{U}^{(t)}\,|\,\mathbf{U}^{(0)})}}{\mathbb{E}} \left[\sum_{x,\mu,a}\left(s^a_{\theta;x,\mu}(\mathbf{U}^{(t)},t) + \frac{\eta^a_{x,\mu}}{\sigma(t)}\right)^2\right]$$



1.Score-Net



$s^a_{\theta;x,\mu}(\mathbf{U})$

Favoni et al., "Lattice Gauge Equivariant Convolutional Neural Networks"

2.MAALA

$$U^{(t)}_{(\tau+\epsilon)} = \exp\left[i\left(\epsilon\frac{\beta}{\beta_0}\hat{s}_\theta(\mathbf{U}^{(t)}_{(\tau)},t) + \sqrt{2\epsilon}\,\hat{\eta}\right)\right]U^{(t)}_{(\tau)}$$

$$p_{\text{accept}} = \min\left\{1, \frac{p(\tilde{\mathbf{U}}^{(t)}_{(\tau)})}{p(\mathbf{U}^{(t)}_{(\tau)})}\frac{q(\mathbf{U}^{(t)}_{(\tau)}\,|\,\tilde{\mathbf{U}}^{(t)}_{(\tau)})}{q(\tilde{\mathbf{U}}^{(t)}_{(\tau)}\,|\,\mathbf{U}^{(t)}_{(\tau)})}\right\}$$

DM for U(N)/SU(N) Gauge Fields

Group-Preserving Forward Process

T^a group generators, a color indices

$$\mathrm{d}U_{x,\mu}^{(t)} = i\sum_a T^a\Big[K_{x,\mu}^a(\mathbf{U}^{(t)})\mathrm{d}t + g(t)\mathrm{d}W_{x,\mu}^{(t),a}\Big]\circ U_{x,\mu}^{(t)}$$

$K_{x,\mu}^a = 0$

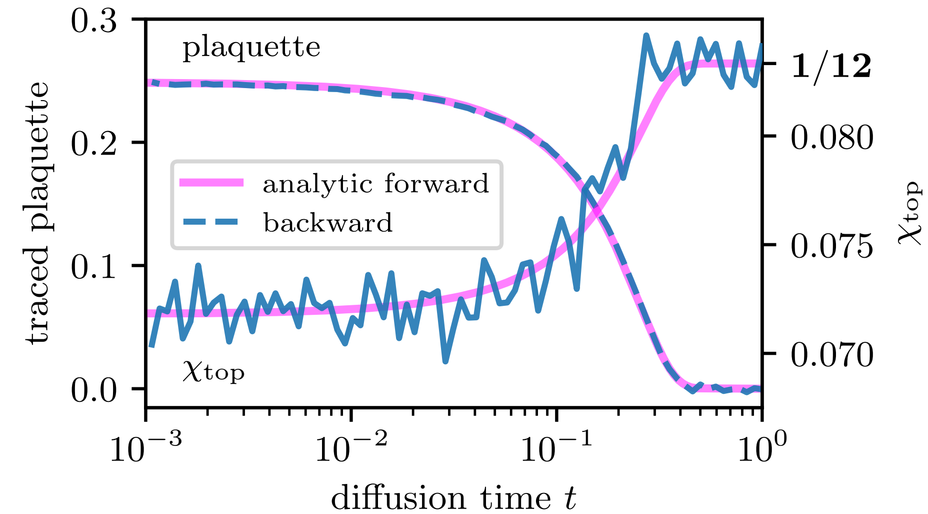
$\eta_{x,\mu}^a \sim \mathcal{N}(0,1)$

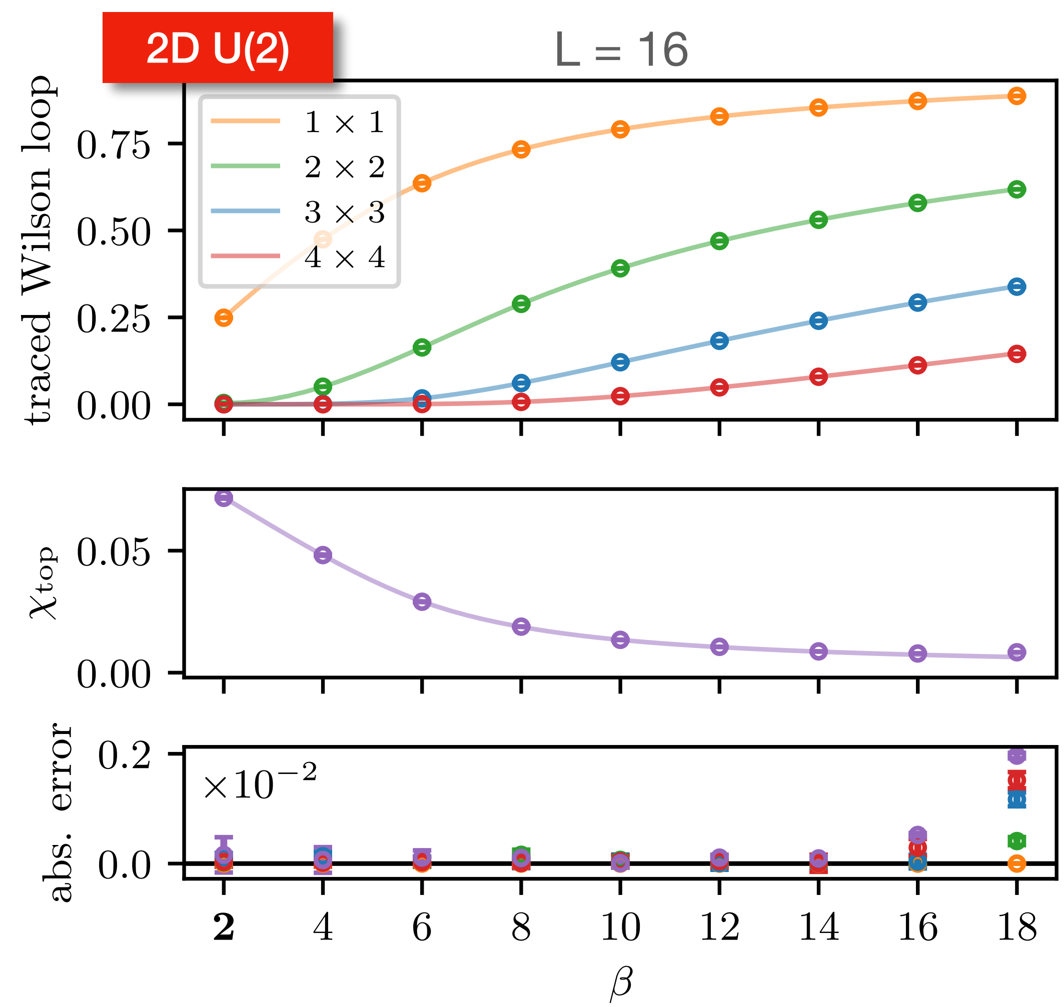
$\sigma(t) = \sqrt{(s^{2t}-1)/(2\ln s)}$

$$U_{x,\mu}^{(t)} = \exp\Big[i\sigma(t)\sum_a T^a\eta_{x,\mu}^a\Big]U_{x,\mu}^{(0)}$$

Loss Function

$$\int_0^T \mathrm{d}t \mathop{\mathbb{E}}_{\substack{p_0(\mathbf{U}^{(0)}) \\ p_{0t}(\mathbf{U}^{(t)}|\mathbf{U}^{(0)})}} \left[\sum_{x,\mu,a}\left(s_{\theta;x,\mu}^a(\mathbf{U}^{(t)},t)+\frac{\eta_{x,\mu}^a}{\sigma(t)}\right)^2\right]$$

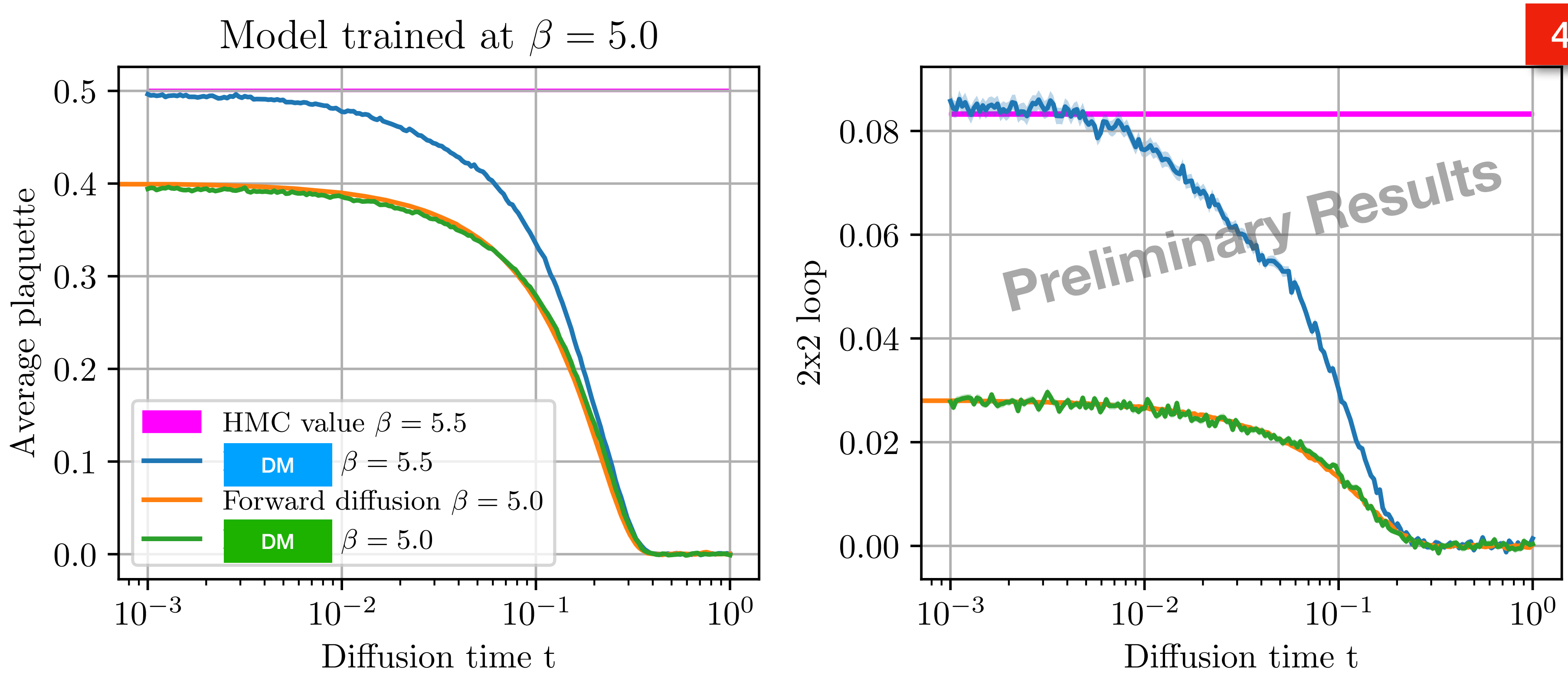




DM for U(N)/SU(N) Gauge Fields

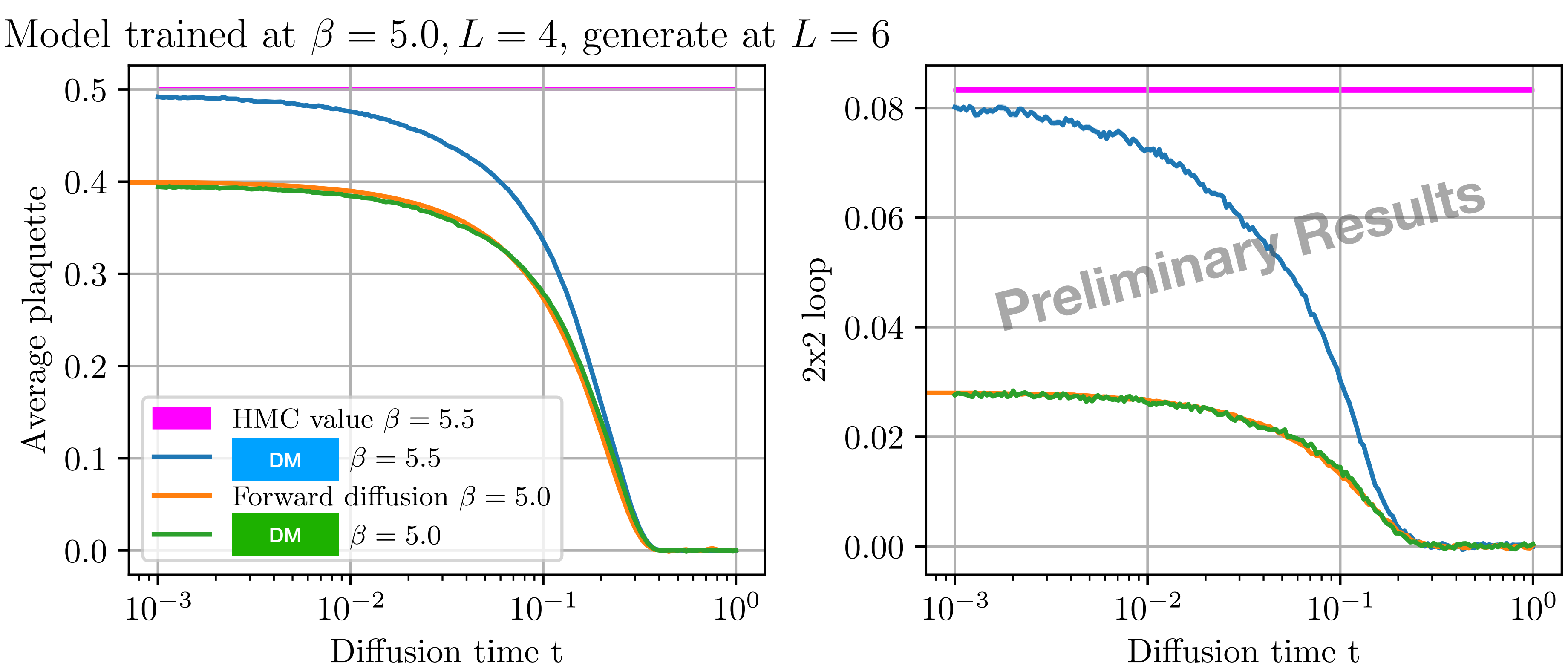
DM-QFT [260x.xxx [hep-lat]]

4D SU(3)



$V = 4^4, \beta_0 = 5$

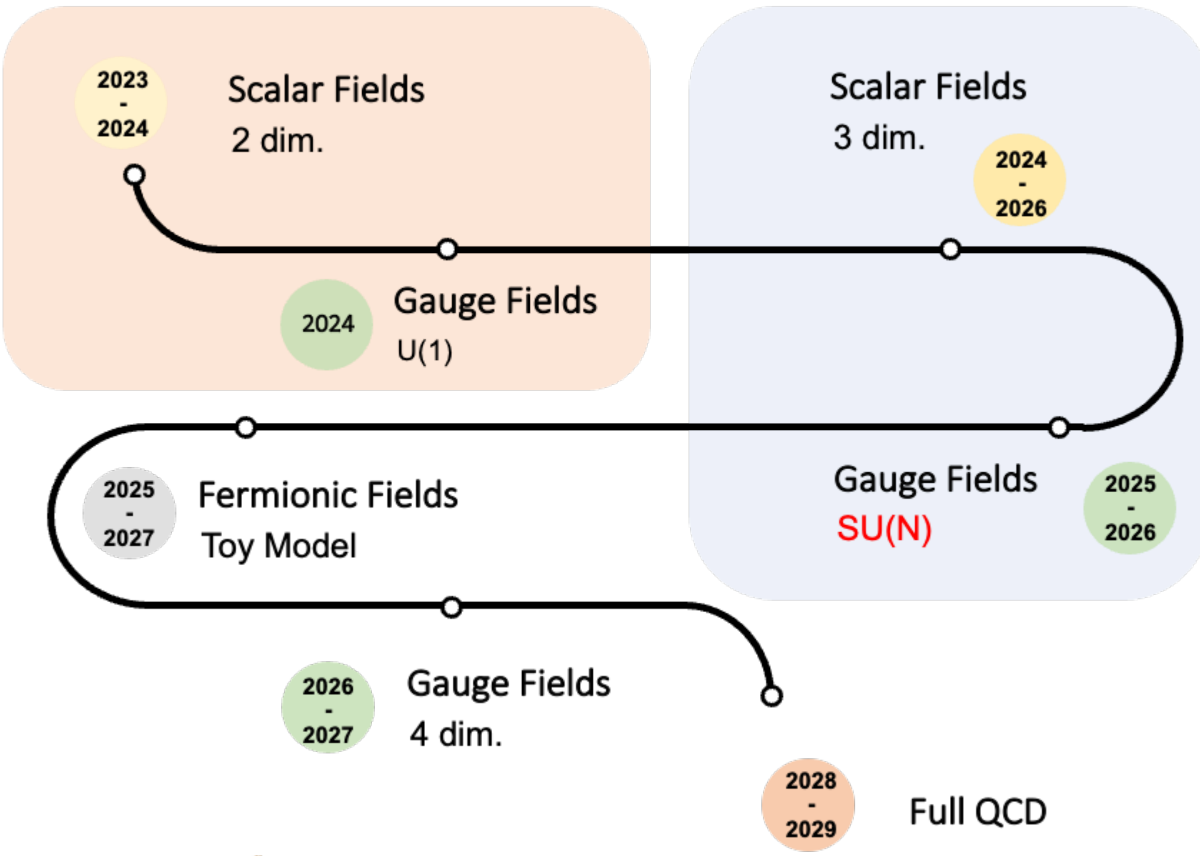
$V = 4^4, \beta = 5.5$

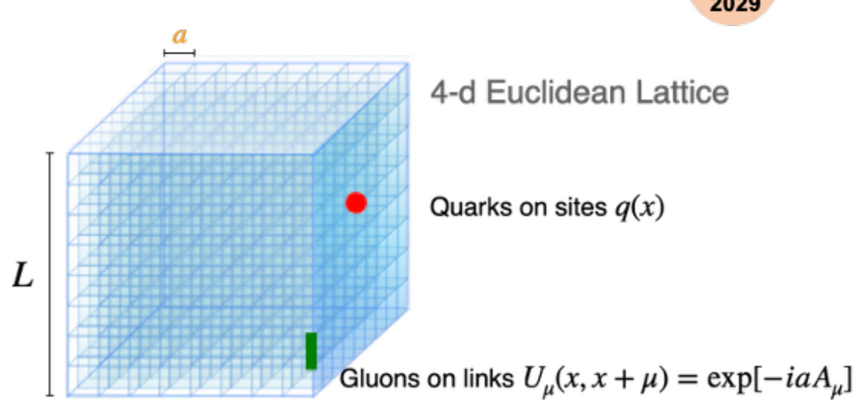


$V = 6^4, \beta = 5.5$

Roadmap toward full QCD

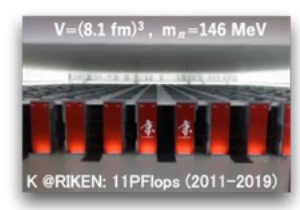
Where shall we go

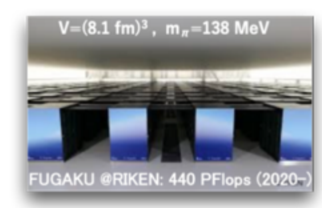




Faster and Reliable Simulation

- **Systematical 4D Gauge Field Simulations**
- **Evaluations and Improvements**
  - Faster Generation
- **Fermions**
  - Pseudo-Fermion, ect.
- **Sign Problem**
  - Complex Langevin Method, etc.





Simulations	Lattice QCD
Size	96×96×96×96
Supercomputer	Fugaku
Resource Consumption	**512 cores**
Time Consumption	**~ 1 day/configuration**

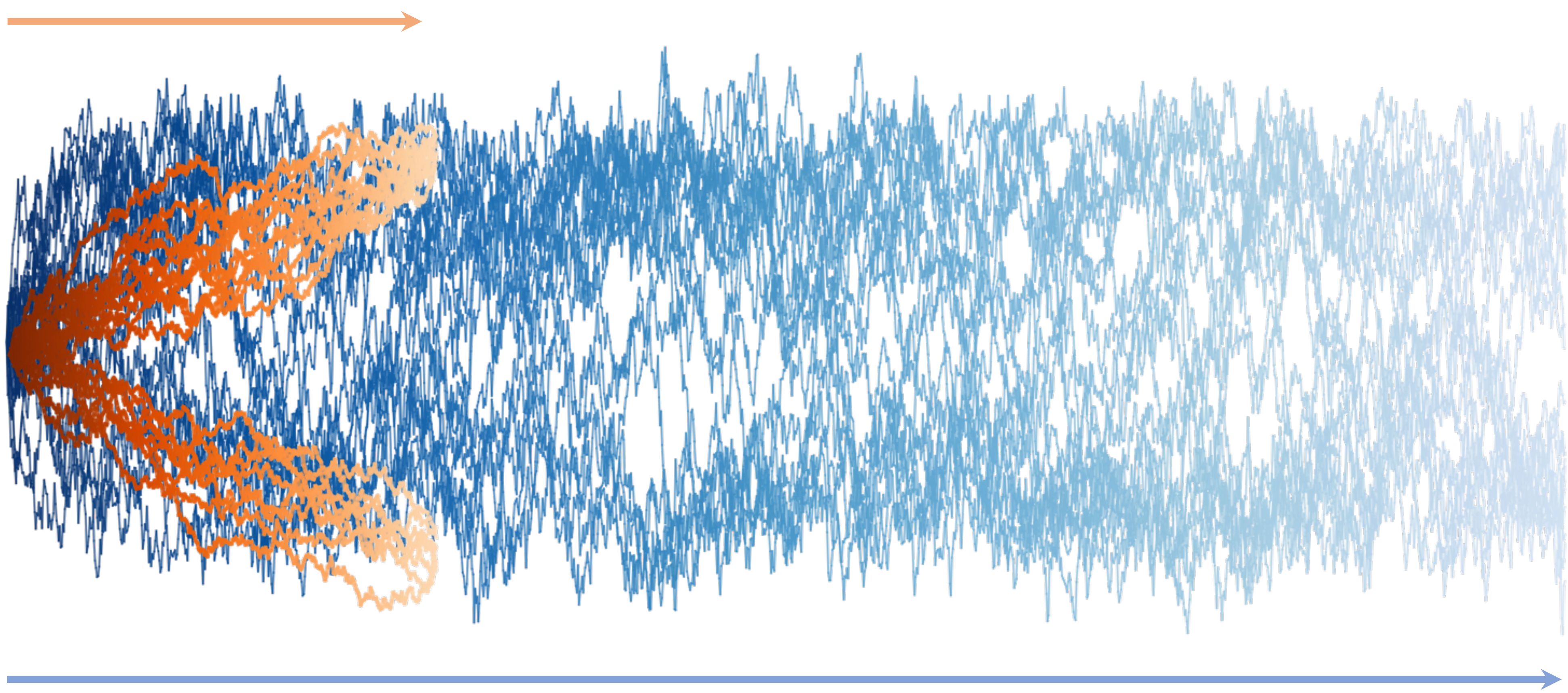
3-5 years to 1-2 days

Optimal Stochastic Quantization

LW [260x.xxx [hep-lat]]

Denoising Sampler

$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0, T]$$



$$d\phi_t = -\frac{\delta S}{\delta \phi}dt + g dw_t, \quad t \in [0, \infty]$$

Stochastic Quantization

Optimal Stochastic Quantization

Denoising Sampler

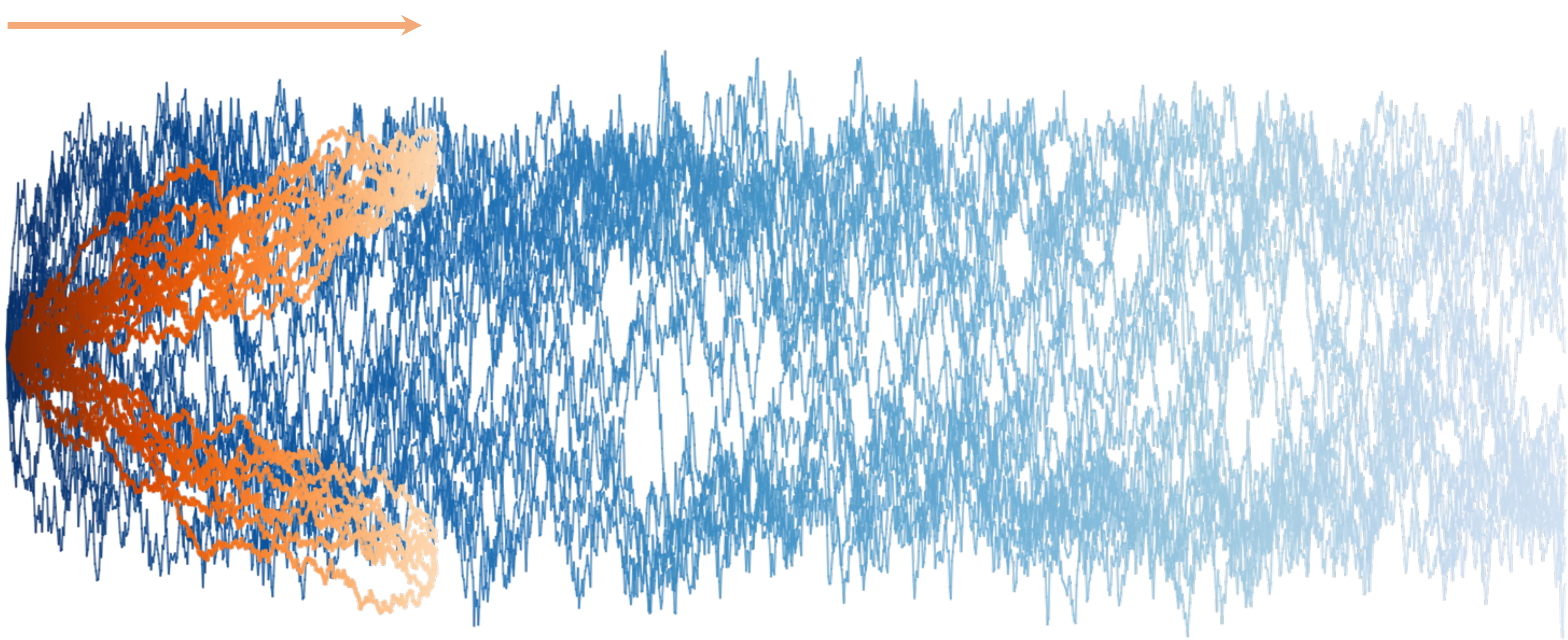
$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0,T]$$

Physics-driven Decomposition

$$u_\theta(\phi_t, t) \equiv -\delta S/\delta\phi + v_\theta(\phi_t, t)$$

Optimization Objective

$$\mathscr{L}(\theta) = \mathbb{E}_{\phi_0 \sim p_0}\left[\frac{1}{2}\int_0^T g^{-2}|v_\theta(\phi_t, t)|^2 dt + S(\phi_T)\right]$$



Optimal Stochastic Quantization

Learning a residual force for finite-time equilibration

$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0,T]$$

Physics-driven Decomposition

$$u_\theta(\phi_t, t) \equiv -\delta S/\delta\phi + v_\theta(\phi_t, t)$$

$$\pi \xrightarrow{f+v_\theta,\ 0\le t\le T} q_T^\theta \approx p_{\text{target}}$$

$$\mathbb{P}_{\text{ref}} : d\phi_t = f(\phi_t)\, dt + g\, dw_t, \quad p_0 = \pi$$
$$\mathbb{P}_\theta : d\phi_t = u_\theta(\phi_t, t)\, dt + g\, dw_t, \quad p_0 = \pi$$

$$\mathcal{L}_{\text{Fisher}}(\theta) = \text{KL}(\mathbb{P}_\theta \| \mathbb{P}_{\text{ref}})$$

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{Fisher}}(\theta) + \mathbb{E}_{\mathbb{P}_\theta}\left[S[\phi_T]\right]$$

Optimization Objective

$$\mathcal{L}(\theta) = \mathbb{E}_{\phi_0 \sim p_0}\left[\frac{1}{2}\int_0^T g^{-2} |v_\theta(\phi_t, t)|^2 dt + S(\phi_T)\right]$$

Running cost: keep close to physical SQ
Terminal cost: drive endpoint to low-action region

No density q_t^θ, **no score** $\nabla_\phi \log q_t^\theta$, **no divergence** $\nabla_\phi \cdot v_\theta$

$$v^*(\phi, t) = g^2 \nabla_\phi \log h_t(\phi)$$

$$h_t(\phi) = \mathbb{E}_{\text{ref}}\left[e^{-S[\phi_T]} \mid \phi_t = \phi\right]$$

Optimal residual drift is a Doob-transform force which comes from expected future Boltzmann weight.

Optimal Stochastic Quantization

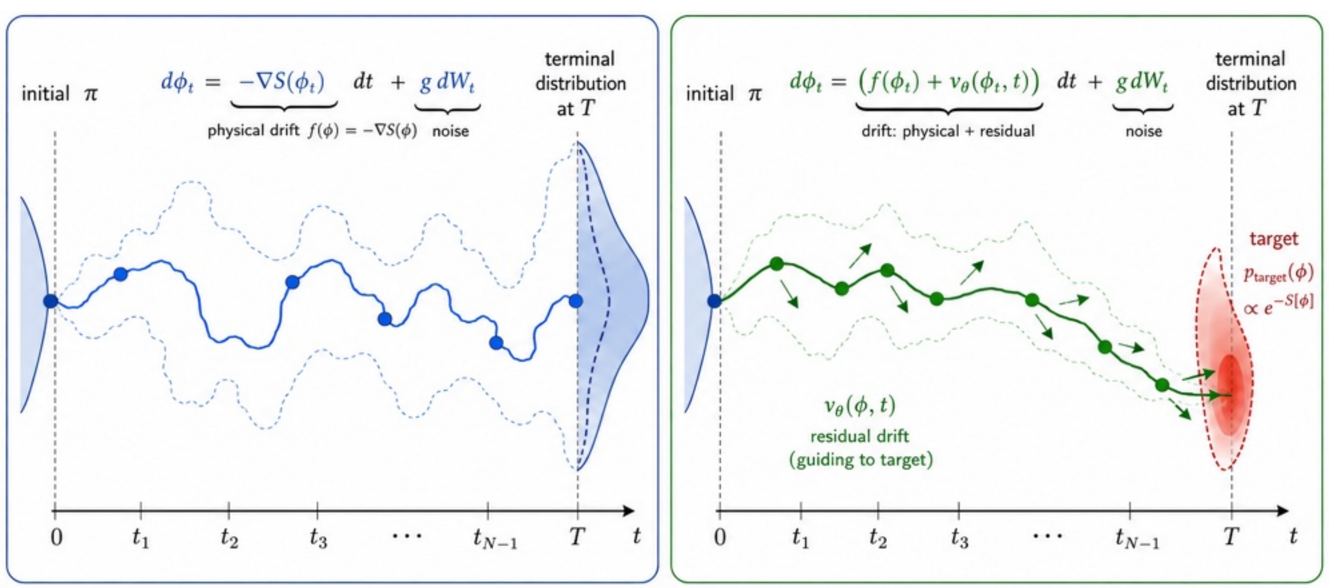
Learning a residual force for finite-time equilibration

$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0,T]$$

Physics-driven Decomposition

$$u_\theta(\phi_t, t) \equiv -\delta S/\delta\phi + v_\theta(\phi_t, t)$$

$$\pi \xrightarrow{f+v_\theta,\ 0\le t\le T} q_T^\theta \approx p_{\text{target}}$$



Optimization Objective

$$\mathcal{L}(\theta) = \mathbb{E}_{\phi_0\sim p_0}\left[\frac{1}{2}\int_0^T g^{-2}|v_\theta(\phi_t, t)|^2 dt + S(\phi_T)\right]$$

Running cost: keep close to physical SQ
Terminal cost: drive endpoint to low-action region

No density q_t^θ, **no score** $\nabla_\phi \log q_t^\theta$, **no divergence** $\nabla_\phi \cdot v_\theta$

$$v^*(\phi, t) = g^2 \nabla_\phi \log h_t(\phi)$$

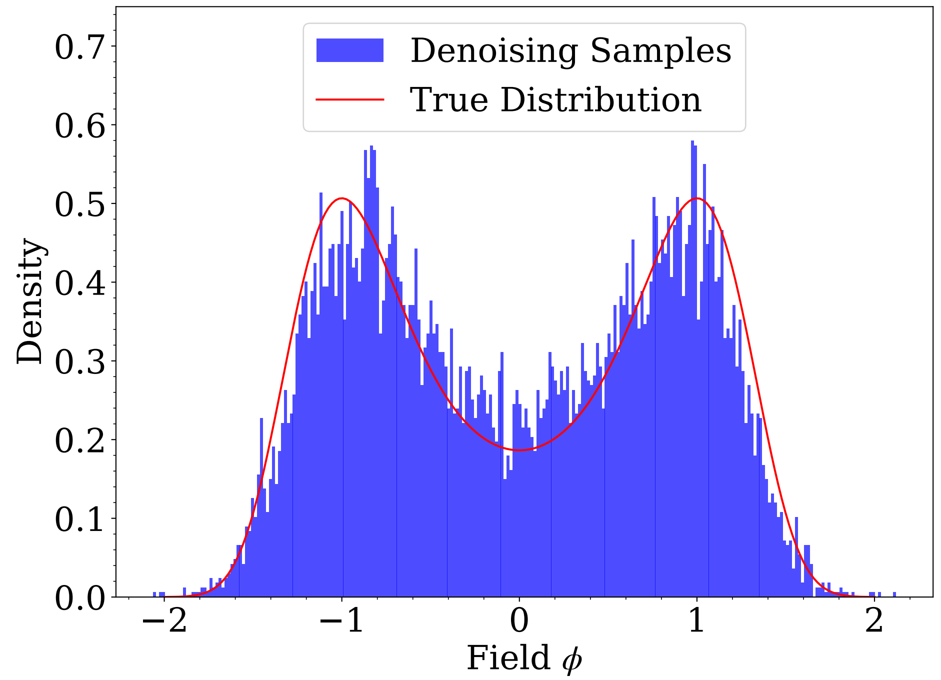
$$h_t(\phi) = \mathbb{E}_{\text{ref}}\left[e^{-S[\phi_T]} \mid \phi_t = \phi\right]$$

Optimal residual drift is a Doob-transform force which comes from expected future Boltzmann weight.

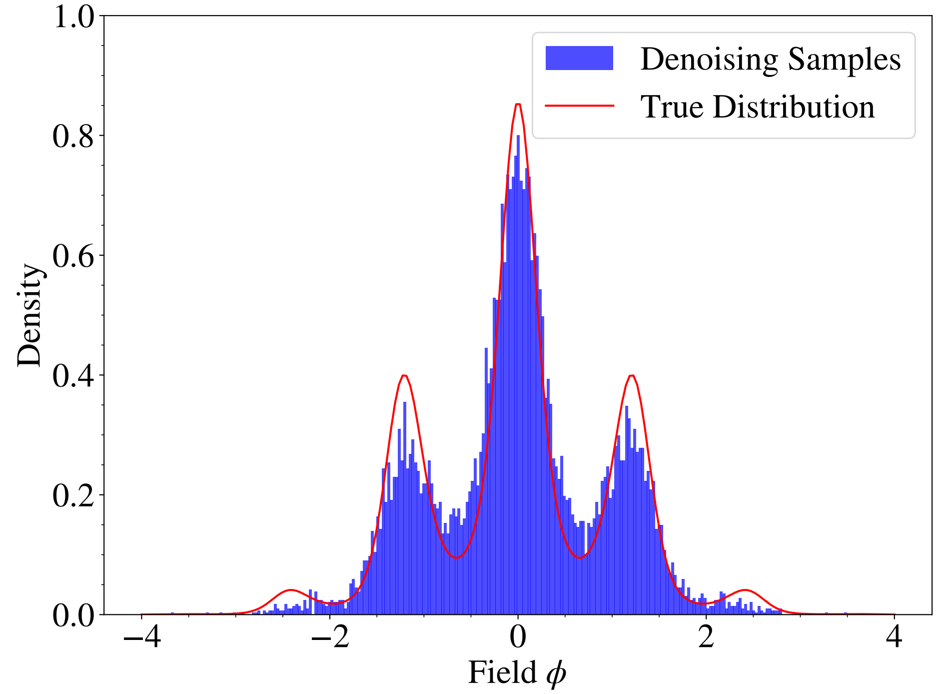
Denoising Sampler

0+0 dimension QFT

$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0,T]$$



$$S(\phi) = (\phi^2 - 1)^2$$



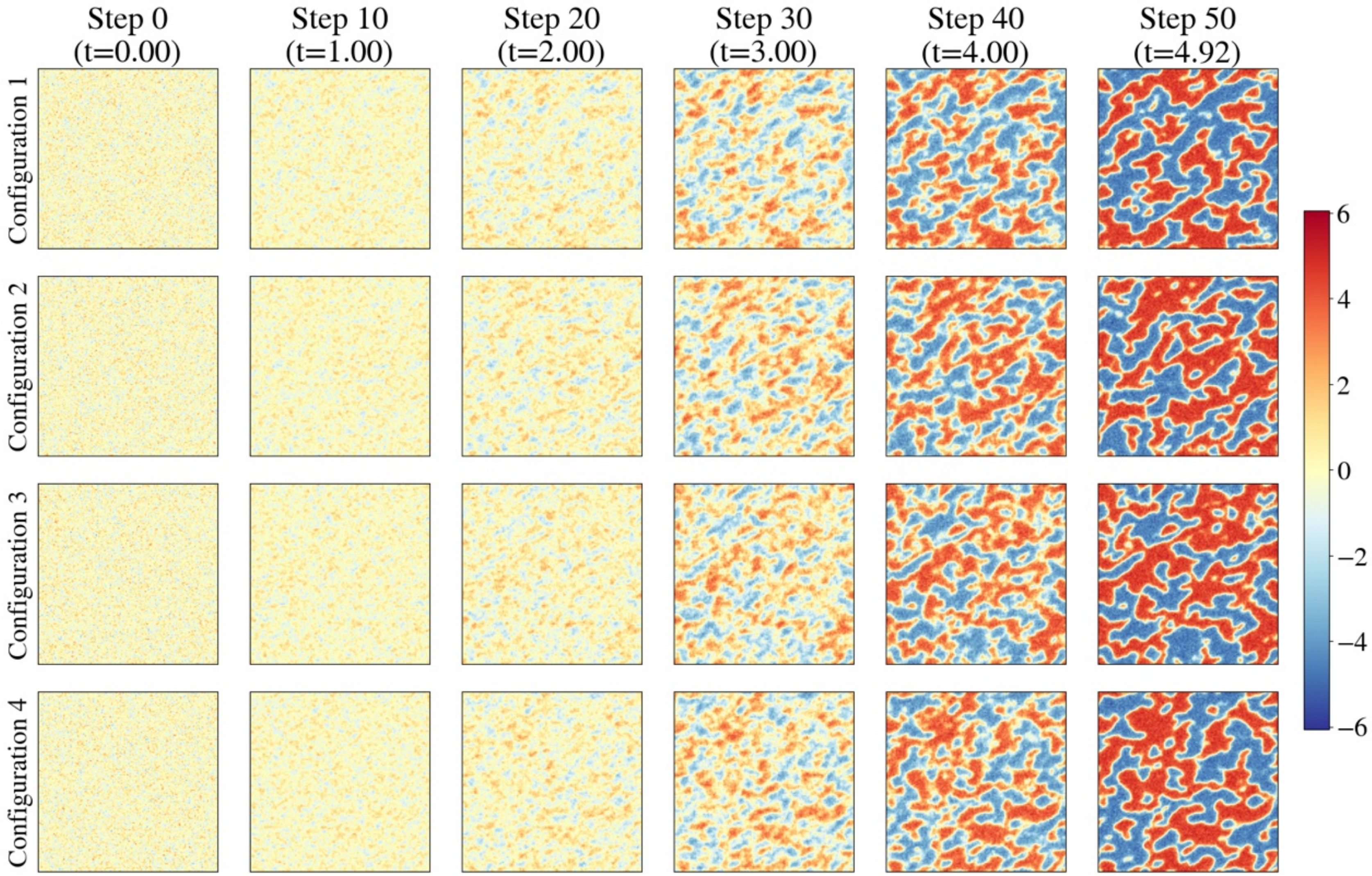
$$S(\phi) = \frac{1}{2}\phi^2 - \cos(k\phi) \quad k = 5$$

Marginal density of $N = 10^4$ denoising samples (blue)
compared with the exact target distributions
$p(\phi) \propto e^{-S(\phi)}$ (red)

Denoising Sampler

Lattice Field Theory

$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0, T]$$



Evolution of a single 128×128 lattice field configuration during the denoising process, at $\kappa = 0.5, \lambda = 0.022$ (broken phase)

Optimal Stochastic Quantization

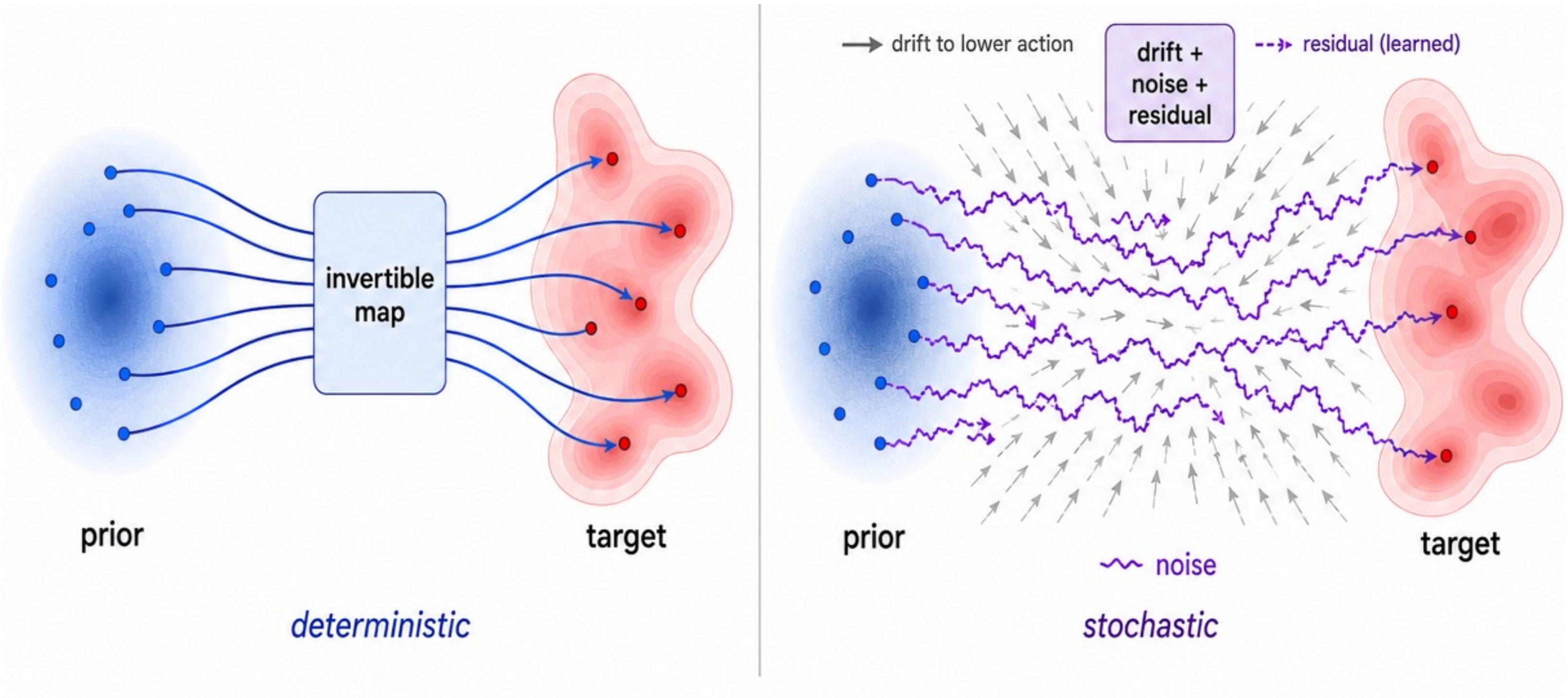
Why Stochasticity Matters

$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0,T]$$

Stochasticity generates **quantum fluctuations** that prevent the system from **collapsing** to *a classical saddle point*.

Diffeomorphism of **Flow** makes **Jacobian** reach **singular** somewhere **when** *sample multi-mode systems*.

Fisher information changes **monotonically,** which keeps the **ergodicity**.

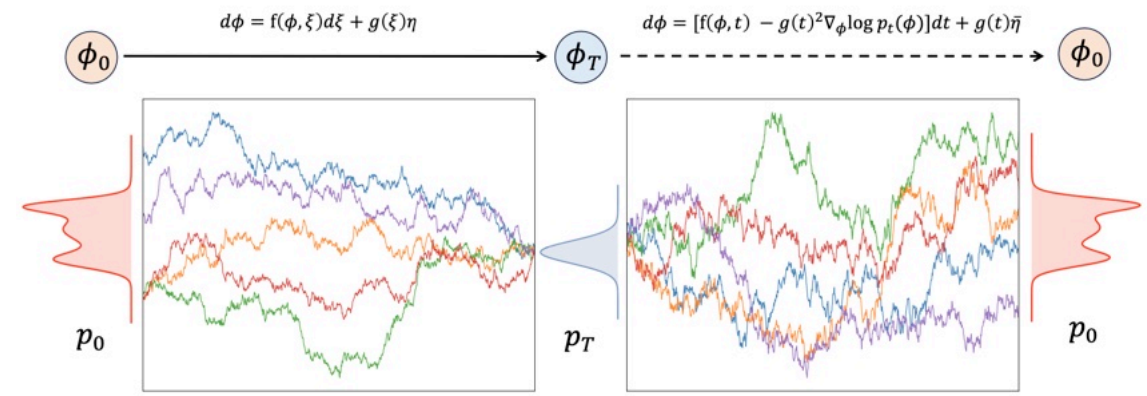


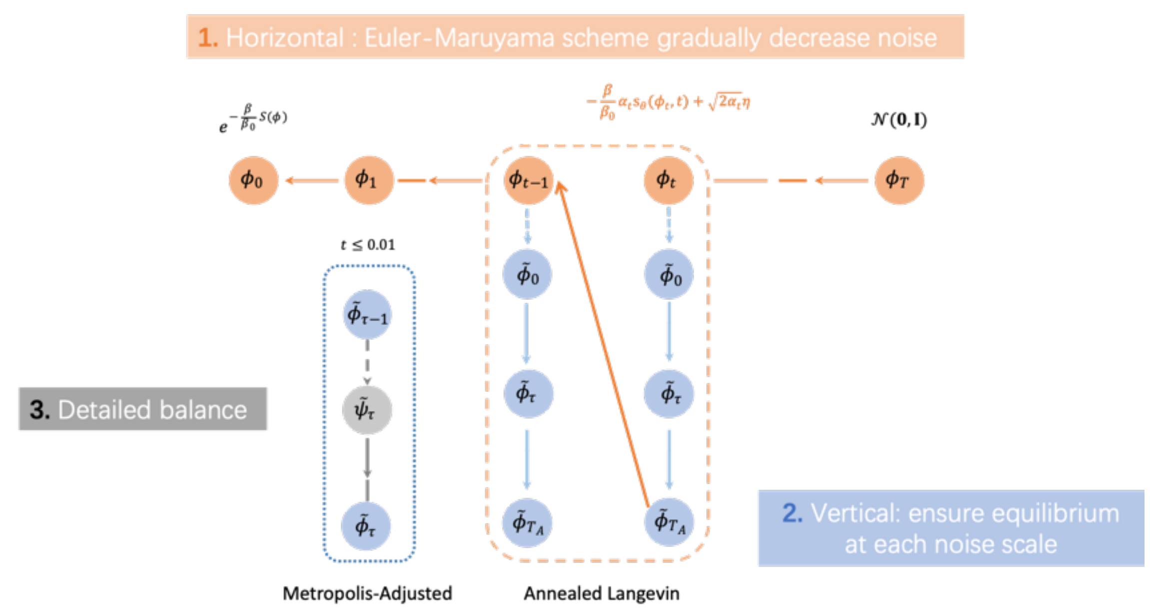
Summary

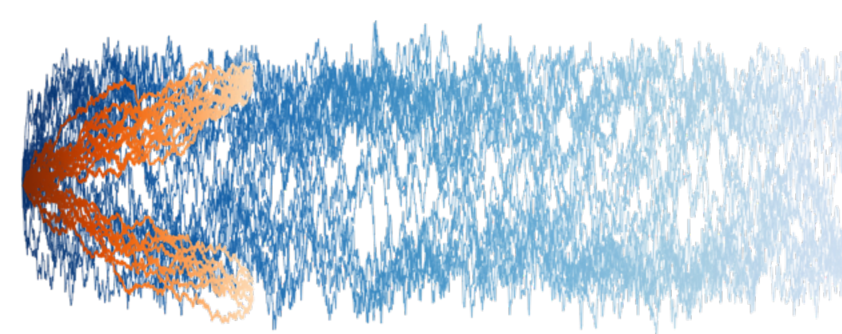
Take-home Messages

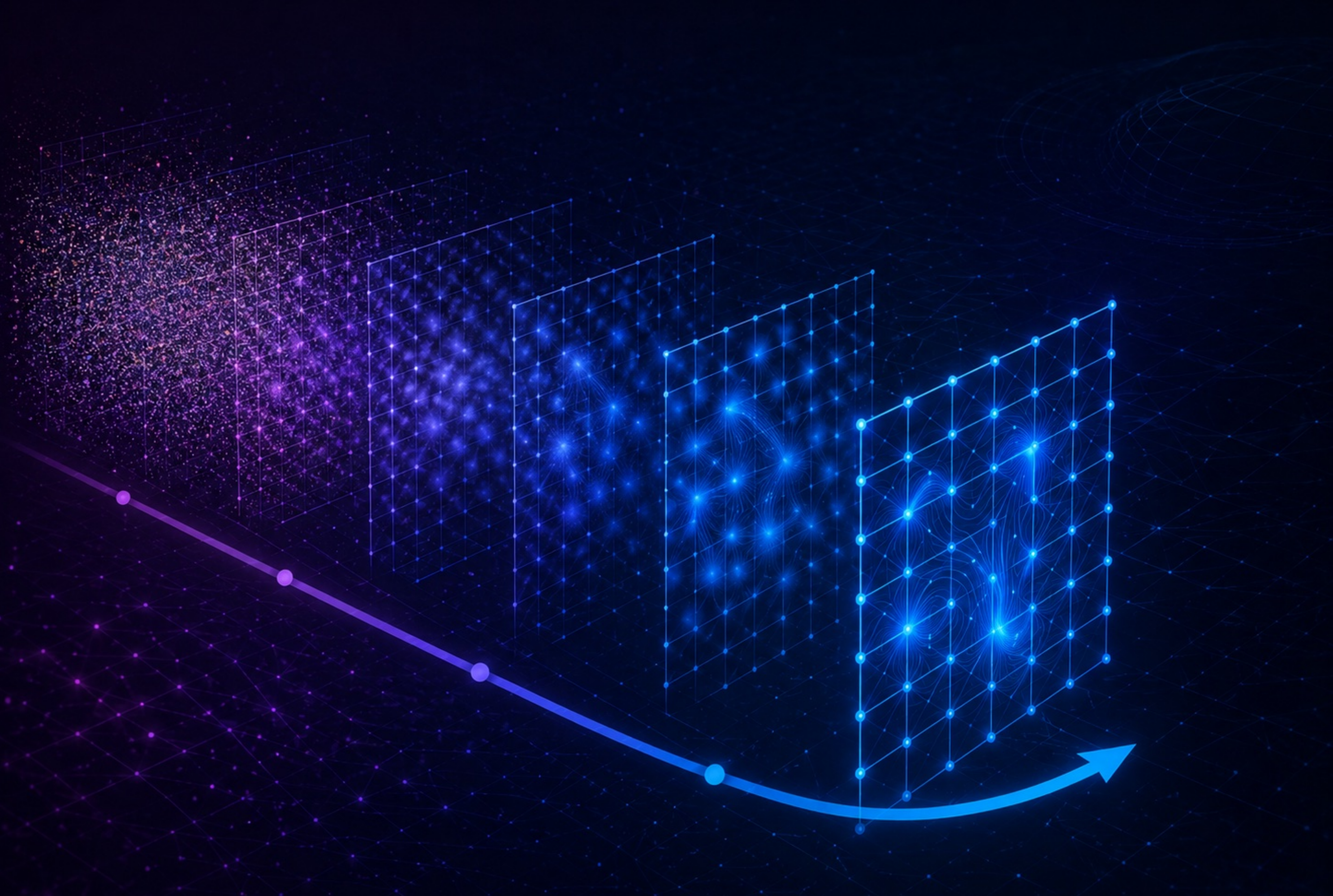
$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)} + \eta(x,\tau)$$

- **Physicists can understand DM well**
  - Langevin process
  - Diffusion models as SQ
- **Lattice fields can be well-simulated**
  - Scalar and Gauge Fields
  - From 2D to 4D
- **Learn-to-Sample**
  - Starting from existing configs
  - **Expandable, Exact, Efficient**
- **Optimal Stochastic Quantization**









DM meets LQFT, opportunities and challenges

Architecture

Y Tang, **LW**, etc., arXiv:2606.xxx

NSCN++ for Scalar Fields

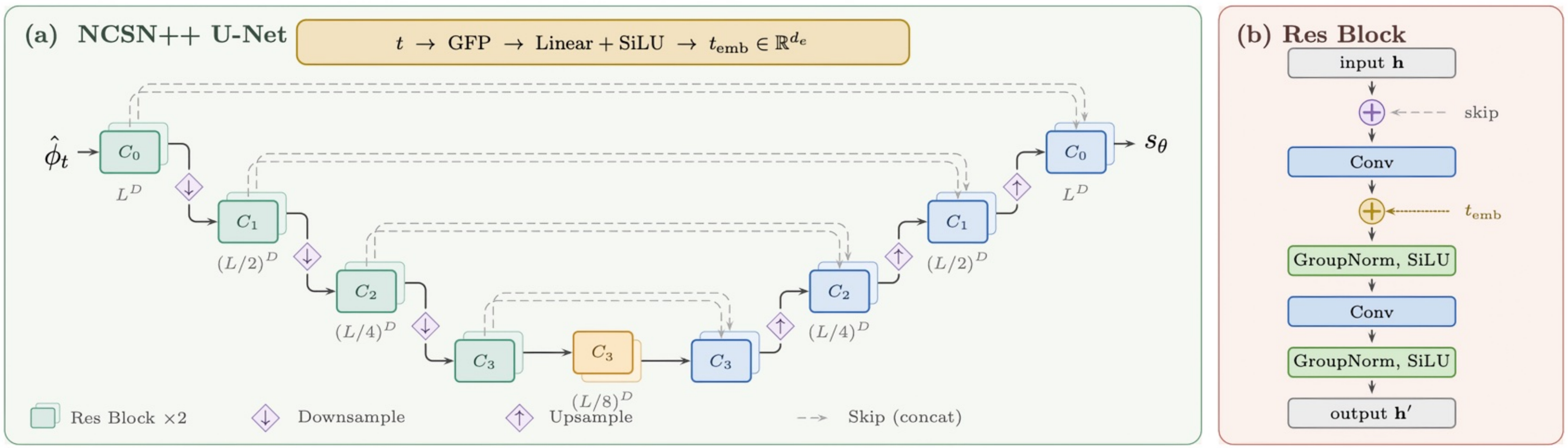
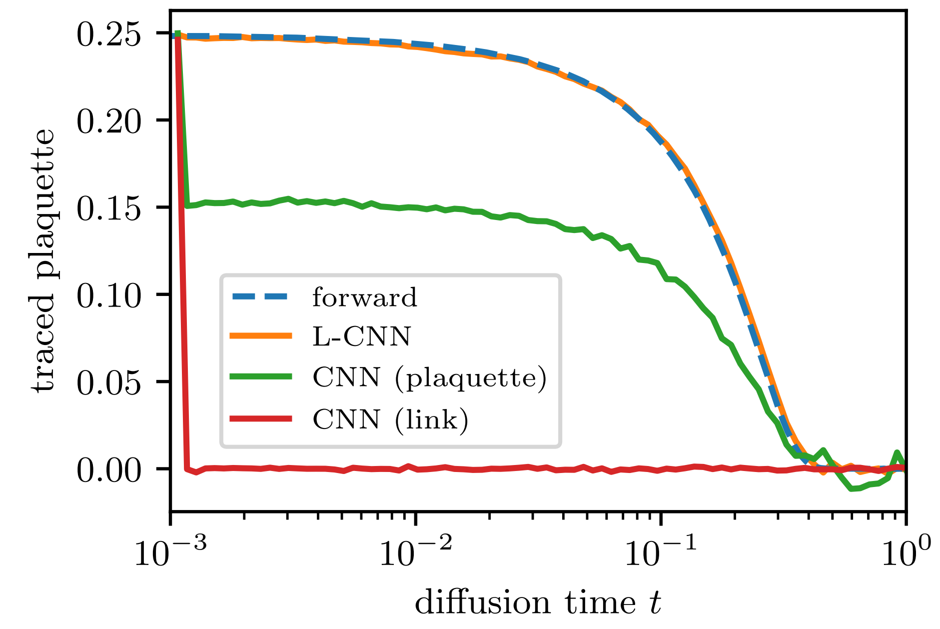


FIG. 31. Modified NCSN++ score network used for the lattice ϕ^4 ensembles. Panel (a) shows the U-Net map from a noised field on a periodic L^D lattice to a score field on the same lattice. The encoder applies three stride two downsamplings. For the larger lattices that are multiples of eight, this reduces the spatial resolution to $(L/8)^D$. The smallest training lattice $L = 4$ is handled by the same map, with its bottleneck reduced to a single site (App. E 1). The decoder restores the original resolution using features stored at the corresponding encoder scales. The fully convolutional map is therefore evaluated with shared weights on all lattice sizes used in the cross-L tests. The scalar diffusion time t is encoded by Gaussian Fourier projection and injected into every block. The final output is divided by the VE marginal width $\Sigma(t)$. Panel (b) shows the time conditioned convolutional block. It follows the two convolution layout and time conditioning of the NCSN++ ResBlock, with the internal residual addition omitted for training stability. All spatial convolutions use circular padding, matching the periodic boundary conditions of the lattice theory.

Architecture



L-CNN for Gauge Fields

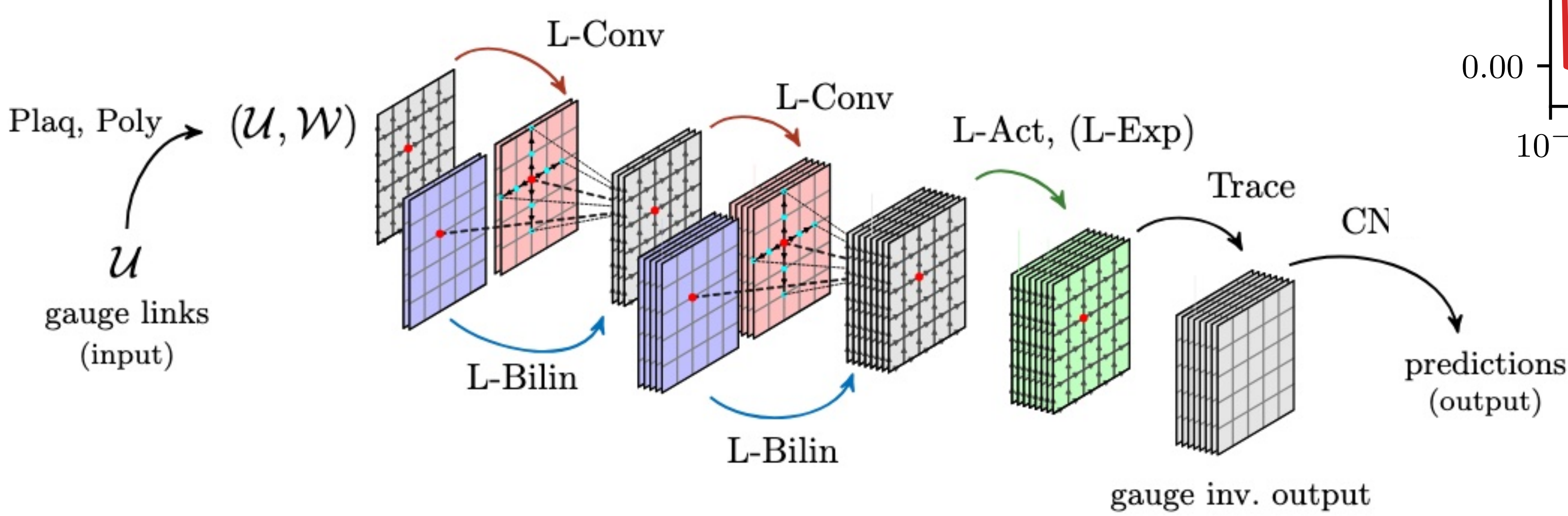


Figure 2. Schematic representation of a lattice gauge-equivariant convolutional neural network (L-CNN). This architecture processes data defined on a lattice, representing quantum field theory in discretized space-time. Each layer in the L-CNN is carefully constructed to preserve gauge symmetry, a property crucial for ensuring physically consistent predictions in quantum field theory. The network first processes the input lattice data given by gauge links $\mathscr{U}$ using simple Wilson loops on the lattice such as plaquettes (Plaq) or Polyakov loops (Poly) to create objects $\mathscr{W}$ that transform locally. These objects are then combined into progressively more complex Wilson loops while maintaining the symmetry through specialized convolutional (L-Conv) and bilinear (L-Bilin) operations. Additional gauge equivariant activation functions (L-Act) or exponentiation layers (L-Exp) can modify the local fields in a gauge equivariant manner. Finally, the network generates gauge-invariant outputs through a trace layer (Trace) that can be processed by standard convolutional layers to produce the desired physical predictions. Unlike conventional CNNs, this design is robust to random and adversarial gauge transformations, making it essential for simulations of fundamental physics. Image from[33].

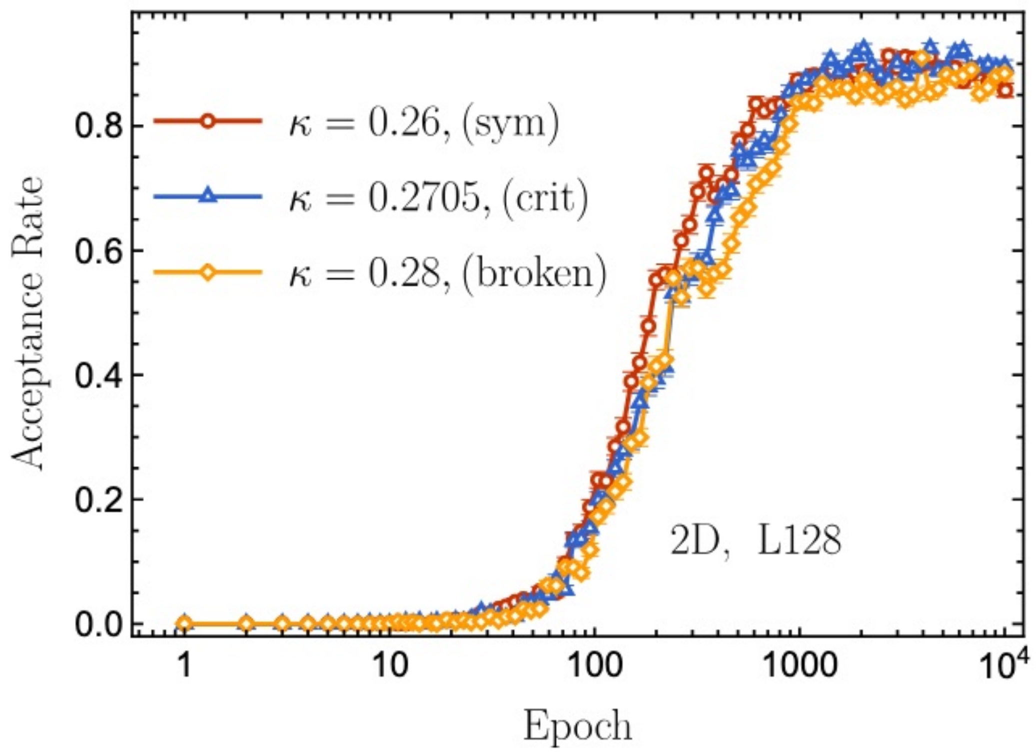
G. Aarts, K. Fukushima, T. Hatsuda, A. Ipp, S. Shi, **L. Wang***, and K. Zhou,
Physics-Driven Learning for Inverse Problems in Quantum Chromodynamics,
Nature Reviews Physics volume 7, pages154–163 (2025)

Efficiency

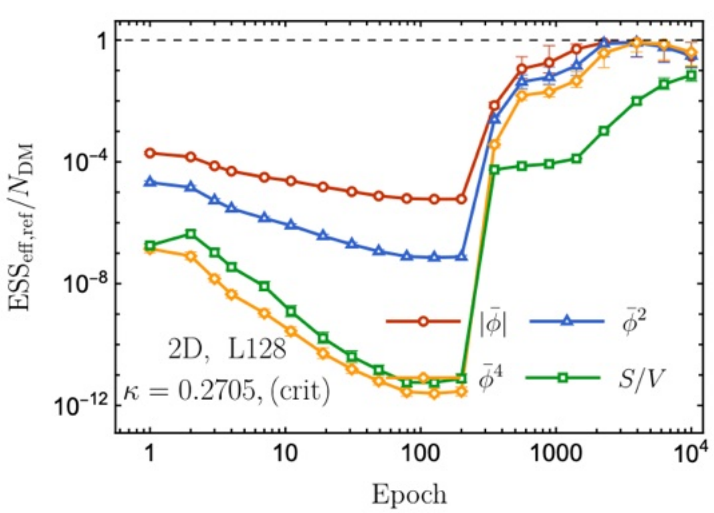
Y Tang, **LW**, etc., arXiv:2606.xxx

2D ϕ^4

$L = 128, \lambda = 0.022$

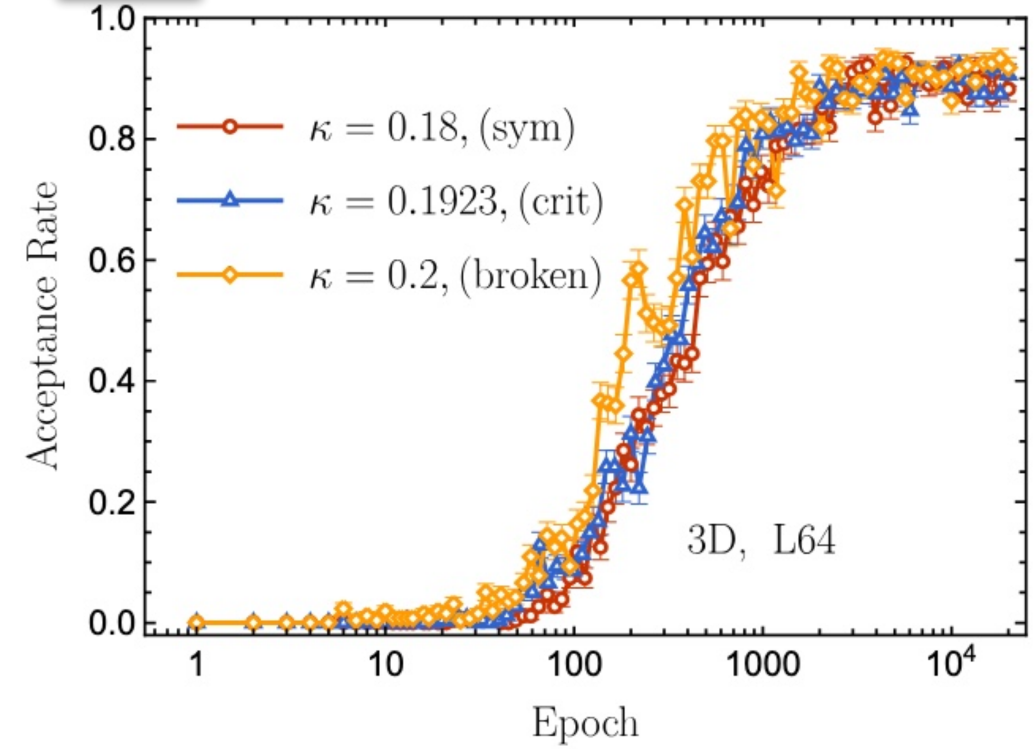


$c = 0.2, t_{mh} = 10^{-4}, N = 1024$

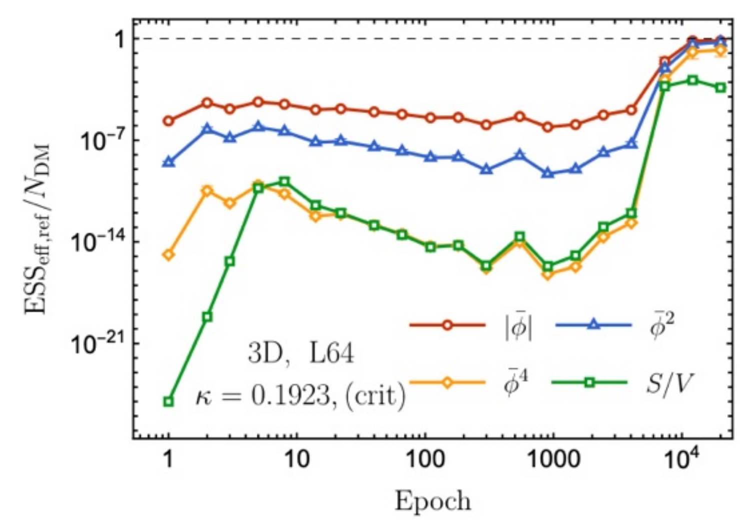


3D ϕ^4

$L = 64, \lambda = 0.9$



$c = 0.5, t_{mh} = 10^{-4}, N = 256$



Good Global Sampling can always reduce the auto-correlation time

Efficiency

Q Zhu, G Aarts, W Wang, K Zhou, **LW**, JHEP(2026)

Metropolis-Adjusted Annealed Langevin Sampler

U(1) in two dim.

- Autocorrelation time

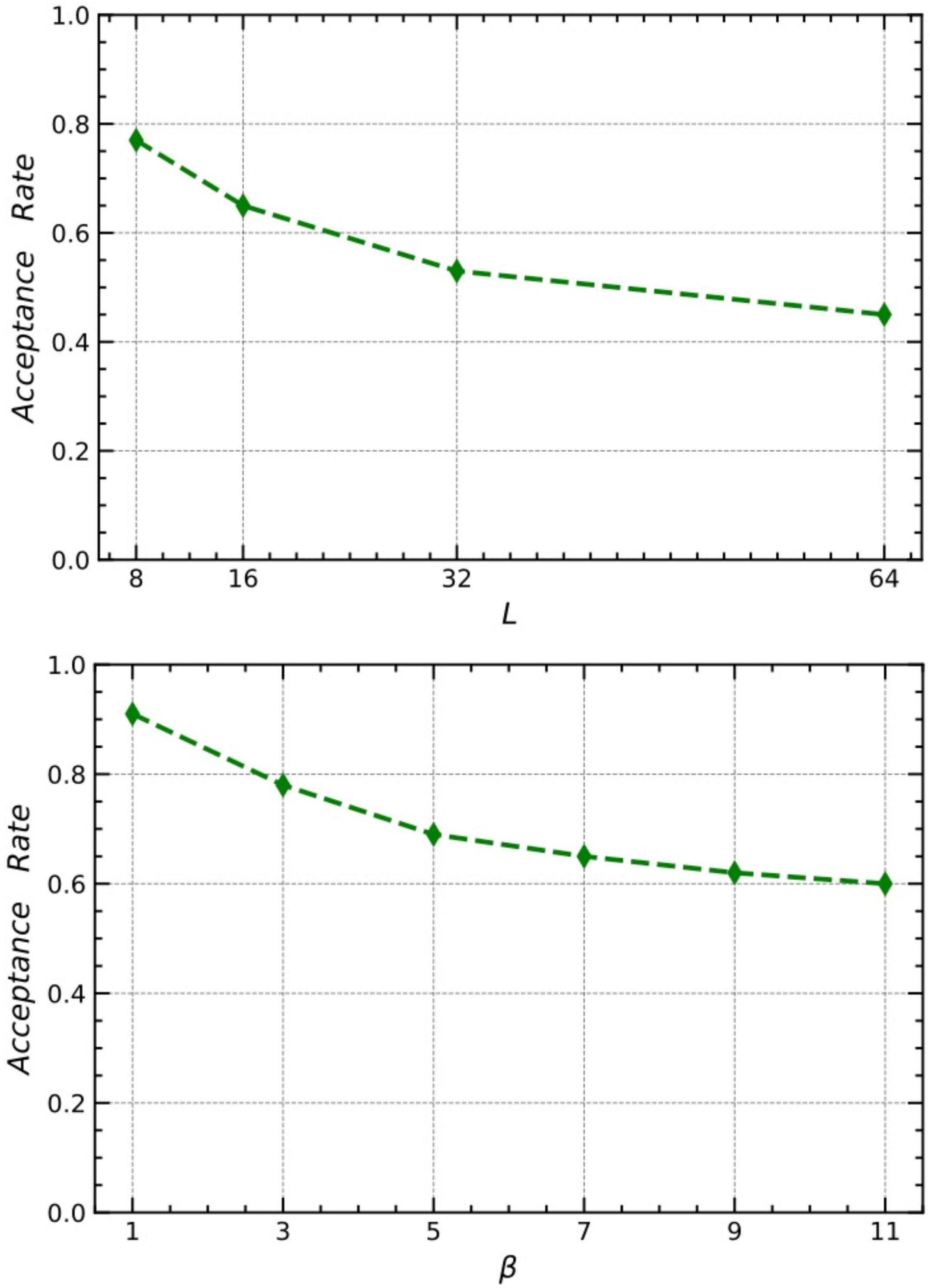
$$C_O(t) = \left\langle O_{t_0} O_{t_0+t} \right\rangle - \left\langle O_{t_0} \right\rangle \left\langle O_{t_0+t} \right\rangle$$

- Integrated autocorrelation time

$$\Gamma_O(t) \equiv \frac{C_O(t)}{C_O(0)}$$

$$\tau_O^{int} = \frac{1}{2} + \sum_{t=1}^{N} \Gamma_O(t)$$

J. Statist. Phys. 50 (1988) 109.



Efficiency

Q Zhu, G Aarts, W Wang, K Zhou, **LW**, JHEP(2026)

Metropolis-Adjusted Annealed Langevin Sampler

U(1) in two dim.

- Autocorrelation time

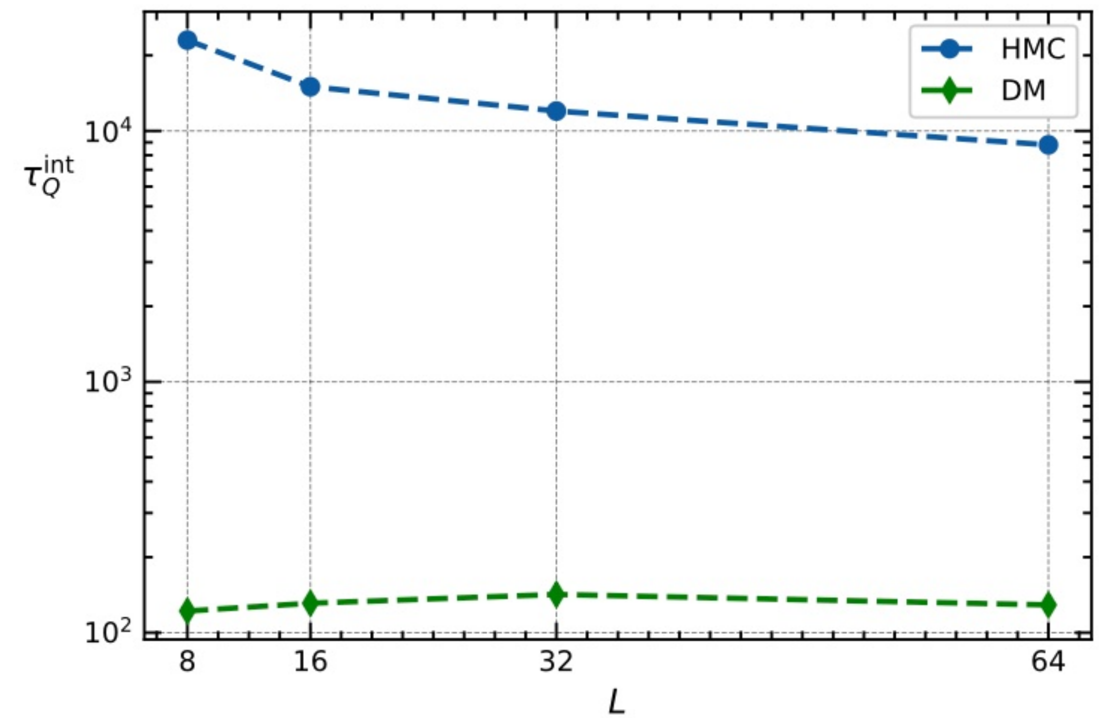
$$C_O(t) = \left\langle O_{t_0} O_{t_0+t} \right\rangle - \left\langle O_{t_0} \right\rangle \left\langle O_{t_0+t} \right\rangle$$

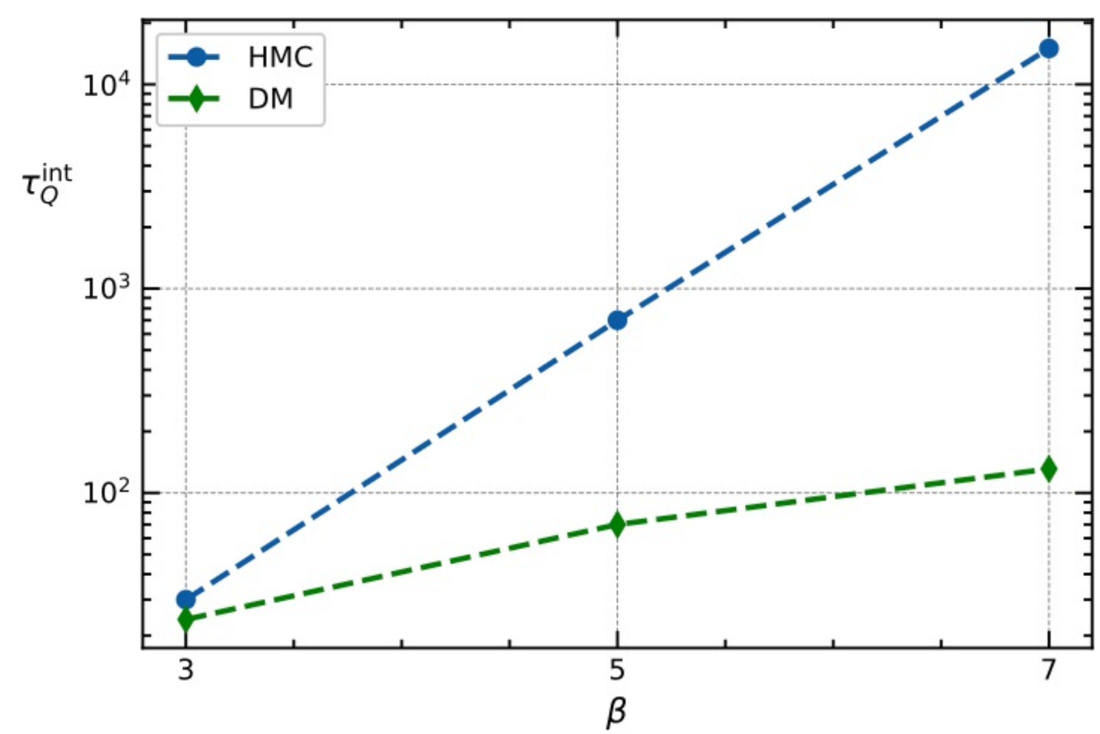
- Integrated autocorrelation time

$$\Gamma_O(t) \equiv \frac{C_O(t)}{C_O(0)}$$

$$\tau_O^{int} = \frac{1}{2} + \sum_{t=1}^{N} \Gamma_O(t)$$

J. Statist. Phys. 50 (1988) 109.





3E for U(1) Gauge Field

Q Zhu, G Aarts, W Wang, K Zhou, **LW**, JHEP(2026)

Comprehensive Comparison for 2D U(1)

Learned at $\beta = 1$ with **30k** cofigurations, L = 16

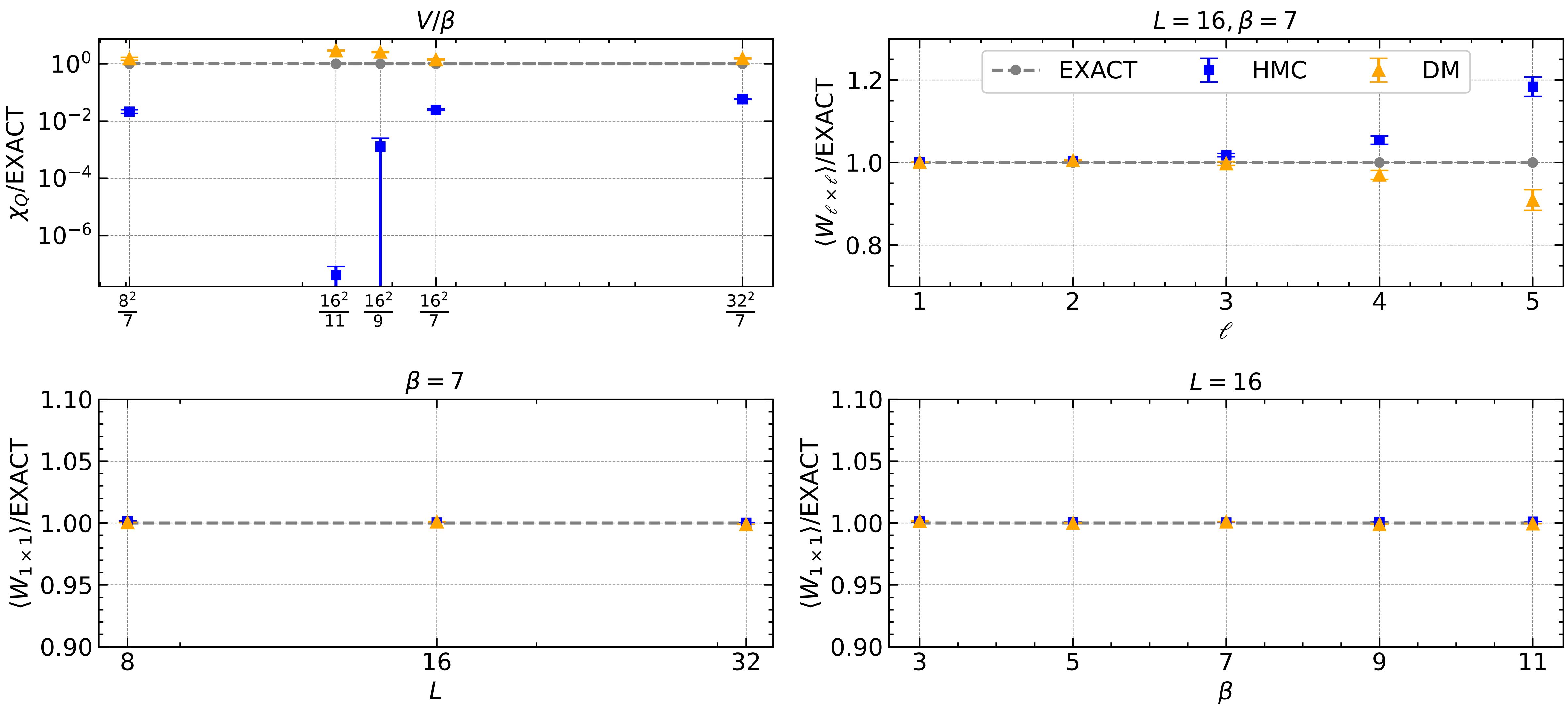


Table 1. Comparison of observables for $\beta = 1$ at different lattice sizes

Lattice Size (L)	1 × 1 Wilson Loop				Topological Susceptibility			
	HMC	DM	Langevin	Exact	HMC	DM	Langevin	Exact
8	0.447(72)	0.445(74)	0.443(80)	0.446	0.0402(17)	0.0413(18)	0.0418(18)	0.0406
16	0.447(37)	0.446(37)	0.444(36)	0.446	0.0416(16)	0.0422(17)	0.0421(20)	0.0406
32	0.446(18)	0.445(19)	0.445(18)	0.446	0.0428(19)	0.0415(18)	0.0412(17)	0.0406
64	0.446(9)	0.446(11)	0.445(9)	0.446	0.0426(19)	0.0427(20)	0.0420(19)	0.0406

Table 4. Comparison of observables for $L = 16$ at different couplings

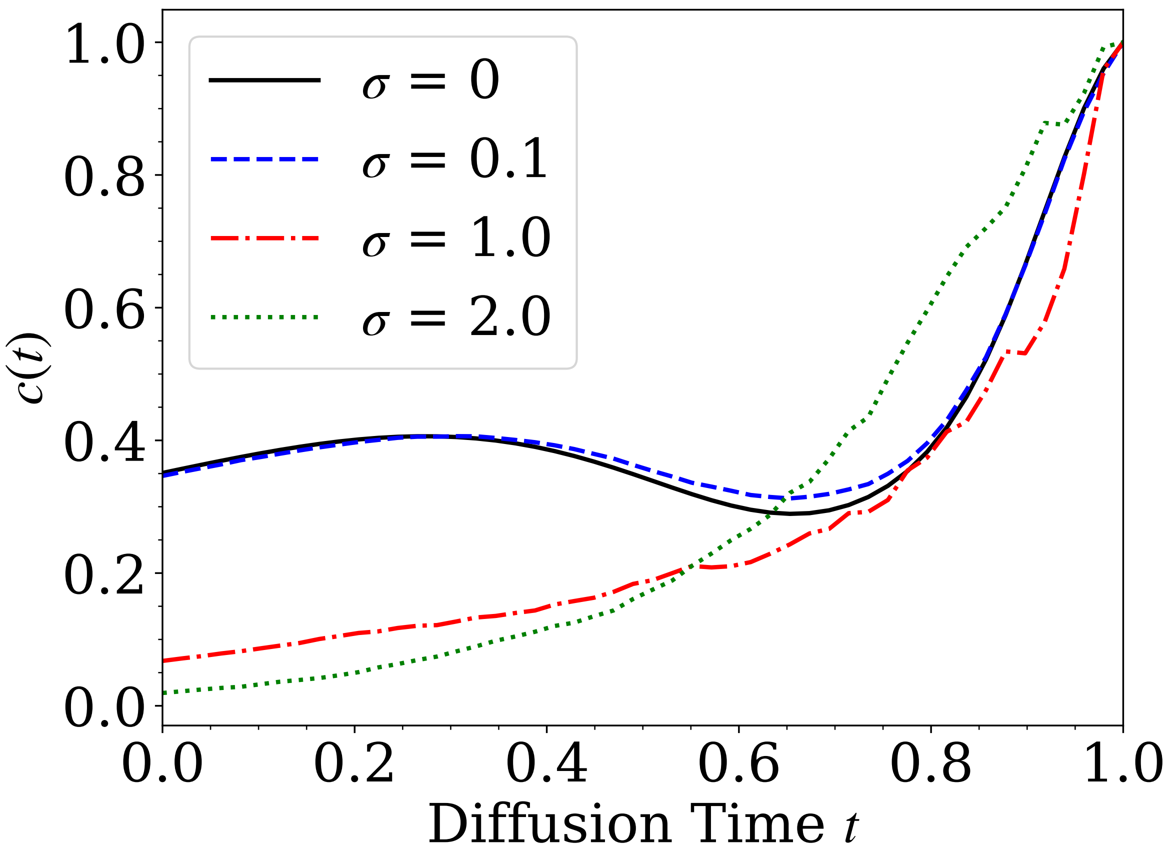
coupling (β)	1 × 1 Wilson Loop				Topological Susceptibility			
	HMC	DM	Langevin	Exact	HMC	DM	Langevin	Exact
3	0.811(17)	0.811(17)	0.809(17)	0.810	0.0096(4)	0.0114(6)	0.0106(14)	0.0111
5	0.894(9)	0.894(9)	0.891(10)	0.894	0.0048(2)	0.0058(5)	0.0075(3)	0.0057
7	0.926(7)	0.926(7)	0.924(6)	0.926	0.00013(2)	0.0045(5)	0.0131(5)	0.0039
9	0.944(3)	0.942(4)	0.940(6)	0.942	0	0.0031(4)	0.0154(7)	0.0029
11	0.954(3)	0.953(4)	0.950(5)	0.953	0	0.0025(3)	0.0165(13)	0.0024

Optimal Stochastic Quantization

Emergent Geometry

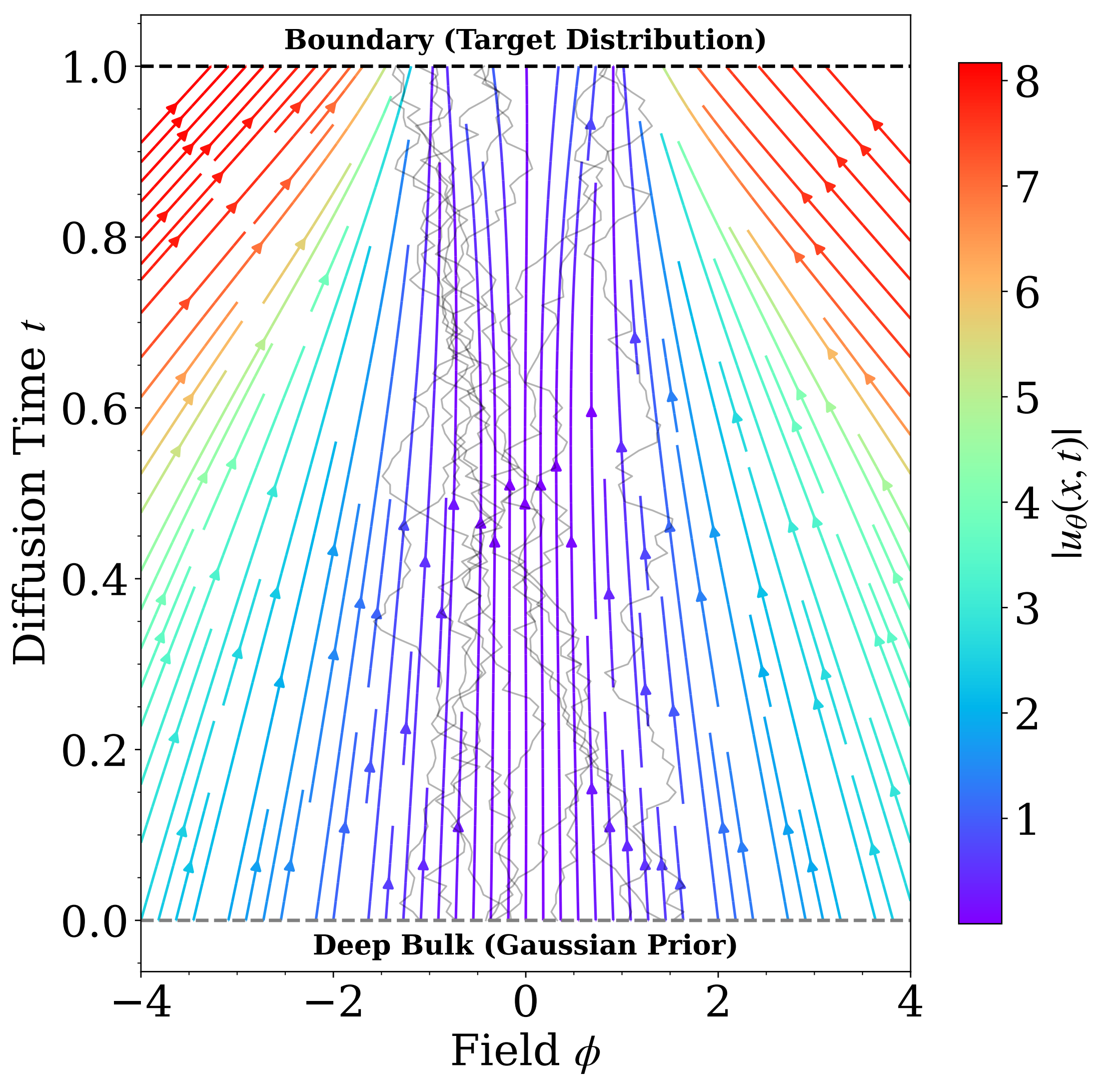
$$d\phi_t = u_\theta(\phi_t, t)dt + g dw_t, \quad t \in [0,T]$$

$$c(t) = \mathcal{N}\, \mathbb{E}_{q_t}\left[\, \|\nabla \ln q_t(\phi)\|^2 \right]$$



Keep monotonicity only when the noise is reasonable

$$S(\phi) = (\phi^2 - 1)^2$$



Back-ups

Diffusion Models

Score-based model

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Anderson, in Stochastic Processes and their Applications, 1982

- **Forward Diffusion SDE**
  - **Drift term**: pulls towards mode
  - **Diffusion term**: injects noise

$$\frac{d\phi}{d\xi} = \boxed{f(\phi, \xi)} + \boxed{g(\xi)\eta(\xi)}$$

- **Reverse Generative Diffusion SDE**
  - Drift term is adjusted with a "**Score Function**"
  - But how to get the score function ?

$$\frac{d\phi}{dt} = \boxed{\left[f(\phi, t) - g^2(t)\, \nabla_\phi \log p_t(\phi)\right]} + \boxed{g(t)\bar{\eta}(t)}$$

Model the score function with neural networks!

$$\frac{d\phi}{dt} = -\sigma^{2t} \mathbf{s}_{\hat{\theta}}(\phi, t) + \sigma^t \bar{\eta}(t)$$

$$\mathcal{L}_\theta = \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_0(\phi_0)} \mathbb{E}_{p_i(\phi_i|\phi_0)} \left[\left\| \mathbf{s}_\theta(\phi_i, \xi) - \nabla_{\phi_i} \log p_i(\phi_i \,|\, \phi_0) \right\|_2^2 \right]$$

One example:

Variance Expanding

$$f(\phi, \xi) = 0$$

$$g(\xi) = \sigma^\xi$$

$$\xi \in [0, T]$$

Diffusion Models

Probabilistic FlowODE

[Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021](#)

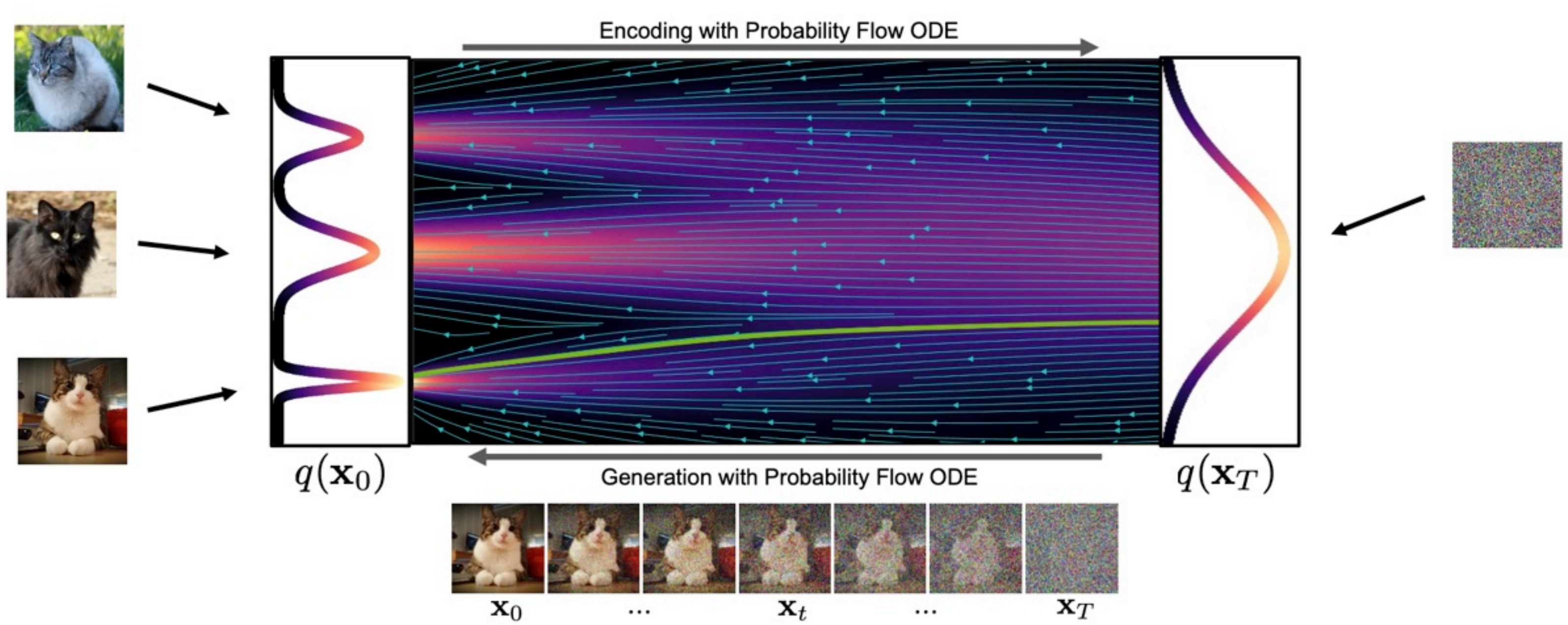
[Anderson, in Stochastic Processes and their Applications, 1982](#)

- **Forward Diffusion SDE**
  - **Drift term**: pulls towards mode
  - **Diffusion term**: injects noise

$$\frac{d\phi}{d\xi} = f(\phi, \xi) + g(\xi)\eta(\xi)$$

- **Reverse Generative Diffusion SDE**
  - Drift term is adjusted with a “**Score Function**”
  - Sampling from the SDE or ODE

$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\, \nabla_\phi \log p_t(\phi) \right] + g(t)\bar{\eta}(t)$$



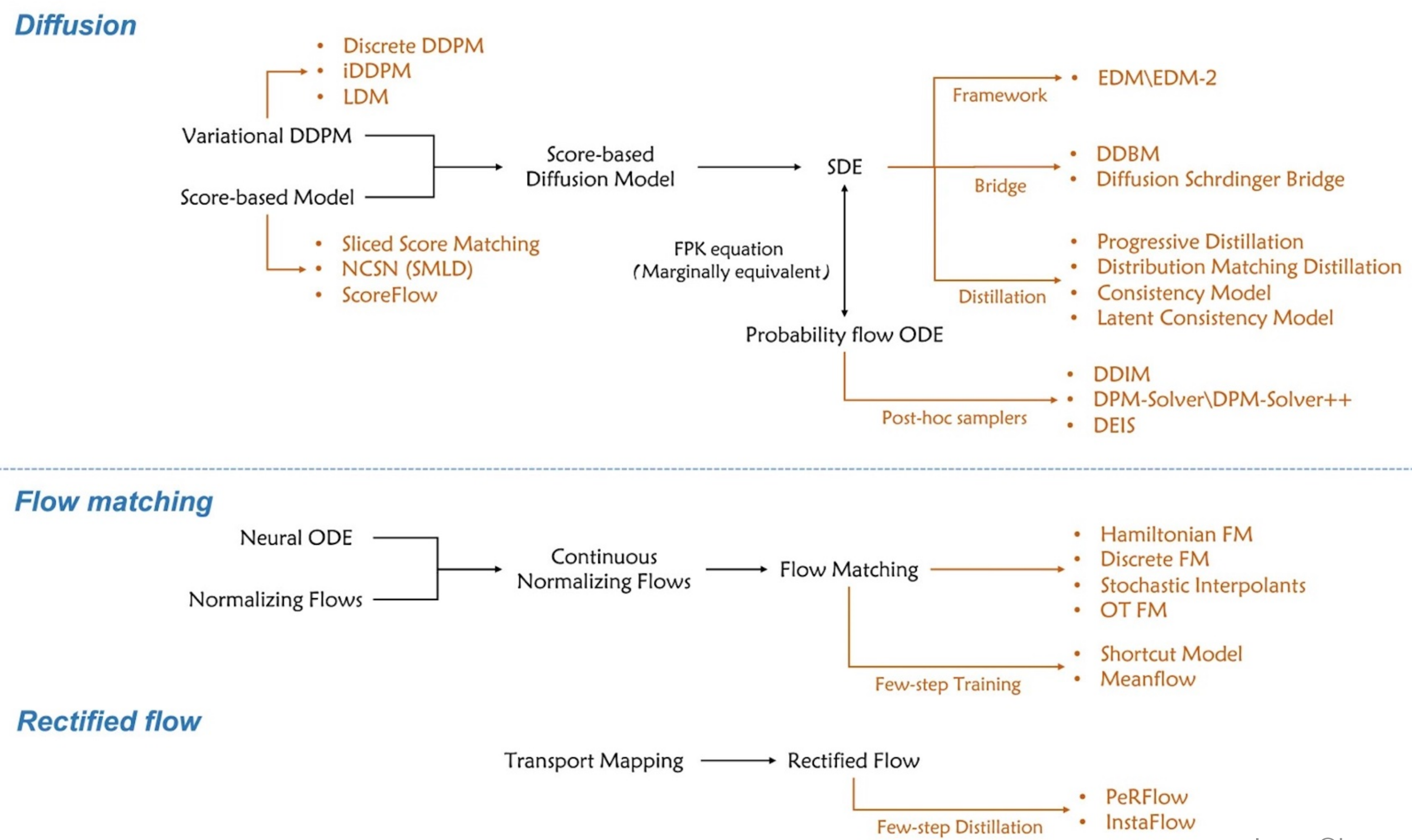
$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\, \nabla_\phi \log p_t(\phi) \right]$$

Probability FlowODE

- Enables use of advanced ODE solvers
- Exact Log-Likelihood computation
- **Recover the Flow-based models**

Diffusion Models

Probabilistic Flow Zoo



lenny@buaa.edu.cn

Physics of Diffusion Models

Slide from Gert@ETH, June 2025

Cumulants

- Correlations are being destroyed and retrieved
- If score is determined exactly, full theoretical control

Diffusion models: generation of correlations

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

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 $\qquad \tau = T - t$

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

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

BEST 'PHYSICS FOR AI' PAPER AWARD 🏅

Higher-order cumulants in diffusion models

Gert Aarts, Diaa Eddin Habibi, Lingxiao Wang, Kai Zhou

[paper] [poster]

Machine Learning and the Physical Sciences, NeurIPS 2024

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

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

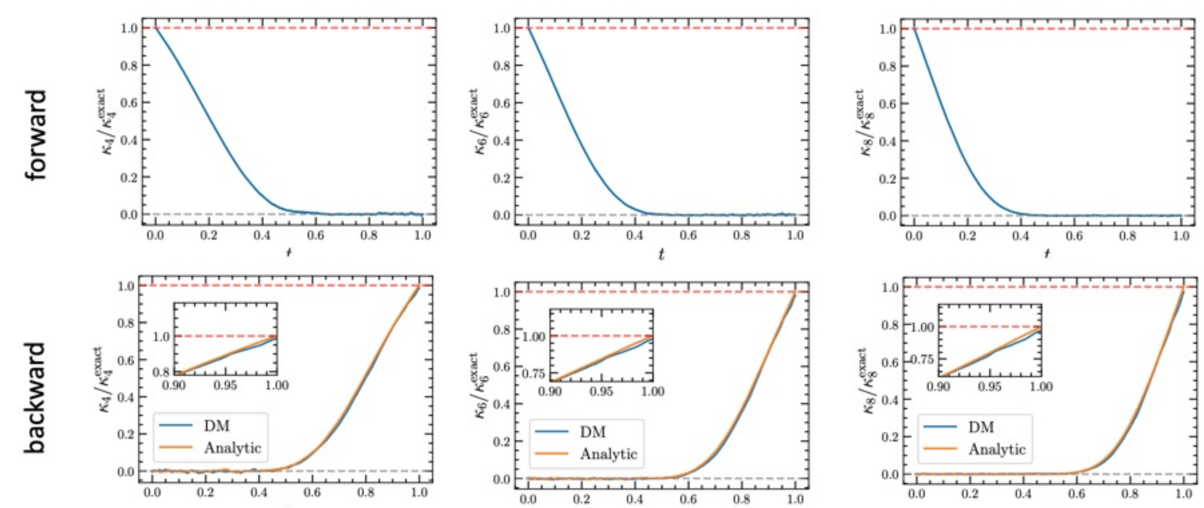
$$\kappa_{n>2}(t) = \frac{\delta^n W[J]}{\delta J(x,t)^n}\Big|_{J=0} = \frac{\delta^n}{\delta J(x,t)^n} \log \mathbb{E}_{P_0}[e^{J(x,t)\phi_0(x)f(t,0)}]\Big|_{J=0}$$

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

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

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

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



analytic = analytic score