Machine Learning for Lattice Field Theory

Lingxiao Wang(王凌霄) (FIAS/RIKEN)

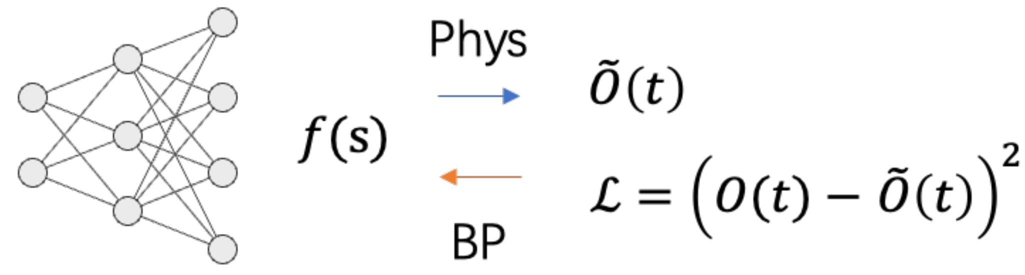
arXiv:2303.15136;
Phys. Rev. D 106, L051502 (2022), Comput. Phys. Commun. 282, 108547 (2023);
Chin. Phys. Lett. 39, 120502 (2022), Phys. Rev. D 107, 056001 (2023), arXiv:2309.17082.

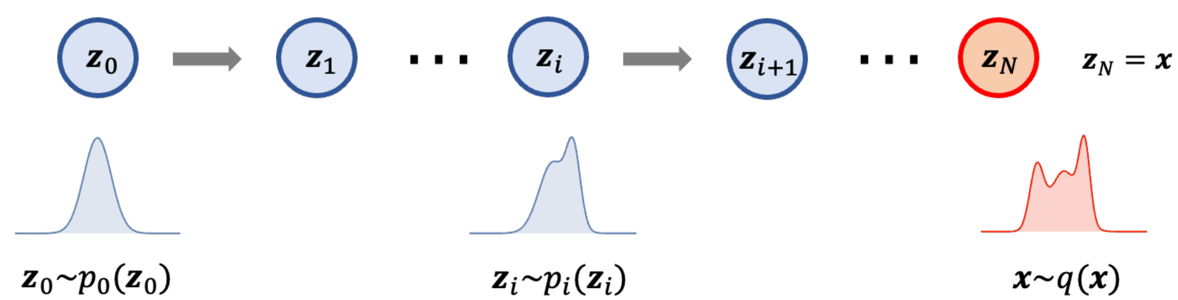
Collaborators: Kai Zhou(FIAS), Shuzhe Shi(THU), Shile Chen(THU), Yin Jiang(BeihangU), Gert Aarts(Swansea U), …

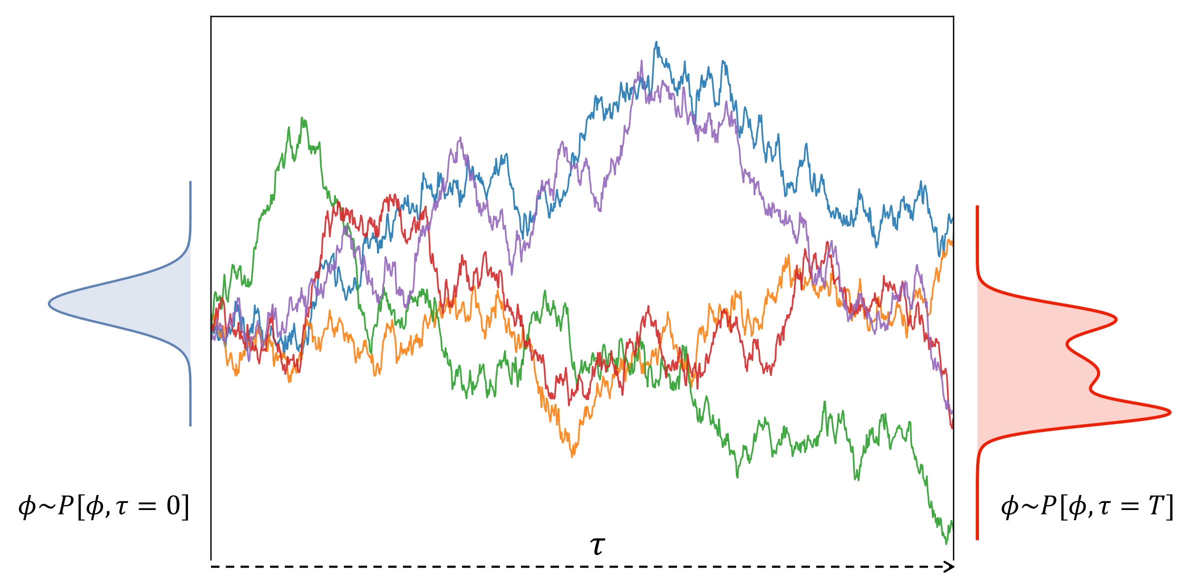
2023年10月9日, 第三届中国格点量子色动力学研讨会，北京

Outline

- **Why ML?**
- **Inverse Problems**
  - Spectral Reconstruction
- **Generative models**
  - Autoregressive Networks
  - Fourier-Flow Model
- **Diffusion Models**
- **Summary**







What is ML?

Machine Learning (from Data)

Artificial Intelligence (AI)

Machine Learning (ML)

Deep Learning (DL)

Booming in theory and applications!

1. Big Data
2. GPU parallel
3. New architecture

2006

Geoffrey Hinton

Machine Learning (ML) is a subset of artificial intelligence that involves the creation of algorithms that allow computers to learn from and make decisions or predictions based on data. It's essentially a way for computers to "learn" from data without being explicitly programmed to do so.

—— **ChatGPT4**

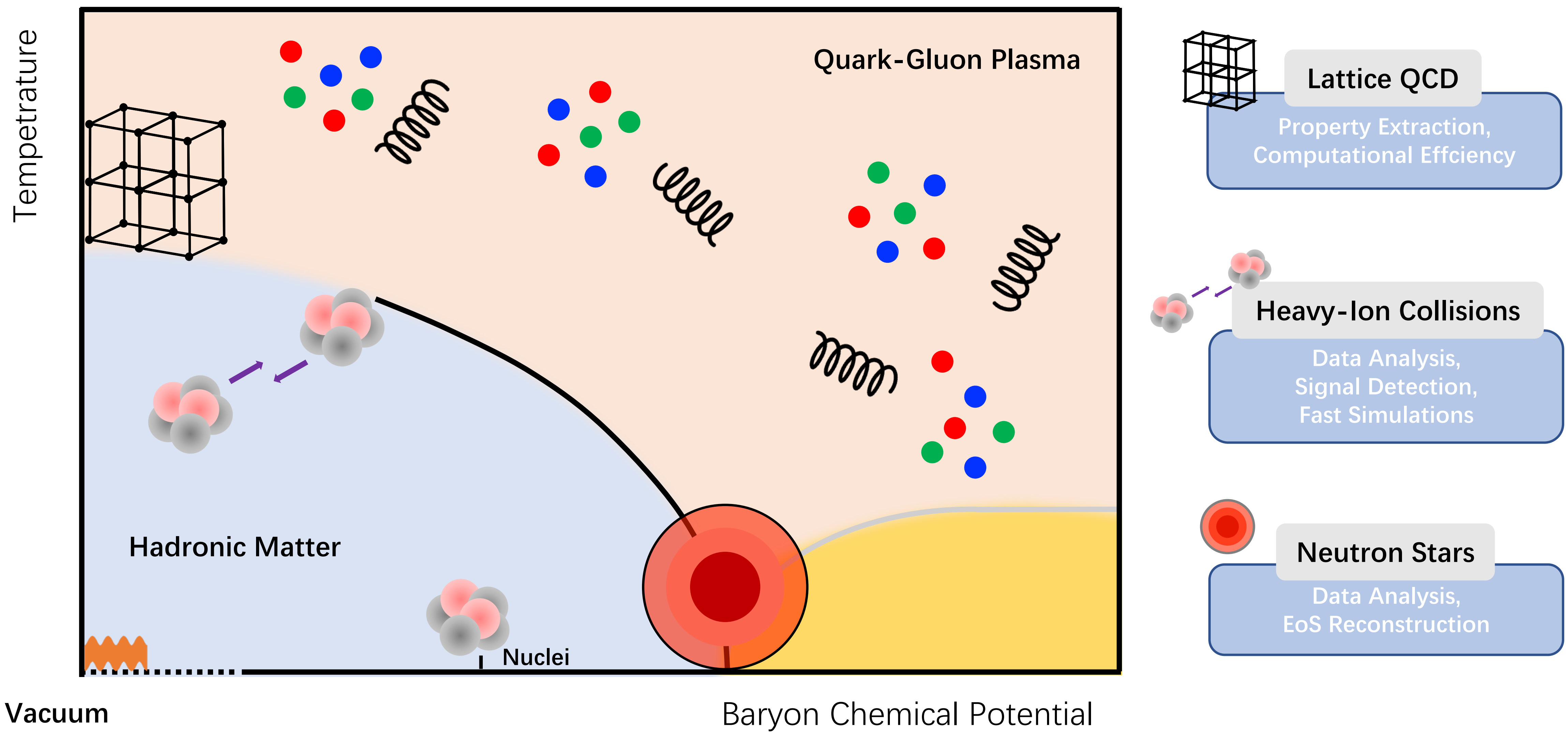
Big Data + Deep Models

↓ GPU

Successful Deep Learning!

Why ML?

arXiv:2303.15136 (invited review on *PPNP*)



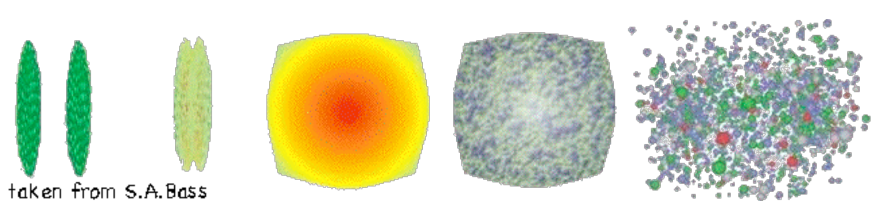
- **Heavy-Ion Collisions** : Large number of data! Complicated simulations!
- **Neutron Star** : Accumulating observations! Poor signal-noise ratio!
- **Lattice QCD** : **Computationally consuming! Data analysis!**

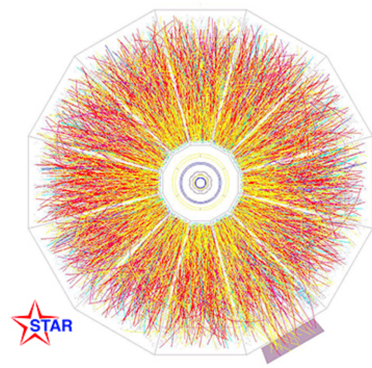
Inverse Problems

Inverse Problems

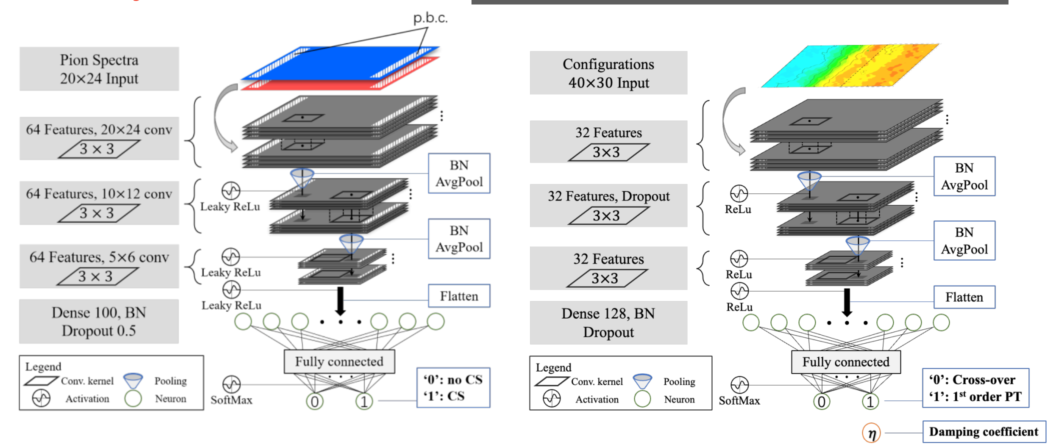
Heavy-Ion Collisions

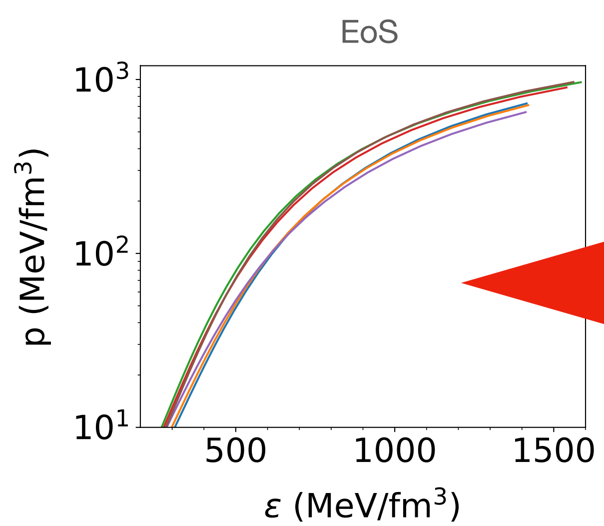
Phys. Rev. C 106, L051901; Phys. Rev. D 103, 116023

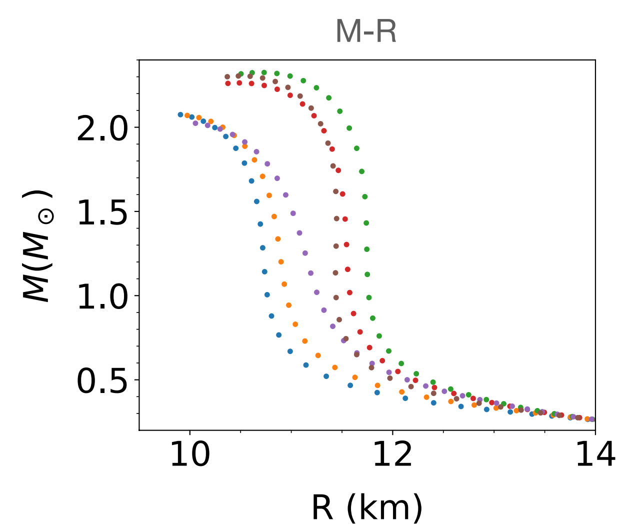




Strongly interaction matter properties

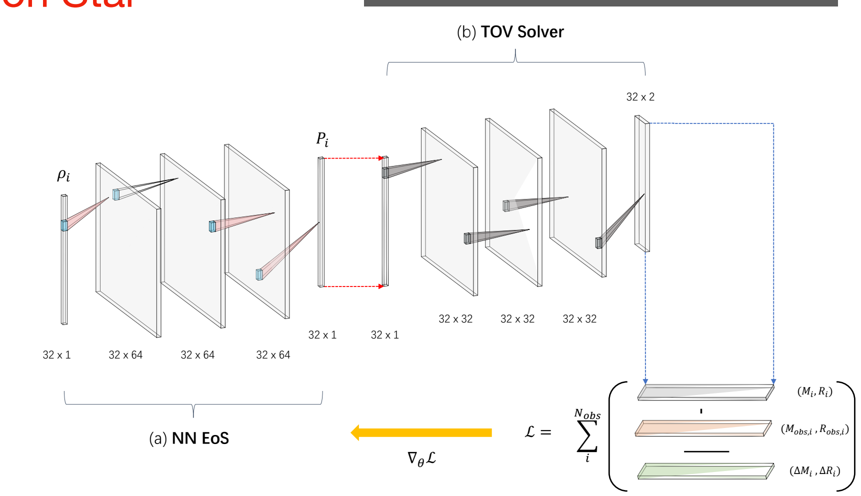


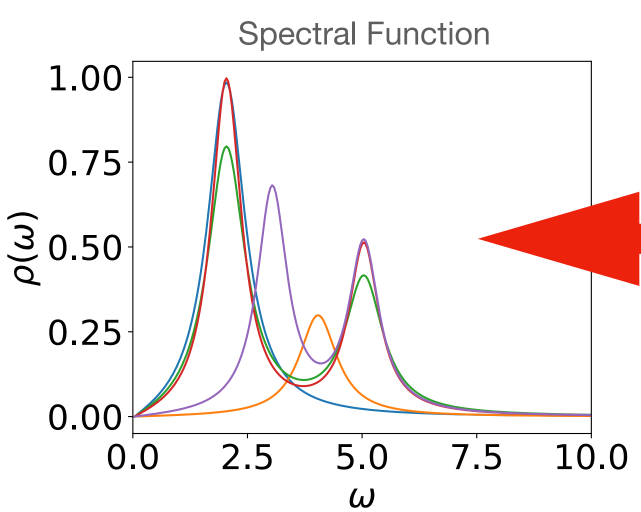


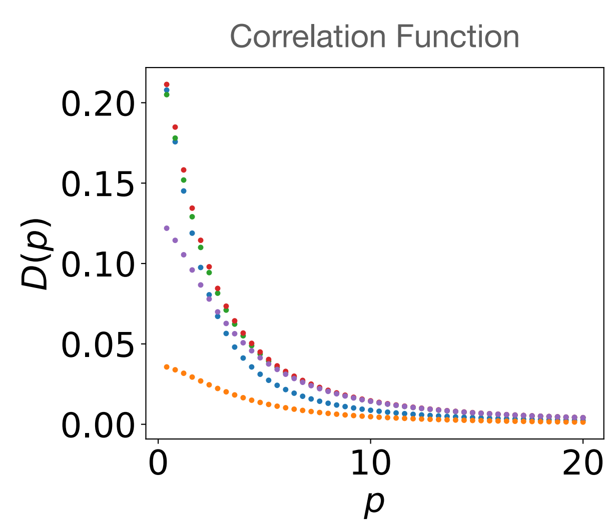


Neutron Star

Phys. Rev. D 107, 083028; JCAP08 (2022) 071

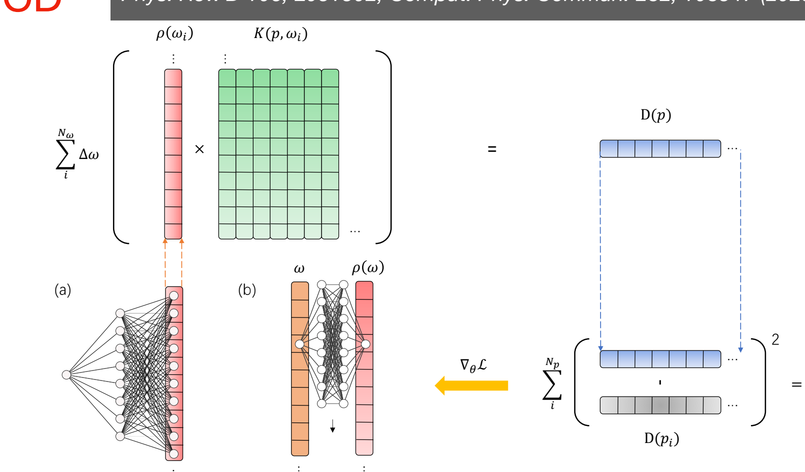






Lattice QCD

Phys. Rev. D 106, L051502; Comput. Phys. Commun. 282, 108547 (2023);



Spectral Functions

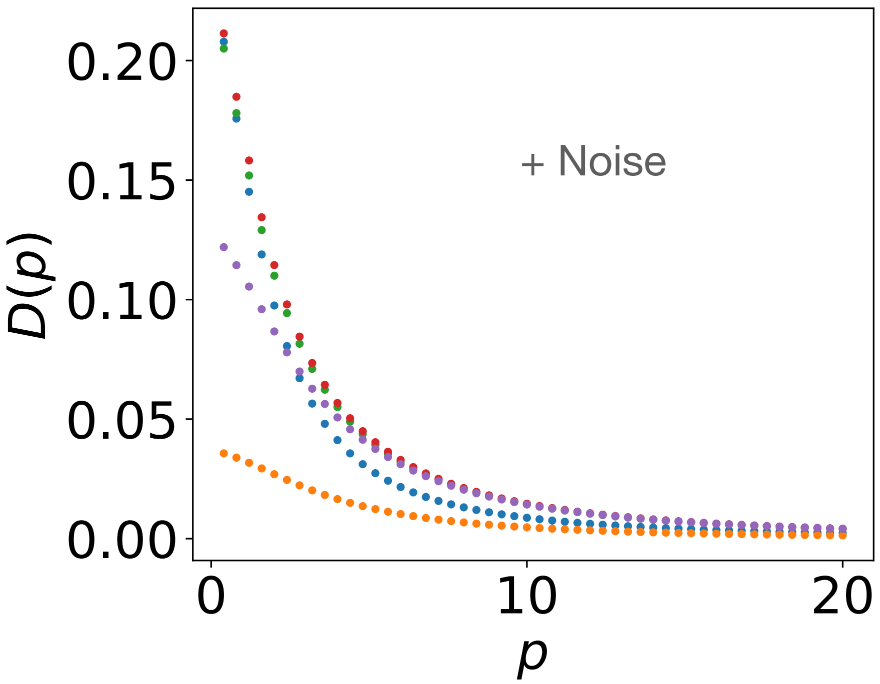
Rebuilding spectral functions

$$D(p) \equiv \int_0^\infty K(p,\omega)\rho(\omega)\mathrm{d}\omega$$

$$K(p,\omega) = \frac{\omega}{\pi(\omega^2 + p^2)}$$

Kallen—Lehmann(KL) representation

Forward process

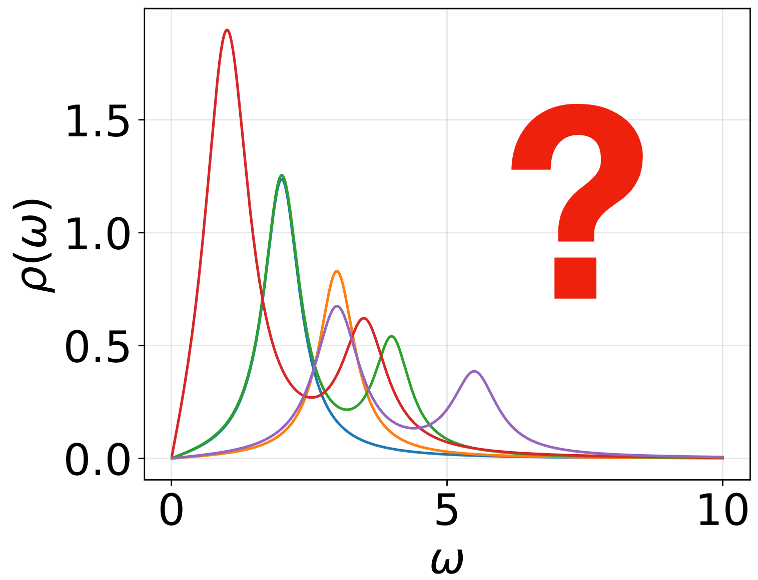


Model

Physical rules/properties, unclear

Observables

Measurements, data



Reconstruct the spectral function from **noisy Euclidean propagator** data (e.g.,Lattice QCD) to extract their physical structures.

Spectral reconstruction

Why ill-posed?

- In practice, the Euclidean correlations have **finite number of points** and **with finite precision**;
- The ill-posedness of the spectral reconstruction **fundamentally exists even for** *continuous correlation functions(infinite observations)*;
- It's caused by the **numerical inaccuracy** of the correlation measurements (induced high degeneracy in solution space).

$$K_{ij}, i \in N_x, j \in N_\omega, N_x < N_\omega$$

$$\vec{D} \equiv K \vec{\rho}$$

highly rectangular

↑ vectorization

$$D(x) \equiv \int_0^\infty K(x,\omega)\rho(\omega)\mathrm{d}\omega$$

↓ eigenvalue problem

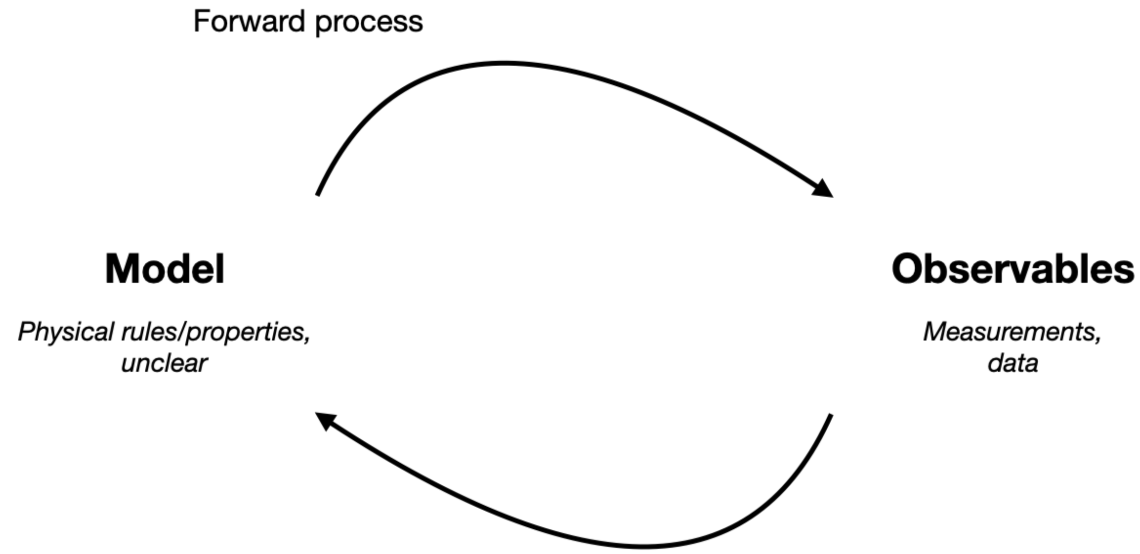
$$\int_0^\infty \psi_s(\omega)\, K(x,\omega)\, \mathrm{d}\omega = \lambda_s \psi_s(x)$$

Comput. Phys. Commun. 282, 108547

J. Phys. A: Math. Gen., Vol. 11, No. 9, 1978. Printed in Great Britain.

Spectral Functions

Methodology



- **Classical methods**

  Truncated Singular Value Decomposition (TSVD)

  Tikhonov regularization, …

  Inverse and Ill-Posed Problems (Academic Press, Boston, 1987).

- **Baysian methods**

  Maximum Entropy Method(MEM)

  Maximum Entropy Analysis of the Spectral Functions in Lattice QCD, Progress in Particle and Nuclear Physics **46**, 459 (2001).

  Bayesian Reconstruction(BR)

  Phys. Rev. Lett. 111, 182003 (2013); arXiv:2208.13590.

Prior → Regularization

- **Supervised Learning**
  the inverse mapping

  Phys. Rev. B **98**, 245101 (2018).
  Phys. Rev. D **102**, 096001 (2020).
  Phys. Rev. Lett. **124**, 056401 (2020).

- **New developments**

  Gausian process

  Phys. Rev. D 105, 036014 (2022).

  Sparse modeling method JHEP07(2020)007

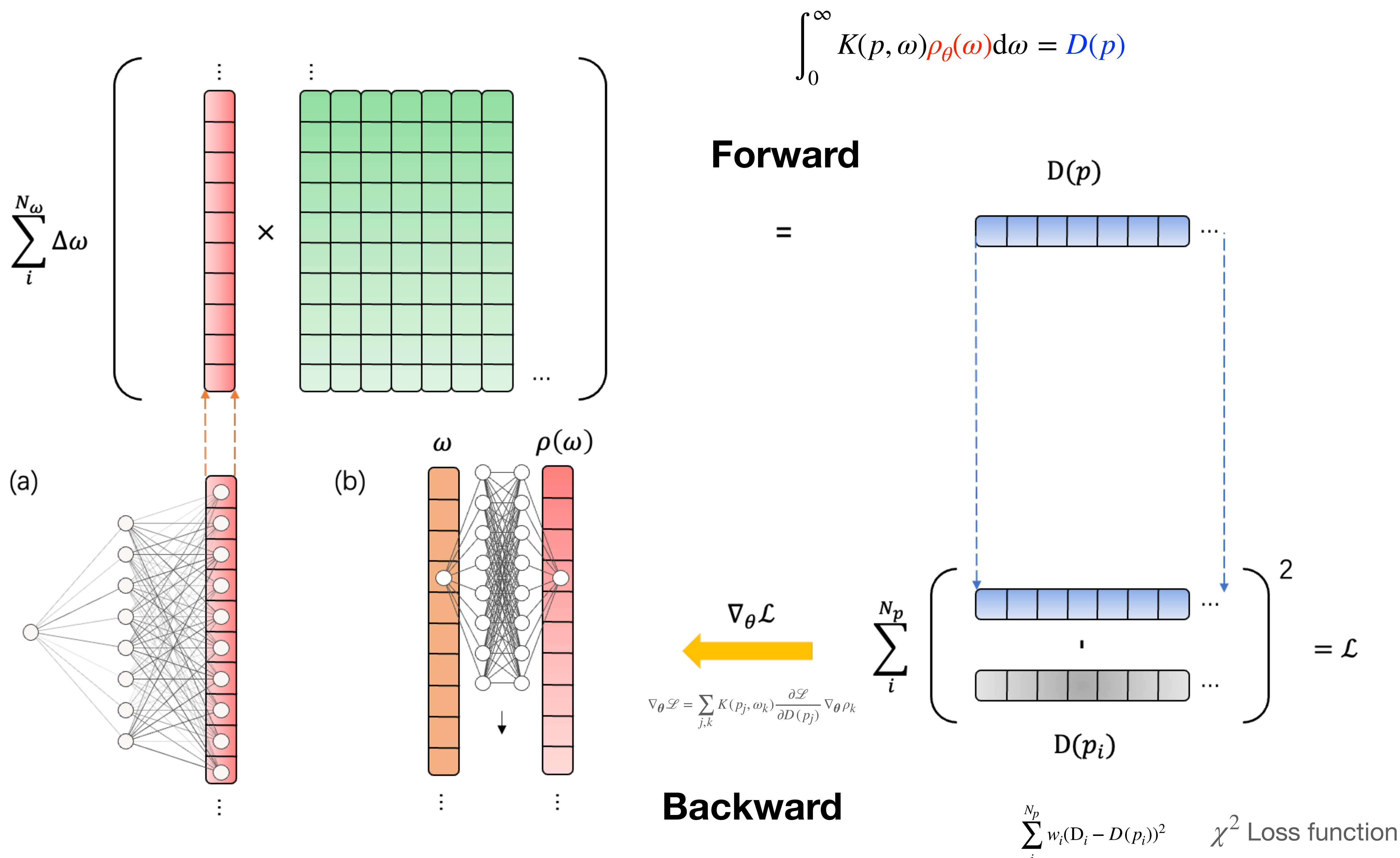
  Radial Basis Functions(RBF)
  Phys. Rev. D 104, 076011

  sVAE(Variational AutoEncoder) arXiv:2110.13521; PoS LATTICE2021 (2022) 148

AD Framework

Automatic differentiation

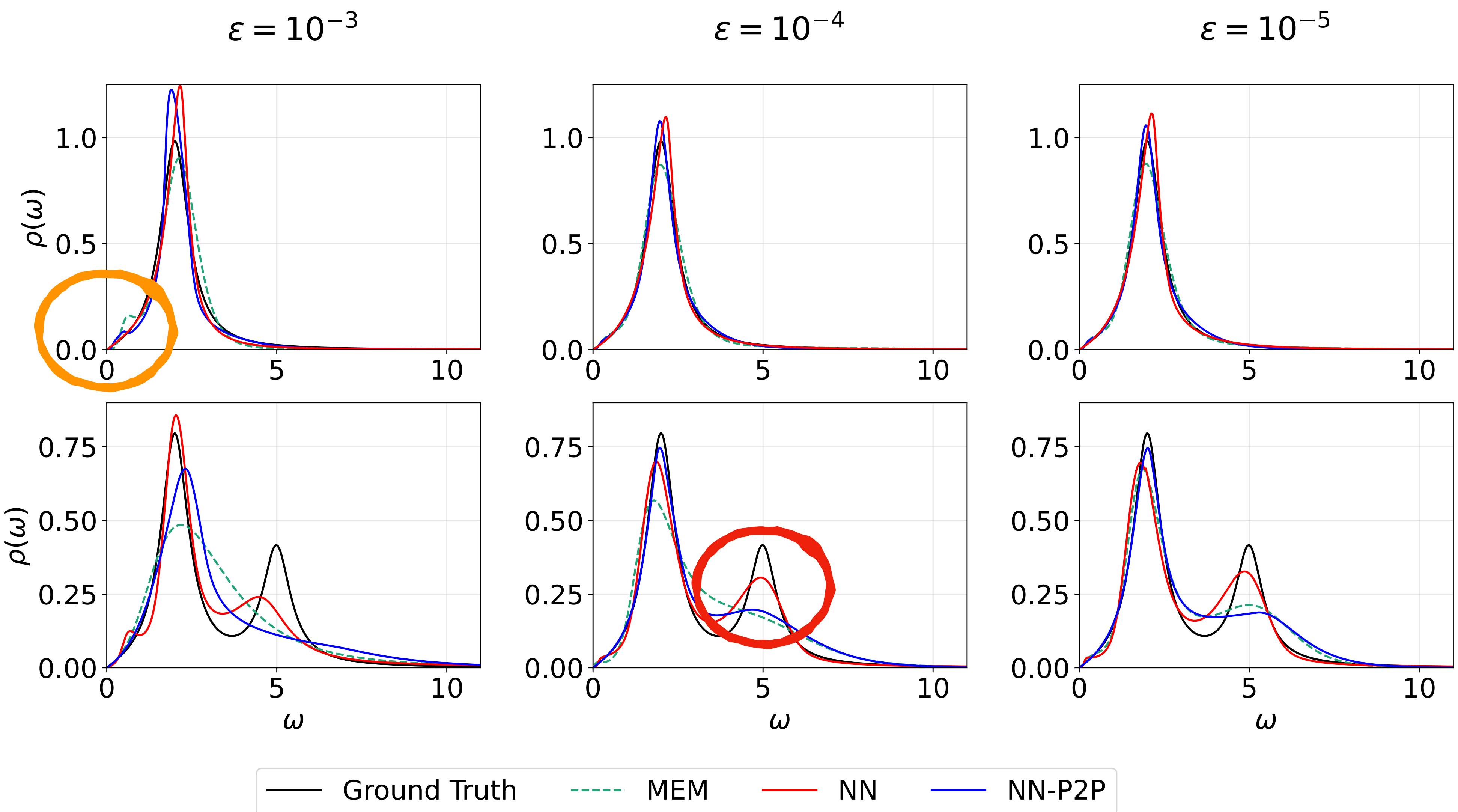
Phys. Rev. D 106, L051502



Benchmark

Phys. Rev. D 106, L051502

Mock data



$$\rho^{(\mathrm{BW})}(\omega) = \frac{4A\Gamma\omega}{\left(M^2+\Gamma^2-\omega^2\right)^2 + 4\Gamma^2\omega^2}$$

$$A = 1.0, \Gamma = 0.5, M = 2.0$$

$$A_1 = 0.8, A_2 = 1.0, \Gamma_1 = \Gamma_2 = 0.5$$
$$M_1 = 2.0, M_2 = 5.0$$

Reconstruction performance increases with noise decreasing

NN-P2P gets the best consistency near the zero-frequency

NN can represent a more diverse spectrum in double-peak case

Regularization

Why NN helps

Comput. Phys. Commun. 282, 108547

$$\frac{\delta J}{\delta \rho(\omega)} = 0$$ **the optimal solution exists!**

$$\frac{f_a/\sigma'(f_a)}{\left(\sum_b f_b^2\right)^{\frac{l-1}{l}}} = \frac{\mathrm{DM}_a}{\alpha}\sum_i \Delta_i K_{ia}$$

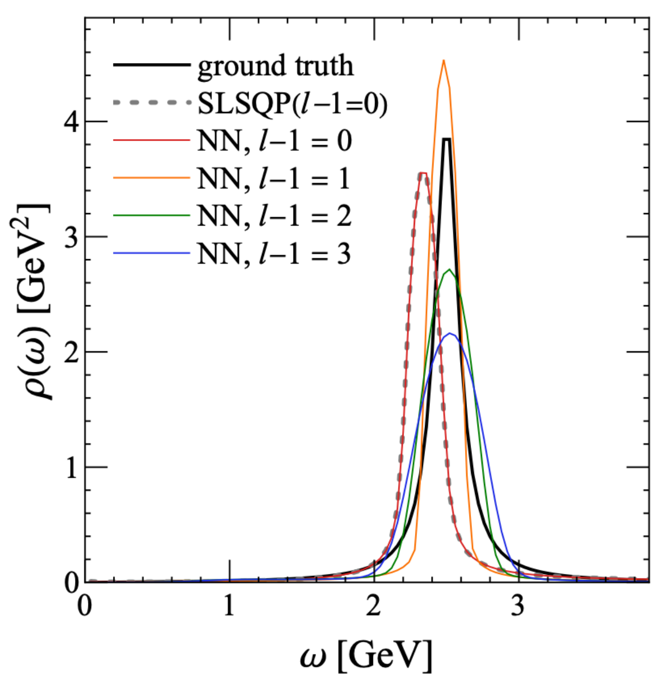
non-local constraints from NN!

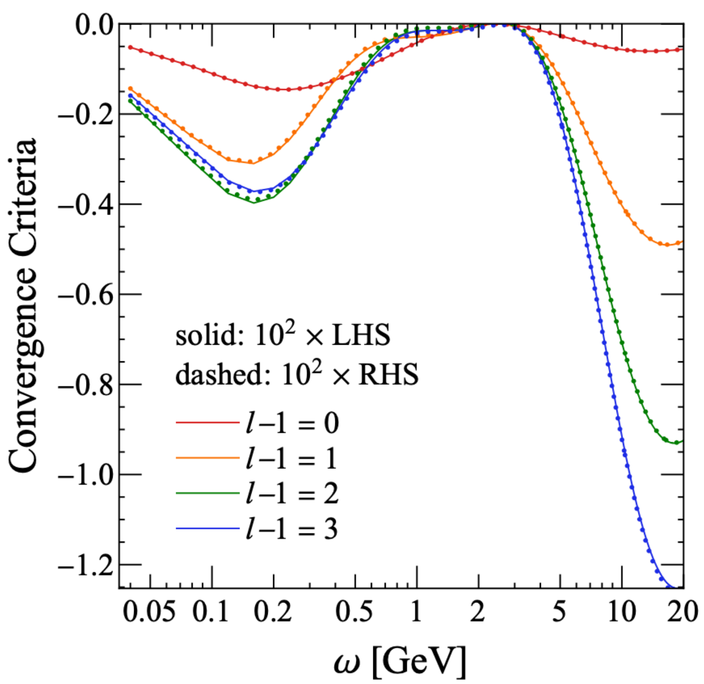
- **Neural Networks (e.g., NN representation)**

$$\rho_a \equiv \rho(\omega_a)$$

  - Output layer, $\rho_a = \mathrm{DM}_a\, \sigma^{(l)}(f_a^{(l)})$
  - Default Model(DM), DM_a
  - Activation functions, $f_a^{(n)} = \sigma^{(n)}(x_a^{(n)})$
  - Hidden layers, $x_a^{(n)} = \sum_b W_{ab}^{(n)} f_b^{(n-1)}$
    width $a = 1,2,\cdots,N^{(n)}$; $n = 1,2,\cdots,l$
- **Set-ups**
  - **Width,** $N^{(0)} = 1, N^{(l)} = N_\omega$
  - **Input layer,** $a_1^{(0)} = 1$
  - **Hidden layer, no activation functions**
  - **Output layer,** $\sigma^{(l)}(x) = \sigma(x),\ f_a \equiv f_a^{(l)}$
  - **L2 regulation,** $L_2 \equiv \alpha \Delta\omega \sum_{l,a,b} \left(W_{ab}^{(l)}\right)^2$

$$\alpha\, f_a\,(1 + e^{-f_a}) \equiv \mathrm{DM}_a\left(\sum_b f_b^2\right)^{\frac{l-1}{l}} \sum_i \Delta_i K_{ia}$$





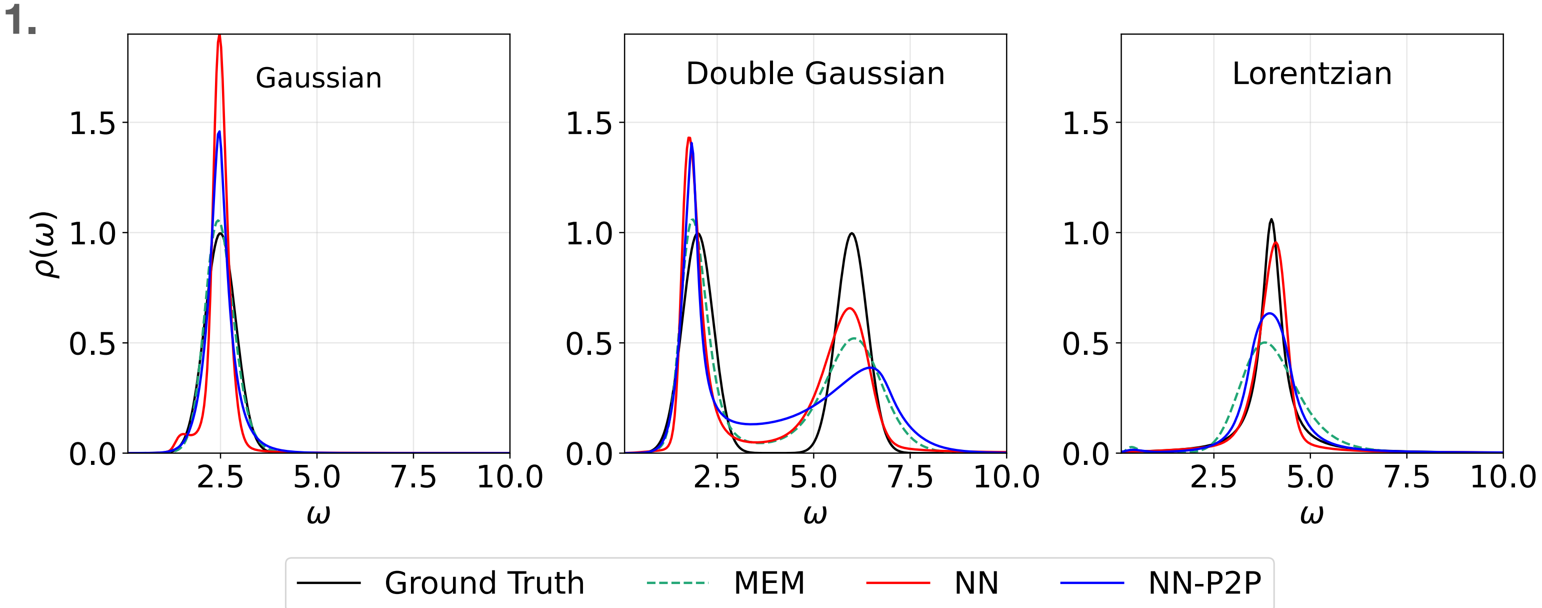
Results

Other cases

Phys. Rev. D 106, L051502

noise level $\epsilon = 10^{-4}$ with $N_p = 25$ points

1.

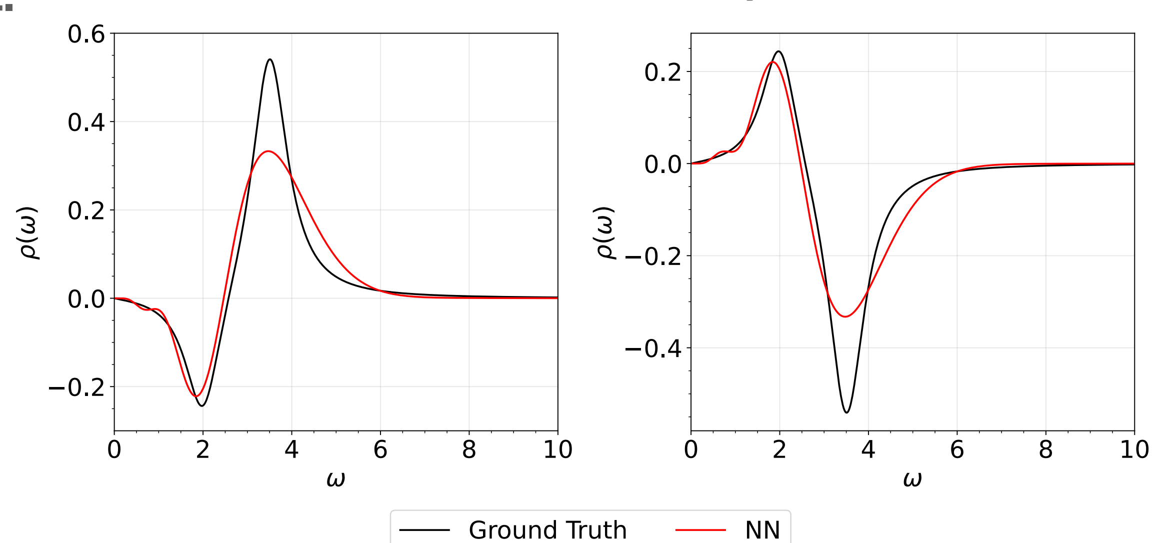


1. **Single-peak functions**
2. **Non-positive-definite SPs**
3. Lattice QCD mock data

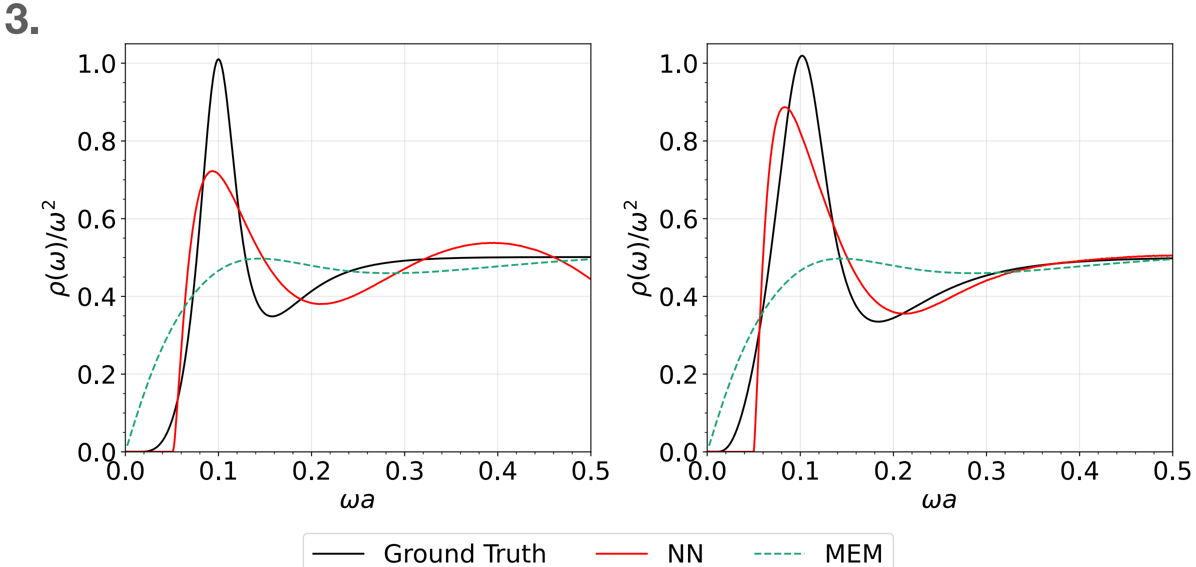
(see arXiv:2110.13521)

noise level $\epsilon = 10^{-4}$ with $N_p = 25$ points

2.



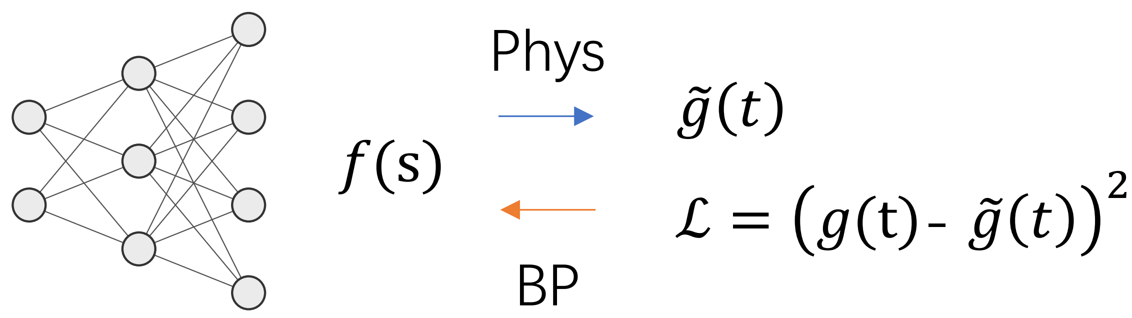
3.

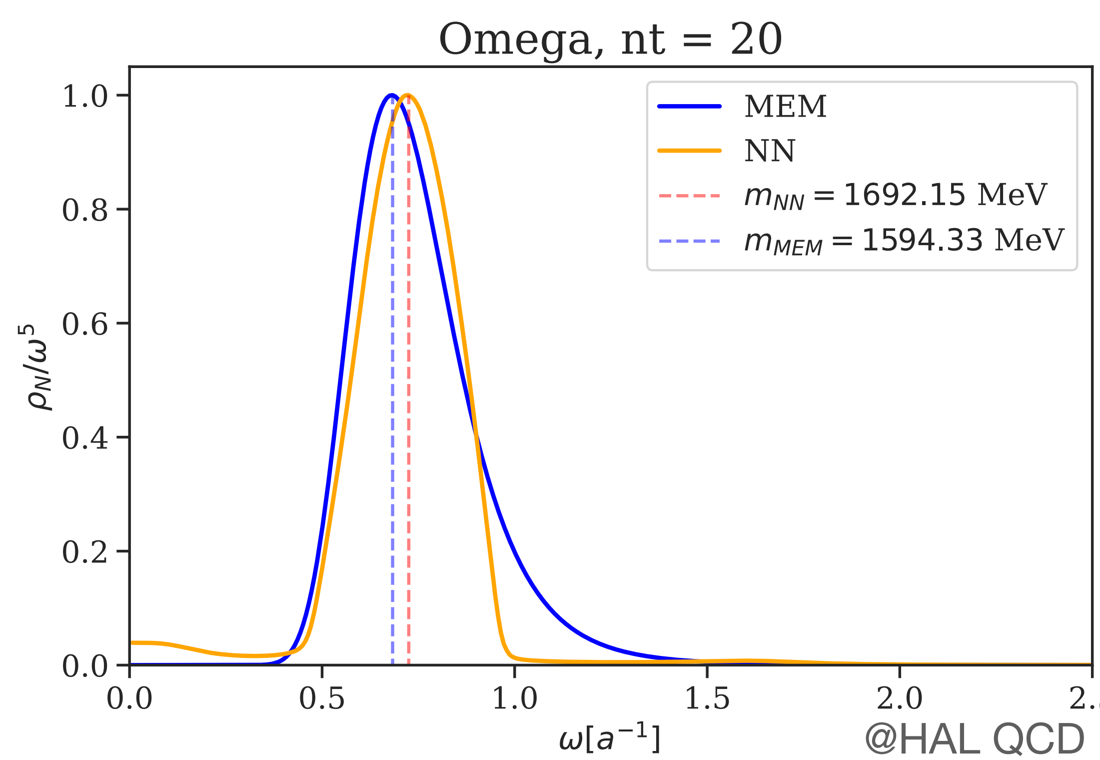


Summary I

- **Inverse Problems**
  - Neural network
    - Flexible representations and regularizations
  - Auto-differentiation framework
    - High-effcient gradient-based optimization
- **Future works**
  - Open codes [github1, github2]
  - **Easy-to-use Python packages**
  - **Real Lattice QCD data !**
    - HAL QCD collaboration
      - Omega, Xi, Lambda,…
    - FASTSUM collaboration

***arXiv:2303.15136* (invited review on *PPNP*)**

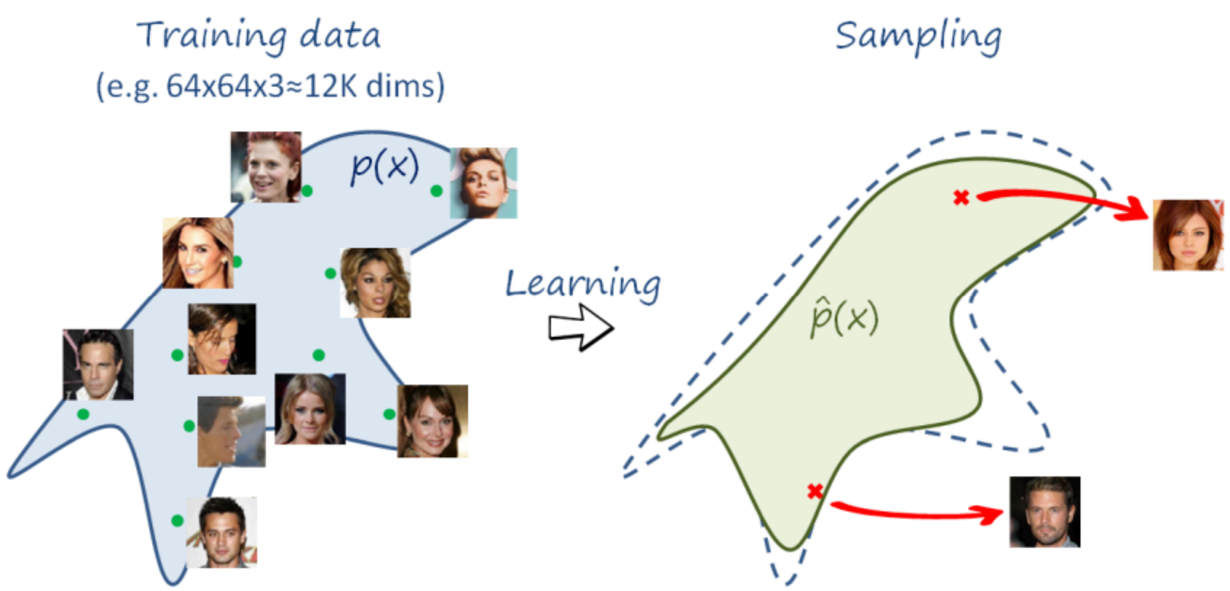


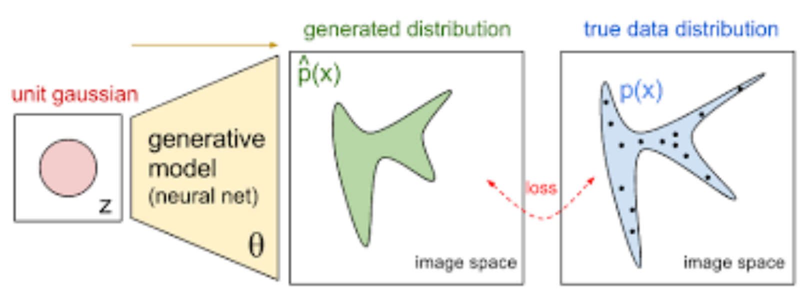


Generative models

Generative Models

for Lattice calculations



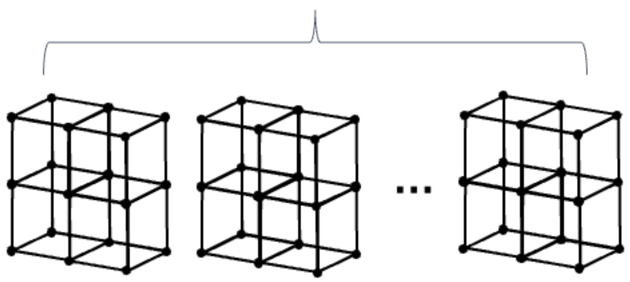


@blogs of OpenAI

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

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

Ensemble



Generative models

→ **underlying distributions in data**

Lattice calculations

→ **physical distributions, sampling**

Markov-Chain MC

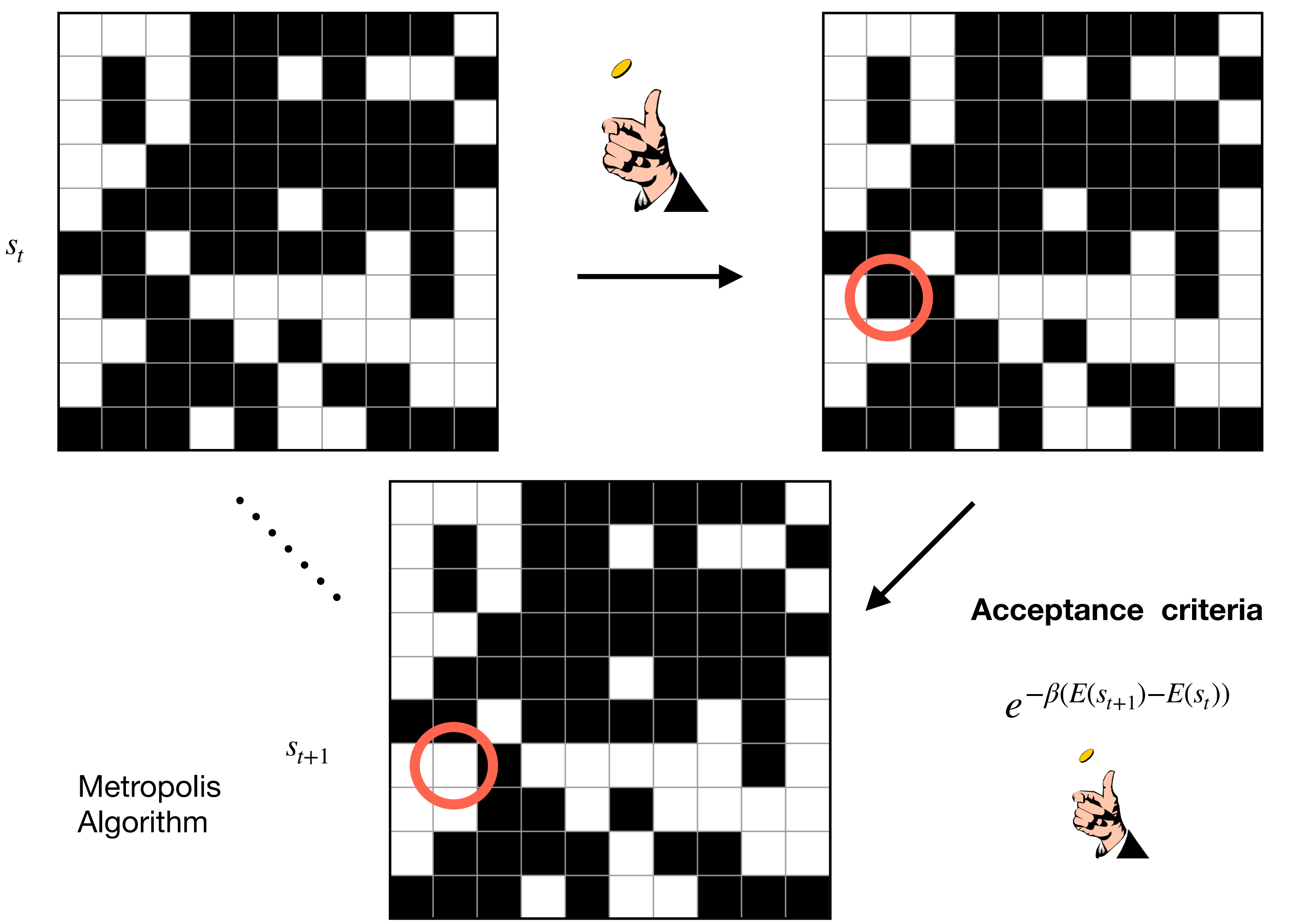
Revisiting

- Generate configurations ϕ_i **independently** from,

$$p(\phi) = \frac{e^{-\beta E(\phi)}}{Z}$$

- Metropolis Method
- Shortcomings
  - Local update, low-effciency
  - Critical Slowing Down
    [U. Wolff, Nucl. Phys. B 17, 93 (1990)]
- **Need global update (proposal)!**

Monte Carlo-Metropolis Algorithm for 2D Ising Model (L=10)



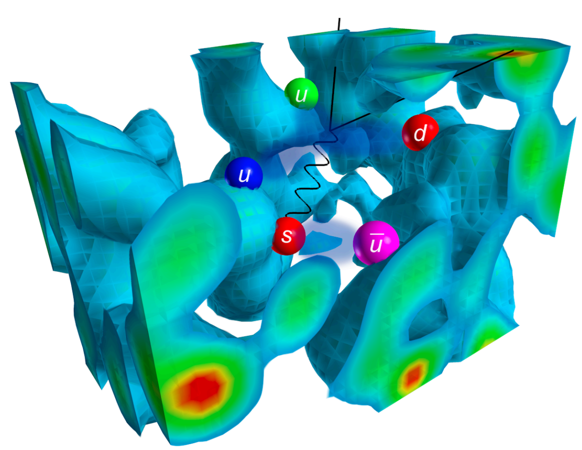
Generative Models

for Lattice calculations

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

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

→ **physical distribution, sampling** through Generative Moodels



Lattice QCD © Derek Leinweber/CSSM/University of Adelaide

- **Implicit Likelihood Estimation (needs training data-set)**
  - 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).

  - Diffusion Models (arXiv: 2309.17082)
- **Explicit Likelihood Estimation**
  - 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. S. Albergo, G. Kanwar, and P. E. Shanahan, Phys. Rev. D 100, 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).
    R. Abbott et al., Phys. Rev. D 106, 074506 (2022).
    A. Singha, et al., Phys. Rev. D 107, 014512 (2023).
    S. Chen, et al., Phys. Rev. D 107, 056001(2023).
    …

Hands-on notebook

M. S. Albergo et al., arXiv:2101.08176.

Review

K. Cranmer, G. Kanwar, S. Racanière, D. J. Rezende, and P. E. Shanahan, **Advances in Machine-Learning-Based Sampling Motivated by Lattice Quatum Chromodynamics**, Nat Rev Phys 1 (2023).

CANs

Continuous Autoregressive Networks

- **Autoregressive Networks** can model probability distribution $q_\theta(s)$ explicitly

$$q_\theta(s) \rightarrow p(s) = \frac{e^{-E(s)}}{Z}$$

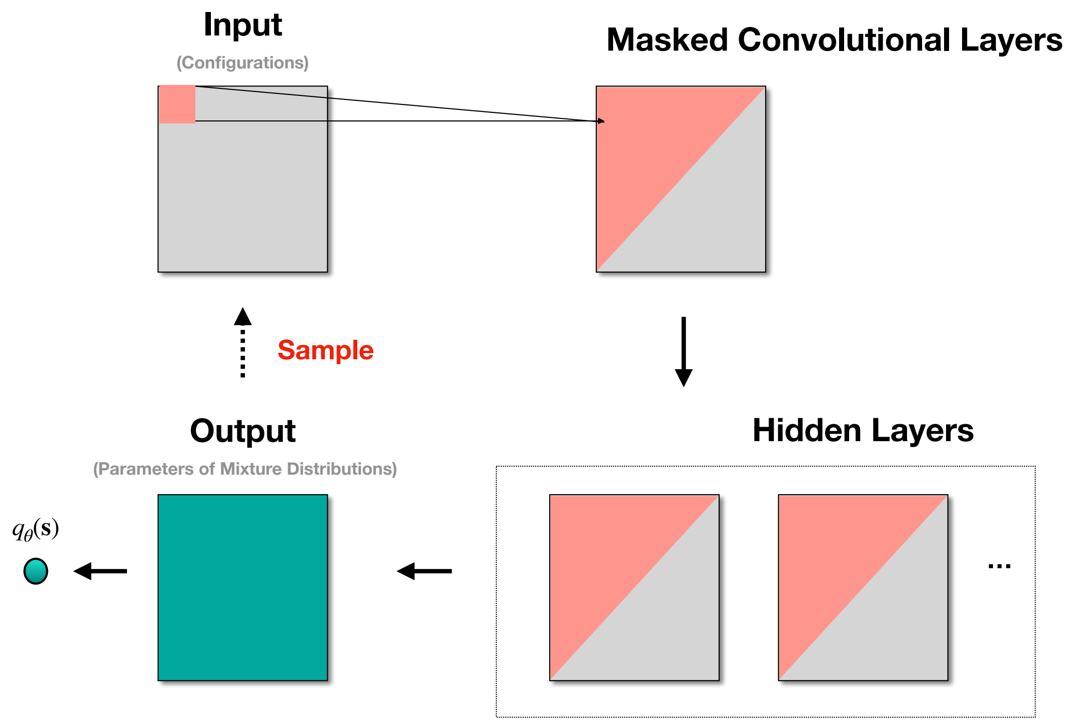
- Optimization
  - Loss function, **variational free energy**

$$F_q = \sum_s q_\theta(s)(E(s) + (ln q_\theta(s))/\beta)$$

  - Kullback-Leibler (KL) divergence

$$D_{KL}(q_\theta \quad p) = \sum_s q_\theta(s) ln(\frac{q_\theta(s)}{p(s)}) = \beta(F_q - F) \geq 0$$

  - Neural network parameters θ



Chinese Phys. Lett. 39, 120502 (2022)

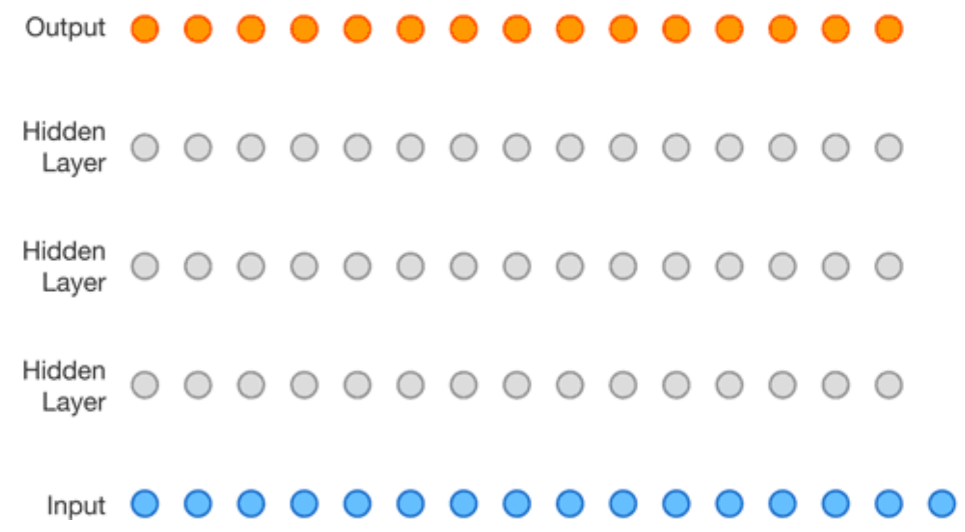
CANs
for 2D XY model

Chinese Phys. Lett. 39, 120502 (2022)

- **2-dimensional(2D) XY model**

$$H = -J \sum_{<i,j>} s_i s_j = -J \sum_{<i,j>} \cos(\phi_i - \phi_j)$$

- Neural network: **PixelCNN**

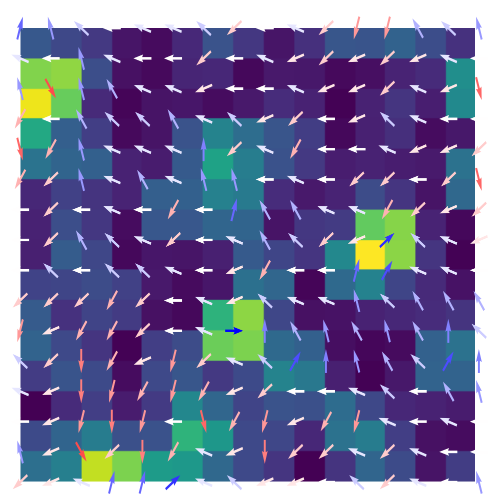


Autoregressive properties@DeepMind Blog

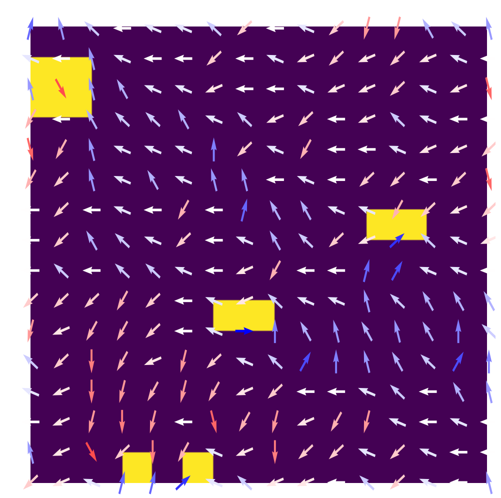
- Joint distribution:

$$q_\theta(s) = \prod_{i=1}^{N} f(s_i \;\; s_1, \ldots, s_{i-1})$$

- Kosterlitz-Thouless(KT) transition
    - Vortices

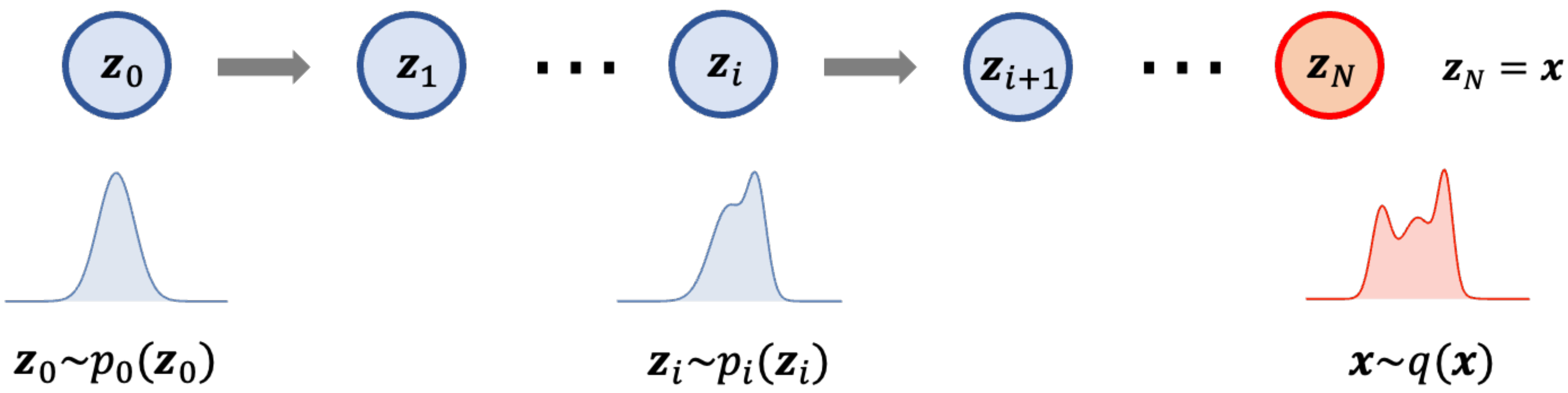


Probability Distributions from CANs



Vortices

Flow-Based Model



- **Flow-based model**
  build a bijective transformation T

$$q_\theta(\mathbf{x}) = p_0(\mathbf{z}) \left| \det J_T(\mathbf{z}) \right|^{-1}$$

- **Jacobian**

  Invertible and tractable

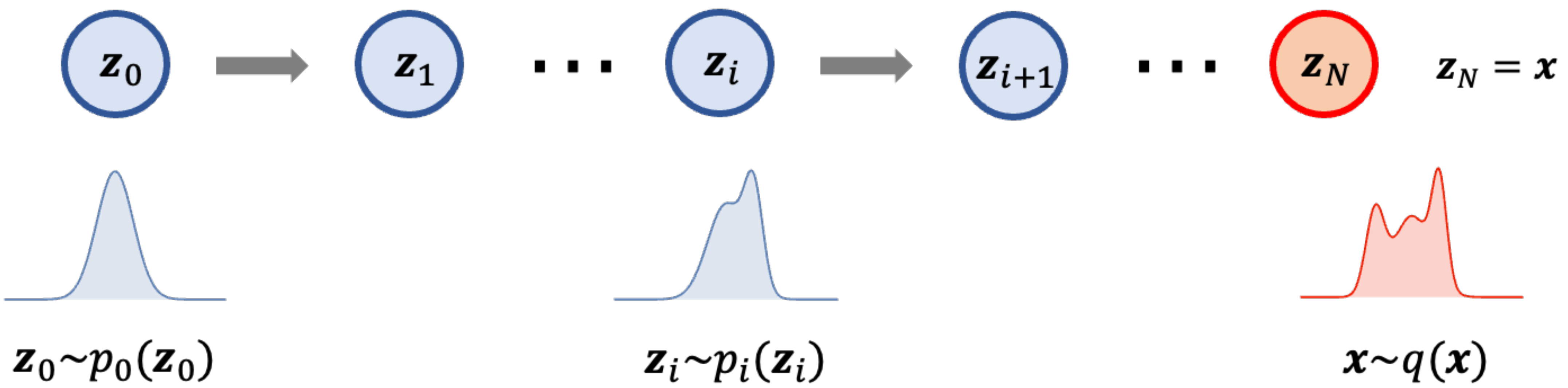
- **Loss function**
  Kullback-Leibler (KL) divergence

$$D_{KL}(q_\theta \| p) \quad p(\phi) = e^{-S(\phi)}/Z$$

- **Optimization**
  - Trainable parameters θ
  - Gradient-based algorithms

Flow-Based Models

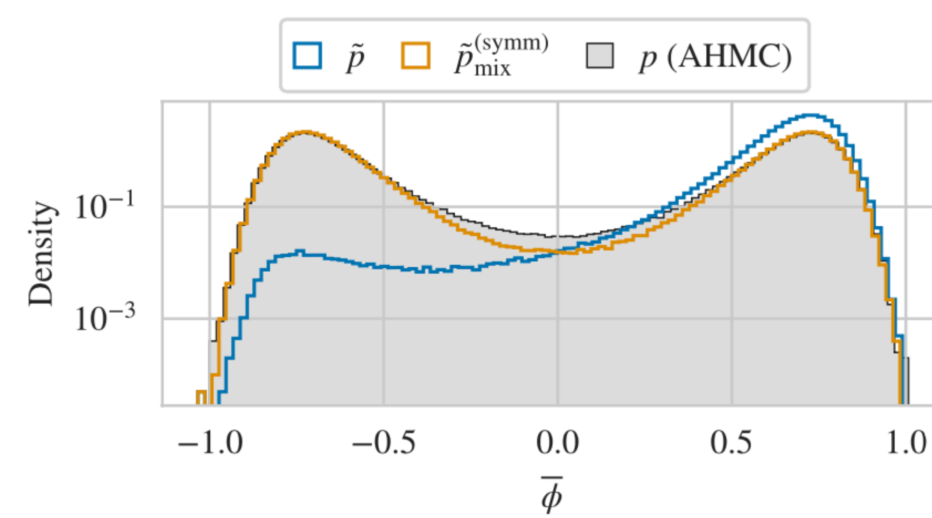
Mode-collapse



- **Why do we need learn in a new representation?**

Flow-based models will encounter **multimodal-distribution**, but the model perfers to choose **one mode of target distributions,**

“Mode-Collapse”

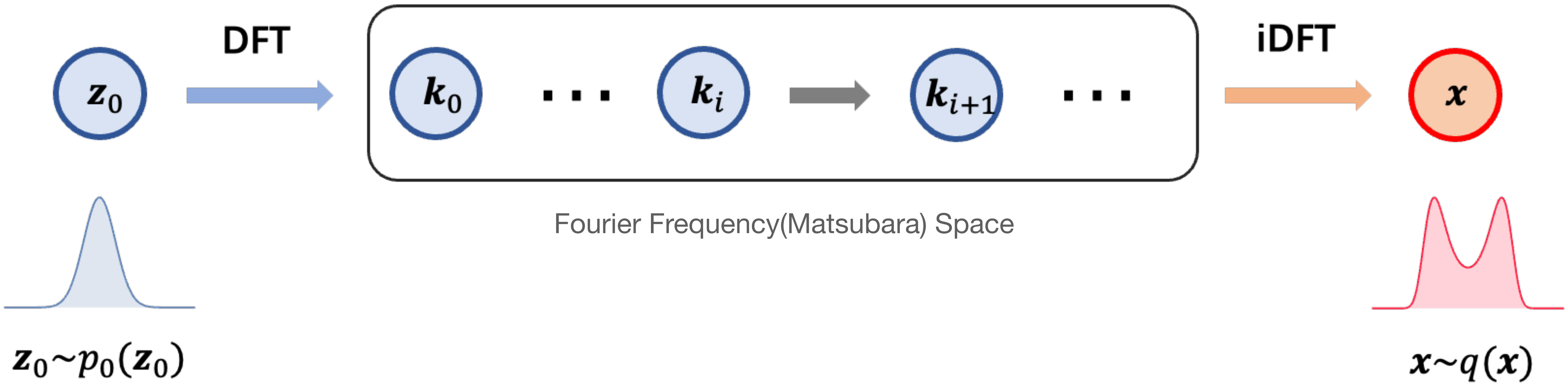


comprehensive discussions in arXiv:2107.00734, arXiv:2302.14082.

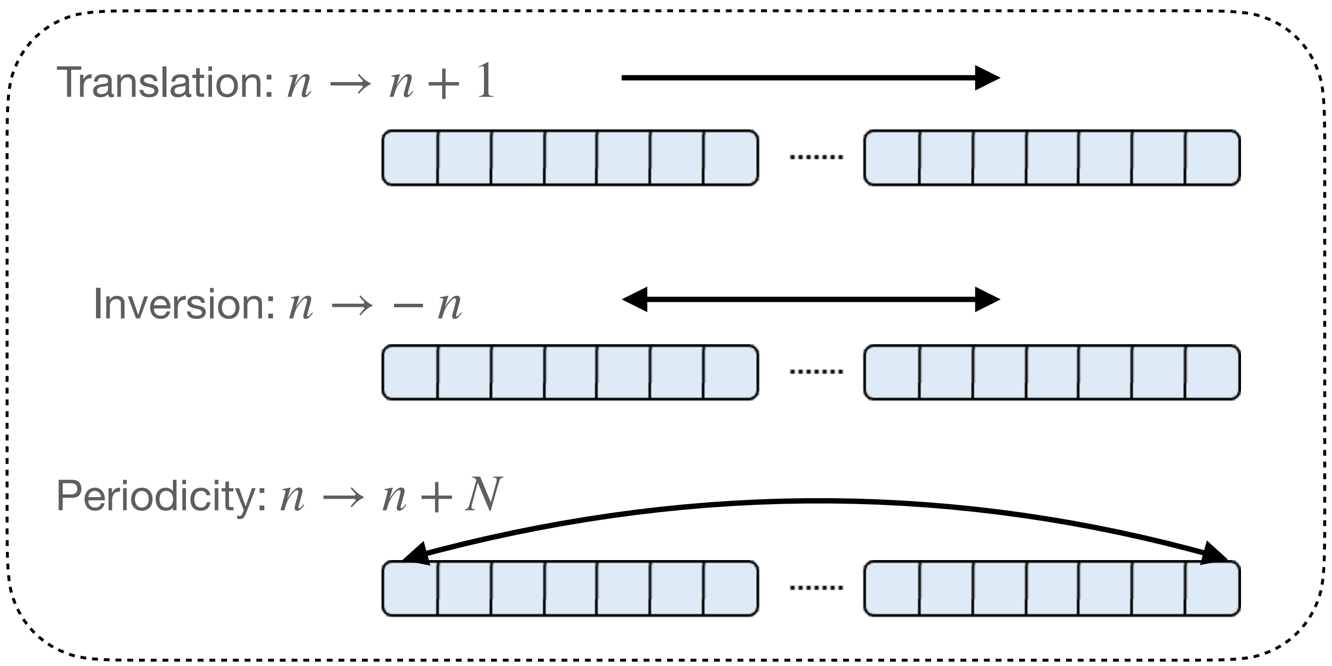
Fourier-Flow Model

Phys. Rev. D 107, 056001

Mode-collapse



Coordinate Space



More priors. More stable!
for training neural networks

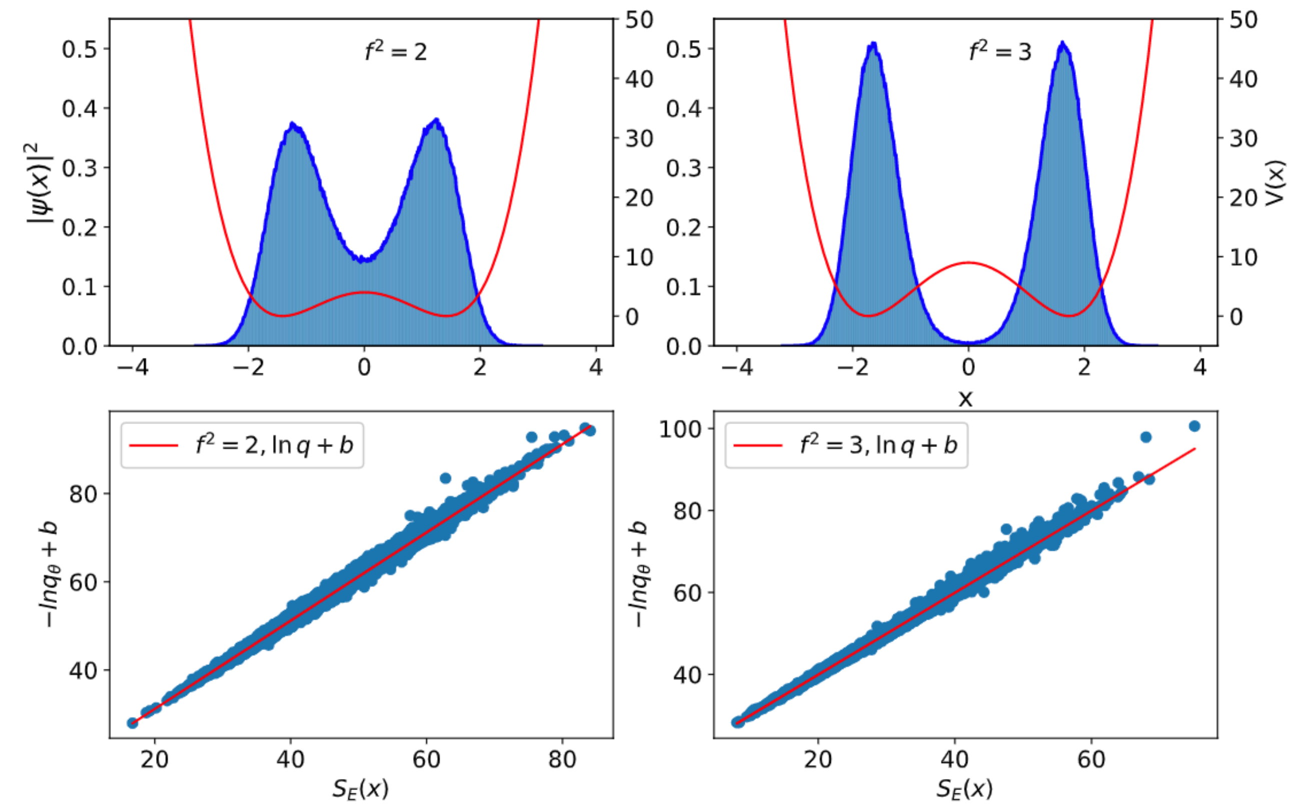
Discrete Fourier transformation (DFT): $X_k = \sum_{n=0}^{N-1} e^{-i\frac{2\pi}{N}kn} x_n$

Inverse DFT (iDFT): $x_n = \frac{1}{N}\sum_{n=0}^{N-1} e^{i\frac{2\pi}{N}kn} X_k$

F-Flow Model

for anharmonic oscillator

Phys. Rev. D 107, 056001

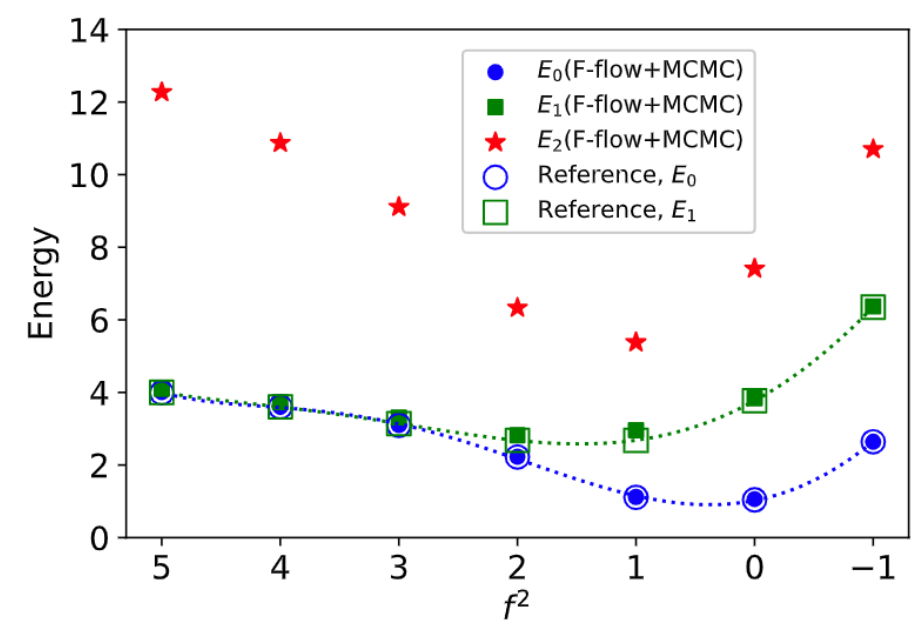


- **1-dimensional(1D) double-well potential**

$$S_E(\{x_n\}) = \frac{\beta}{N}\sum_{n=0}^{N-1}\left[\frac{m(x_{n+1}-x_n)^2}{2a^2} + \lambda(x_n^2 - f^2)^2\right]$$

$$m = 0.5, \lambda = 1$$

- **No analytical solution!**
  - MCMC
    [S. Mittal et al., Eur. J. Phys. 41, 055401 (2020)]
  - Moment methods
    [R. Blankenbecler et al., Phys. Rev. D 21, 1055 (1980)]



- Ground state: $E_0 = 3\lambda\langle x^4\rangle - 4\lambda f^2\langle x^2\rangle + \lambda f^4$
- Excited states

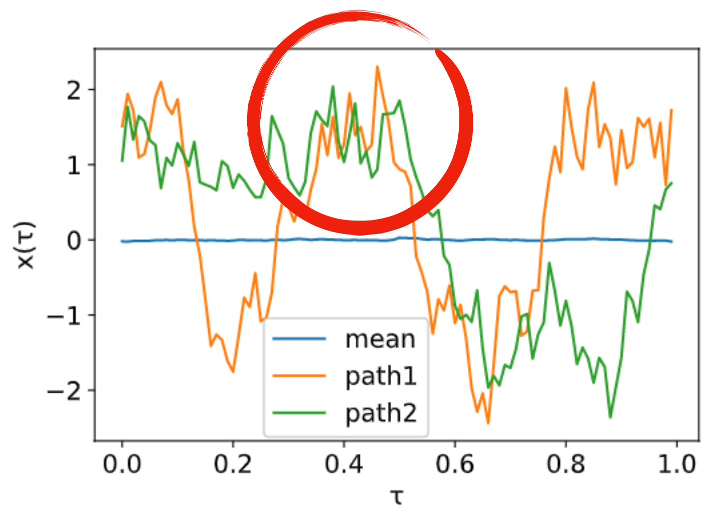
$$E_1 - E_0 = -\lim_{\tau\to\infty}\frac{d\log G_2(\tau)}{d\tau},$$
$$E_2 - E_0 = -\lim_{\tau\to\infty}\frac{d\log G_4(\tau)}{d\tau}$$

F-Flow Model

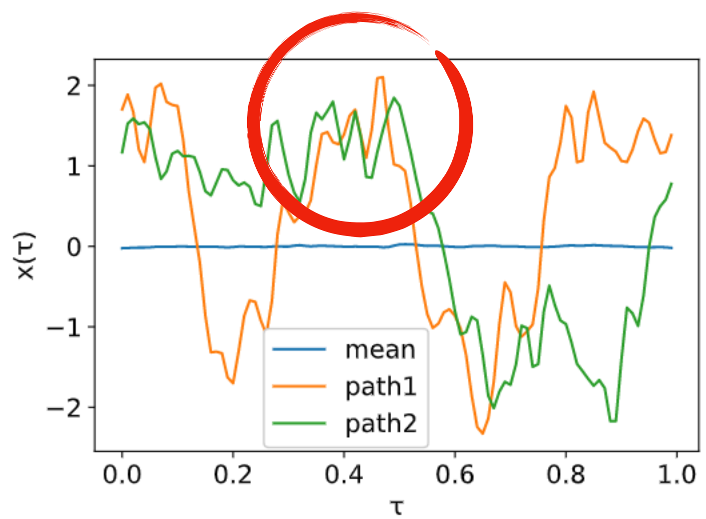
Matsubara frequency

Phys. Rev. D 107, 056001

full frequency modes



70% frequency modes



$$S_E(\{x_n\}) = \frac{\beta}{N}\sum_{n=0}^{N-1}\left[\frac{m(x_{n+1}-x_n)^2}{2a^2} + V(x_n)\right]$$

$$X_k = \sum_{n=0}^{N-1} e^{-i\frac{2\pi}{N}kn}x_n$$

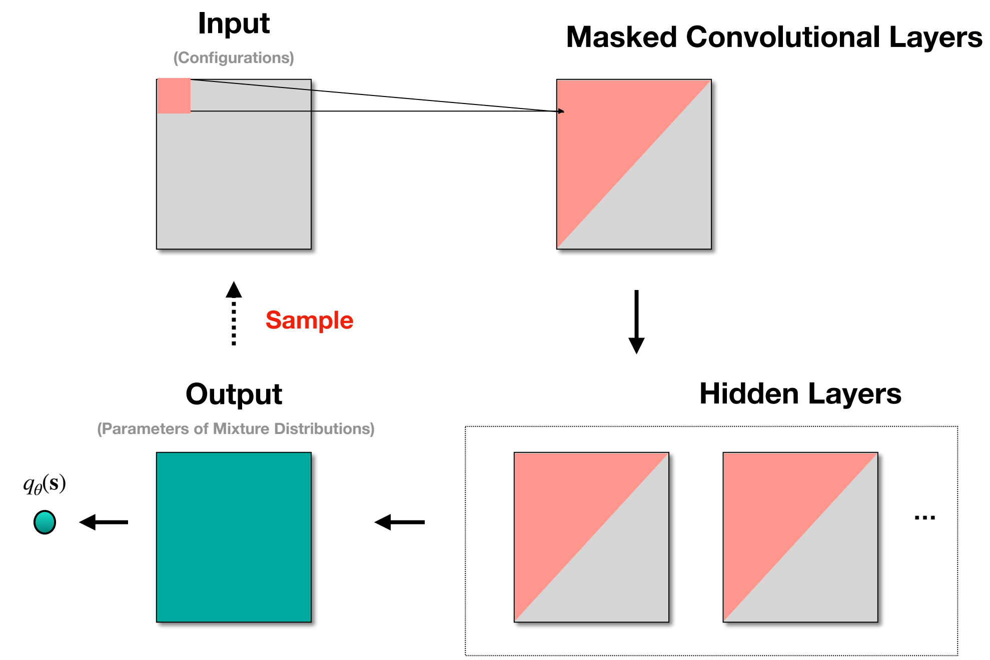
$$S(\{X_k\}) = \frac{\beta}{N^2}\sum_{k=0}^{N-1}\left[\frac{m(1-\cos\frac{2\pi k}{N})}{a^2}\ X_k^{\ 2} + V(X_k)\right]$$

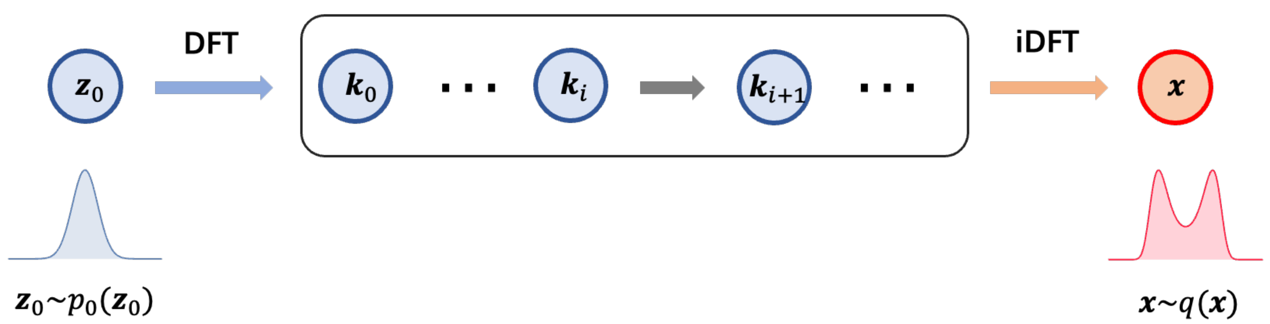
Quantum fluctuations

The kinetic term decouples in Matsubara frequency sapce

Summary II

- **Generative Models**
  - Improve MCMC performance
  - Continuous autoregressive networks
    can reproduce KT phase transition
  - F-Flow model
    can handle multi-mode systems
- **Future works**
  - F-Flow in reduced frequency space
    - Super-resolution
  - Few-body systems
  - From SU(2) to QCD





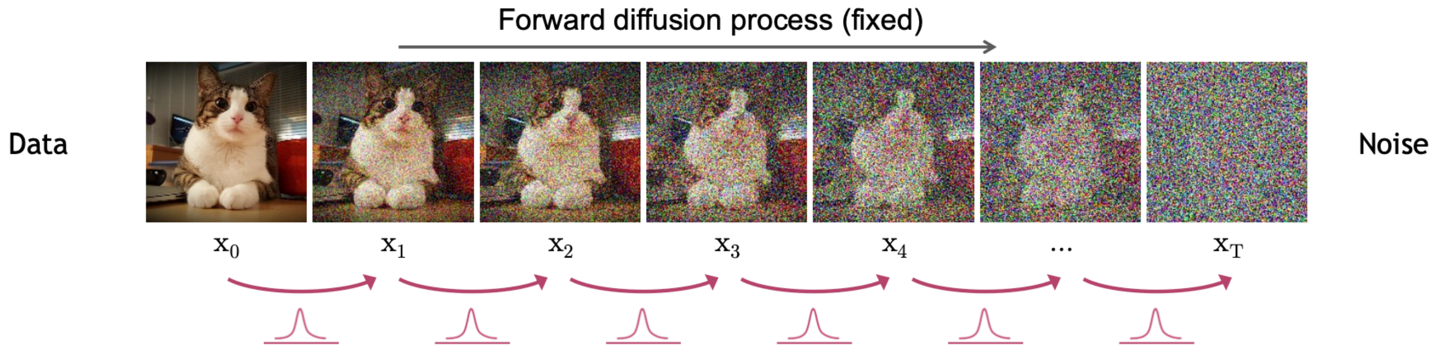
Diffusion Models

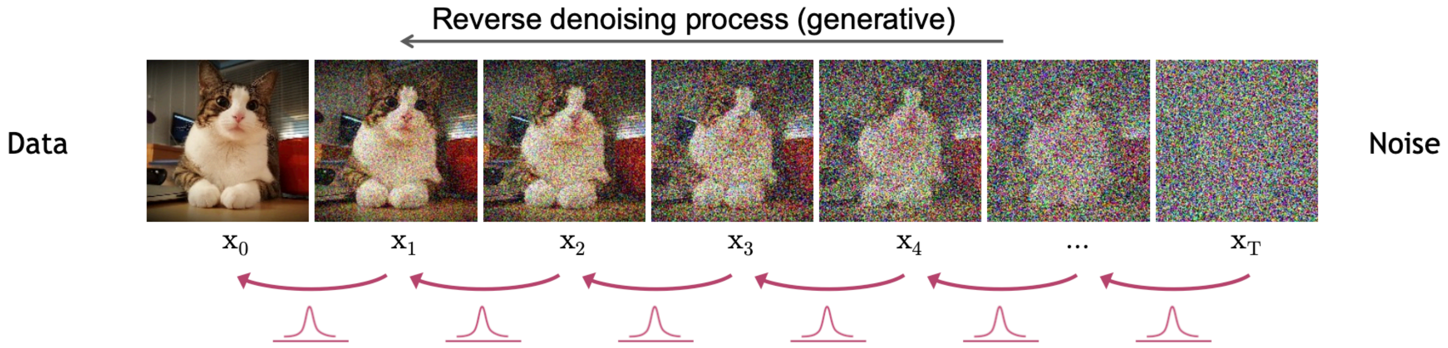
Diffusion Models

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

Denoising Diffusion Models

- Forward diffusion process **gradually adds noise** to input
- Reverse denoising process learns to **generate data by denoising**
- **Probabilistic Models** to learn how to **denoise from a simple distribution to a target distribution**





Space Opera

Jason Allen via Midjourney

Diffusion Models

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

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

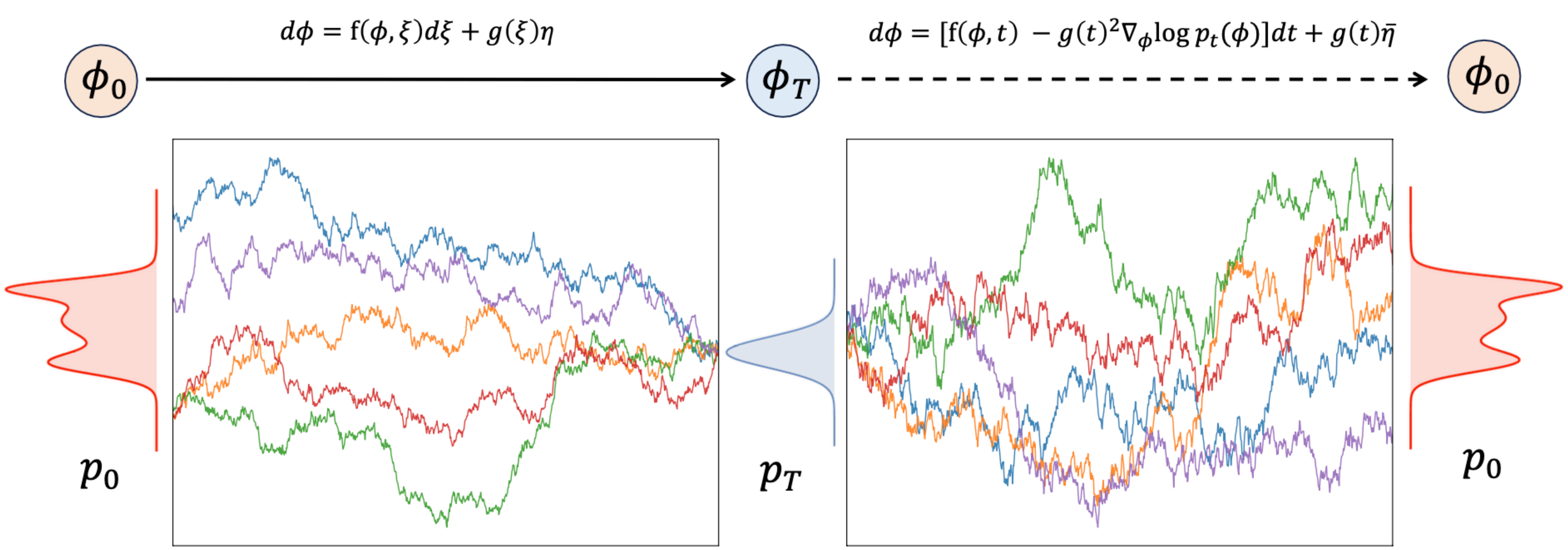
Stochastic Differential Equation(SDE)

- **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**"
  - But how to get the score function ?

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



Diffusion Models

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

Anderson, in Stochastic Processes and their Applications, 1982

Stochastic Differential Equation(SDE)

- **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!

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

Ho et al, NeurIPS, 2020
Song et al., NeurIPS, 2021
Kingma et al., NeurIPS, 2021
Vahdat et al., NeurIPS, 2021
Huang et al., NeurIPS, 2021
Karras et al., arXiv, 2022

Diffusion Models

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

Anderson, in Stochastic Processes and their Applications, 1982

Score-based model

- **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 ?

  Model the score function with neural networks!

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

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

Variance Expanding

$$f(\phi, \xi) = 0$$
$$g(\xi) = \sigma^{\xi}$$
$$\xi \in [0, T]$$

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

Lattice Field Theory

Scalar ϕ^4 field

arXiv: 2309.17082

- **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

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

- Dimensionless form

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

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

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

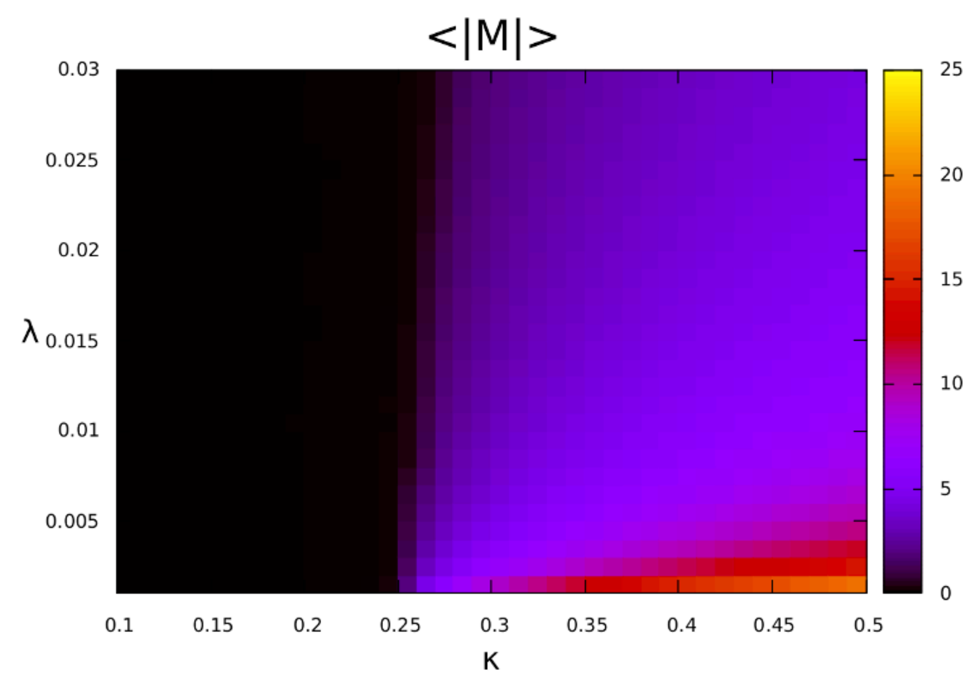
- Hopping parameter κ, Coupling constant λ

Phase Transition

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

$$\kappa_\mathrm{C}(\lambda) = \frac{1-2\lambda}{2d}$$

Order parameter: magnetization



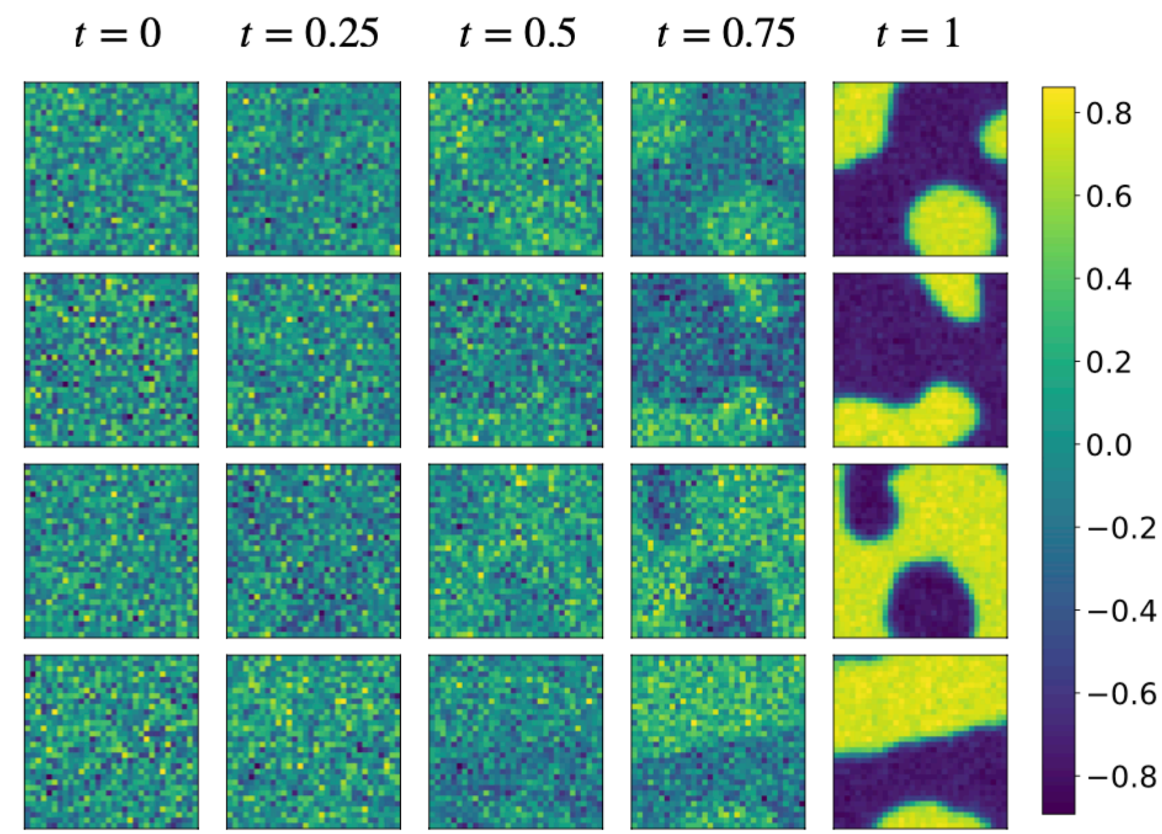
Phase diagram at $d = 2$ @Julian Urban

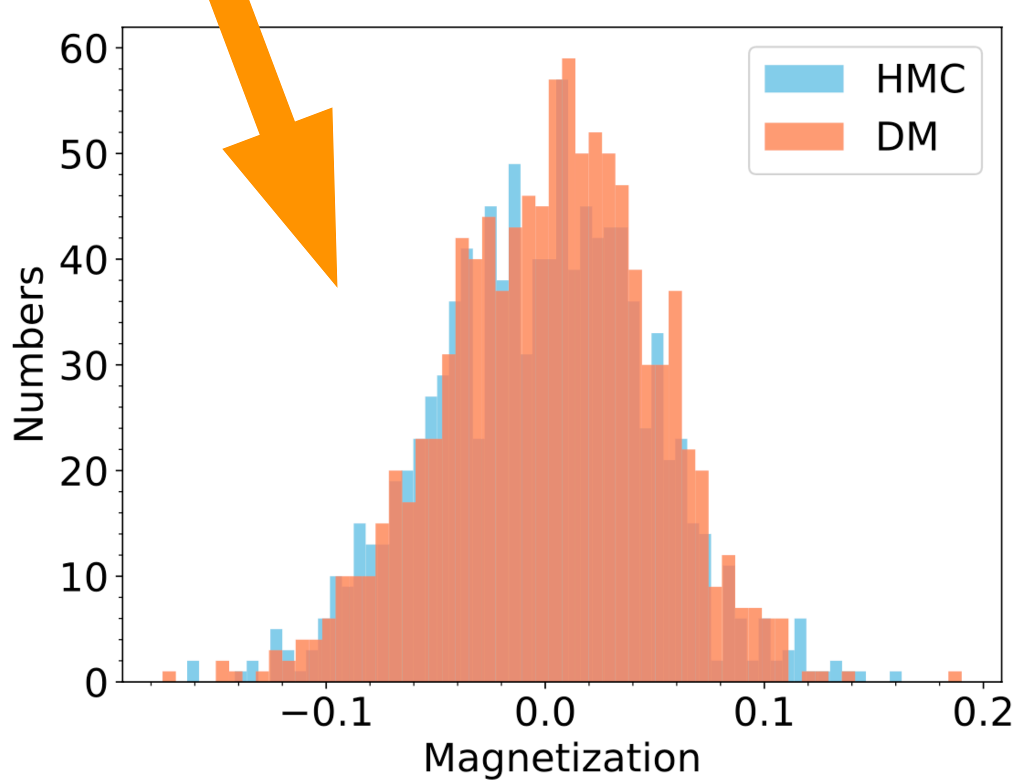
Lattice Field Theory

DM in Different Phases

arXiv: 2309.17082

- **Diffusion models**
  - $T = 1.0, \sigma = 25$
- **Data generation**
  - 2-d 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

Stochastic Quantization

Parisi G. and Wu Y. S., Sci. China, A **24**, ASITP-80-004 (1980).
P. H. Damgaard and H. Hüffel, Stochastic Quantization, Phys. Rept. 152, 227 (1987).
M. Namiki, Basic Ideas of Stochastic Quantization, PTPS 111, 1 (1993).

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

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

τ: fictitious time, α: diffusion constant

- **Fokker-Planck equation**

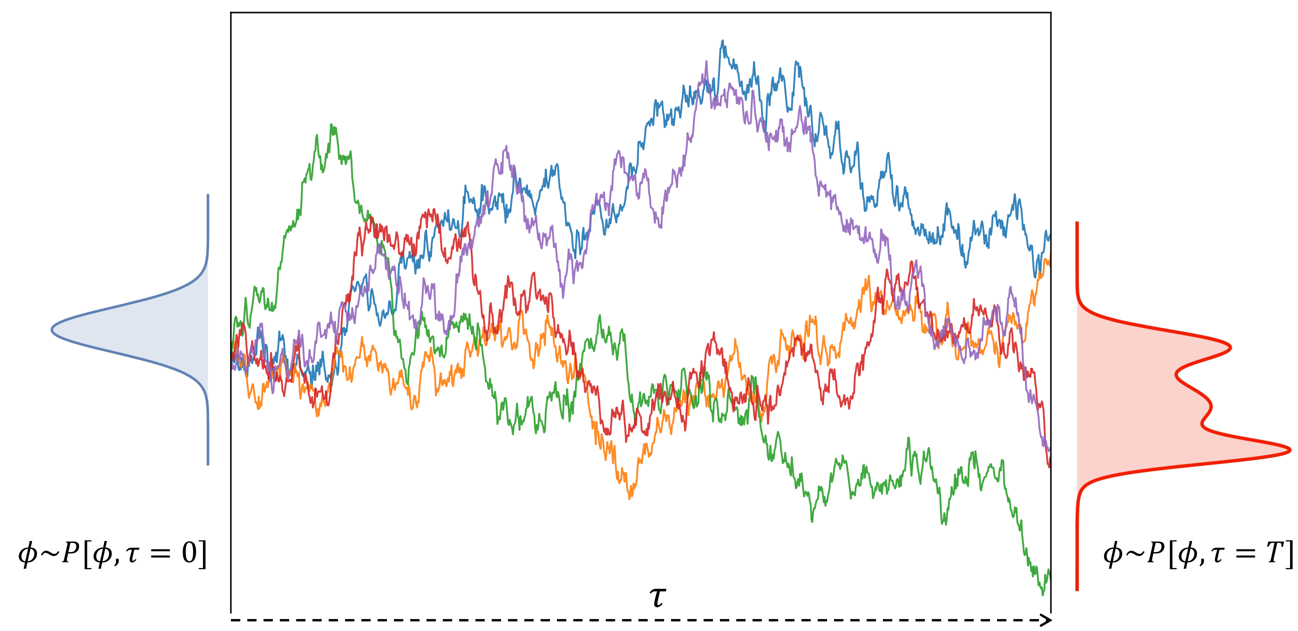
$$\frac{\partial P[\phi,\tau]}{\partial \tau} = \alpha \int d^n x \left\{ \frac{\delta}{\delta \phi} \left(\frac{\delta}{\delta \phi} + \frac{\delta S_E}{\delta \phi} \right) \right\} P[\phi,\tau]$$

Equilibrium solution (long-time limit),

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

- Set the diffusion constant as $\alpha = \hbar$

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



Thermal equilibrium limit
→ Quantum distribution

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

Stochastic Quantization

DMs as SQ

arXiv: 2309.17082

- **Diffusion models(Reverse SDE):**

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

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

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

$$\phi(\tau_{n+1}) = \phi(\tau_n) + g_\tau^2 \nabla_\phi \log q_{\tau_n}[\phi(\tau_n)]\Delta\tau + g_\tau \sqrt{\Delta\tau}\bar{\eta}(\tau_n)$$

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

- **FP equation**

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

$$\nabla_\phi S_{\mathrm{DM}} \equiv -\nabla_\phi \log q_\tau(\phi)$$

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

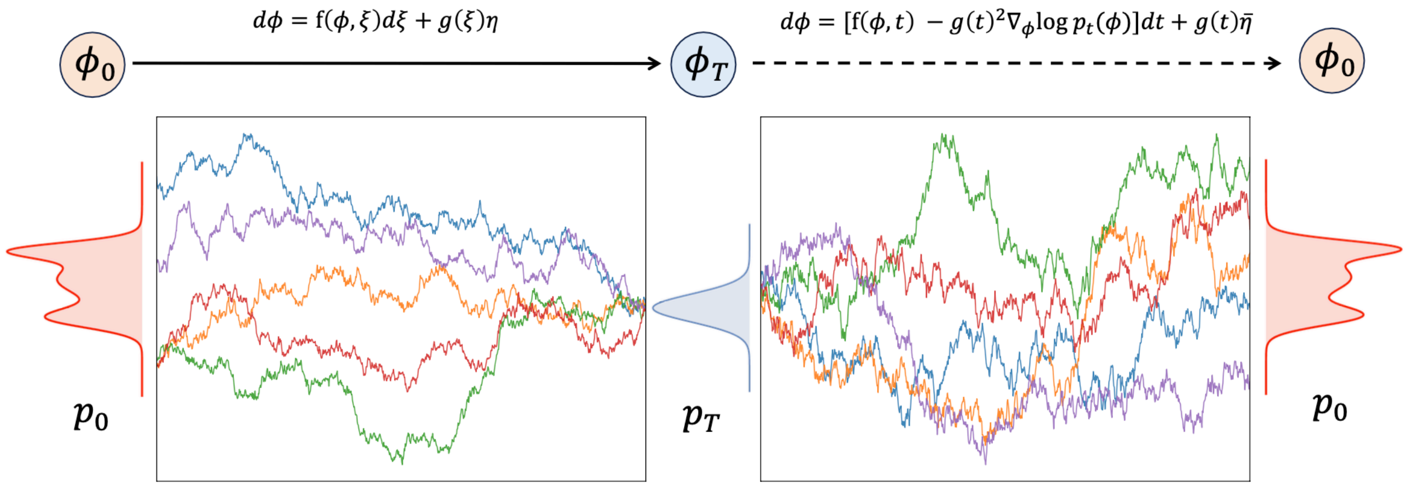
$$p_{\tau=T}(\phi) \to P[\phi, T]$$

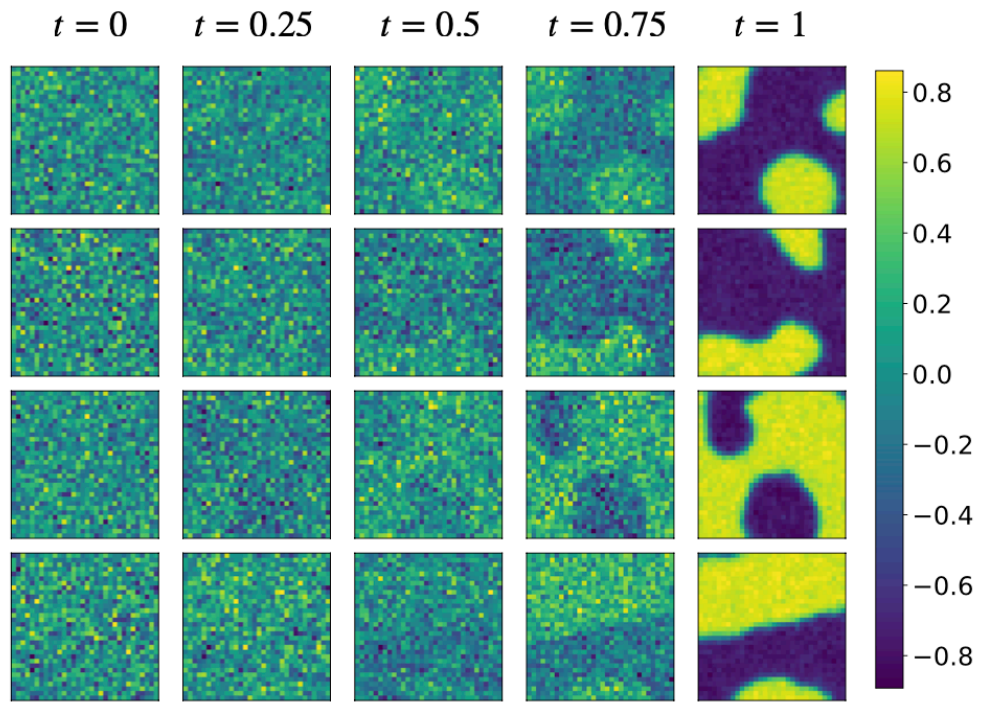
$$O(\bar{\alpha}) \sim O(\hbar)$$

The reverse mode of
a well-trained diffusion model
at $\tau \to T$ serves as
the stochastic quantization
for the input

Summary III

- Serve as an efficient sampler
- **Diffusion models as SQ**
- **Future works**
    - Gauge Field
    - Complex Langevin method(CLM) for fermions
    - Renormalization group(RG) flow

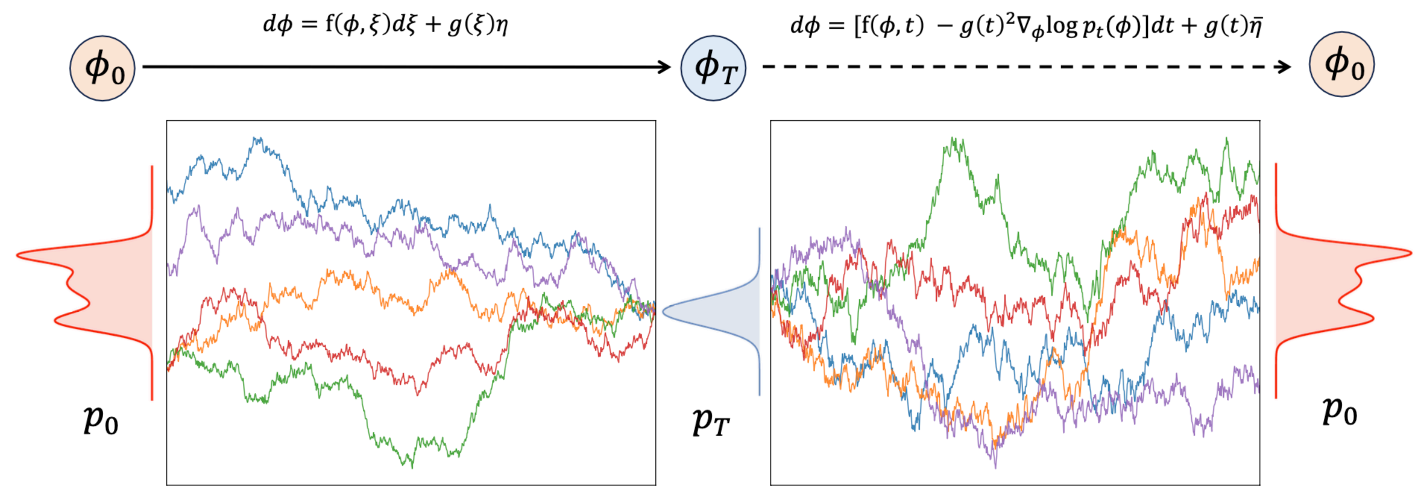


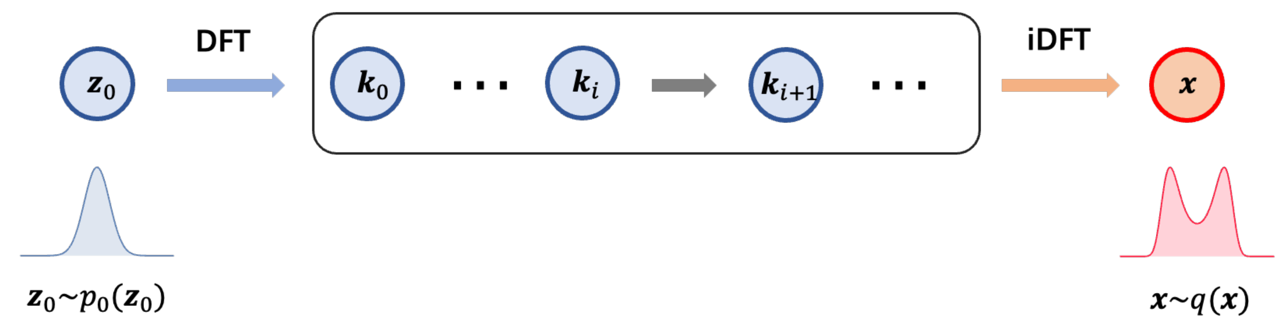


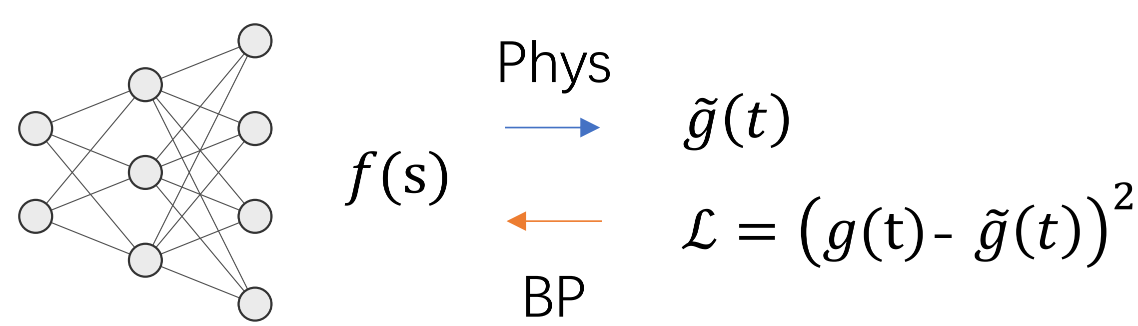
$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)} + \eta(x,\tau)$$

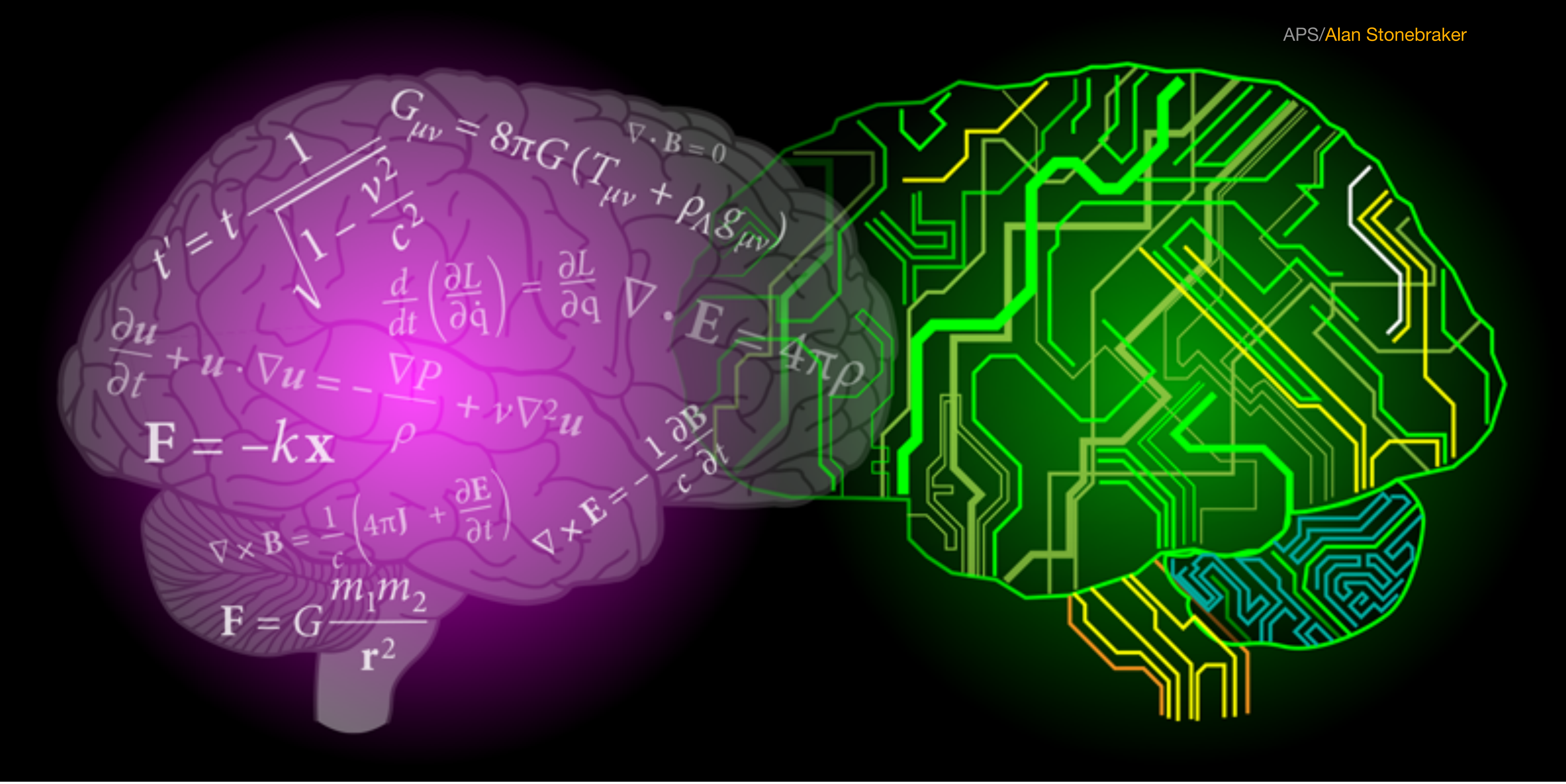
Q&A

- Lattice QFT
  - Theory
    - Stochastic Quantization
  - Configurations
    - Generative Model-based MC
  - Observables
    - Inverse Problems









APS/Alan Stonebraker

Future

ML meets LQFT, opportunities and challenges