Machine Learning for Lattice Field Theory

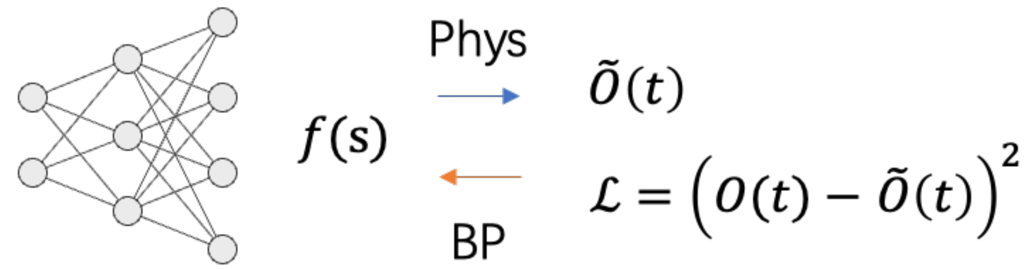
Lingxiao Wang(王凌霄) (FIAS)

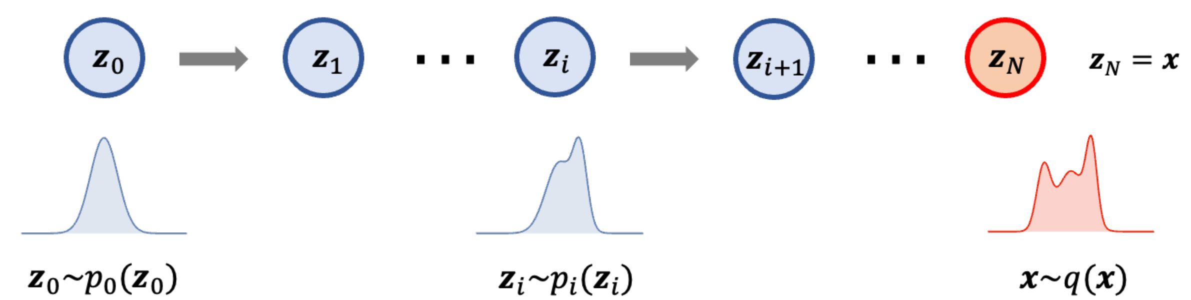
arXiv:2303.15136, arXiv:23xx.xxxxx
Phys. Rev. D 106, L051502 (2022), Comput. Phys. Commun. 282, 108547 (2023);
Chin. Phys. Lett. 39, 120502 (2022), Phys. Rev. D 107, 056001 (2023).

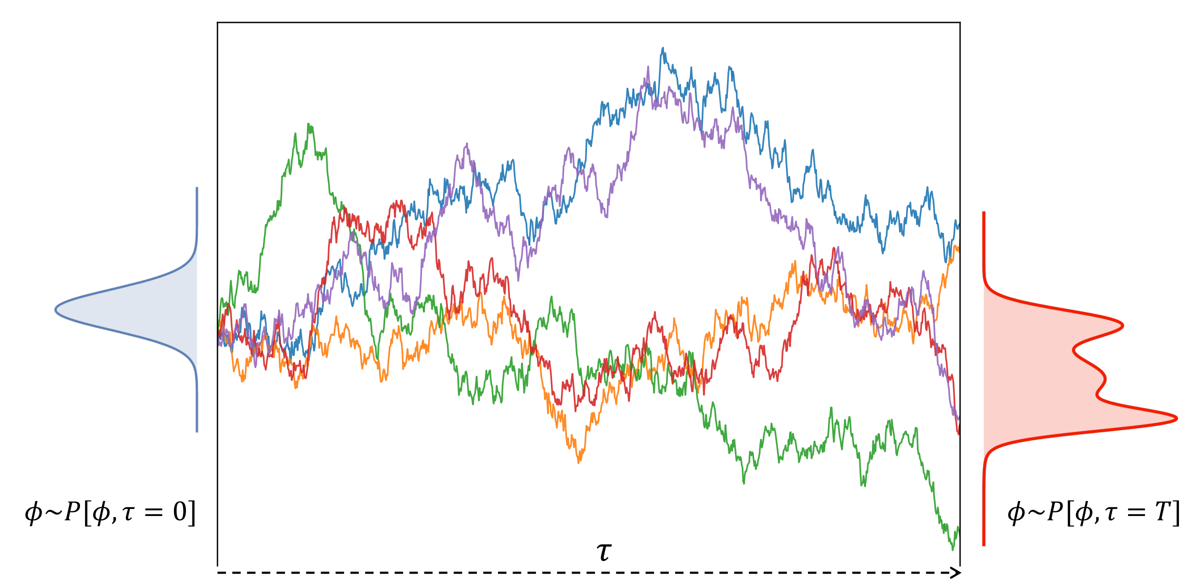
May 26, 2023, UCAS

Outline

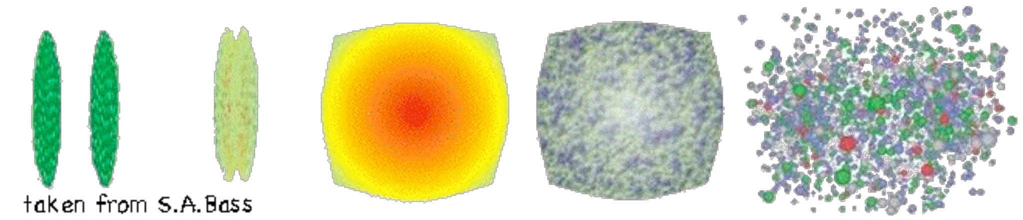
- **ML for QCD matter**
- **Why ML?**
- **ML for LQFT**
- **Inverse Problems**
    - Spectral function reconstruction
- **Generative models**
    - Revisit MCMC
    - Continuous Autoregressive Networks
    - Fourier-Flow Model
- **Diffusion Models**
- **Summary**



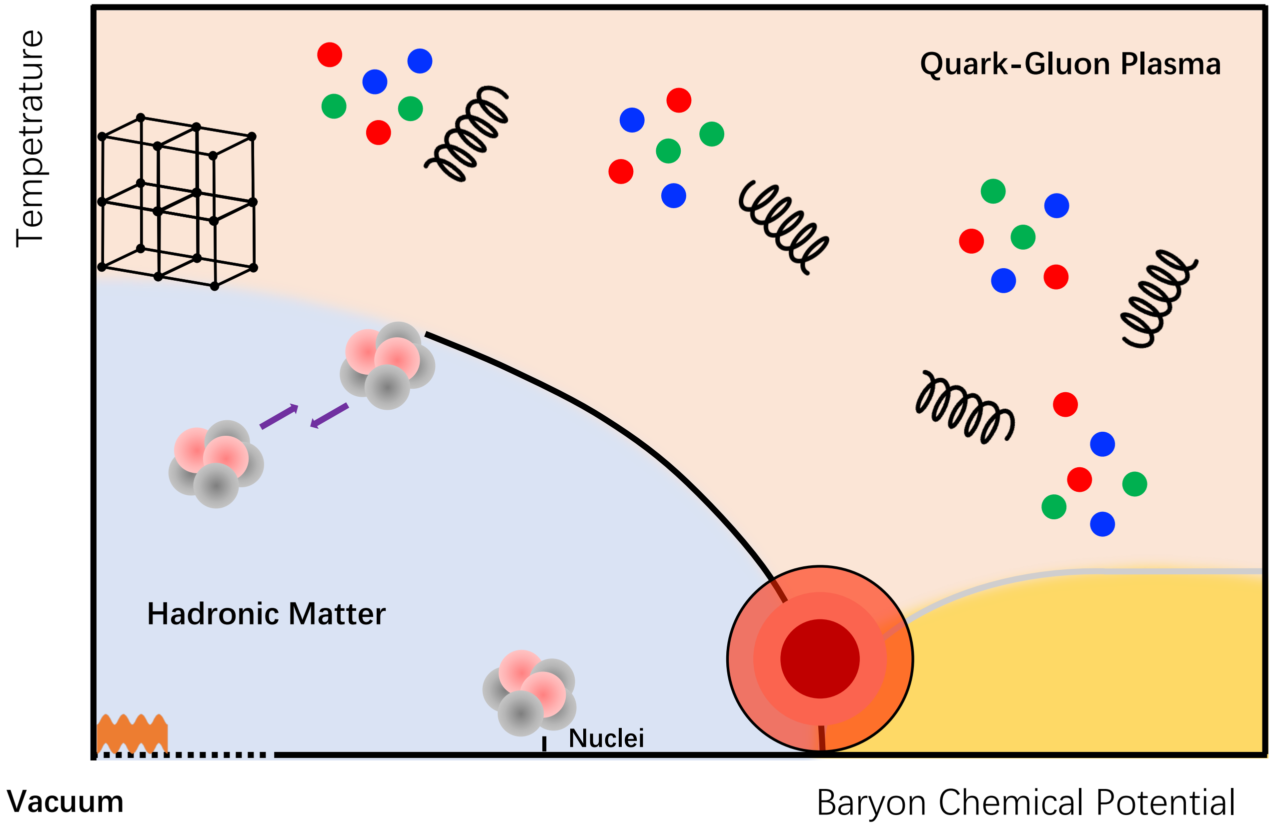


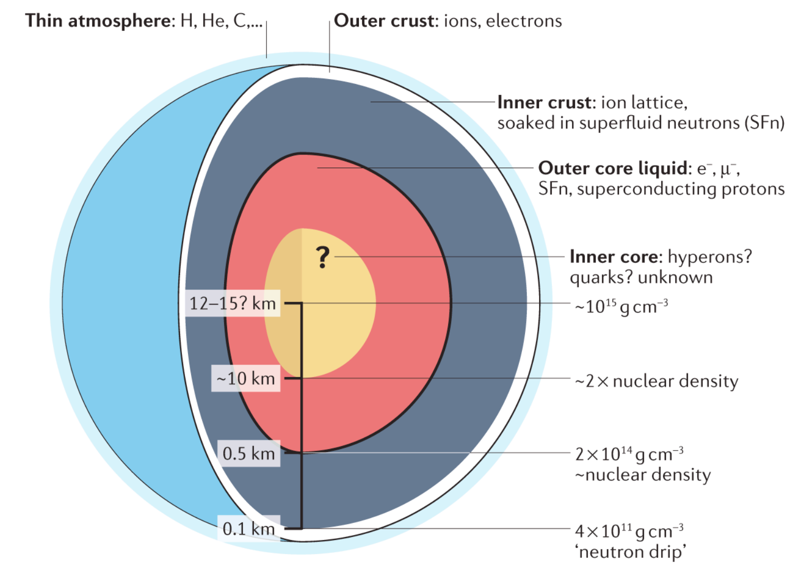


QCD matter



Heavy-Ion Collisions

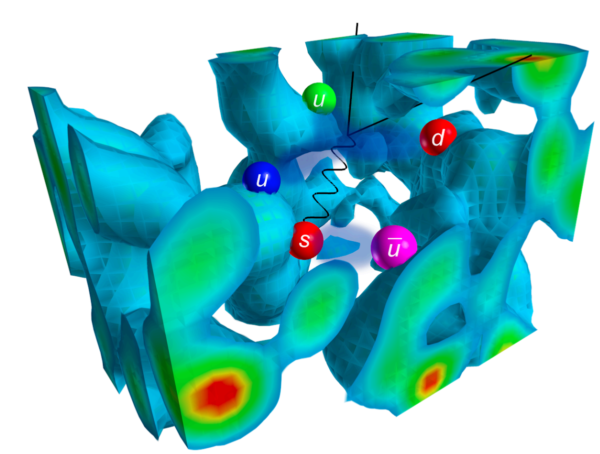




Nat. Rev. Phys. 4, 237–246 (2022)

Exploring QCD matter in three "**labs**",

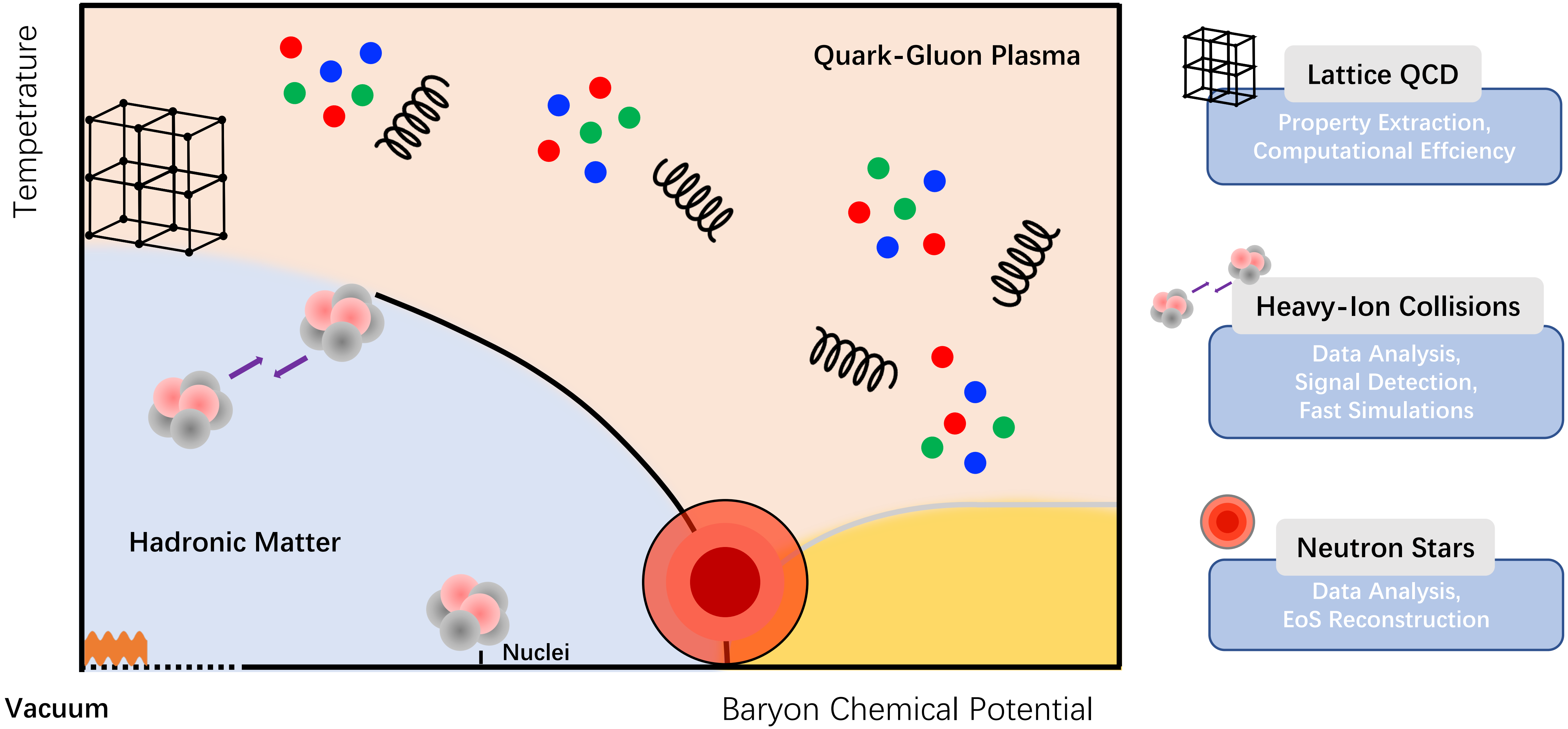
- **Heavy-Ion Collisions** : compress matter to **high-T** and **high-$\mu_{\mathbf{B}}$**
- **Neutron Star** : dense matter, merger events at **low-T** and **high-$\mu_{\mathbf{B}}$**
- **Lattice QCD** : numerically solve QCD Lagrangian at **finite-T** and $\mu_{\mathbf{B}} \sim 0$



Lattice QCD © Derek Leinweber/CSSM/University of Adelaide

Why ML?

arXiv:2303.15136



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

What is ML?

Machine Learning (from Data)

Big Data + Deep Models

↓ GPU

Successful Deep Learning!

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 is a subfield of artificial intelligence that involves enabling computers to learn from data, without being explicitly programmed. It uses statistical algorithms and models to analyze and identify patterns in data, and then uses these patterns to make predictions or decisions about new data inputs.

—— ChatGPT3.5

ML for LQFT

List applications of Machine Learning in Lattice QCD.

1. Predicting the onset of the QCD phase transition: Machine learning techniques are being used to study the behavior of quark-gluon plasma, which is a necessary step in predicting the onset of the QCD phase transition.

2. Modeling phase transitions in lattice QCD simulations: Machine learning methods can be used to model phase transitions in lattice QCD simulations, which can lead to significant improvements in our understanding of the behavior of the strong force.

3. Improving Monte Carlo simulations: Monte Carlo simulations are widely used in lattice QCD calculations. Machine learning can be used to improve the efficiency and accuracy of these simulations, leading to more precise predictions.

4. Extracting information from lattice QCD data: Machine learning algorithms can be used to extract useful information from lattice QCD data, helping to identify patterns and trends that might be difficult to discern using traditional techniques.

5. Developing new QCD-inspired models: Machine learning can be used to develop new models inspired by QCD, which can give insight into the behavior of the strong force under different conditions.

6. Speeding up calculations: Machine learning can be used to speed up calculations in lattice QCD simulations, allowing researchers to simulate larger systems and study more complex phenomena.

7. Studying the equation of state of the quark-gluon plasma: Machine learning can be used to study the equation of state of the quark-gluon plasma, a crucial step in understanding the behavior of matter at extreme temperatures and densities.

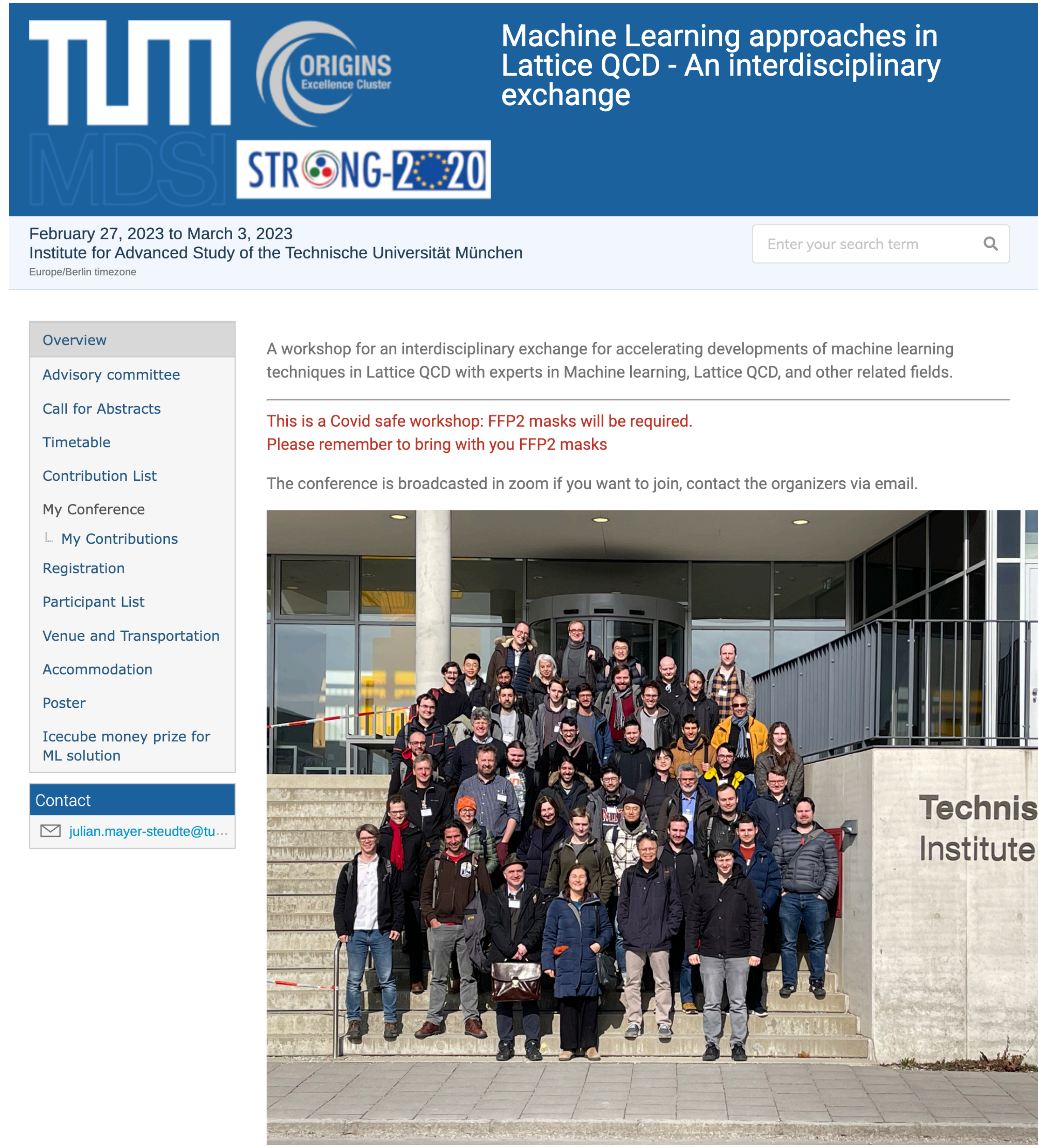
8. Developing improved lattice gauge theories: Machine learning can be used to develop improved lattice gauge theories, which can help to solve some of the outstanding problems in lattice QCD calculations.

9. Predicting the outcomes of lattice QCD experiments: Machine learning can be used to predict the outcomes of lattice QCD experiments, helping to guide experimental design and interpretation of results.

CHATGPT

ML for LQFT

First Workshop



Panels

- Generative models
- Gauge field generation
- ML for Sign Problem
- ML for Phase Transitions
- ML for EFT
- Physical interpretation
  - Symmetry
  - Automatic Differentiation
  - Symbolic Learning
- Inverse problems
  - Spectral Reconstruction

...

ML for LQFT

Inverse Problems

Spectral function reconstruction

Generative models

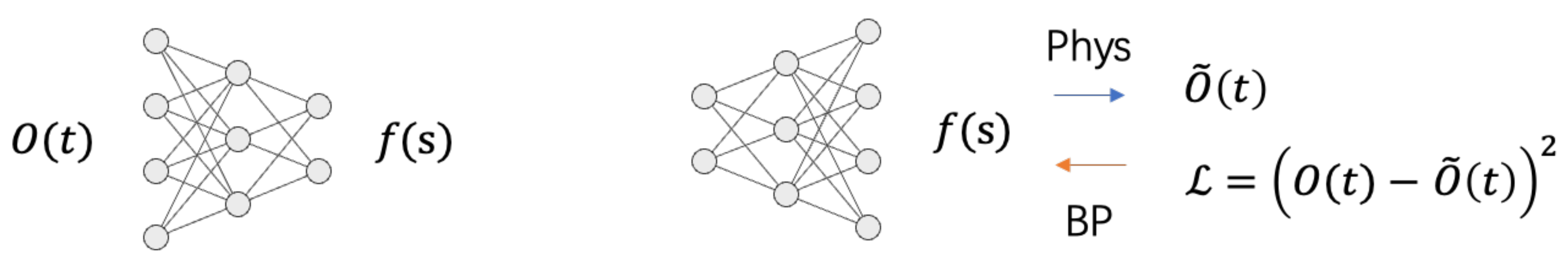
Revisit MCMC

Continuous Autoregressive Networks

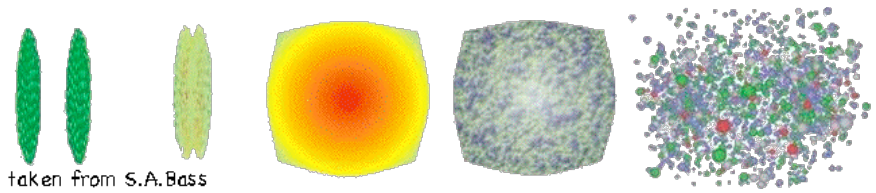
Fourier-Flow Model

Diffusion Models

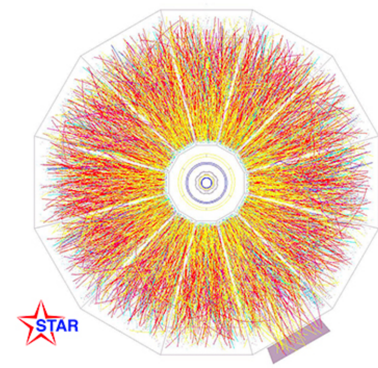
Inverse Problems



Inverse Problems

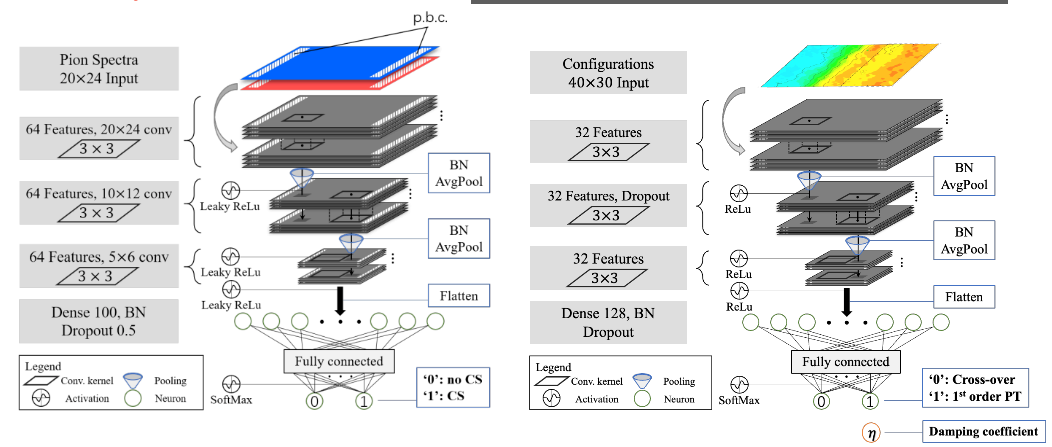


Strongly interaction matter properties

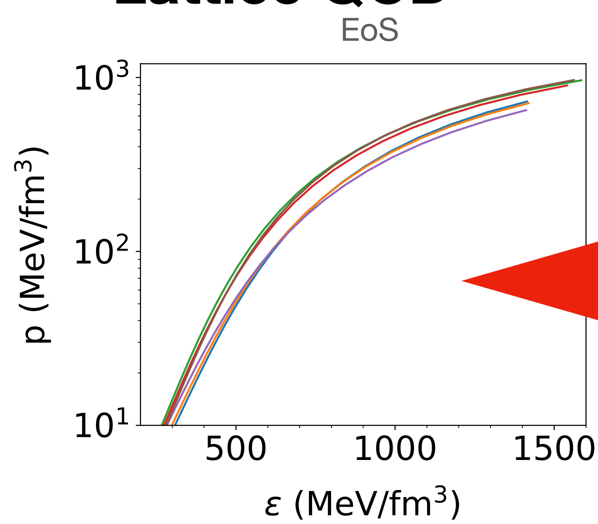


Heavy-Ion Collisions

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



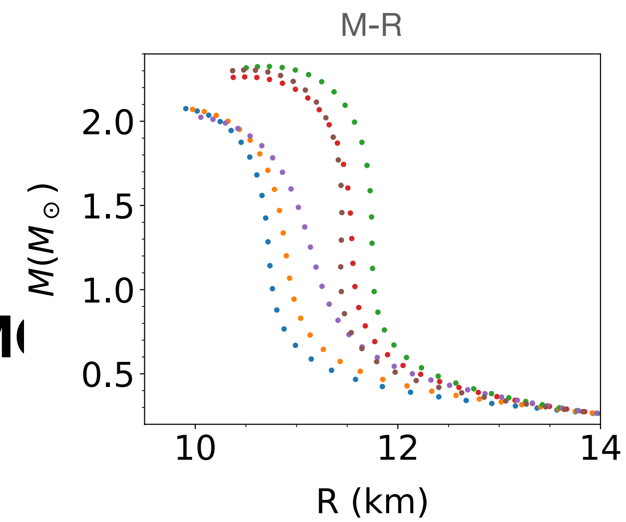
- **Lattice QCD**



CEP)

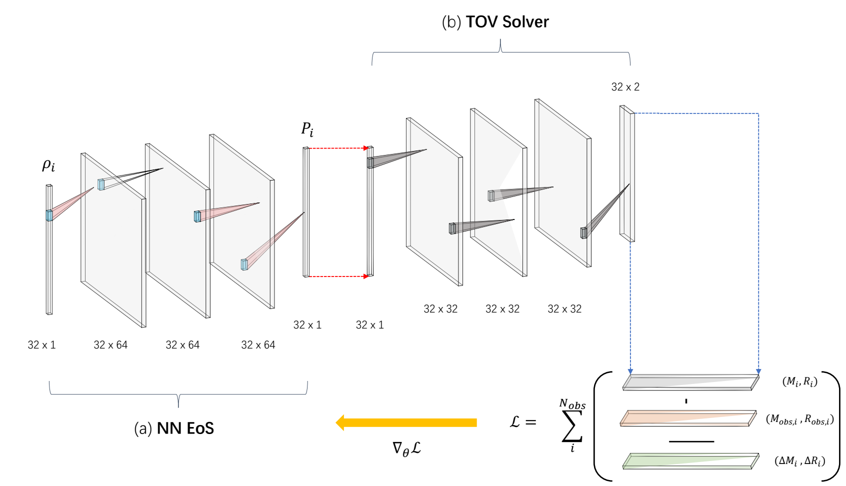
ition

Carlo(QM

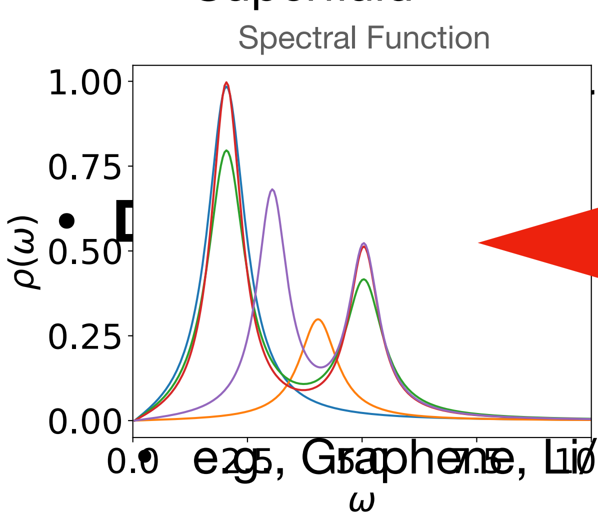


Neutron Star

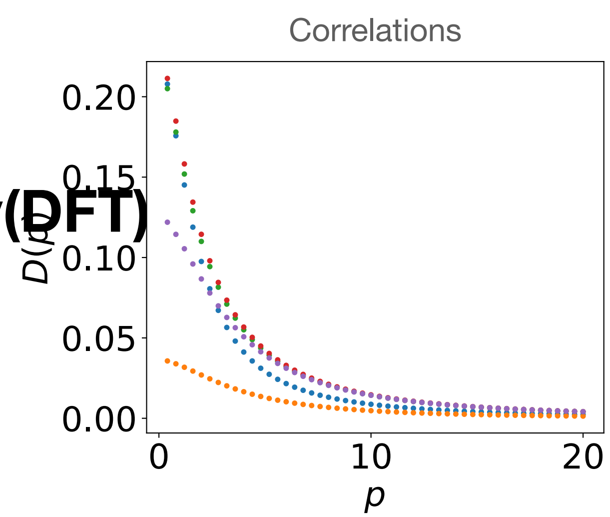
Phys. Rev. D 107, 083028; JCAP08 (2022) 071



- Superfluid



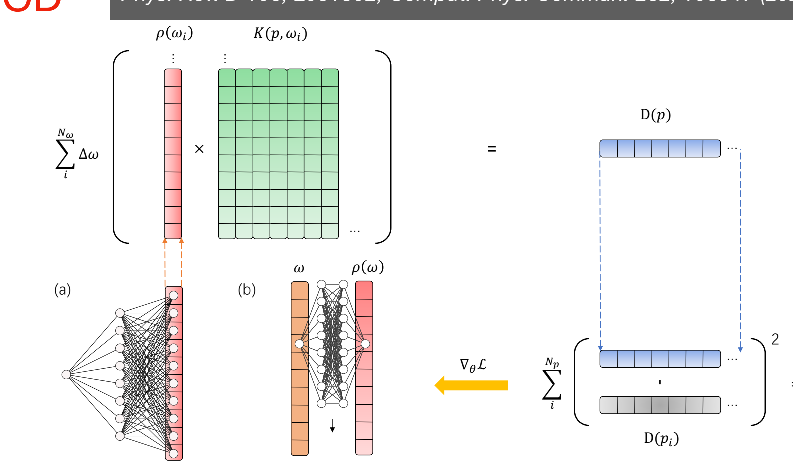
- D… Theory(DFT)



- e.g., Graphene, Li/Ca/Fe…

Lattice QCD

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



Spectral Functions

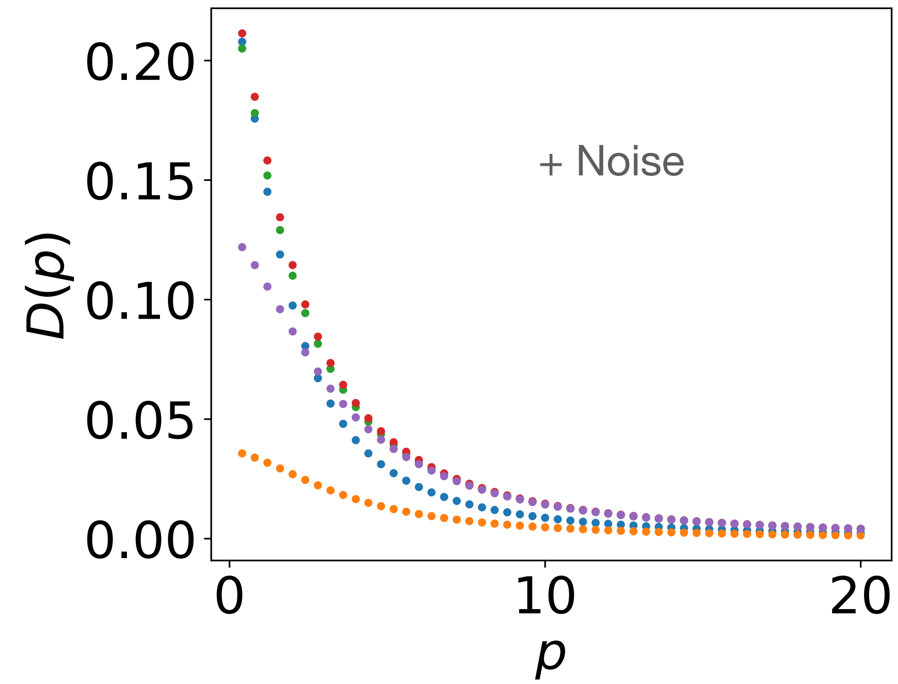
Rebuilding spectral functions

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

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

Kallen—Lehmann(KL) representation

Forward process

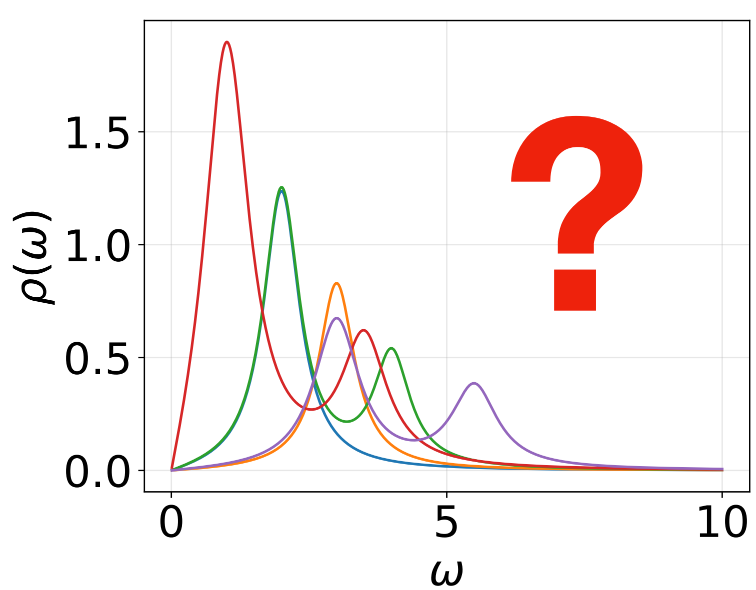


Model

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? Fundmental difficulties

- 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**;
- It's caused by the **numerical inaccuracy** of the correlation measurements (induced high degeneracy in solution space).

Comput. Phys. Commun. 282, 108547

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

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

highly rectangular

vectorization

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

eigenvalue problem

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

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

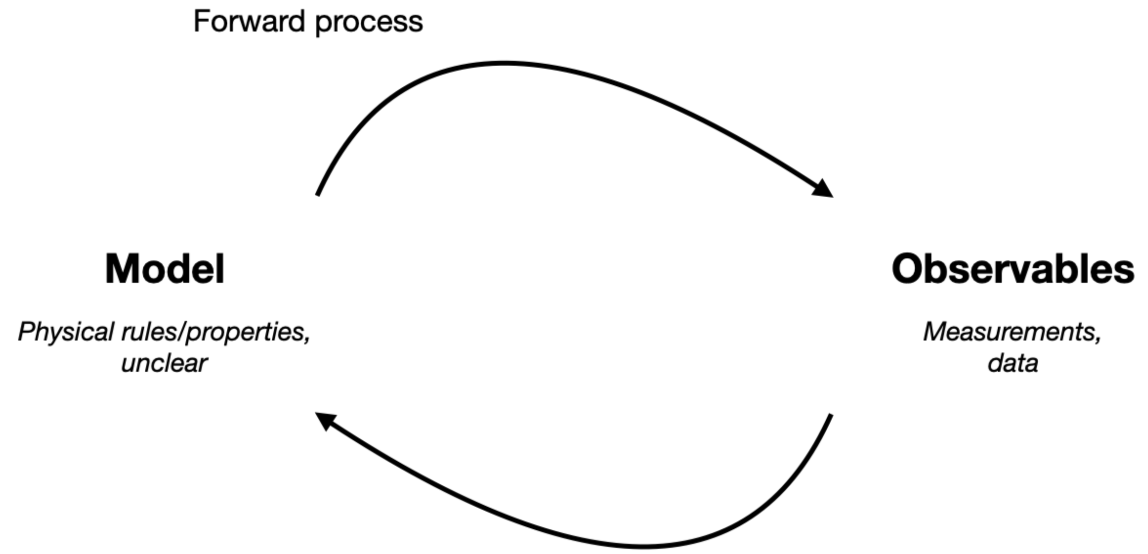
On the numerical inversion of the Laplace transform and similar Fredholm integral equations of the first kind

J G McWhirter and E R Pike
The Royal Signals and Radar Establishment, St Andrews Road, Great Malvern, Worcs WR14 3PS, UK

Received 27 February 1978

Spectral Functions

Methodology



- **Classical methods**
  - Truncated Singular Value Decomposition (TSVD)
  - Tikhonov regularization, …

  H. W. Engl and C. W. Groetsch, editors , ***Inverse and Ill-Posed Problems*** (Academic Press, Boston, 1987).

- **Baysian methods**
  - **Maximum Entropy Method(MEM)**

  M. Asakawa, Y. Nakahara, and T. Hatsuda, ***Maximum Entropy Analysis of the Spectral Functions in Lattice QCD***, Progress in Particle and Nuclear Physics **46**, 459 (2001).

  - Bayesian Reconstruction(BR)

  Y. Burnier and A. Rothkopf, **Bayesian *Approach to Spectral Function Reconstruction for Euclidean Quantum Field Theories***, Phys. Rev. Lett. 111, 182003 (2013).
  A. Rothkopf, ***Bayesian Inference of Real-Time Dynamics from Lattice QCD***, arXiv:2208.13590.

- **Supervised Learning** the inverse mapping

  L. Kades, J. M. Pawlowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. P. G. Ziegler, ***Spectral Reconstruction with Deep Neural Networks***, Phys. Rev. D **102**, 096001 (2020).

  R. Fournier, L. Wang, O. V. Yazyev, and Q. Wu, ***Artificial Neural Network Approach to the Analytic Continuation Problem***, Phys. Rev. Lett. **124**, 056401 (2020).

  H. Yoon, J.-H. Sim, and M. J. Han, ***Analytic Continuation via Domain Knowledge Free Machine Learning***, Phys. Rev. B **98**, 245101 (2018).

- **New developments**

  Gausian process

  J. Horak, J. M. Pawlowski, J. Rodríguez-Quintero, J. Turnwald, J. M. Urban, N. Wink, and S. Zafeiropoulos, ***Reconstructing QCD Spectral Functions with Gaussian Processes***, 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

Spectral reconstruction

Why ill-posed? Fundmental difficulties

- 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 <u>continuous correlation functions</u>**;

- It's caused by the **numerical inaccuracy** of the correlation measurements (<u>induced high degeneracy in solution space</u>).

<u>Comput. Phys. Commun. 282, 108547</u>

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

eigenvalue problem

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

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

On the numerical inversion of the Laplace transform and similar Fredholm integral equations of the first kind

J G McWhirter and E R Pike

The Royal Signals and Radar Establishment, St Andrews Road, Great Malvern, Worcs WR14 3PS, UK

Received 27 February 1978

$$\lambda_{\pm,s}^{[\text{Fourier}]} = \pm\sqrt{\pi/2}\,,$$

$$\lambda_{\pm,s}^{[\text{Laplace}]} = \pm\sqrt{\pi}\cosh^{-\frac{1}{2}}(\pi s)$$

where $s \in (-\infty, +\infty)$ is the real-valued, continuous label of the eigenstate

$\psi_s(x)$ can serve as a complete set of basis for Hilbert space

Spectral reconstruction

Why ill-posed? Fundmental difficulties

- 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**;
- It's caused by the **numerical inaccuracy** of the correlation measurements (induced high degeneracy in solution space).

Comput. Phys. Commun. 282, 108547

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

eigenvalue problem

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

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

Kallen—Lehmann(KL) kernel

$$\psi_{+,s}(x) = \frac{\cos\left(s\ln(x/a)\right)}{\sqrt{\pi}\, x/a}$$

$$\psi_{-,s}(x) = \frac{\sin\left(s\ln(x/a)\right)}{\sqrt{\pi}\, x/a}$$

eigenvectors

$$\lambda_{\pm,s} = \frac{1}{2\cosh(\pi s/2)}$$

Will contribute **"null-mode" (zero-mode)**

where $s \in (-\infty, +\infty)$ is the real-valued, continuous label of the eigenstate

Spectral reconstruction

Comput. Phys. Commun. 282, 108547

Why ill-posed? Fundmental difficulties

- The ill-posedness of the spectral reconstruction **fundamentally exists even for continuous correlation functions**;
- It's caused by the **numerical inaccuracy** of the correlation measurements (induced high degeneracy in solution space).

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

$$\lambda_{\pm,s} = \frac{1}{2\cosh(\pi s/2)}$$

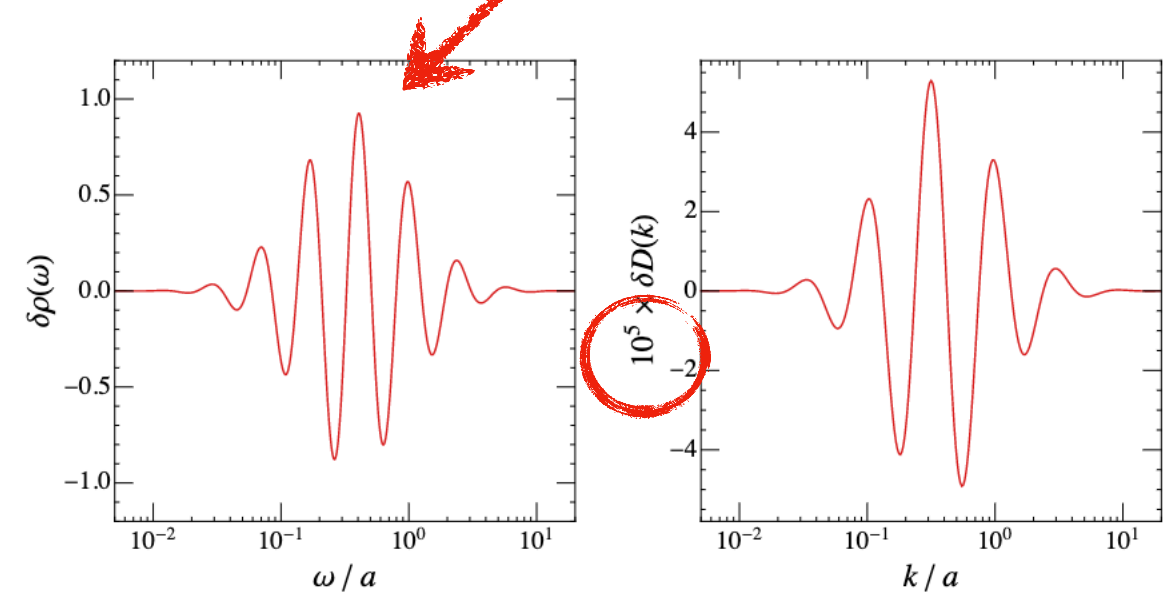
Will contribute **"null-mode" (zero-mode)**

$$\rho(\omega) = \int_0^\infty \frac{D(k)\,\mathrm{d}k}{\pi\,\omega} \int_{-\infty}^\infty \mathrm{d}s\ \cos(s\ln\frac{k}{\omega})\cosh(\frac{\pi s}{2})$$

$$\psi_{+,s}(x) = \frac{\cos\left(s\ln(x/a)\right)}{\sqrt{\pi\, x/a}}$$

$$\psi_{-,s}(x) = \frac{\sin\left(s\ln(x/a)\right)}{\sqrt{\pi\, x/a}}$$

eigenvectors



Spectral reconstruction

[*Comput. Phys. Commun. 282, 108547*]

Why ill-posed? Fundmental difficulties

- The ill-posedness of the spectral reconstruction **fundamentally exists even for <u>continuous correlation functions</u>**;
- It's caused by the **numerical inaccuracy** of the correlation measurements (<u>induced high degeneracy in solution space</u>).

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

$$\lambda_{\pm,s} = \frac{1}{2\cosh(\pi s/2)}$$

Will contribute **"null-mode" (zero-mode)**

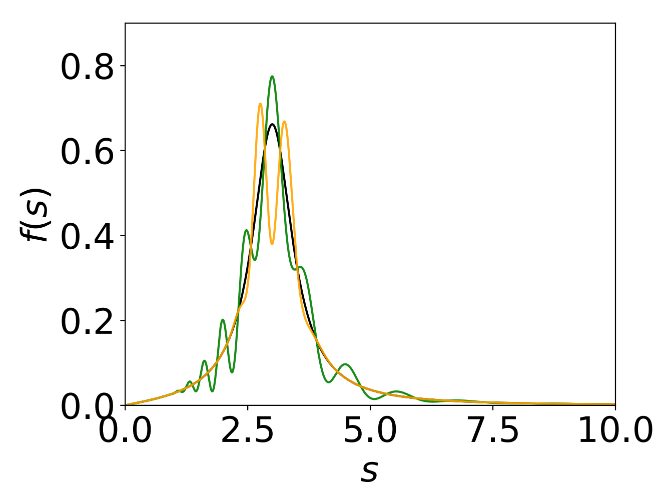
$$\rho(\omega) = \int_0^\infty \frac{D(k)\,\mathrm{d}k}{\pi\,\omega} \int_{-\infty}^{\infty} \mathrm{d}s\ \cos(s \ln \frac{k}{\omega})\cosh(\frac{\pi s}{2})$$

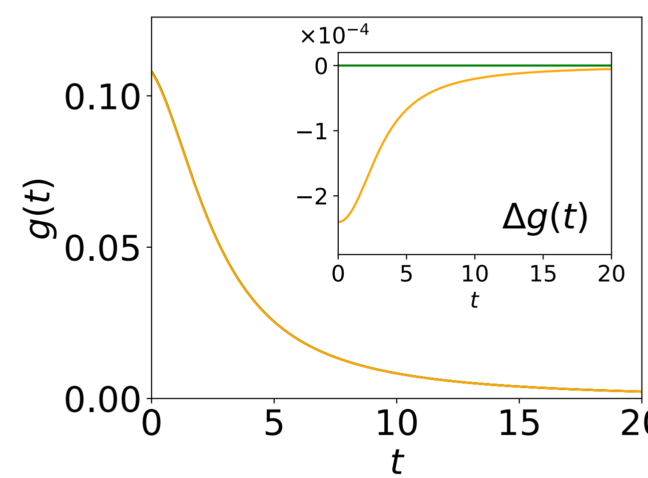
$$\psi_{+,s}(x) = \frac{\cos\left(s\ln(x/a)\right)}{\sqrt{\pi}\, x/a}$$

$$\psi_{-,s}(x) = \frac{\sin\left(s\ln(x/a)\right)}{\sqrt{\pi}\, x/a}$$

eigenvectors

Regulators should kill "null-modes"!



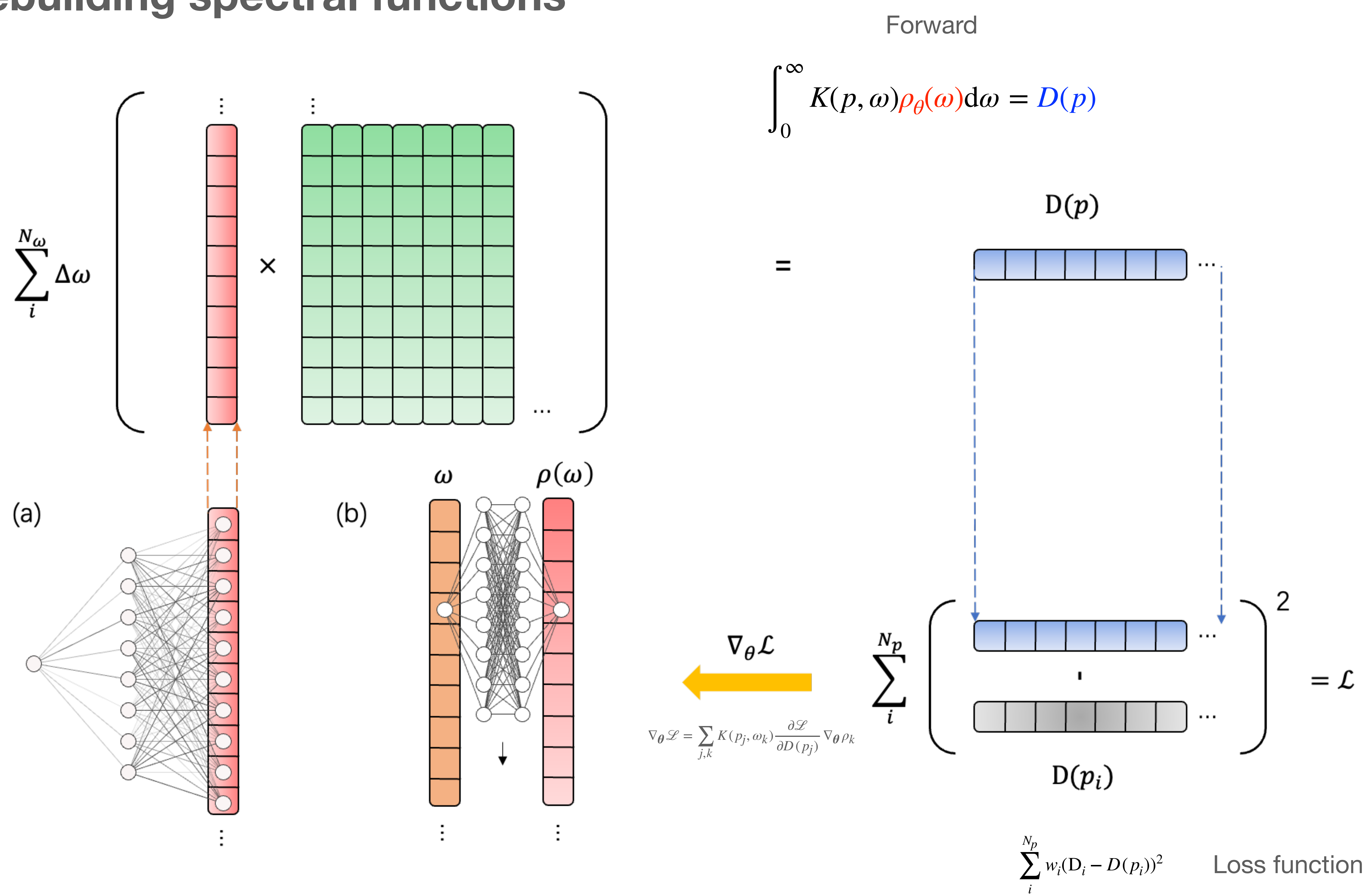


$g(t) = \int_a^b K(t,s) f(s) ds$

AD Framework

Rebuilding spectral functions

Phys. Rev. D 106, L051502



Framework

- **Automatic differentiation (AD)**
  - It refers to a general way of taking a program which **computes a value**, and **automatically constructing a procedure for computing derivatives of that value.**

Chain rule: $h'(x) = f'(g(x))g'(x).$

  - Example

How we compute the derivatives of logistic least squares regression in a net

ω weights, b bias, $\sigma(z)$ activation function
x input, y output, t target, $\mathscr{L}$ loss function

Computing the loss:

$$z = wx + b$$
$$y = \sigma(z)$$
$$\mathscr{L} = \frac{1}{2}(y - t)^2$$

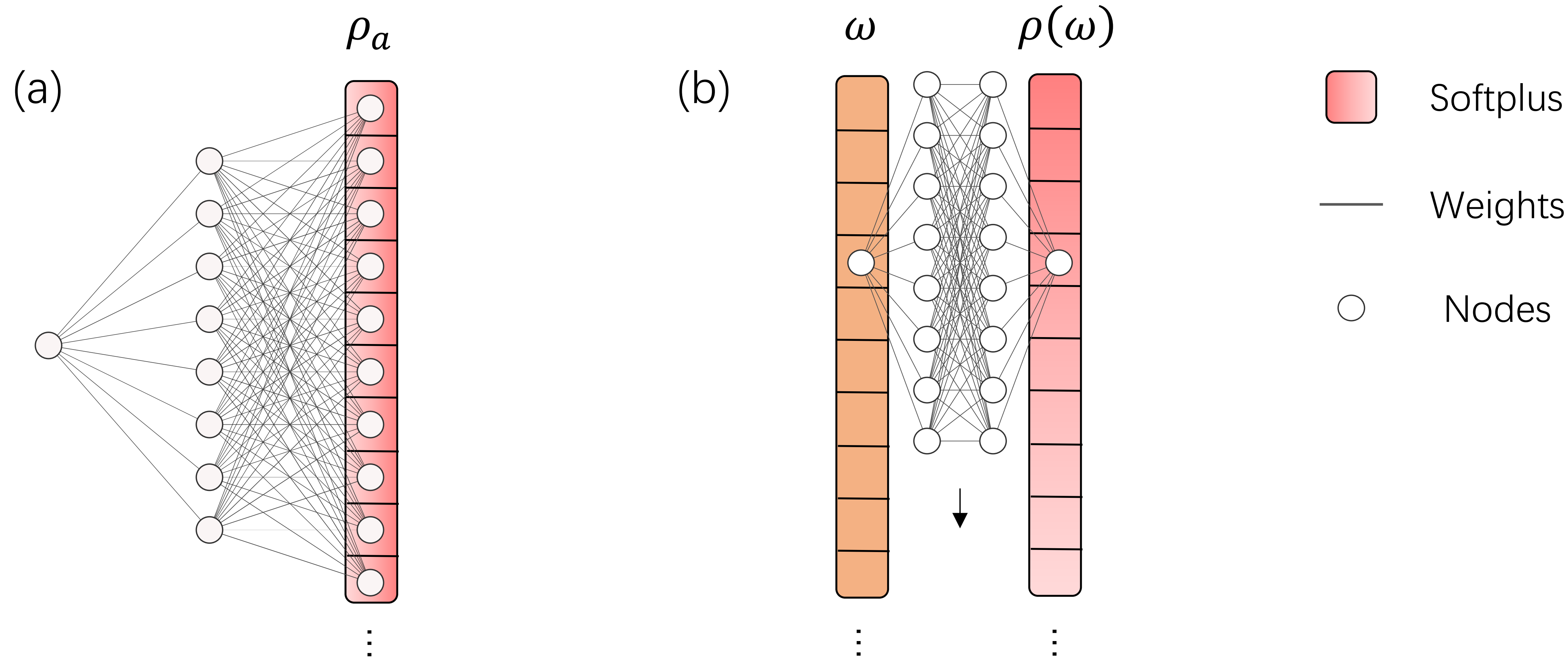
Computing the derivatives:

$$\overline{\mathscr{L}} = 1$$
$$\bar{y} = y - t$$
$$\bar{z} = \bar{y}\sigma'(z)$$
$$\bar{w} = \bar{z}x$$
$$\bar{b} = \bar{z}$$

1. *Automatic Differentiation in Machine Learning: A Survey*, J. Mach. Learn. Res. **18**, 5595 (2017).
2. *Physics-Based Deep Learning*, ArXiv:2109.05237 [Physics] (2021).

Neural Networks

Phys. Rev. D 106, L051502



(a) NN : $(\rho_1, \rho_2, \cdots, \rho_{N_\omega})$

Differentiable variables : Network weights $\{\theta\}$

Adam, L2 ($\lambda = 10^{-3} \rightarrow 10^{-8}$), Smoothness ($\lambda_s = 10^{-4} \rightarrow 0$)

width = 64 and depth = 3 with bias

(b) NN-P2P : $\rho(\omega)$

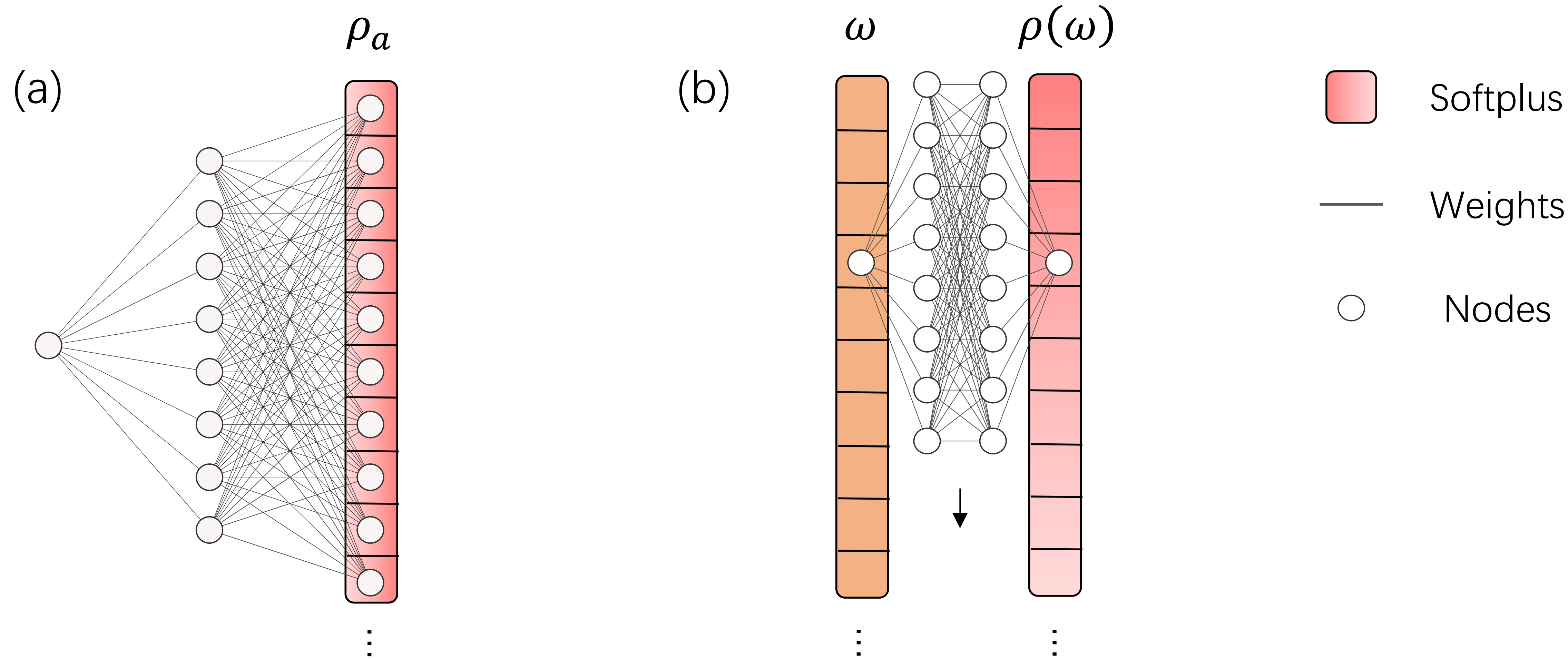
Differentiable variables : Network weights $\{\theta\}$

Adam, L2 ($\lambda = 10^{-6} \rightarrow 0$)

width = 64 and depth = 3 with bias

Neural Networks

Phys. Rev. D 106, L051502



Gradient-based Optimization

Adam : $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$

Regularization

L2 : $\lambda||\theta||_2^2$

Smoothness: $\lambda_s \sum_i^{N_\omega} (\rho_i - \rho_{i-1})^2$

Physical Prior

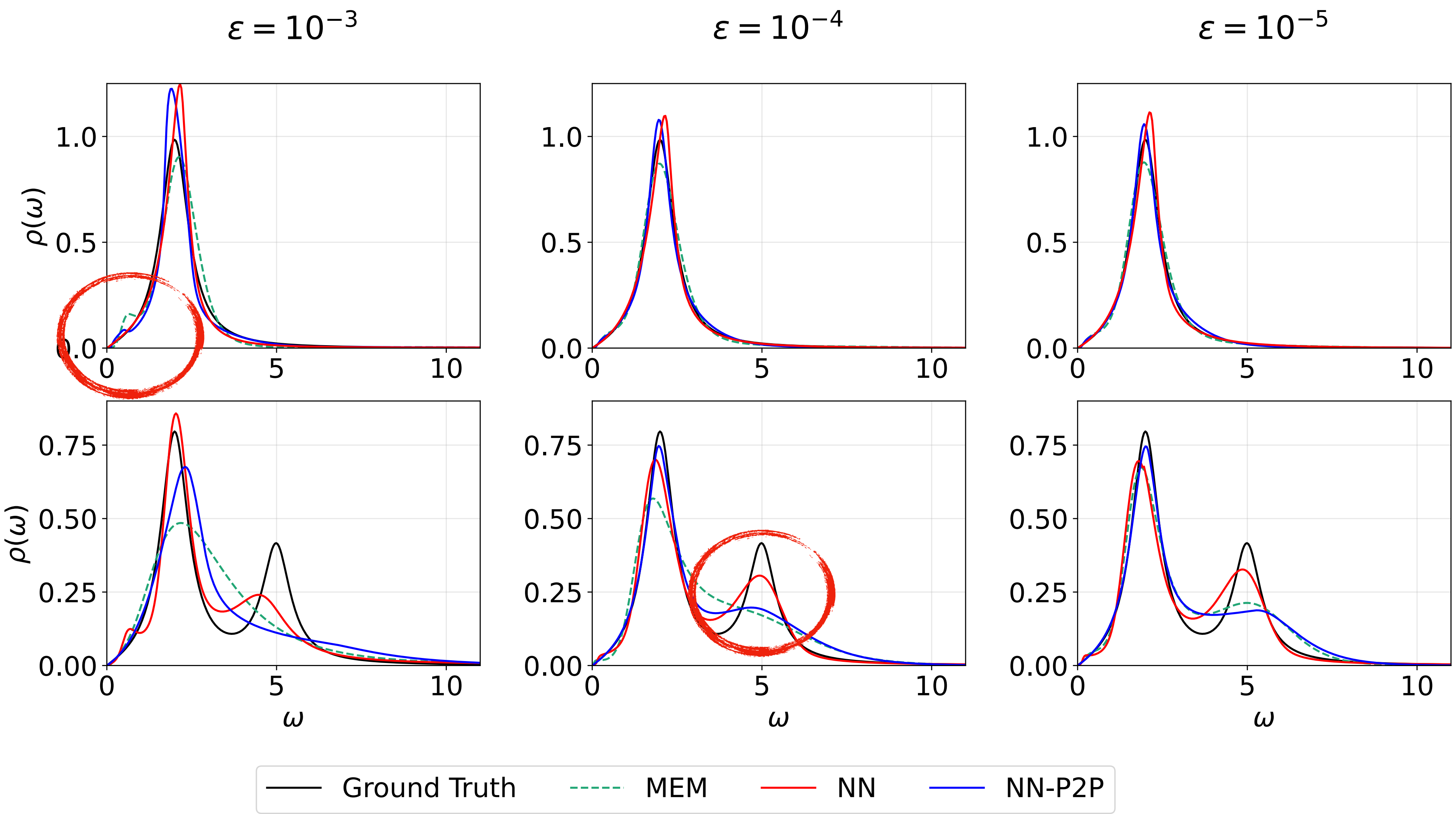
Positive-defined condition: Softplus

$log(1 + e^x)$

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

- **Maximizing Bayesian Posterior**

$$P(\rho\,|\,D, I) = \frac{P(D\,|\,\rho, I)P(\rho\,|\,I)}{P(D\,|\,I)}$$

  - Likelihood, $P(D\,|\,\rho, I) = e^{-\chi^2/2}$
  - Prior, $P(\rho\,|\,I) = e^{\mathcal{S}[\rho]}$

- **Minimization the loss function**

$$J \equiv \frac{\chi^2}{2} - \mathcal{S}[\rho]$$

  - **Chi-square term**

$$\chi^2 \equiv \sum_{i,j=1}^{N} C_{ij}^{-1}(D_i^{\text{obs}} - D_i)(D_j^{\text{obs}} - D_j)$$

the inverse covariance matrix, C^{-1}

$$\Delta_i \equiv -\frac{\delta\chi^2/2}{\delta D(k_i)} = \sum_j C_{ij}^{-1}(D_j^{\text{obs}} - D(k_j))$$

  - **"Entropy" term serves as a regulator**

Three typical "Entropy" terms

$$\mathcal{S}_{\text{TK}} = -\frac{\alpha}{2}\sum_{a=1}^{N_\omega}(\rho_a - \text{DM}_a)^2\Delta\omega\,,$$

$$\mathcal{S}_{\text{MEM}} = \alpha\sum_{a=1}^{N_\omega}\left(\rho_a - \text{DM}_a - \rho_a\ln\frac{\rho_a}{\text{DM}_a}\right)\Delta\omega\,,$$

$$\mathcal{S}_{\text{BR}} = \alpha\sum_{a=1}^{N_\omega}\left(1 - \frac{\rho_a}{\text{DM}_a} + \ln\frac{\rho_a}{\text{DM}_a}\right)\Delta\omega\,.$$

α, a hyper parameter; defaulted model (DM); $\Delta\omega$, step length

$$\frac{\delta J}{\delta\rho(\omega)} = 0$$

$$\rho_a^{\text{TK}} - \text{DM}_a = \frac{1}{\alpha}\sum_i \Delta_i^{\text{TK}}K_{ia}\,,$$

$$\ln\frac{\rho_a^{\text{MEM}}}{\text{DM}_a} = \frac{1}{\alpha}\sum_i \Delta_i^{\text{MEM}}K_{ia}\,,$$

$$\frac{1}{\text{DM}_a} - \frac{1}{\rho_a^{\text{BR}}} = \frac{1}{\alpha}\sum_i \Delta_i^{\text{BR}}K_{ia}\,.$$

the optimal solution exists

Regularization

Why NN helps

Comput. Phys. Commun. 282, 108547

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

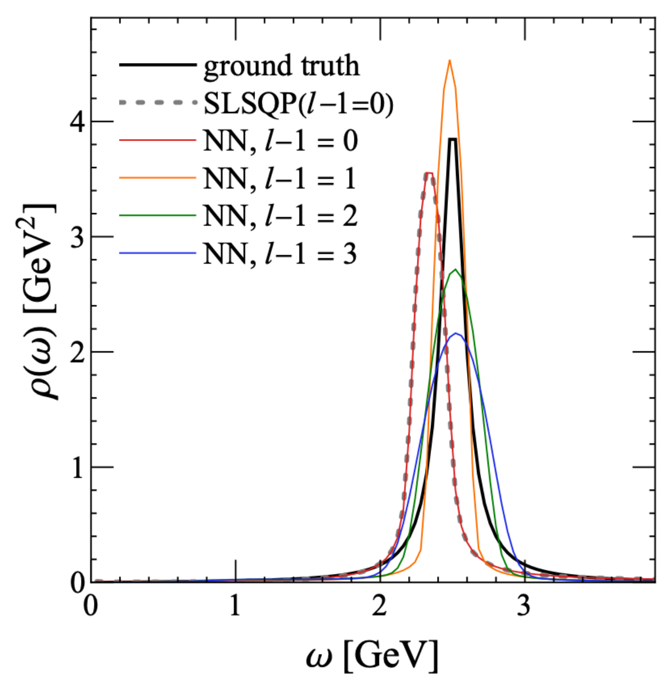
$$\frac{f_a/\sigma'(f_a)}{\left(\sum_b f_b^2\right)^{\frac{l-1}{l}}} = \frac{\mathrm{DM}_a}{\alpha}\sum_i \Delta_i K_{ia}$$

$$\frac{(1+e^{-f_a})f_a}{\left(\sum_b f_b^2\right)^{\frac{l-1}{l}}} = \frac{\mathrm{DM}_a}{\alpha}\sum_i \Delta_i K_{ia}$$ for *Softplus* activation function

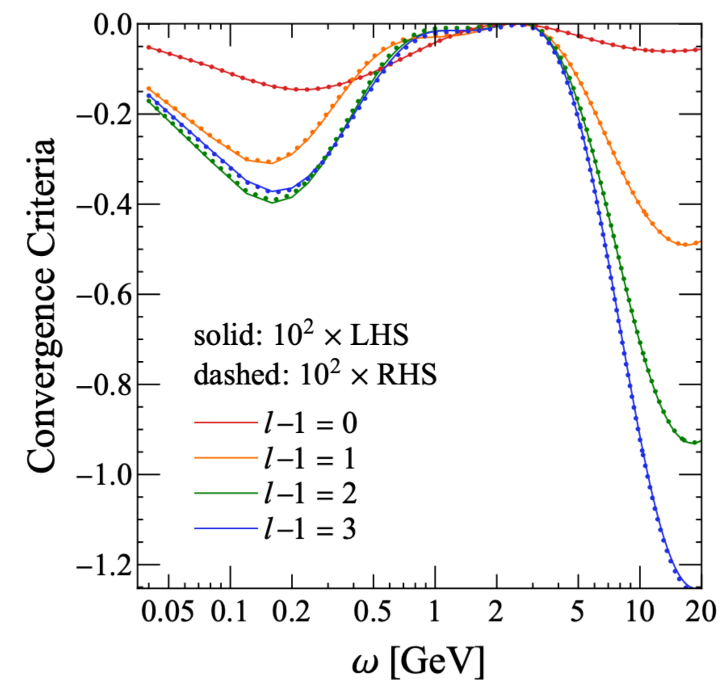
- **Neural Networks (e.g., NN representation)**

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

  - Output layer, $\rho_a = \mathrm{DM}_a\, \sigma^{(l)}(f_a^{(l)})$
  - 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,** $L_2 \equiv \alpha\Delta\omega \sum_{l,a,b}\left(W_{ab}^{(l)}\right)^2$



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



Regularization

Why NN helps

Comput. Phys. Commun. 282, 108547

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

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

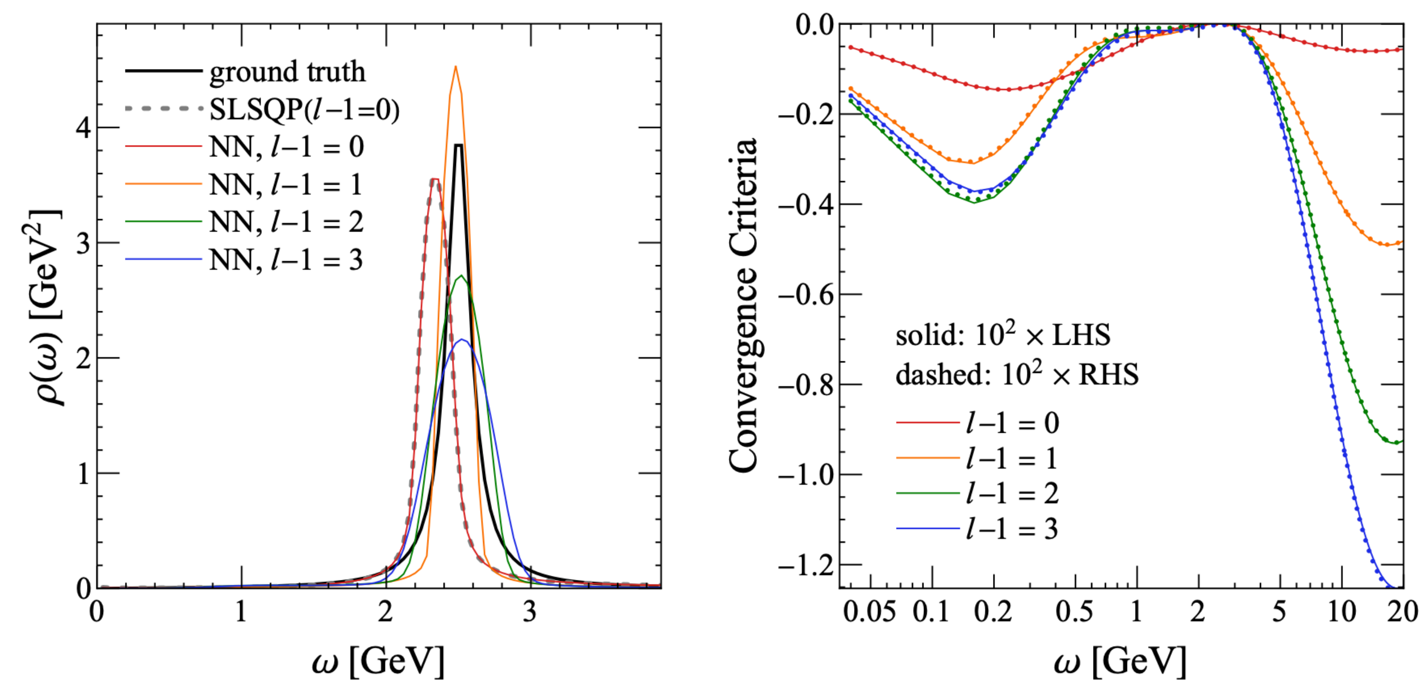
$$\rho_a \equiv \rho(\omega_a)$$

  - Output layer, $\rho_a = \text{DM}_a\, \sigma^{(l)}(f_a^{(l)})$
  - 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,** $L_2 \equiv \alpha \Delta\omega \sum_{l,a,b} \left(W_{ab}^{(l)}\right)^2$

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

non-local constraints from NN!

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



Call-back

Comput. Phys. Commun. 282, 108547

Tackle fundmental difficulties

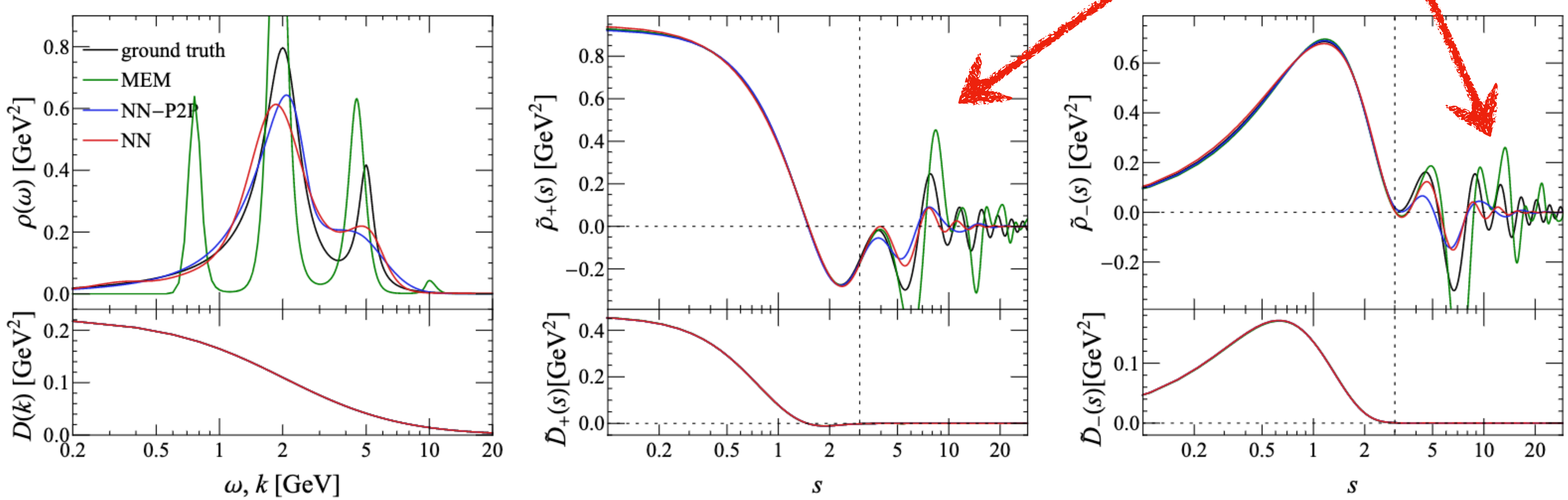
- The ill-posedness of the spectral reconstruction **fundamentally exists even for <u>continuous correlation functions</u>**;
- It's caused by the **numerical inaccuracy** of the correlation measurements (<u>induced high degeneracy in solution space</u>).

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

$$\lambda_{\pm,s} = \frac{1}{2\cosh(\pi s/2)}$$

Will contribute **"null-mode" (zero-mode)**

Regulators suppress "null-modes"!

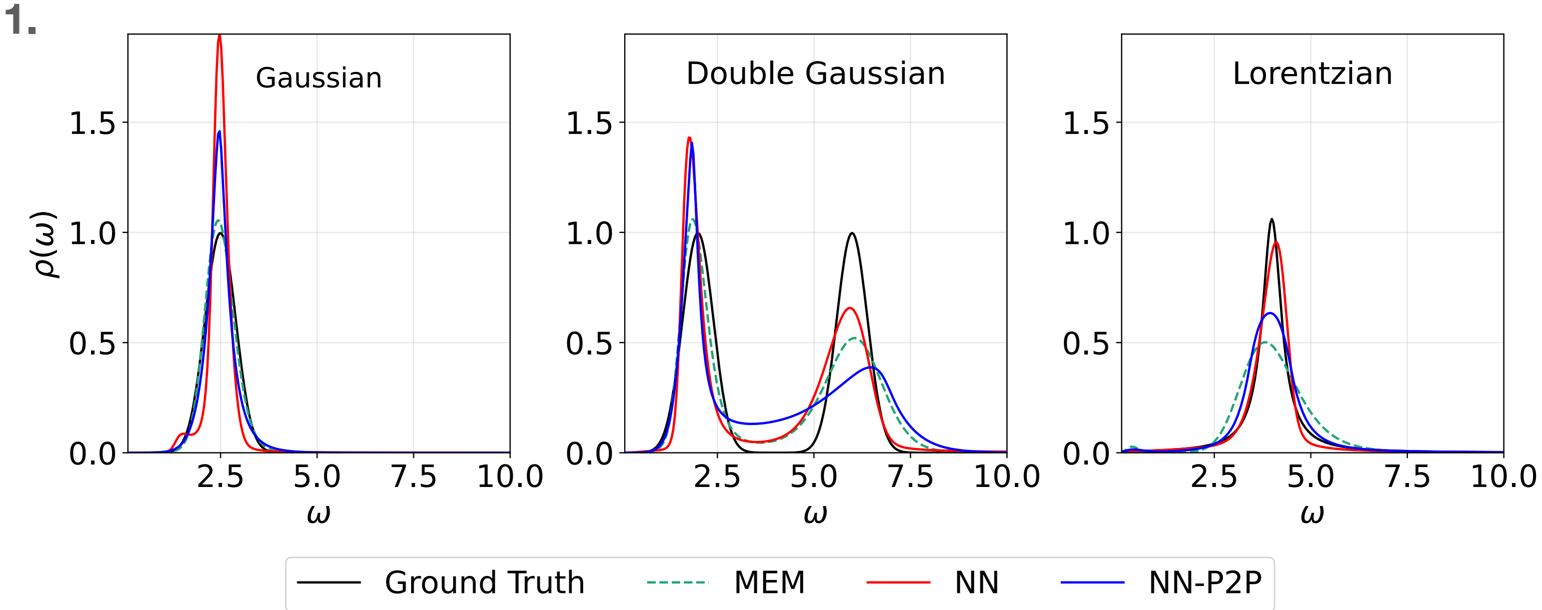


Results

Other cases

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

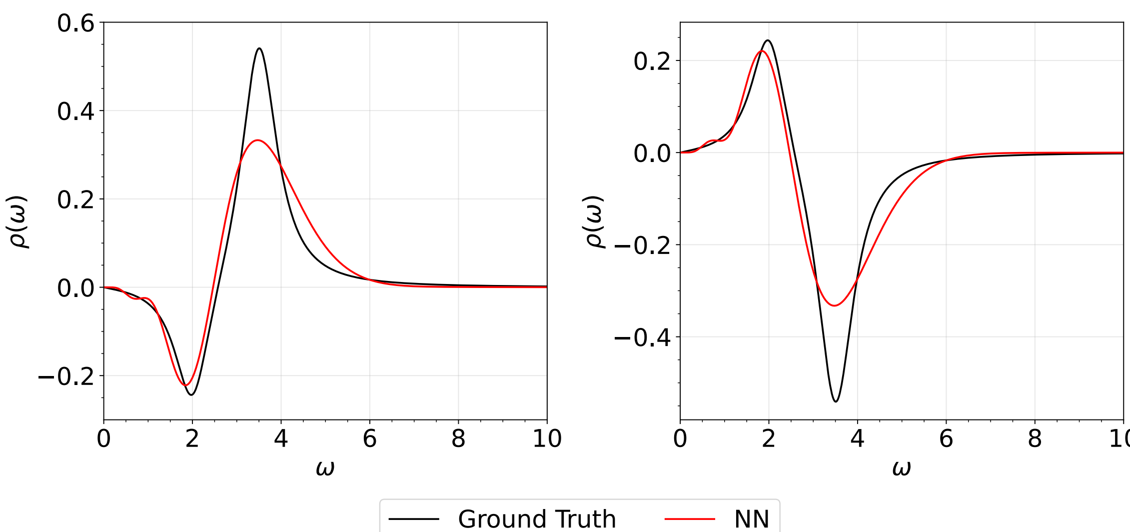
1.



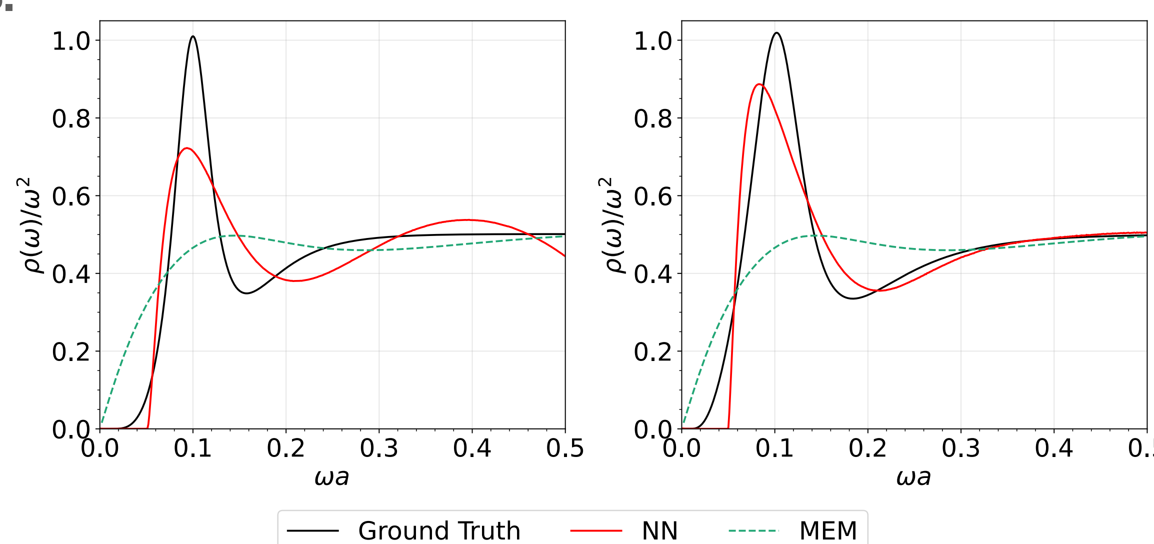
- **1. Single-peak functions**
- **2. Non-positive-definite SPs**
- 3. Lattice QCD mock data

2.

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



3.

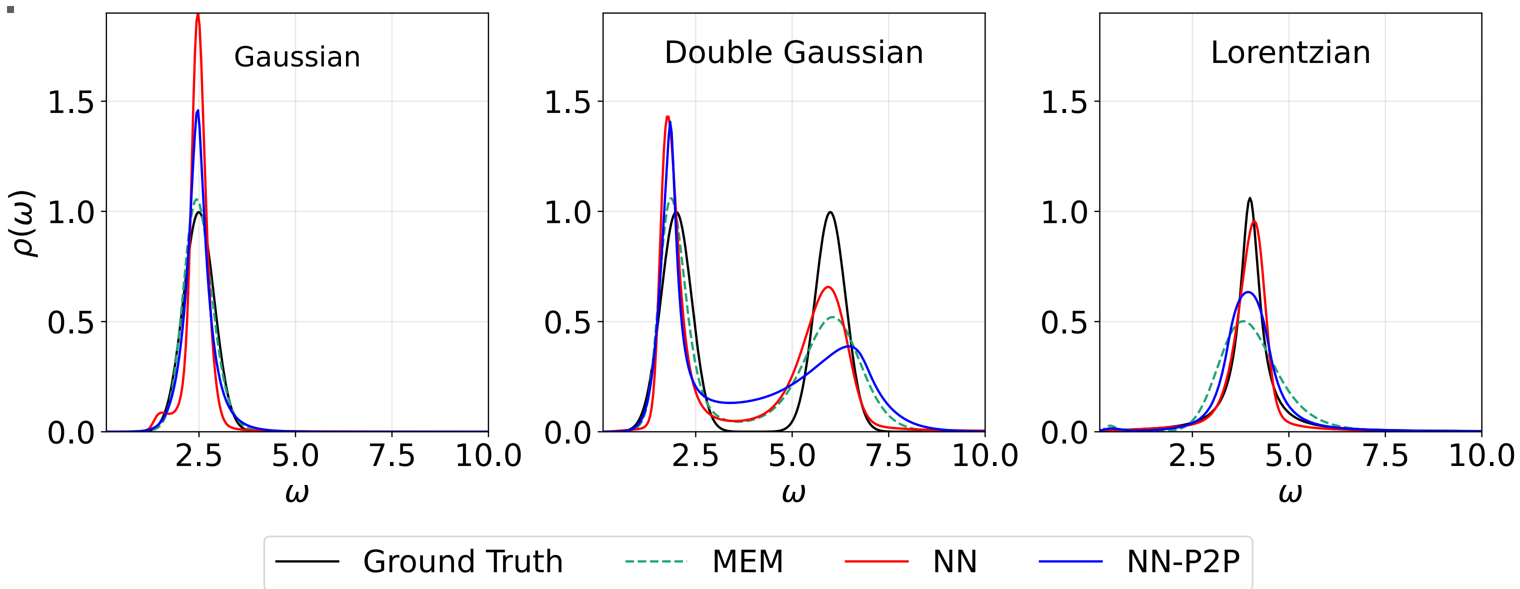


Results

Other cases

[Phys. Rev. D 106, L051502]

1.



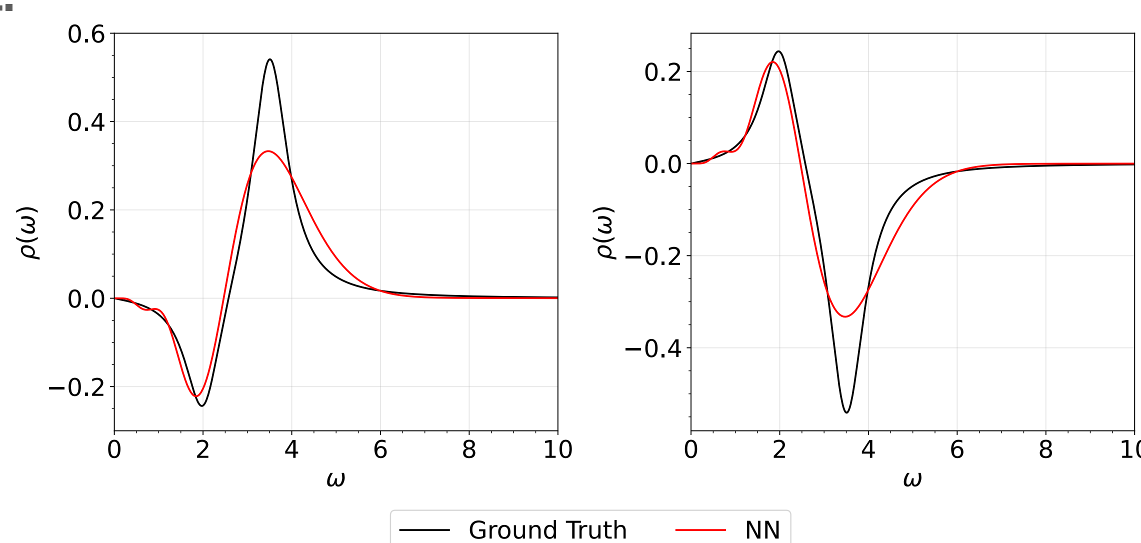
3. Lattice QCD mock data

(details see arXiv:2110.13521)

$$G(\tau, T) = \int_0^\infty \frac{d\omega}{2\pi} K(\omega, \tau, T)\rho(\omega, T)$$

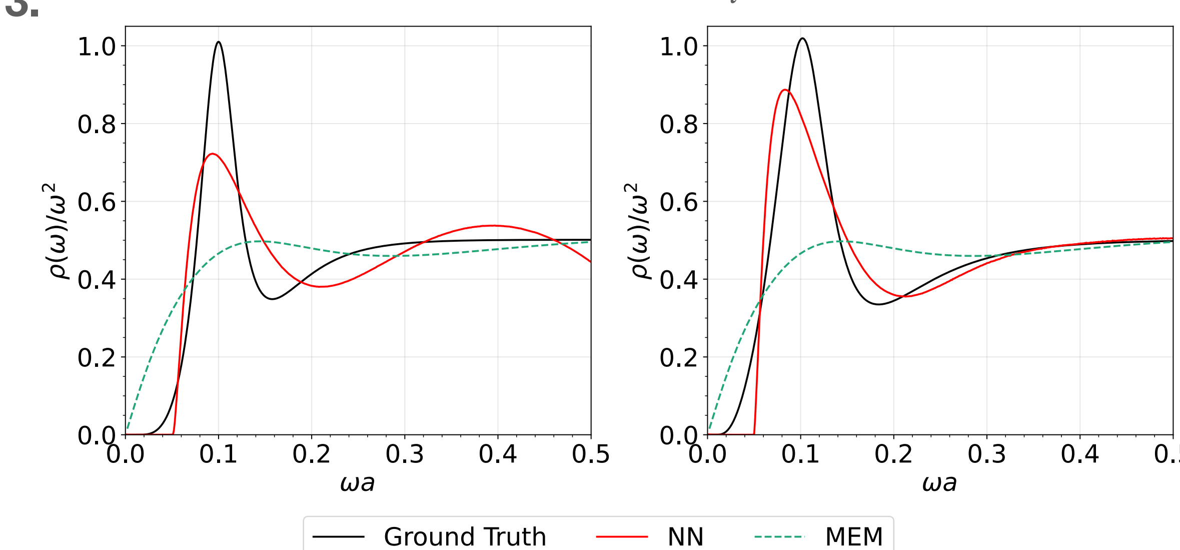
$$K(\omega, \tau, T) = \frac{\cosh \omega(\tau - \frac{1}{2T})}{\sinh \frac{\omega}{2T}}$$

2.



3.

noise level $\epsilon = 10^{-5}$ with $N_\tau = 48$ points

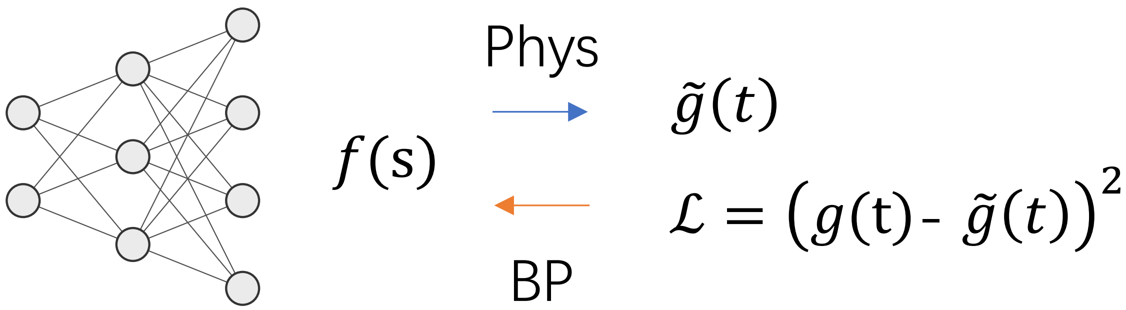


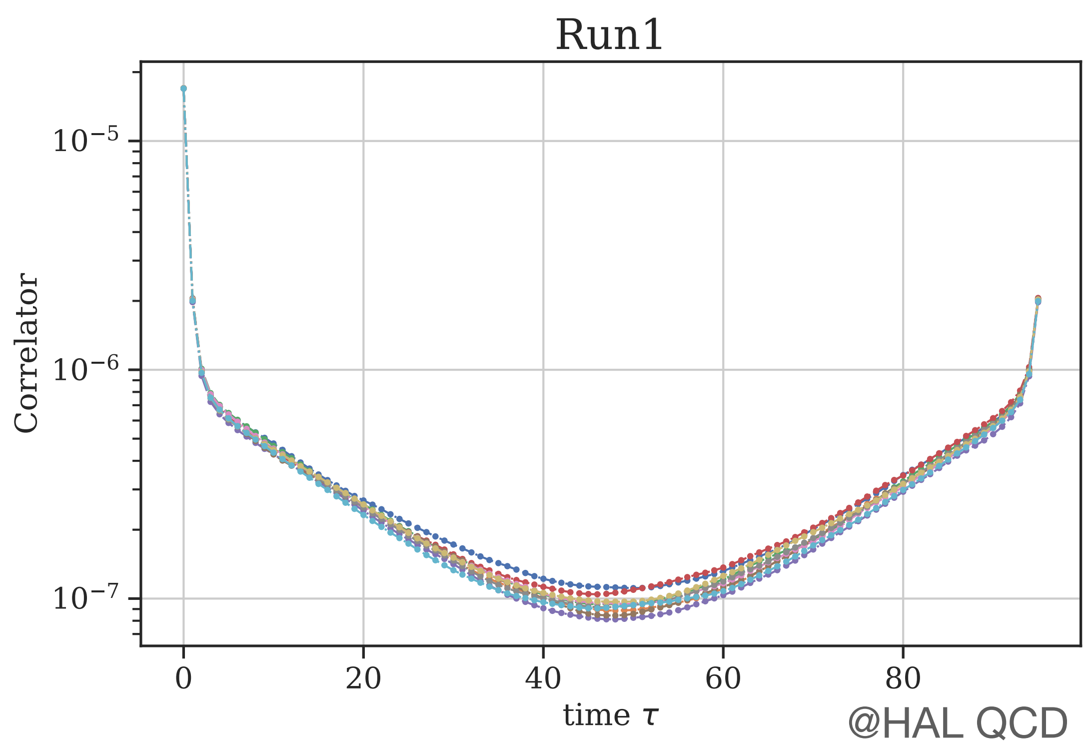
$C_{\text{res}} = 2.0, C_{\text{cont}} = 2.1, M_{\text{res}} = 0.1, M_{\text{cont}} = 0.05$ $\Gamma = 0.06$(left), $\Gamma = 0.09$(right)

Summary I

- **Inverse Problems**
  - Neural network representations
  - Auto-differentiation framework
  - Gradient-based optimization
- **Future works**
  - Open codes [github1, github2]
  - **Easy-to-use Python packages**
  - **Real Lattice QCD data !**
    - HAL QCD collaboration
    - Nucleon, Lambda, Sigma, Xi, Omega

Unbiased! Flexible!

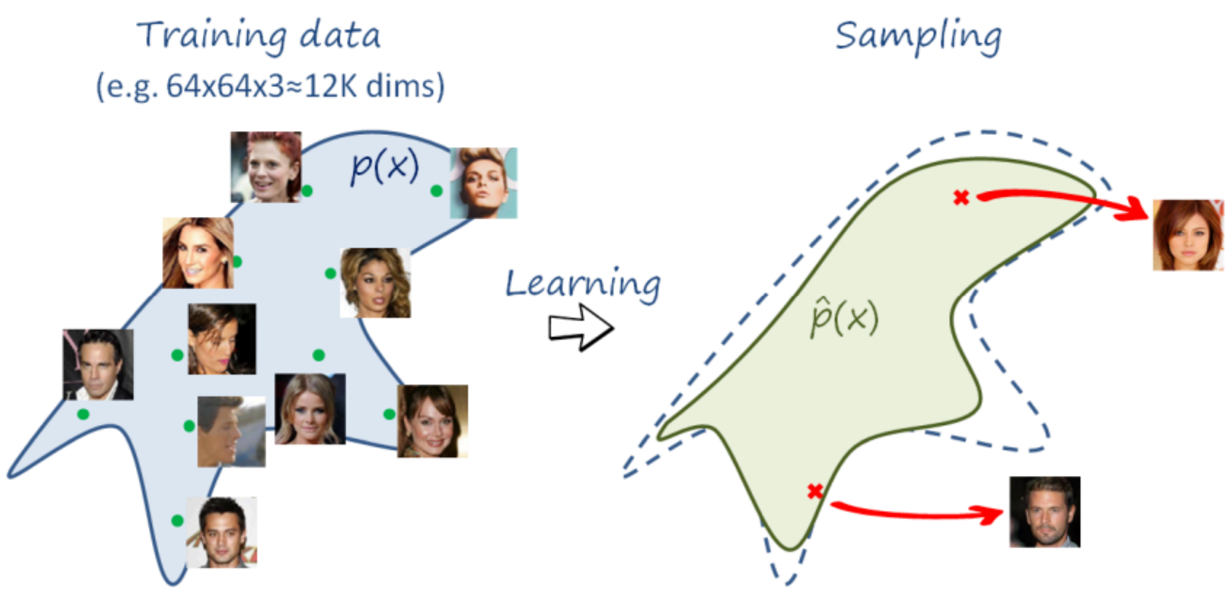


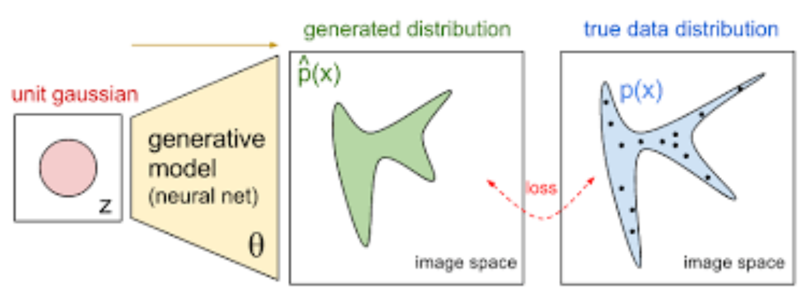


Generative models

Generative Models

for Lattice calculations



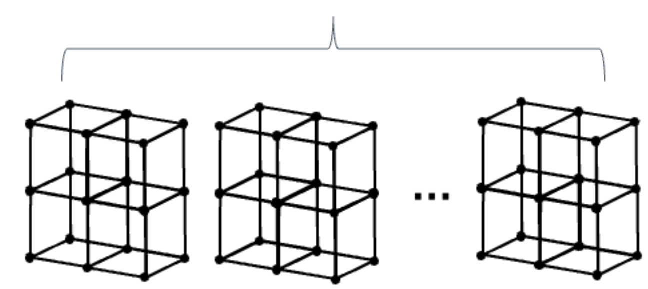


@blogs of OpenAI

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

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

Ensemble



Generative models :
approaching **underlying distributions in data**

Lattice calculations:
approaching **physical distributions, sampling**

Markov-Chain MC

Revisiting

- Generate configurations ϕ_i **independently** from,

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

- Metropolis Method

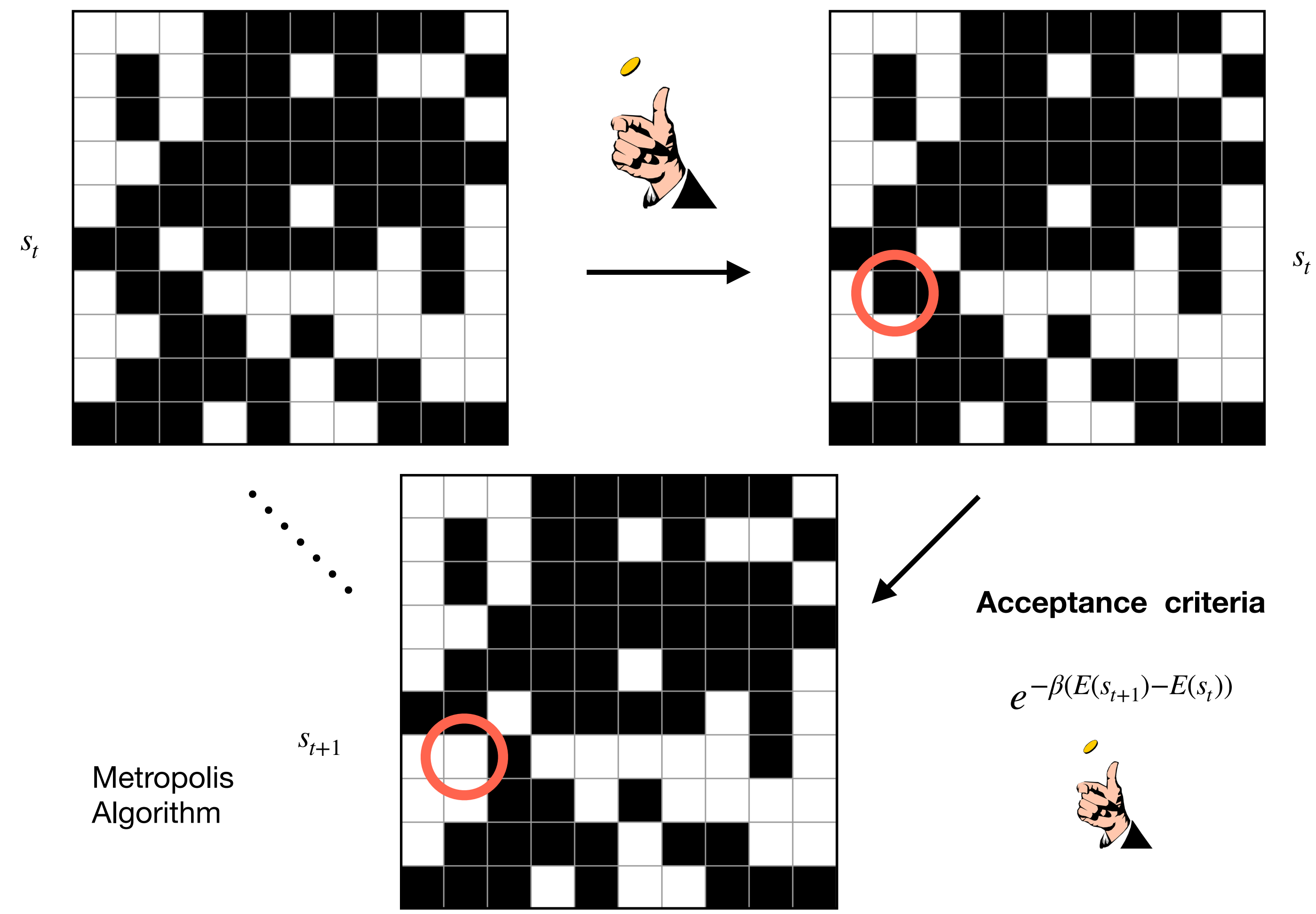
- Shortcomings

  - 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)



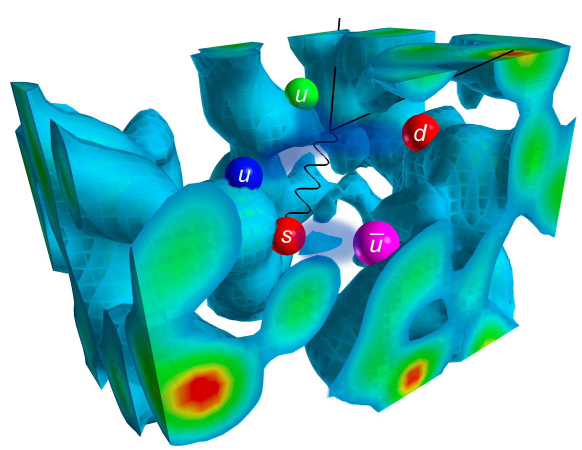
Generative Models

for Lattice calculations

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

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

approaching **physical distribution, sampling** with Generative Moodels



Lattice QCD © Derek Leinweber/CSSM/University of Adelaide

- **Implicit Likelihood Estimation**
  - VAEs

    D. Giataganas, et al., New J. Phys. 24, 043040 (2022).
  - GANs

    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 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, et al., 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).
    M. S. Albergo, et al., Phys. Rev. D 104, 114507 (2021).
    L. Del Debbio, et al., Phys. Rev. D 104, 094507 (2021).
    M. Gerdes, et al., arXiv:2207.00283.
    M. S. Albergo, et al., Phys. Rev. D 106, 014514 (2022).
    R. Abbott et al., arXiv:2211.07541.
    M. Caselle, et al., J. High Energ. Phys. 2022, 15 (2022).
    R. Abbott et al., Phys. Rev. D 106, 074506 (2022).
    S. Bacchio, et al., arXiv:2212.08469.
    A. Singha, et al., Phys. Rev. D 107, 014512 (2023).
    S. Chen, et al., Phys. Rev. D 107, 056001(2023).
    …

White paper
D. Boyda et al., arXiv:2202.05838.
Hands-on notebook
M. S. Albergo et al., arXiv:2101.08176.

CANs

Continuous Autoregressive Networks

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

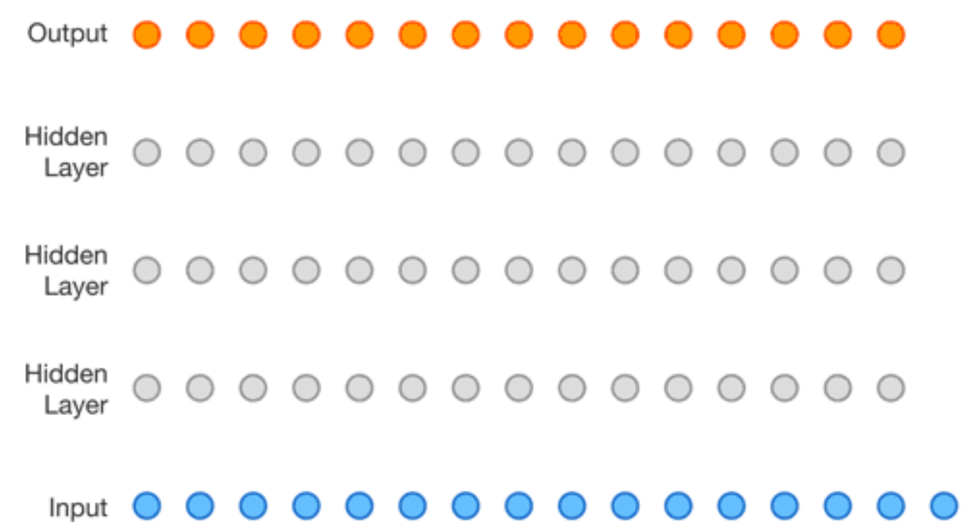
- Kullback-Leibler (KL) divergence

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

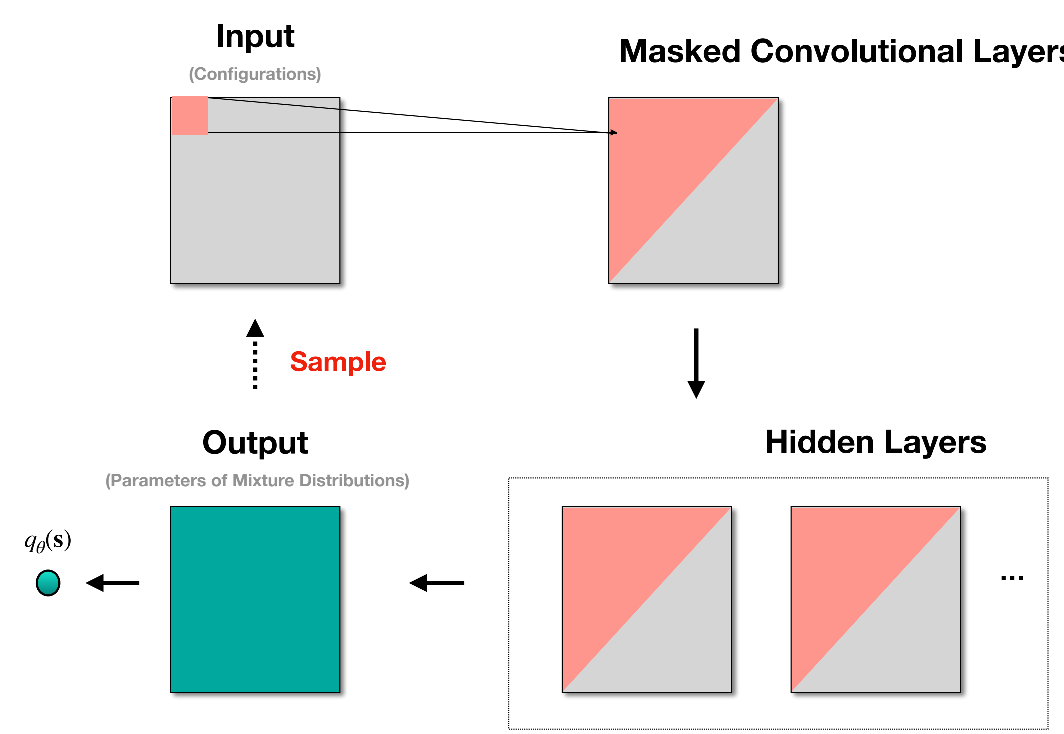
- Optimization
  - Loss function: variational free energy

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

  - Neural network parameters θ



Autoregressive properties@DeepMind Blog



Chinese Phys. Lett. 39, 120502 (2022)

CANs

for 2D XY model

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

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

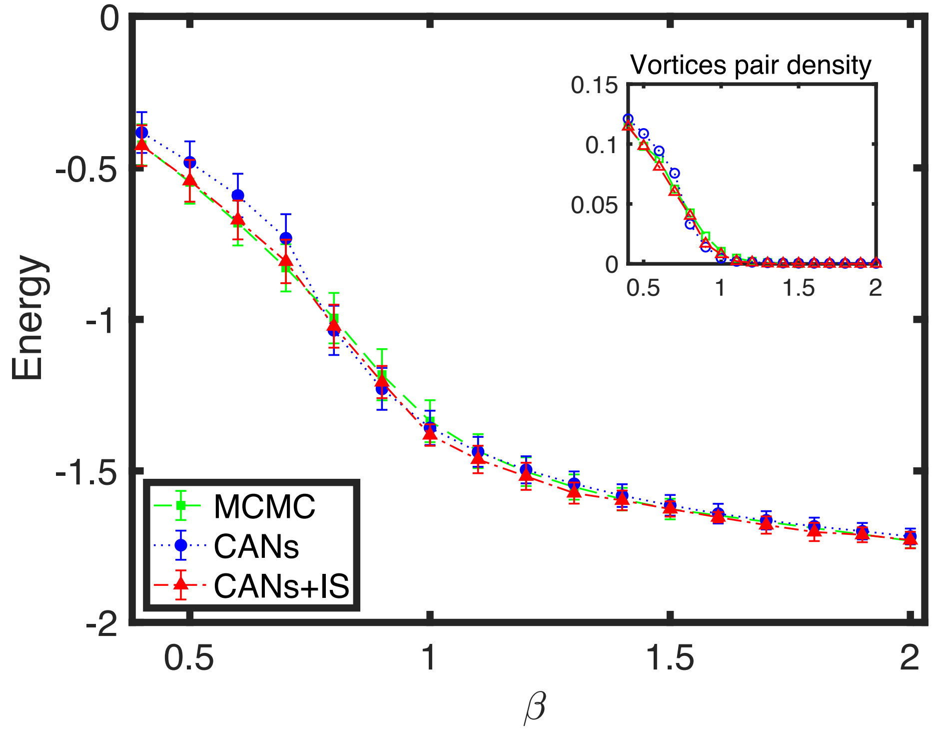
- Neural network: **PixelCNN**
- Prior distribution: **Beta distribution**

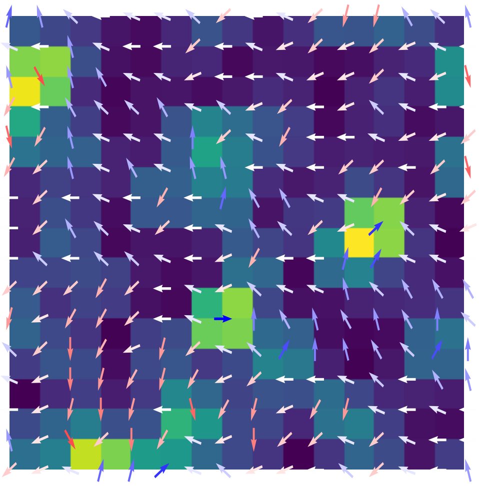
$$q_\theta(s) = \prod_{i=1}^{N} f(s_i \,|\, s_1, \ldots, s_{i-1})$$

$$f(s_i; a_i, b_i) = \frac{\Gamma(a_i + b_i)}{\Gamma(a_i)\Gamma(b_i)} s_i^{a_i - 1}(1 - s_i)^{b_i - 1}$$

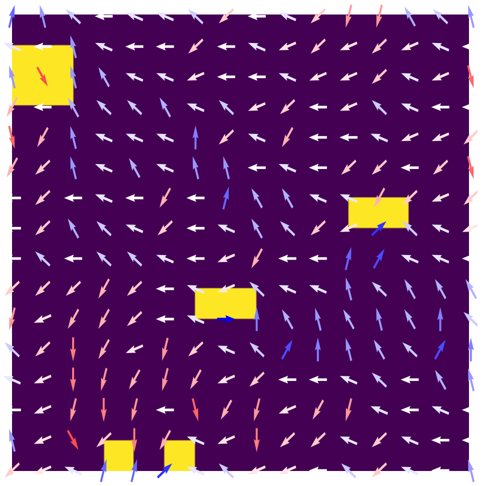
- Kosterlitz-Thouless(KT) transition
  - Vortices

Chinese Phys. Lett. 39, 120502 (2022)





Probability Distributions from CANs



Vortices

CANs

for 2D XY model

Chinese Phys. Lett. 39, 120502 (2022)

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

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

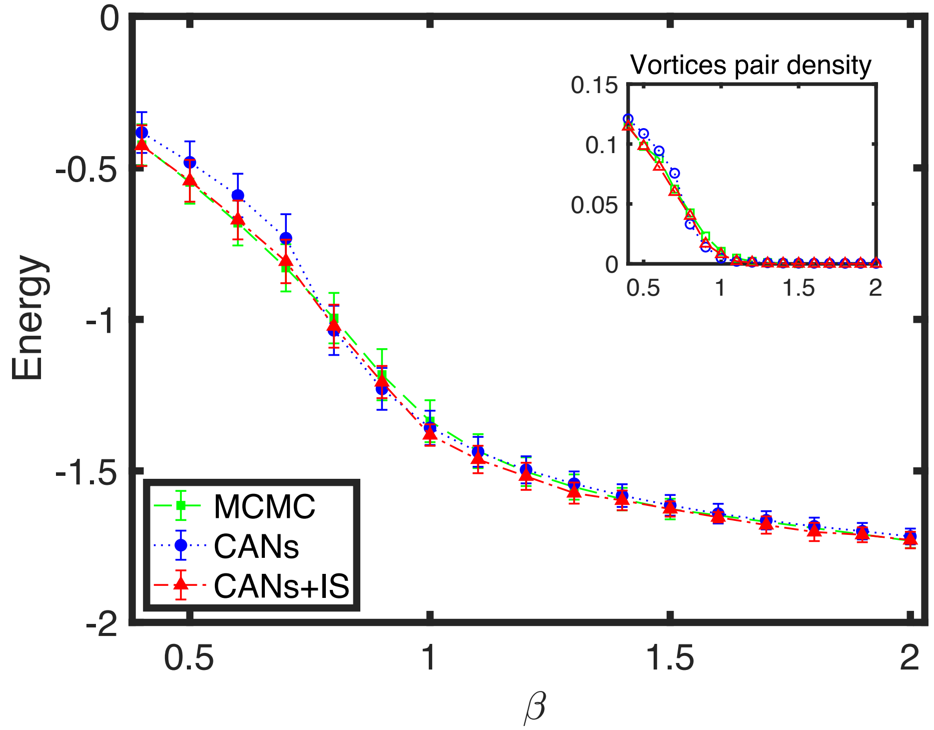
- Neural network: **PixelCNN**

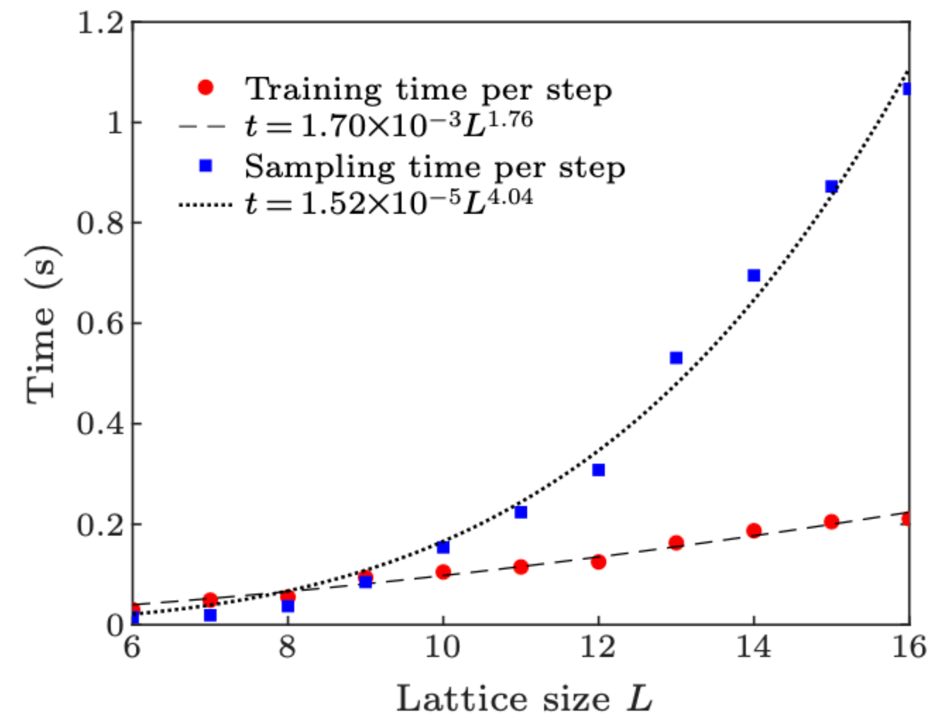
- Prior distribution: **Beta distribution**

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

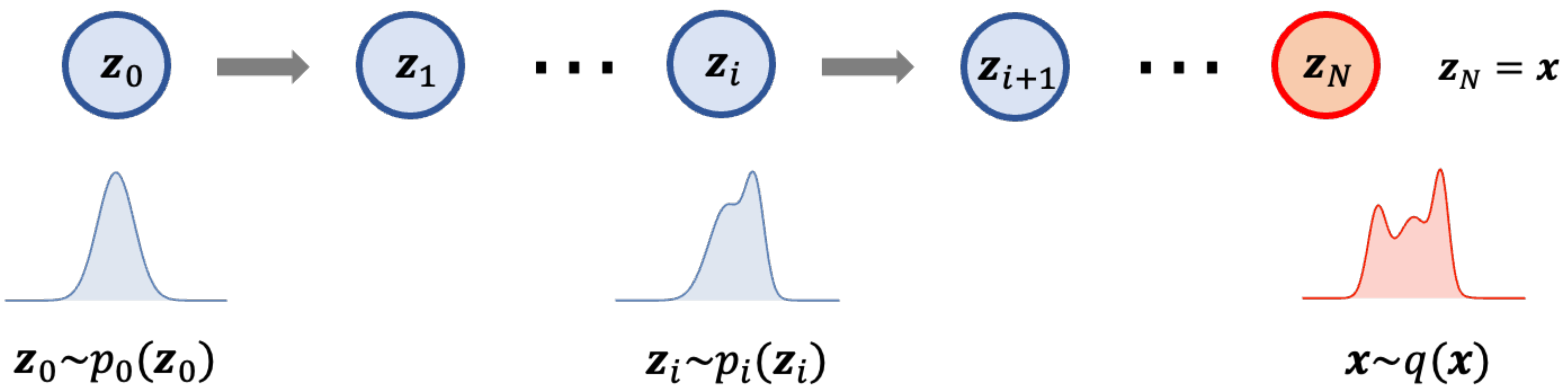
$$f(s_i; a_i, b_i) = \frac{\Gamma(a_i + b_i)}{\Gamma(a_i)\Gamma(b_i)} s_i^{a_i - 1}(1 - s_i)^{b_i - 1}$$

- Kosterlitz-Thouless(KT) transition
    - Vortices





Flow-Based Model



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

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

- **Jacobian**

  Invertible and tractable

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

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

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

Flow-Based Models

Flow models for lattice QCD

- **MIT-led program** to develop flow model architectures for applications across lattice QCD

✔ **First flow architectures for lattice field theory** (scalar field theory) [Albergo et al., 1904.12072]

✔ Gauge field theories
- Flow transformations on compact, connected manifolds [Rezende et al., 2002.02428]
- Gauge-equivariant architectures: Abelian field theories [Kanwar et al., 2003.06413, 2101.08176]
- Gauge-equivariant architectures: non-Abelian field theories [Boyda et al., 2008.05456]

✔ Theories with fermions
- Architectures for theories with fermions [Albergo et al., 2106.05934]
- Combining architectures for gauge fields and fermions [Albergo et al., 2202.11712]
- Techniques to incorporate pseudofermions [Abbott et al., 2207.08945]

✔ Initial application to QCD in 4D
[This talk+upcoming manuscripts on scaling and 4D]

Architectures for QCD at scale [ongoing; Aurora Early Science Project]

[see also tutorial notebook 2101.08176, work on multimodal distributions 2107.00734]

9 Phiala Shanahan, MIT

HPC and AI on Aurora

HPC and AI on Aurora

Save Time!

Improving Normalizing Flows to Sample from Boltzmann Distributions
Vincent Stimper
27 February 2023 15:10
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Complex normalizing flows and subtractions for sign problems
Yukari Yamauchi
27 February 2023 16:50
Zoom
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Stochastic normalizing flows for lattice field theory
Elia Cellini
27 February 2023 11:40
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Conditional Normalizing Flow model for sampling in the Critical region of Lattice Field Theory
Ankur Singha
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Fourier-Flow model generating Feynman paths
Lingxiao Wang
27 February 2023 11:15
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Learning trivializing flows
David Albandea
27 February 2023 10:20
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

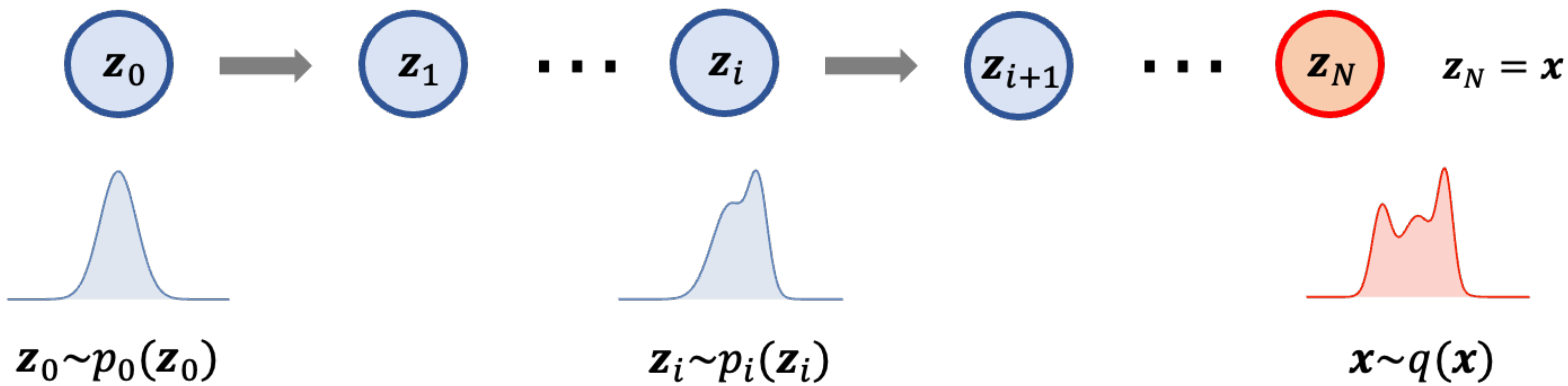
Aspects of scaling and scalability for flow-based samplers
Daniel Hackett
27 February 2023 09:55
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Simulation of the 2D Schwinger Model via machine-learned flows in Global Correction steps
Jacob Finkenrath
28 February 2023 11:15
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

Learning Trivializing Gradient Flows
Simone Bacchio
01 March 2023 09:00
Institute for Advanced Study of the Technische Universität München
Machine Learning approaches in Lattice QCD - An interdisciplinary exchange

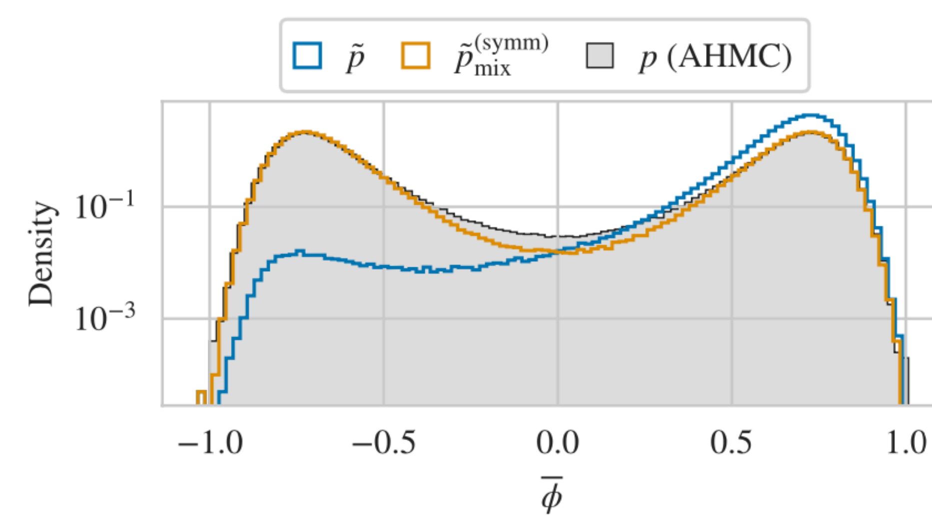
Flow-Based Models

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".

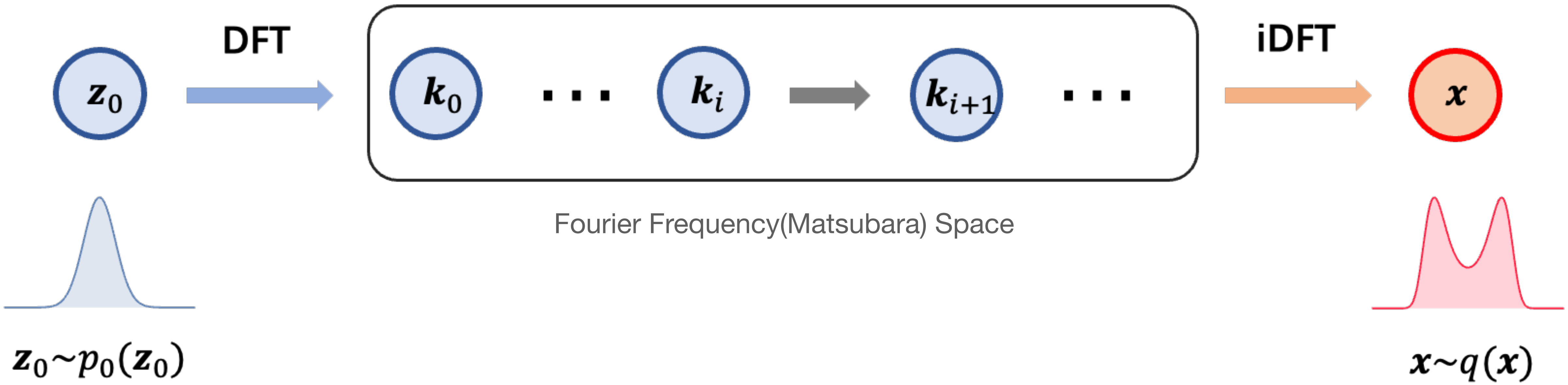


comprehensive discussions in arXiv:2107.00734, arXiv:2302.14082.

Fourier-Flow Model

Phys. Rev. D 107, 056001

Mode-collapse

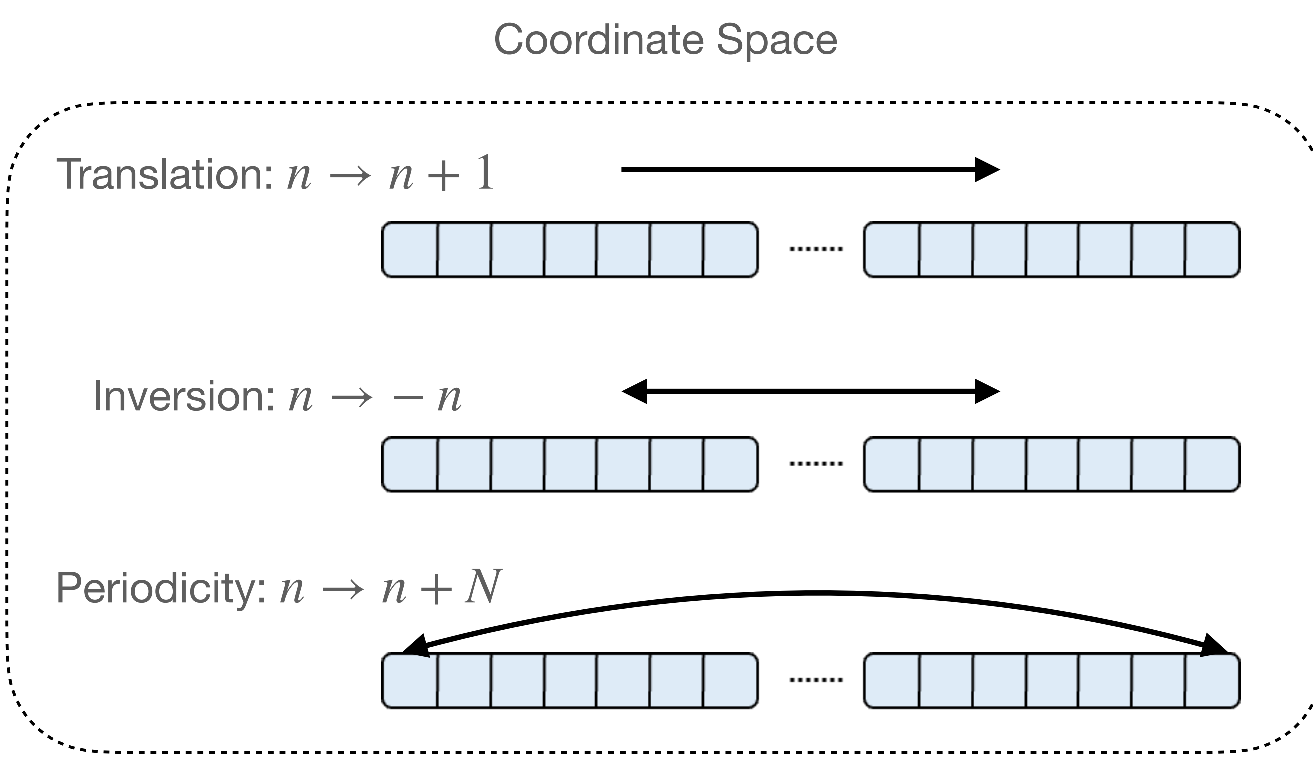


- More priors, more stable when training neural networks
- Discrete Fourier transformation (DFT)

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

- Inverse DFT (iDFT)

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



Fourier-Flow Model

Phys. Rev. D 107, 056001

Real NVP

- **Rreal-valued non-volume preserving (Real NVP)**

  build the flow with **affine transformations**

$$\begin{cases} X^i_{1:k} = X^{i-1}_{1:k} \\ X^i_{k+1:N} = X^{i-1}_{k+1:N} \odot e^{s^i_\theta(X^{i-1}_{1:k})} + t^i_\theta(X^{i-1}_{1:k}), \end{cases}$$

- **Jacobian**

$$\det J^i_T = \Pi_j^{N-k} e^{s^i_\theta(X_{1:k})_j}$$

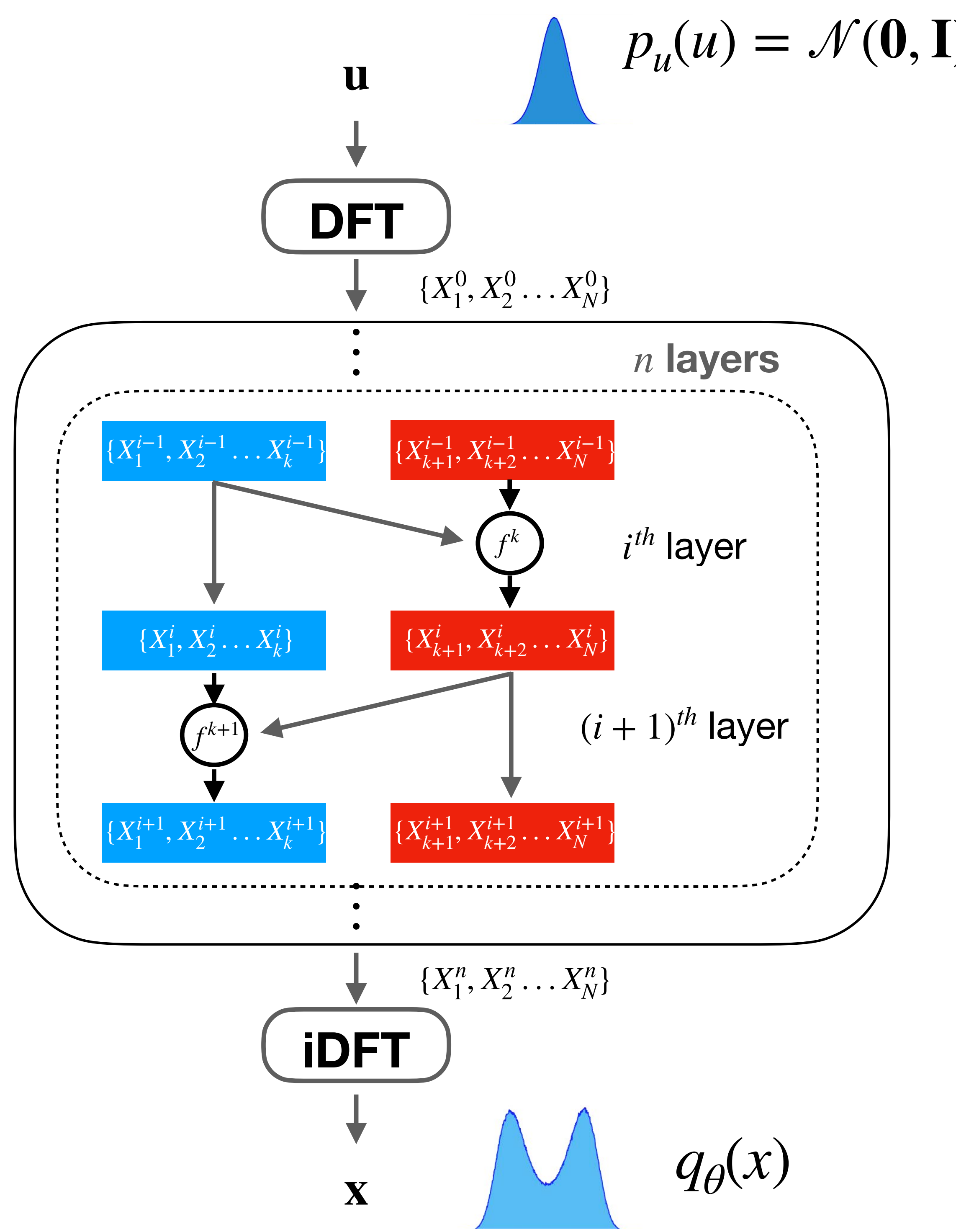
- Neural network parameterization

$$s_\theta : \mathbb{R}^k \to \mathbb{R}^{N-k},\ t_\theta : \mathbb{R}^k \to \mathbb{R}^{N-k}$$

- Loss function

$$L(\theta) = \mathbb{E}_{x\sim q_\theta(x)}[S_E(x) + \ln q_\theta(x)] \cancel{+\ln Z}$$

$$q_\theta(x) \equiv p_{\mathbf{x}}(\mathbf{x};\theta) = p_{\mathbf{u}}(\mathbf{u})\,|\det J_T(\mathbf{u})|^{-1}$$



Fourier-Flow Model

Path integral

- **0 + 1 dimensional QFT**

  Quantum state

$$\psi(x,t) = \int \mathscr{D}x_0(t) K(x,t;x_0,t_0)\psi(x_0,t_0)$$

  Feynman propagator

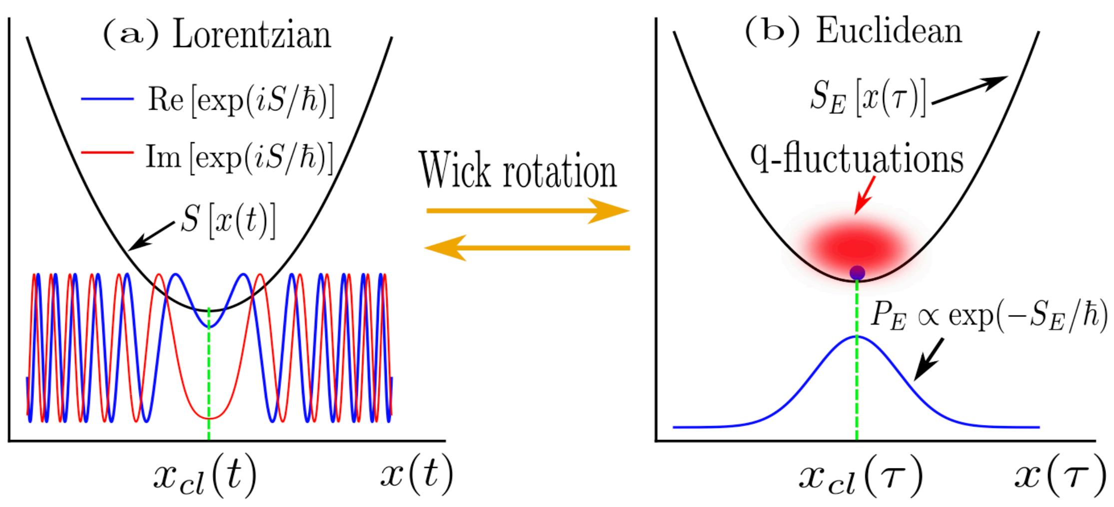
$$K(x,t;x_0,t_0) \propto \sum_{[x(t)]} e^{iS[x(t)]/\hbar}$$

- **Eculidean space**

$$S_E[x(\tau)] = \int_0^{\mathscr{T}} d\tau \{T[x(\tau)] + V[x(\tau)]\}$$

$$K_E(x,T;x_0,\tau_0) \propto \sum_{[x(\tau)]} e^{-S_E[x(\tau)]/\hbar}$$

Phys. Rev. B 105, 214205 (2022).



Discretization

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

$\hbar = 1$, time interval $\beta = 10$, lattice size $N = 100$,
spacing $a = \beta/N$, periodic boundary $x_0 = x_N$

Phys. Rev. D 107, 056001

Fourier-Flow Model

Energy levels

- **Discrete Eculidean action**

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

- **Ground state**

$$E_0 = \langle T \rangle + \langle V \rangle$$

Virial theorem: $2\langle T \rangle = \sum_n \left\langle x_n \frac{dV}{dx_n} \right\rangle$

- **Excited states**

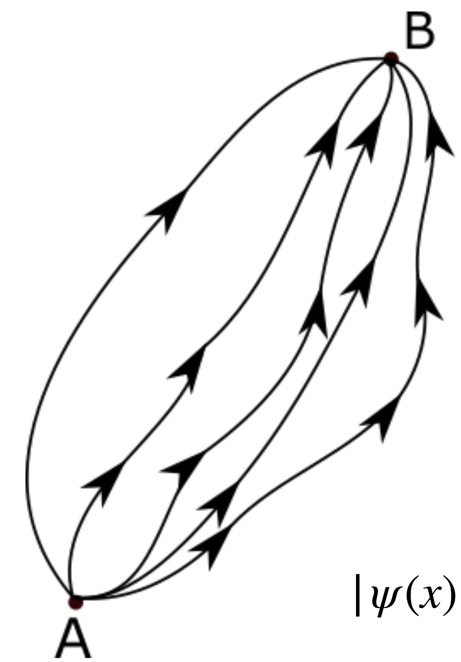
$$E_1 - E_0 = -\lim_{\tau\to\infty} \frac{d\log G_2(\tau)}{d\tau}$$

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

Correlators

$$G_2 = \lim_{\beta\to\infty} \left(\langle x(\tau)x(0)\rangle - \langle x(\tau)\rangle\langle x(0)\rangle\right)$$

$$G_4 = \lim_{\beta\to\infty} \left(\langle x(\tau)^2 x(0)^2\rangle - \langle x(\tau)^2\rangle\langle x(0)^2\rangle\right)$$



$|\psi(x)|^2 \propto K(A=B)$

Wave Function

$$|\psi(x)|^2 \propto K_E(x,\beta;x,\tau_0)$$

$$K_E(x,\beta;x_0,\tau_0) = \sum_{n=0} e^{-\beta E_n}\psi_n(x_0)\psi_n^*(x)$$

$\beta \to \infty$, ground state wave-function

Spectral Representation

F-Flow Model

for harmonic oscillator

Phys. Rev. D 107, 056001

- **1-dimensional(1D) harmonic oscillator**

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

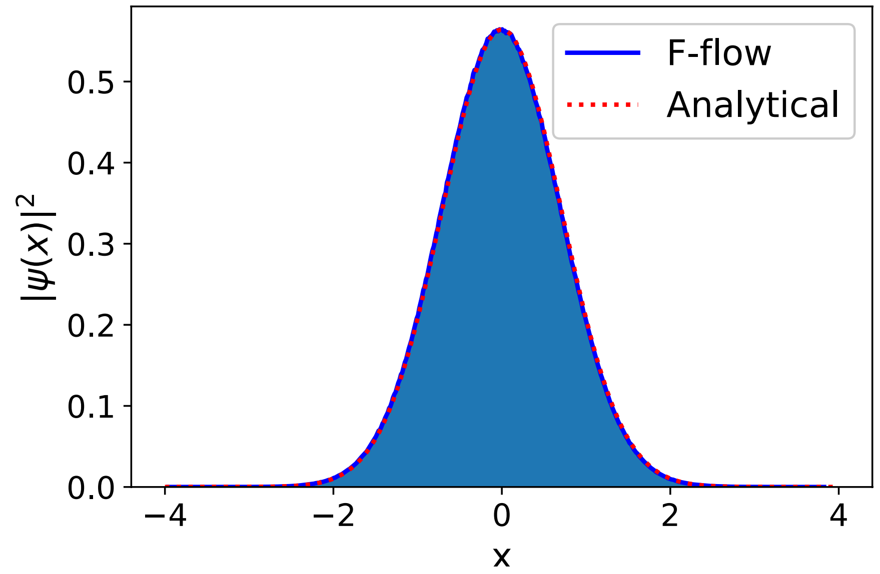
$$m = 1, \mu = 1$$

- **Analytical solutions**

$$E_n = (\frac{1}{2}+n)\mu$$

$$\psi_n(x) = \frac{1}{\sqrt{2^n\, n!}}\left(\frac{m\mu}{\pi\hbar}\right)^{1/4} e^{-\frac{m\mu x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\mu}{\hbar}}x\right)$$

Ground state



Energy	Analytical	F-flow(200k)	F-flow(400k)
E_0	0.5	0.4997($\pm$ 0.0001)	0.4997($\pm$ 0.0001)
E_1	1.5	1.5171($\pm$ 0.0011)	1.4999($\pm$ 0.0004)
E_2	2.5	2.532($\pm$ 0.037)	2.502($\pm$ 0.034)

F-Flow Model

for anharmonic oscillator

Phys. Rev. D 107, 056001

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

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

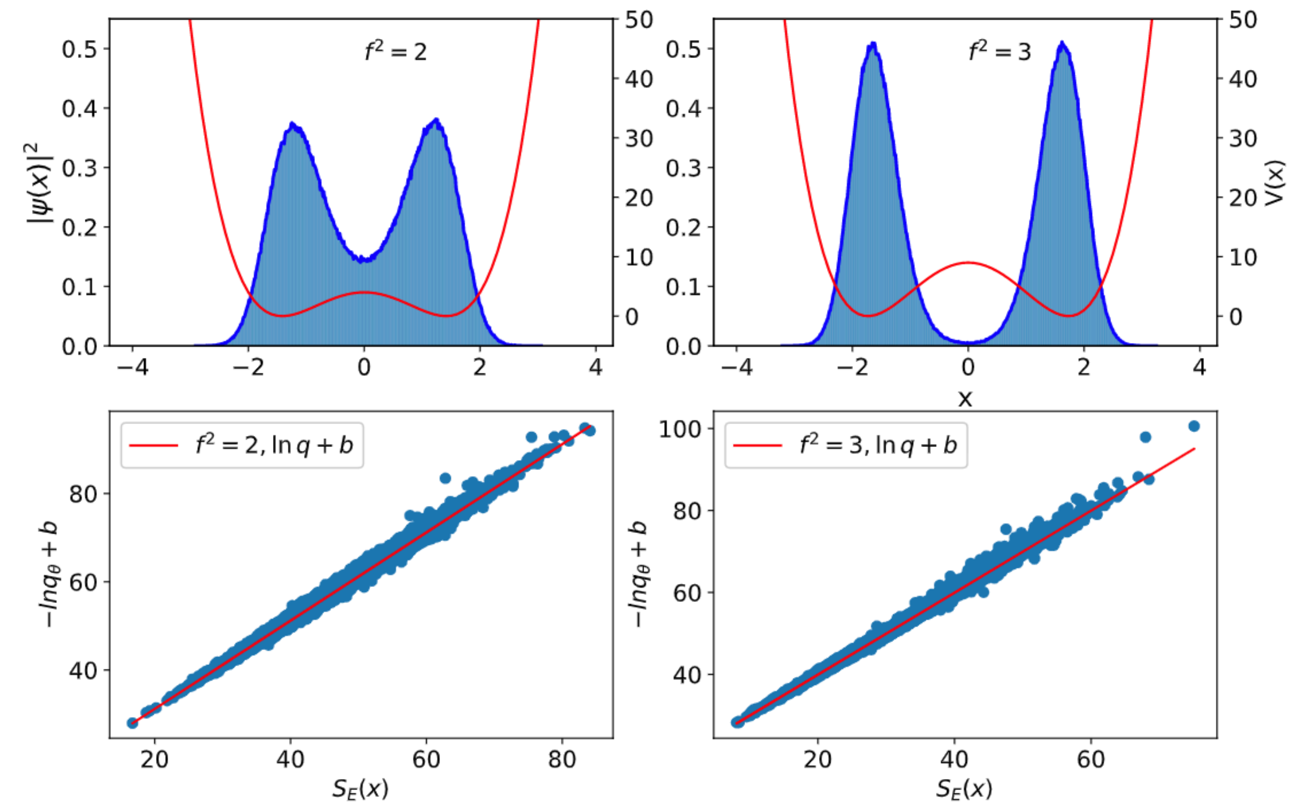
$$m = 0.5, \lambda = 1$$

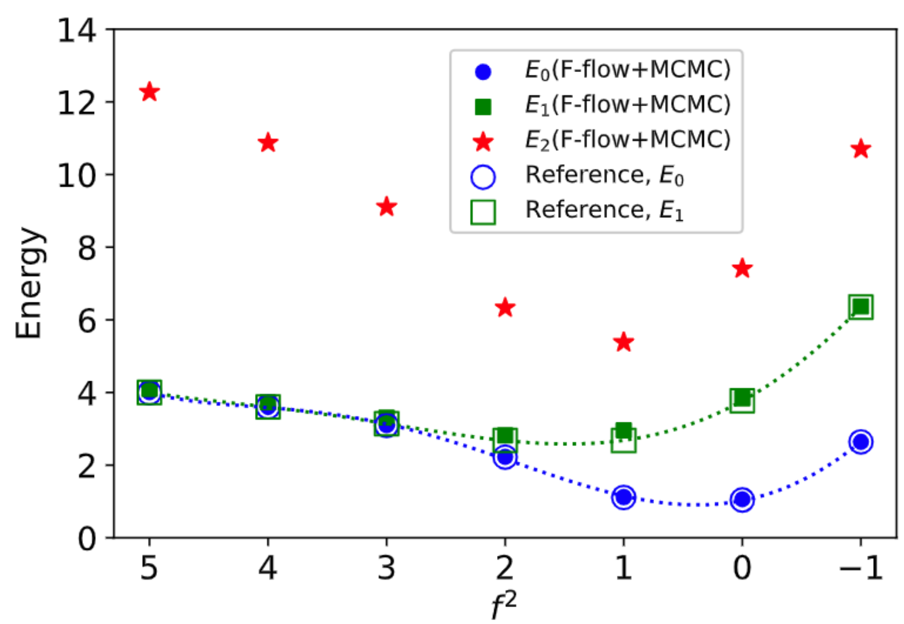
- **No analytical solution!**
  - MCMC
    [S. Mittal et al., Eur. J. Phys. 41, 055401 (2020)]
  - Moment methods
    [R. Blankenbecler et al., Phys. Rev. D 21, 1055 (1980)]
- Ground state

$$E_0 = 3\lambda\langle x^4\rangle - 4\lambda f^2\langle x^2\rangle + \lambda f^4$$

- 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

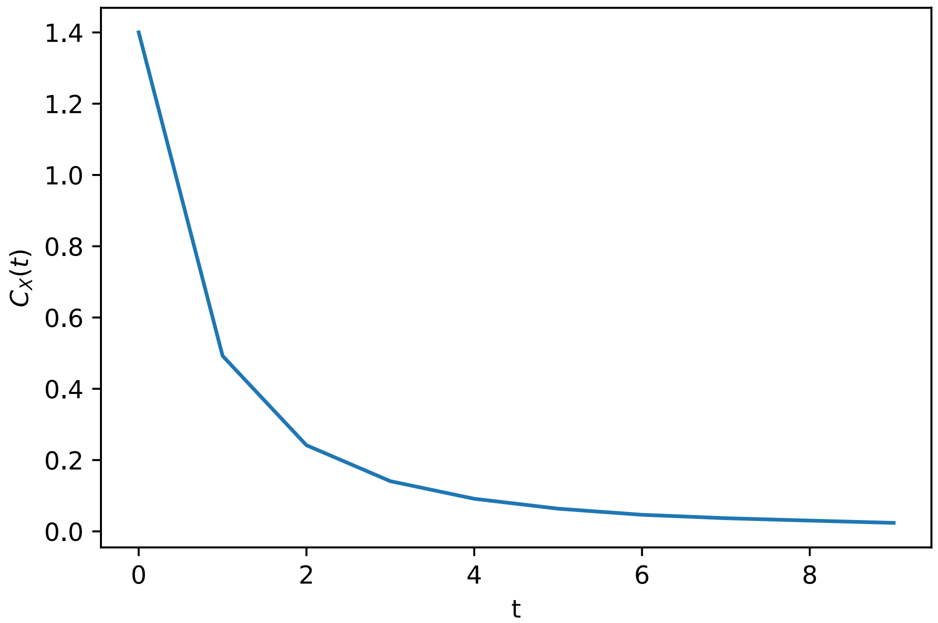
for anharmonic oscillator

Phys. Rev. D 107, 056001

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

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

- **Comparisons**
  - Autocorrelation function:
    $C_X(t) = < X_t X_{i+t} > - < X_t > < X_{i+t} >$
    i is position and t is discrete time on Markov chain
  - MCMC/ F-flow/ **F-flow + MCMC**



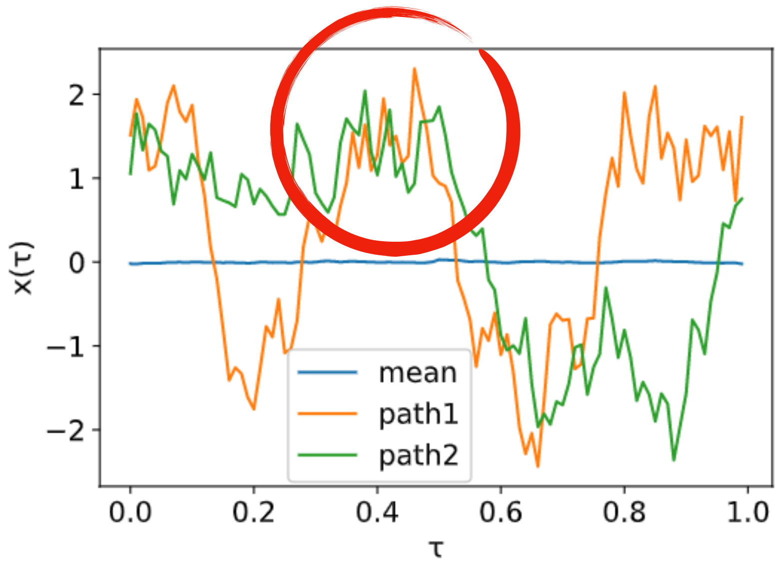
	f^2	-1.0	0.0	1.0	2.0	3.0	4.0	5.0
	F-flow	2.66(±0.001)	1.06(±0.001)	1.15(±0.001)	2.28(±0.001)	3.14(±0.002)	3.60(±0.004)	4.03(±0.01)
E_0	F-flow+MCMC	2.64(±0.001)	1.04(±0.001)	1.11(±0.001)	2.23(±0.001)	3.10(±0.003)	3.60(±0.004)	4.01(±0.002)
	MCMC	2.64(±0.001)	1.04(±0.001)	1.11(±0.001)	2.23(±0.001)	3.10(±0.001)	3.60(±0.001)	4.00(±0.001)
	F-flow	6.37(±0.02)	3.84(±0.02)	2.96(±0.01)	2.87(±0.003)	3.38(±0.003)	3.74(±0.004)	4.13(±0.01)
E_1	F-flow+MCMC	6.35(±0.02)	3.76(±0.01)	2.74(±0.01)	2.82(±0.004)	3.31(±0.003)	3.73(±0.004)	4.10(±0.003)
	MCMC	6.35(±0.008)	3.77(±0.006)	2.71(±0.004)	2.82(±0.003)	3.32(±0.001)	3.73(±0.001)	4.09(±0.001)
	F-flow	10.70(±0.05)	7.41(±0.05)	5.38(±0.03)	6.36(±0.02)	9.12(±0.02)	10.91(±0.03)	12.30(±0.03)
E_2	F-flow+MCMC	10.69(±0.08)	7.44(±0.03)	5.87(±0.04)	6.33(±0.01)	9.11(±0.03)	10.87(±0.03)	12.27(±0.08)
	MCMC	10.68(±0.03)	7.41(±0.02)	5.85(±0.01)	6.35(±0.01)	9.07(±0.01)	10.85(±0.02)	12.22(±0.02)

F-Flow Model

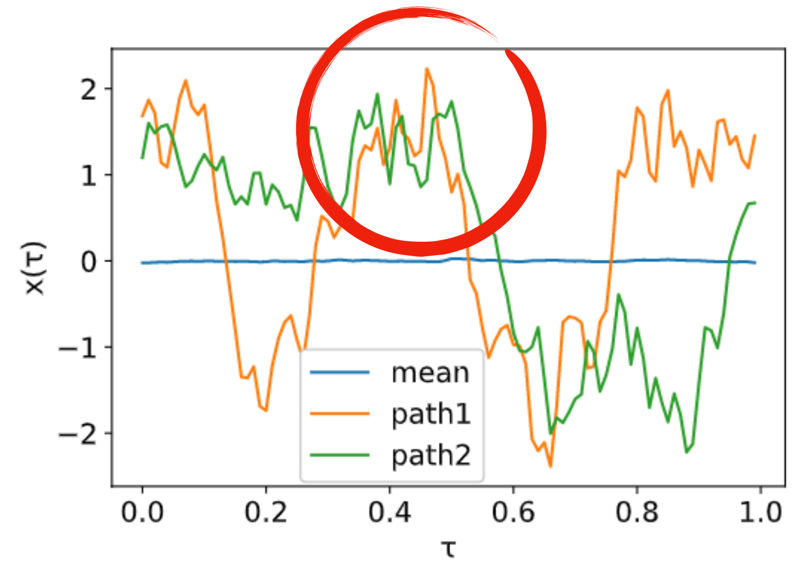
Phys. Rev. D 107, 056001

Feynman paths

full frequency modes

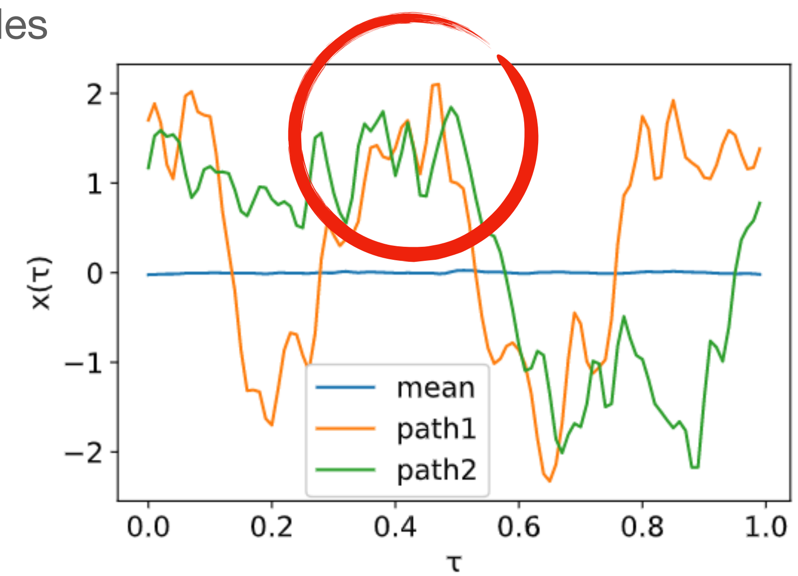


80% frequency modes

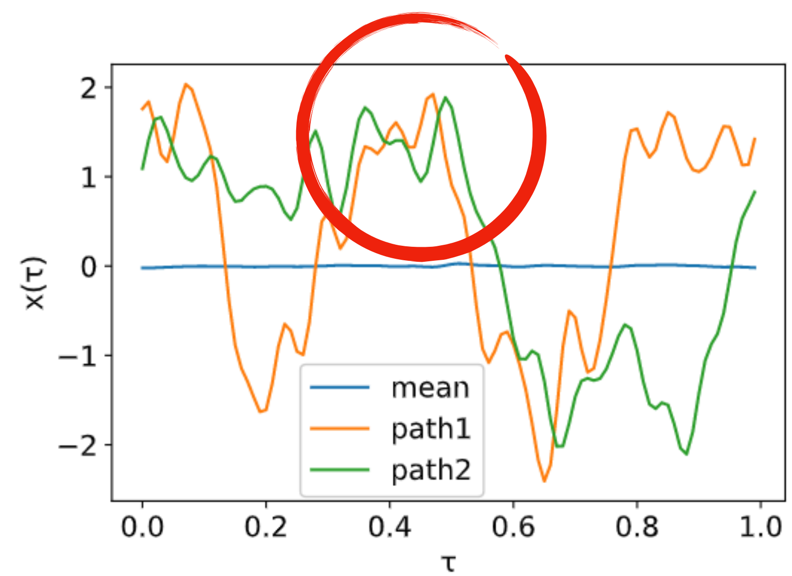


$f^2 = 2$

70% frequency modes



60% frequency modes



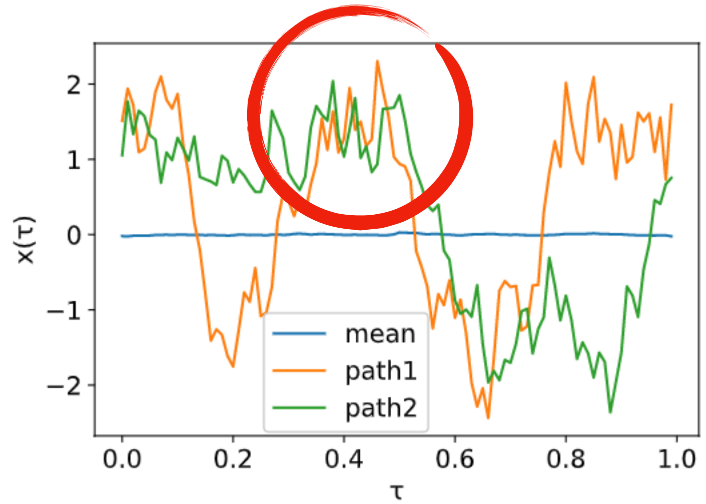
Quantum fluctuations

F-Flow Model

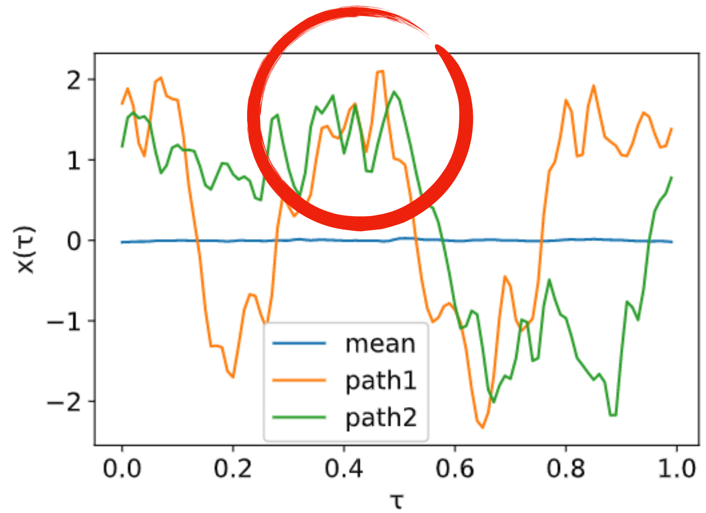
Matsubara frequency

Phys. Rev. D 107, 056001

full frequency modes



70% frequency modes



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

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

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

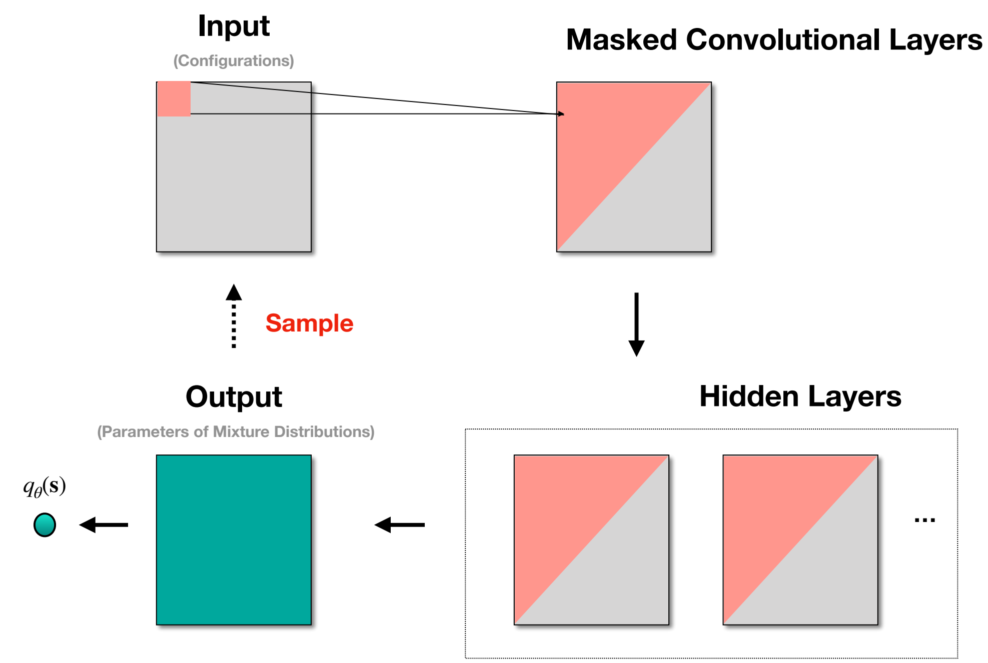
Quantum fluctuations

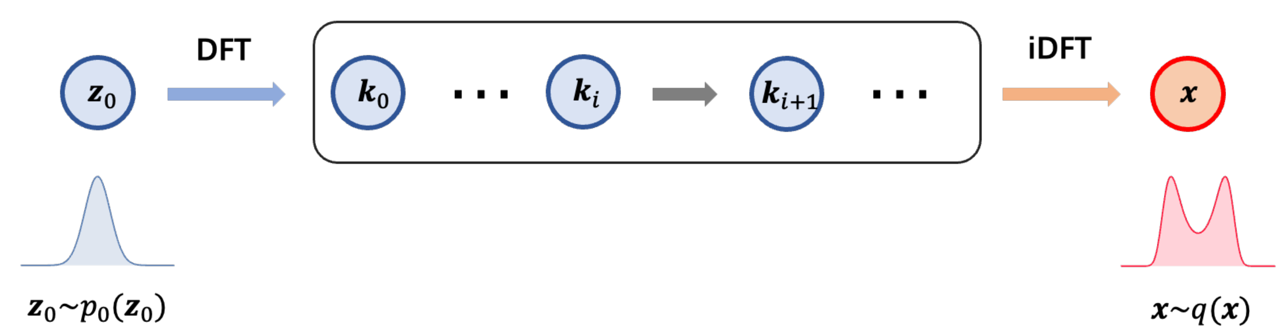
The kinetic term decouples in Matsubara frequency sapce

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
  - Lattice potential/periodic potential
  - Few-body systems
  - From SU(2) to QCD





Diffusion Models

Space Opera

Jason Allen via Midjourney

Diffusion Models

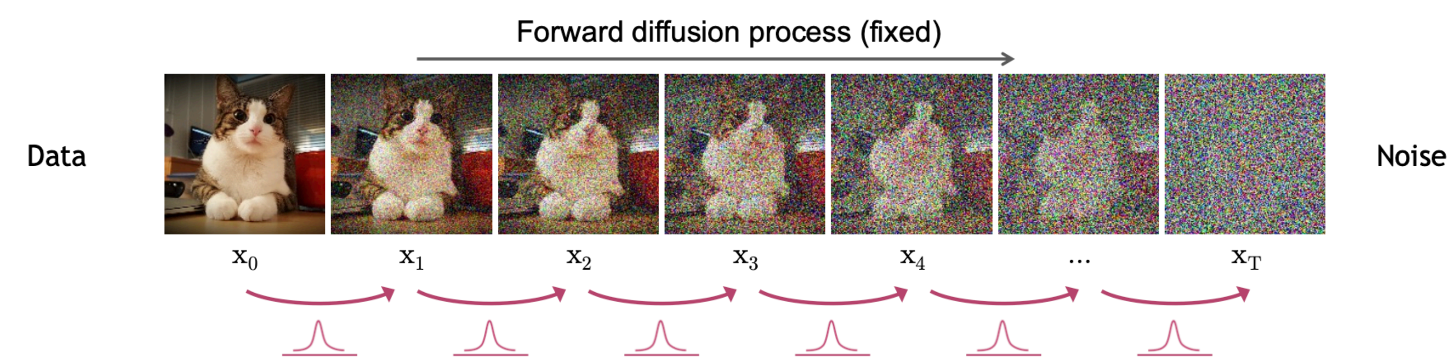
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020

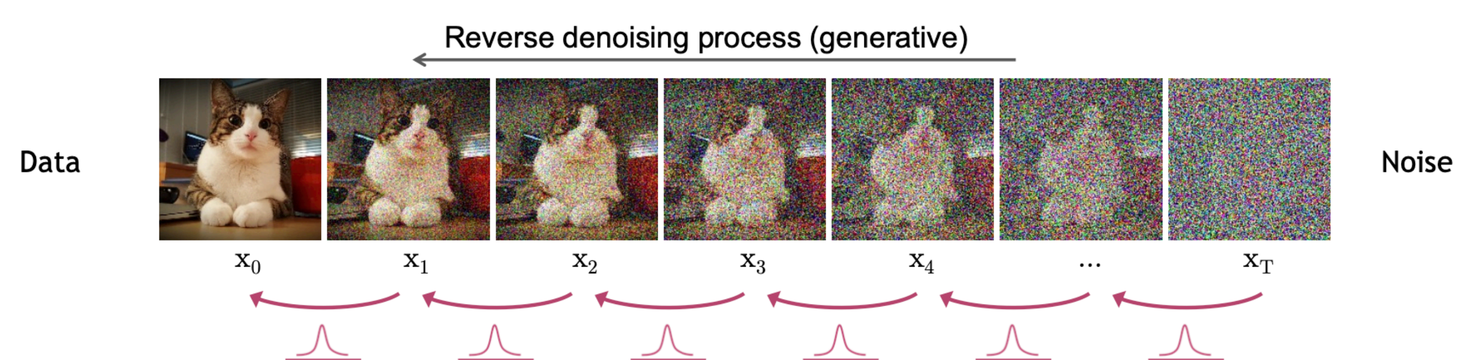
Denoising Diffusion Models

- Forward diffusion process that **gradually adds noise** to input

- Reverse denoising process that learns to **generate data by denoising**

- **Probabilistic Models**
  Desgin a model to learn how to denoise from a simple distribution to a target distribution



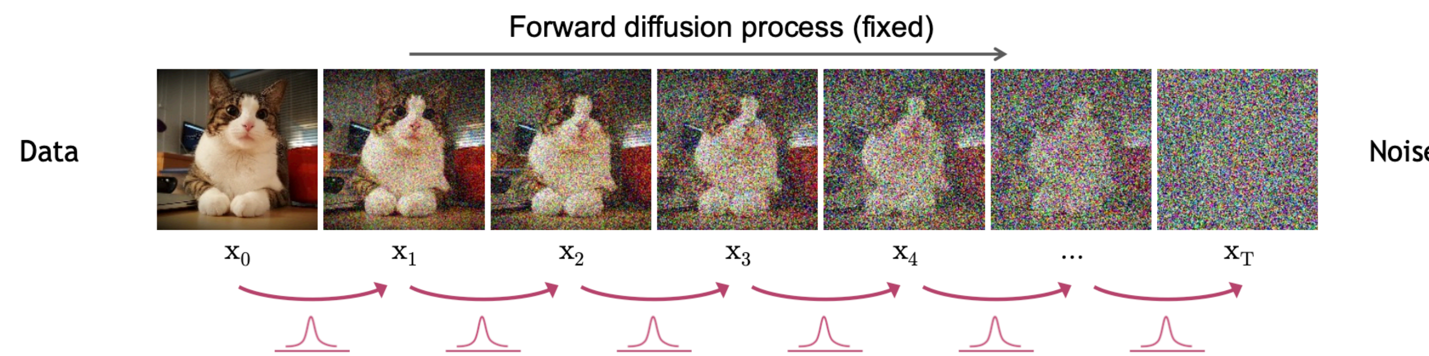


Diffusion Models

Denoising Diffusion Models

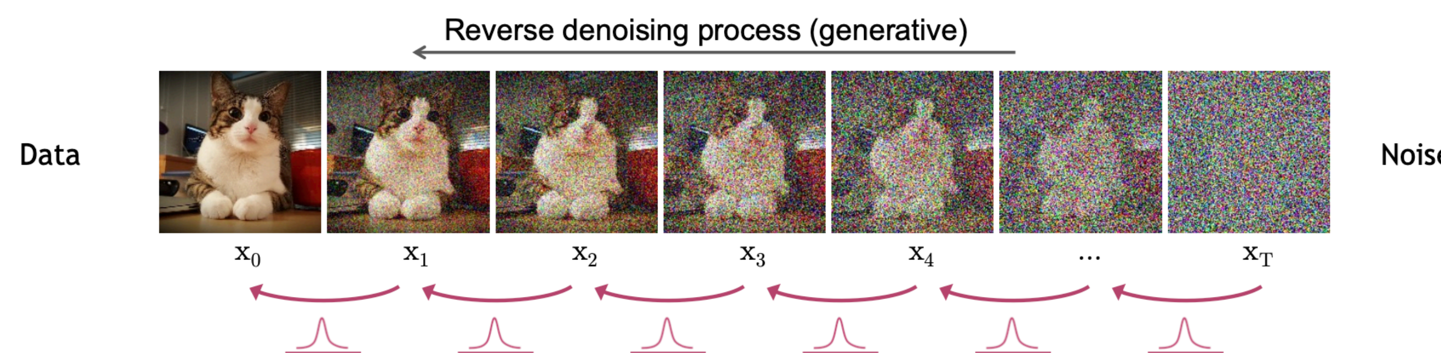
- **Probabilistic Models**
  desgin a model to learn how to denoise from simple distribution to a target distribution

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}\,|\,x_t)$$



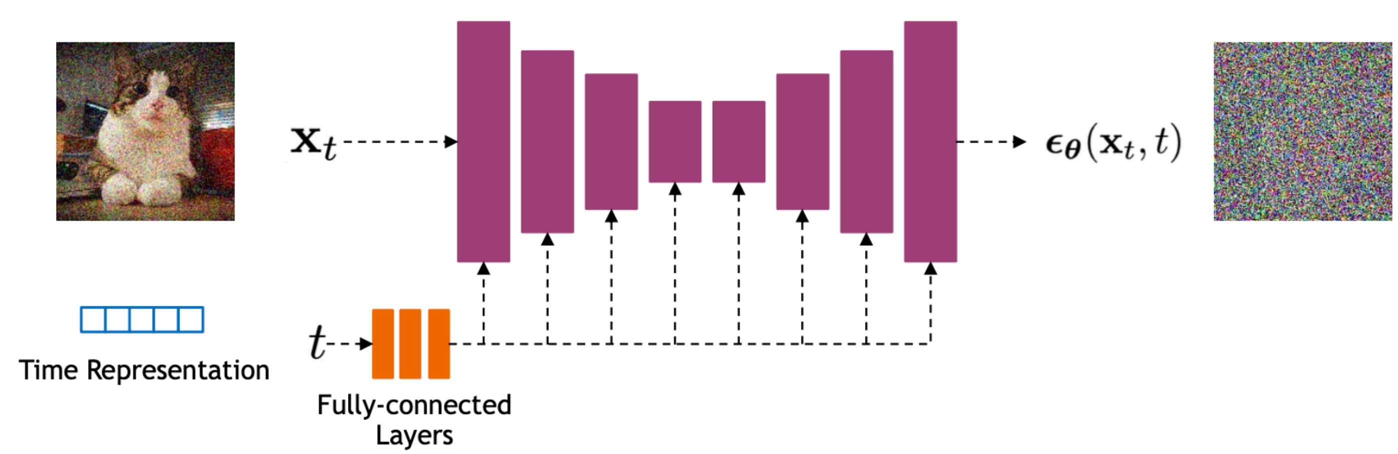
- Optimization
  - Loss function: Variational lower bound(VLB)

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}\,|\,x_0)}]$$



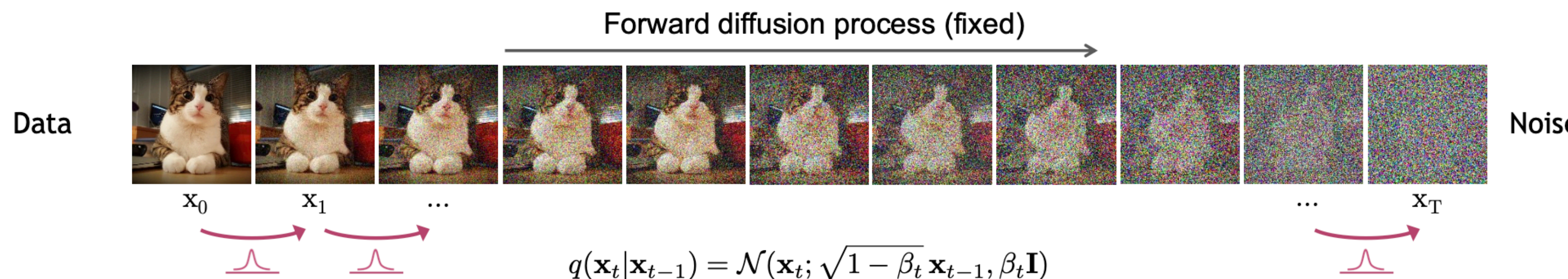
  - Trainable parameters θ : neural networks

- Parameterizing the Denoising Model
  - Trainable neural networks are usually **U-Net** architectures with **ResNet blocks** for representing the standard deviation $\epsilon_\theta(\mathbf{x}_t, t)$



Diffusion Models

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



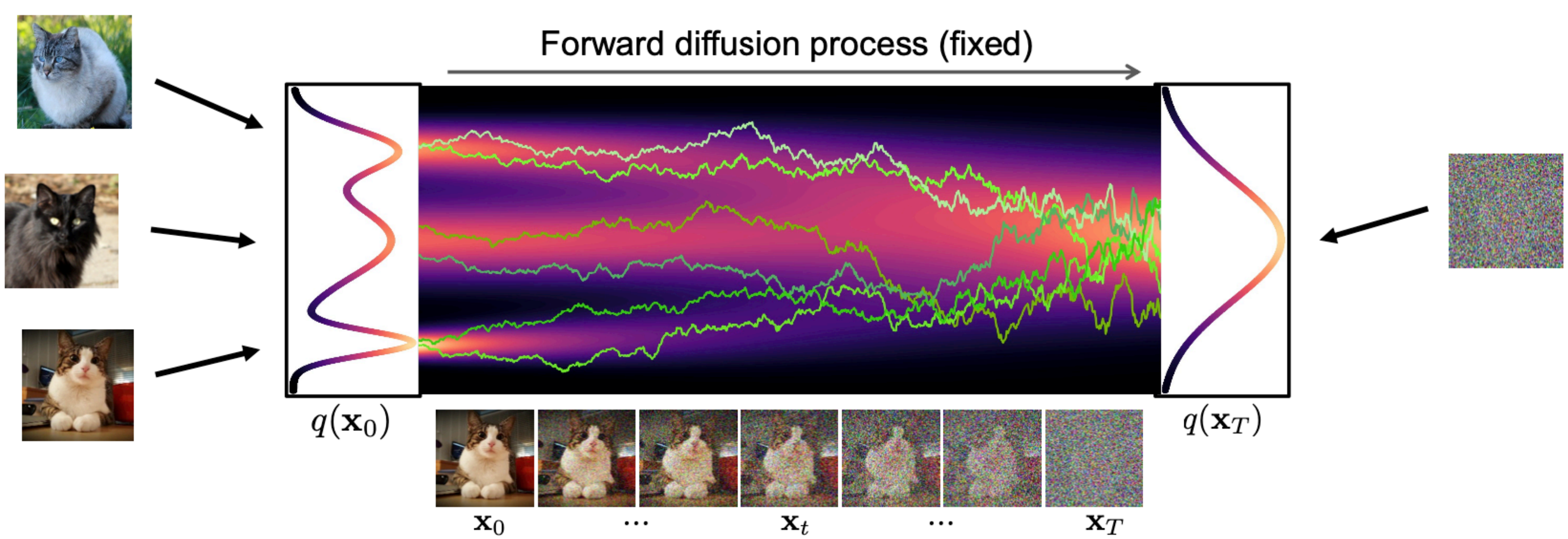
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

- Forward diffusion process is driven by a stochastic differential equation(SDE)
- **Forward Diffusion SDE**
  - Drift term: pulls towards mode
  - Diffusion term: injects noise
- Reverse denoising process ?

$$\begin{aligned} \mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\ &= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && (\beta_t := \beta(t)\Delta t) \\ &\approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && \text{(Taylor expansion)} \end{aligned}$$

$$\mathrm{d}\mathbf{x}_t = \boxed{-\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t} + \boxed{\sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t}$$

Stochastic Differential Equation (SDE)
describing the diffusion in infinitesimal limit



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

$$d\mathbf{x}_t = \boxed{-\frac{1}{2}\beta(t)\mathbf{x}_t\,dt} + \boxed{\sqrt{\beta(t)}\,d\boldsymbol{\omega}_t}$$

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

$$d\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{"Score Function"}}\right]}^{\text{drift term}} dt + \overbrace{\sqrt{\beta(t)}}^{\text{diffusion term}}\,d\bar{\boldsymbol{\omega}}_t$$

Model the score function with neural networks!

$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\substack{\text{diffusion}\\ \text{time } t}} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)}}_{\substack{\text{diffused}\\ \text{data } \mathbf{x}_t}} ||\underbrace{\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\substack{\text{neural}\\ \text{network}}} - \underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\substack{\text{score of}\\ \text{diffused data}\\ \text{(marginal)}}}||_2^2$$

Denoising Score Matching objective with loss weighting $\lambda(t)$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\frac{\lambda(t)}{\sigma_t^2}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)||_2^2$$

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

$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\substack{\text{diffusion}\\ \text{time } t}} \underbrace{\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}}_{\substack{\text{data}\\ \text{sample } \mathbf{x}_0}} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)}}_{\substack{\text{diffused data}\\ \text{sample } \mathbf{x}_t}} ||\underbrace{\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\substack{\text{neural}\\ \text{network}}} - \underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\mathbf{x}_0)}_{\substack{\text{score of diffused}\\ \text{data sample}}}||_2^2$$

Diffusion Models

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

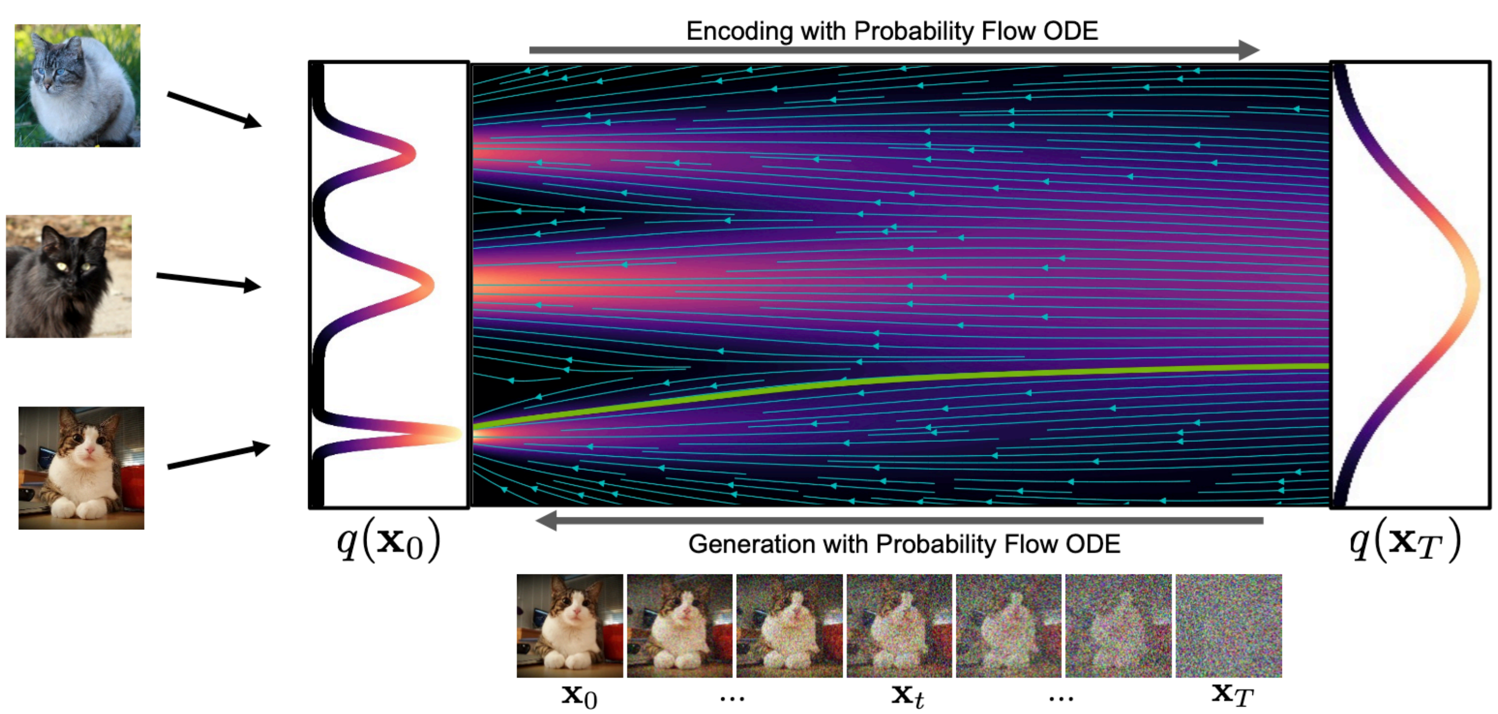
Score-based model

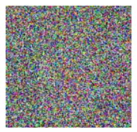
- **Forward Diffusion SDE**
  - Drift term: pulls towards mode
  - Diffusion term: injects noise

$$\mathrm{d}\mathbf{x}_t = \boxed{-\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t} + \boxed{\sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t}$$

- **Reverse Generative Diffusion SDE**
  - Model the score function with neural networks!
  - Sampling from the SDE or ODE

$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]\mathrm{d}t}^{\text{drift term}} + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$





Probability FlowODE

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\left[\mathbf{x}_t + \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\right]\mathrm{d}t$$

- Enables use of advanced ODE solvers
- Deterministic encoding and generation
- Exact Log-Likelihood computation

Lattice Field Theory

Scalar ϕ^4 field

- **Action** of bare fields

$$S = \int \mathrm{d}^d x \mathrm{d}t \mathscr{L} = \int \mathrm{d}^d x \mathrm{d}t (\frac{1}{2}(\partial^2 \phi_0^2 - m^2 \phi_0^2) - \frac{\lambda_0}{4!}\phi_0^4)$$

- Euclidean 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

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

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

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

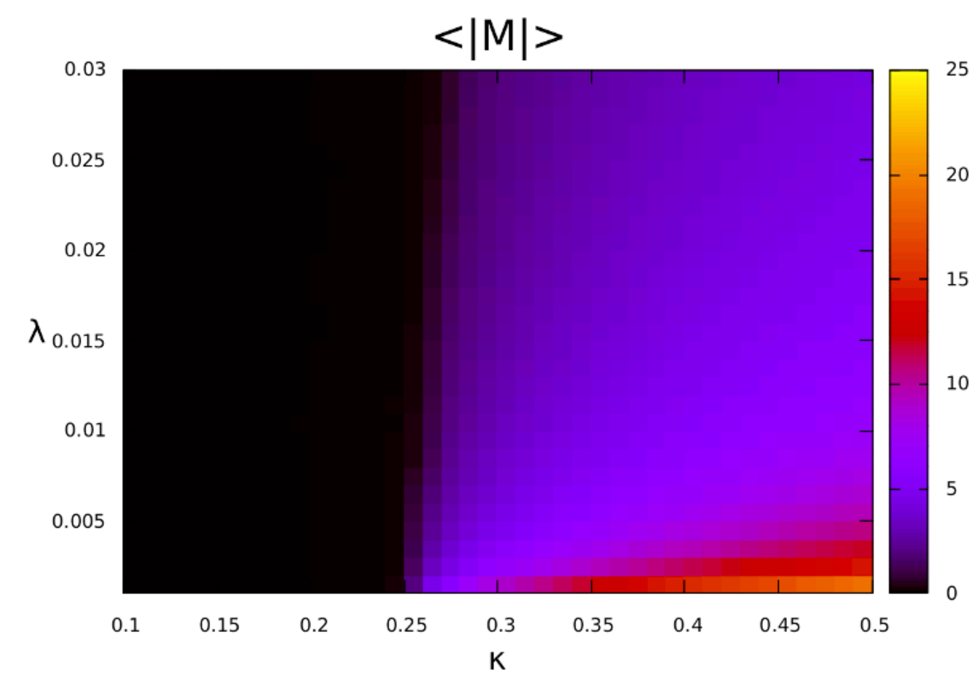
  - Hopping parameter κ
  - Coupling constant λ

Quantum 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 $d = 2$

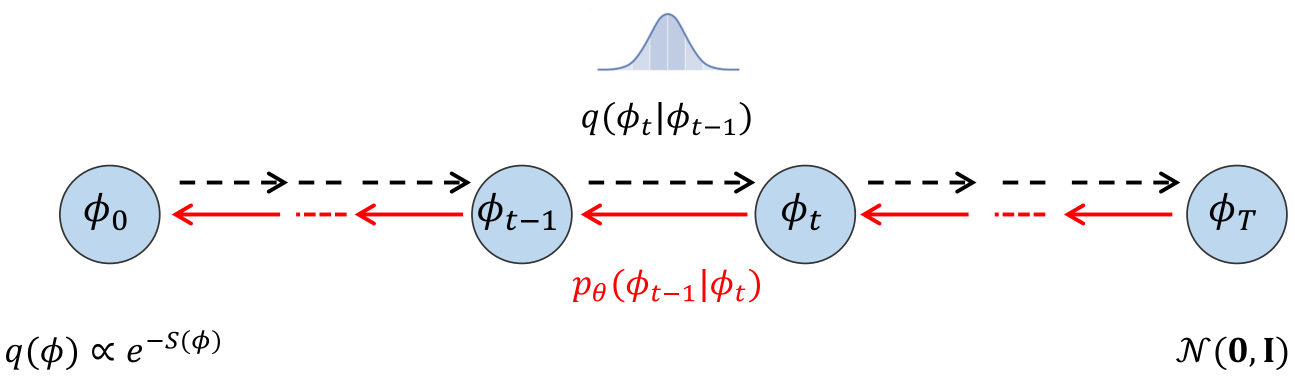
Lattice Field Theory

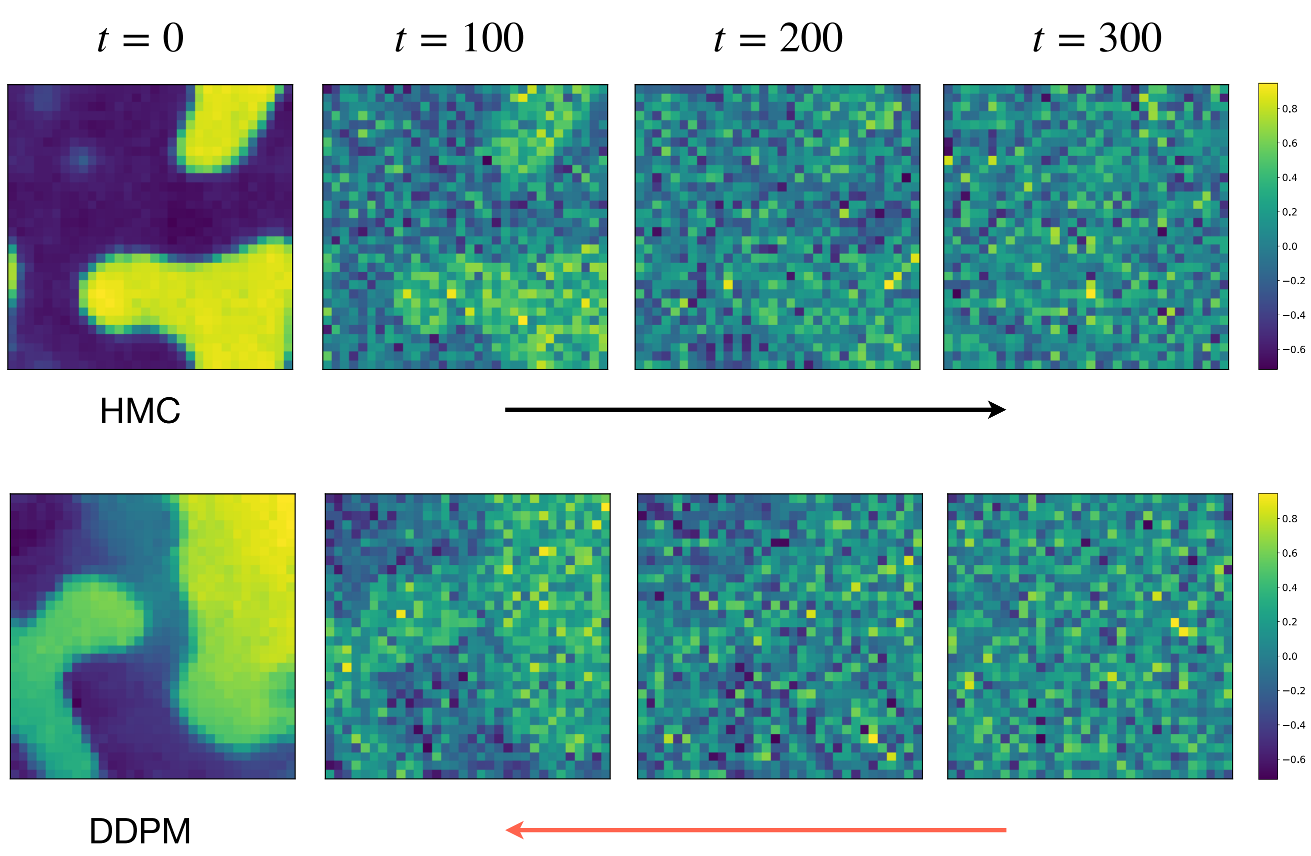
DM in broken phase

in preparation

- **Diffusion models**
  - DDPM
  - $T = 300$
  - Time scheme $\beta \in [10^{-4}, 0.02]$
- **Data generation**
  - 2-dim 32×32 lattice
  - Hamiltonian Monte Carlo(HMC)
    - 100 step burn-in loop
    - 1024 chains
      - 64 pre-equilibrium steps
      - 64 updates
  - 5120 configurations for training

$$\kappa = 1.0, \lambda = 0.022$$





Lattice Field Theory

DM in symmetric phase

data set	M	χ_2	U_L
Training (size=1024)	2.8E-3 ± 3.1E-3	2.48 ± 0.112	-2.76E-2 ± 0.113
Testing (size=5120)	0.4E-3 ± 1.4E-3	2.50 ± 4.89E-2	1.44E-2 ± 4.60E-2
Generated (size=5120)	-4.9E-3 ± 0.9E-3	2.33 ± 3.21E-2	2.30E-2 ± 4.30E-2

- **Diffusion models**
  - $T = 50$
- **Data generation**
  - 1024 configurations for training

$$\kappa = 0.21, \lambda = 0.022$$

  - 5120 configurations for testing
- **Observables**
  - Magnetization

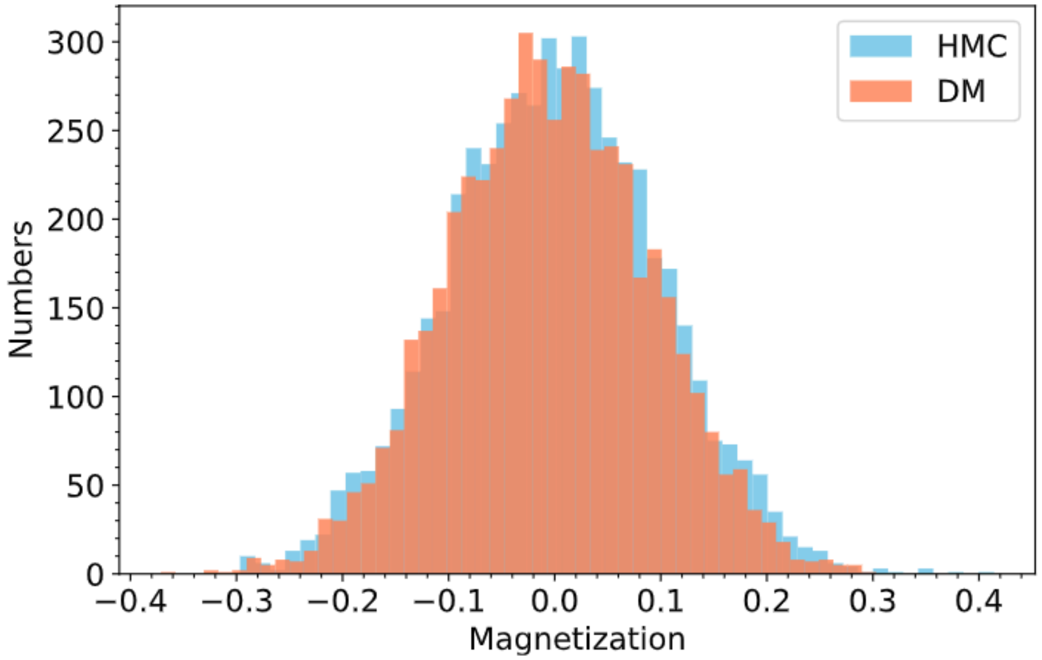
$$\langle M \rangle = \langle \frac{1}{V} \sum_{x \in \Lambda} \phi(x) \rangle$$

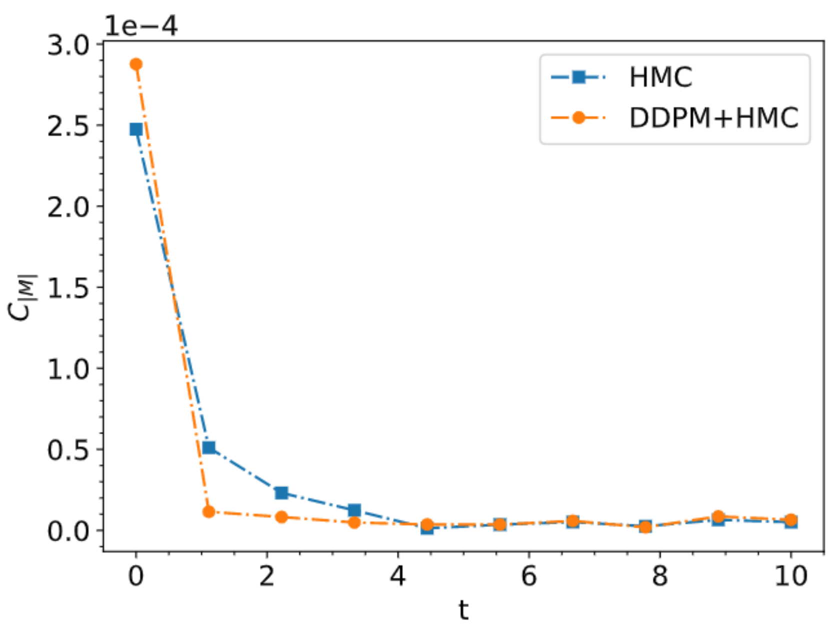
  - Two-point susceptibility

$$\chi_2 = V(\langle M^2 \rangle - \langle M^2 \rangle)$$

  - Binder cumulant

$$U_L = 1 - \frac{1}{3} \frac{\langle M^4 \rangle}{\langle M^2 \rangle^2}$$





Stochastic Quantization

Parisi G. and Wu Y. S., Sci. China, A **24**, ASITP-80-004 (1980).

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

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

τ: fictitious time, α: diffusion constant

- **Fokker-Planck equation**

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

$$\tilde{F} \equiv \alpha\int d^n x \left\{ \frac{\delta}{\delta\phi}\left(\frac{\delta}{\delta\phi} + \frac{\delta S_E}{\delta\phi}\right)\right\},$$

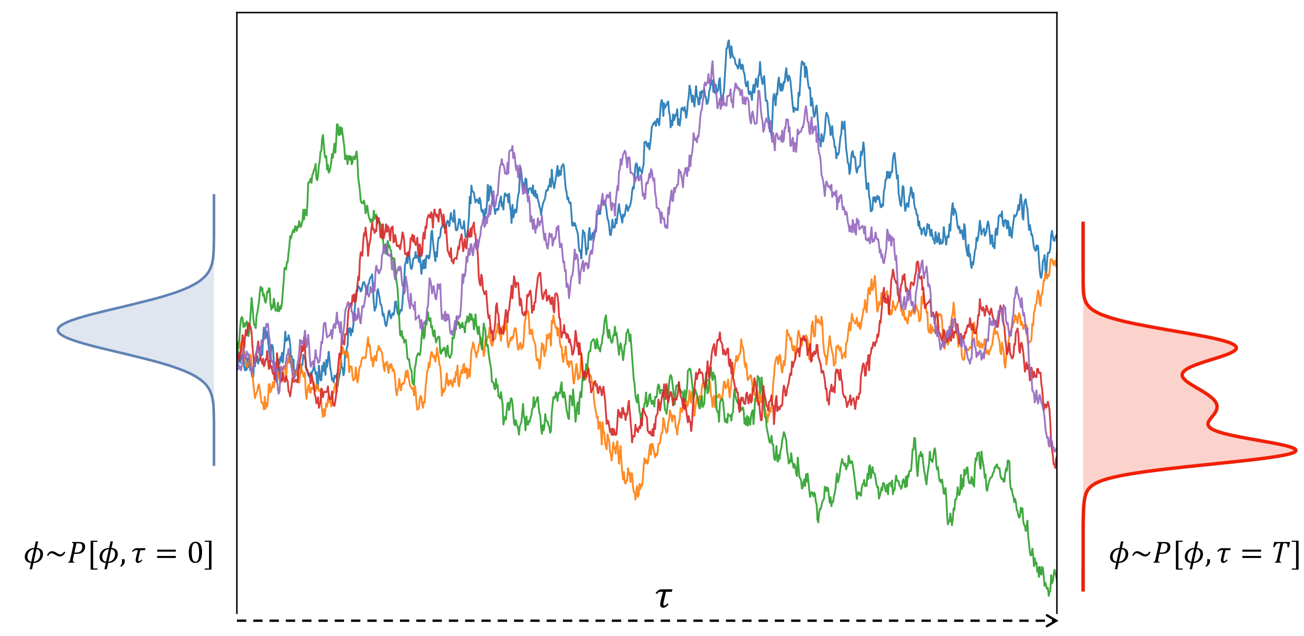
$$\partial_\tau P[\phi,\tau] = \tilde{F}P[\phi,\tau]$$

The equilibrium solution (long-time limit) of the equation,

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

- 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 world

No need gauge-fixing,
Can handle fermionic fields naturally
(Complex Langevin method)
....

Stochastic Quantization

DMs as SQ

- **Diffusion models**
  - **Reverse SDE**

    $$d\phi = [-\frac{1}{2}\beta_t\phi - \beta_t\partial_\phi \log q_t(\phi)]dt + \sqrt{\beta_t}d\bar{w}$$

  - **Redefine** $-\beta_t dt \equiv d\tau'$

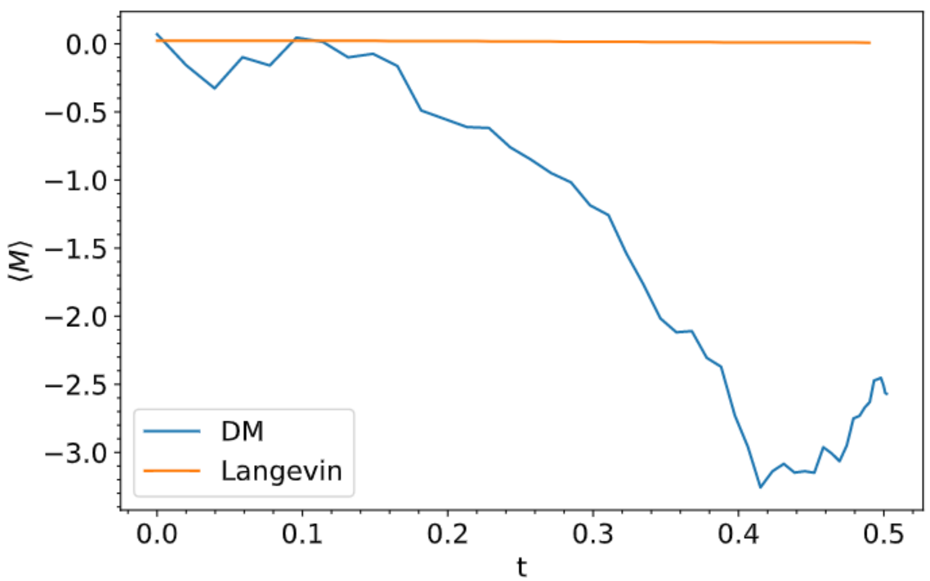
    $$d\phi = [\frac{1}{2}\phi + \partial_\phi \log q_{\tau'}(\phi)]d\tau' + d\bar{w}$$

    $$\frac{\partial\phi}{\partial\tau'} = \boxed{[\frac{1}{2}\phi + \partial_\phi \log q_{\tau'}(\phi)]} + \frac{\partial\bar{w}}{\partial\tau'}$$

- **Stochastic quantization**

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

Converge faster than Langevin dynamics in broken phase



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

$$\log p(\phi) \propto -S(\phi)$$

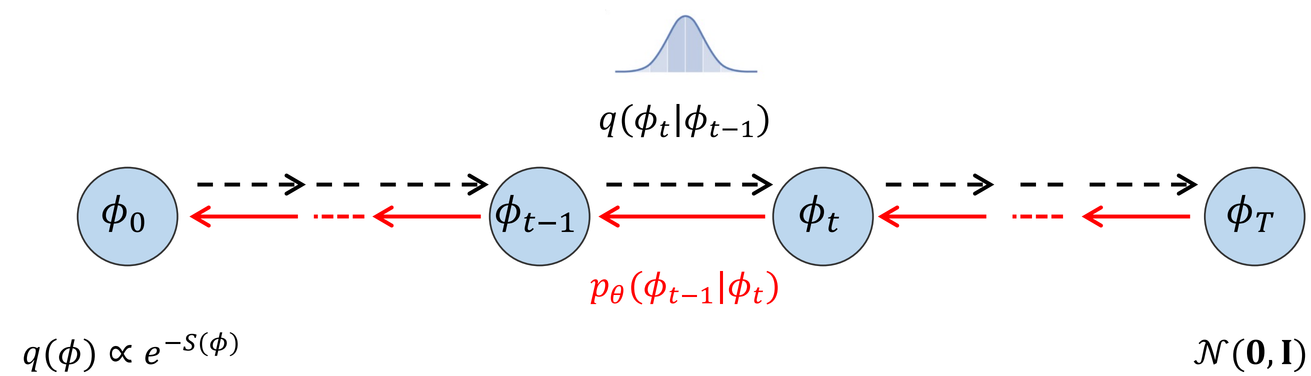
The reverse mode of a well-trained diffusion model at $\tau' \to 0$ is a stochastic quantization representation of the field theory.

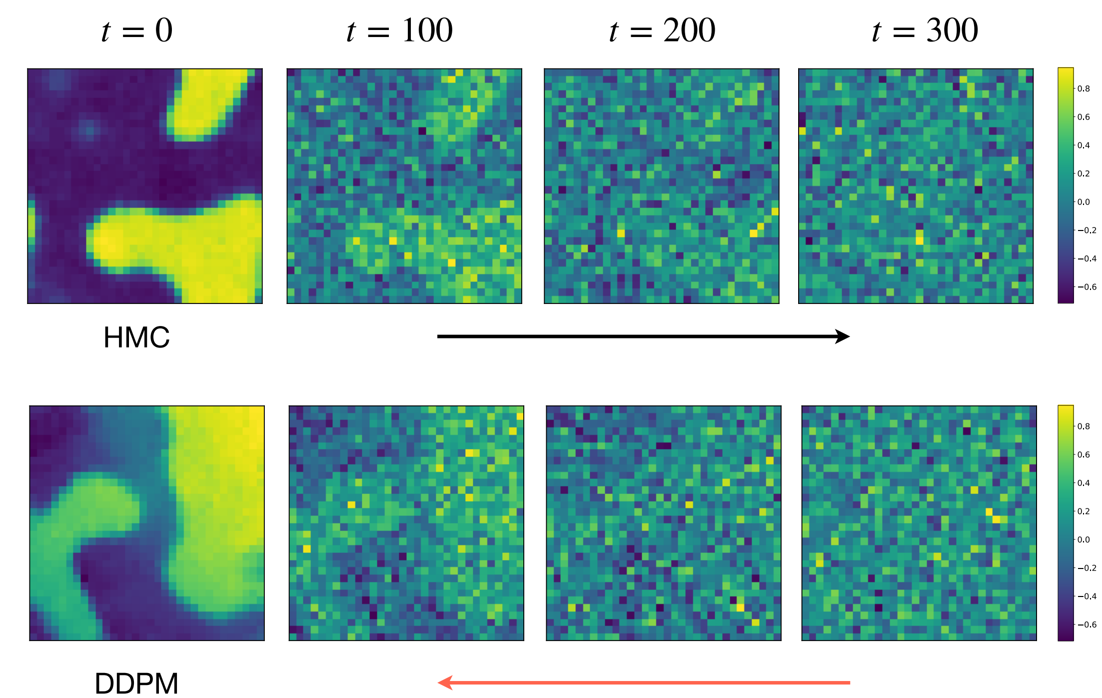
Summary III

- Serve as an efficient sampler
- **Future works**
  - Diffusion models(DMs) as Stochastic Quantization(SQ)
    - Compare drift terms in SQ and DMs
    - Excat likelihood(**free energy**) estimation
  - Variational calculation

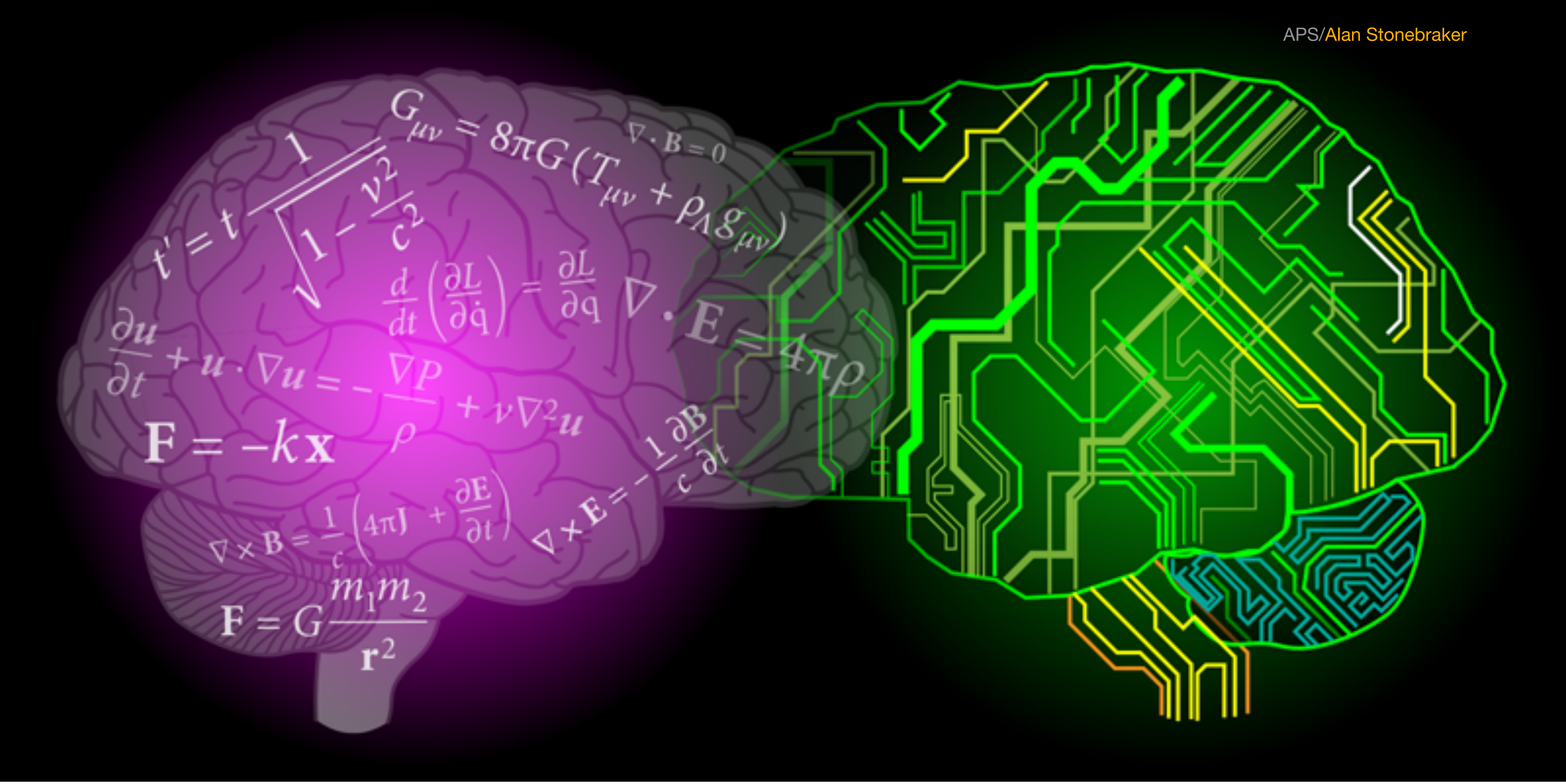
$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\text{diffusion time } t} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)}}_{\text{diffused data } \mathbf{x}_t} || \underbrace{\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\text{neural network}} - \underbrace{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}_{\text{score of diffused data (marginal)}} ||^2$$

- **Complex Langevin dynamics**
  - Touch sign-problem





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



Future

ML meets Physics, opportunities and challenges