# Application of neural networks for calculating path integrals in quantum field theory

Chistyakov V. V. Ivanov A. S. Salnikov D. V. Vasiliev A. V.

MSU, INR

Quarks 2024

# Path integral

$$\langle O \rangle_\beta = \frac{Tr\left(e^{-\beta H} O\right)}{Z} = \sum_n \langle n|O|n\rangle \frac{e^{-\beta E_n}}{Z} \tag{1}$$

$$\langle O \rangle_\beta = \frac{1}{Z} \int D\left[\phi(x)\right] e^{-S_E[\phi]}\, O(\phi) \tag{2}$$

$$\beta \to \infty \qquad \langle O \rangle_\beta \to \langle 0|O|0\rangle \tag{3}$$

# 1+0 QFT and general kinetic term

$$H = T(p) + V(x) \tag{4}$$

$$\langle O \rangle_\beta \approx \int \frac{d^N x}{Z} \mathcal{K}_T(x) \mathcal{K}_V(x) \bar{O}(x) \equiv \int d^N x \frac{e^{-S(x)}}{Z} \bar{O}(x) \tag{5}$$

$$\mathcal{K}_T(x) = \prod_{i=1}^{N} K_T(x_{i+1} - x_i) \qquad \mathcal{K}_V(x) = \prod_{i=1}^{N} K_V(x_i) \tag{6}$$

$$K_V(x) = \exp\left(-\tau V(x)\right) \tag{7}$$

$$K_T(\xi) = \int \frac{dp}{2\pi} \exp\left(-\tau T(p) + ip\xi\right) \tag{8}$$

# Relativistic kinetic term

$$T(p) = \sqrt{p^2 + m^2} - m \tag{9}$$

$$K_T(\xi) = \frac{m}{\pi\sqrt{1 + \left(\frac{\xi}{\tau}\right)^2}} K_1\left(m\tau\sqrt{1 + \left(\frac{\xi}{\tau}\right)^2}\right) \tag{10}$$

$$m \to 0 \qquad T(p) = |p| \tag{11}$$

$$K_T(\xi) = \frac{1}{\pi\tau}\frac{1}{1 + \left(\frac{\xi}{\tau}\right)^2} \tag{12}$$

Ivanov A. S., Novoselov A. A., Pavlovsky O. V. "Relativistic path integral monte carlo: Relativistic oscillator problem"// IJMP C. 2016. 27, 11. 1650133–1–1650133–14.

# Monte-Carlo algorithm

$$P(x) = \frac{e^{-S(x)}}{Z} \tag{13}$$

$$\langle O \rangle = \int d^N x P(x) \bar{O}(x) \approx \frac{1}{M} \sum_{k=1}^{M} \bar{O}[x^{(k)}] \tag{14}$$

$$\{x^{(k)}\} \sim P(x) \tag{15}$$

we need to generate a sample $\{x^{(k)}\}$ from distribution $P(x)$

# Markov chain Monte Carlo (Metropolis)

- generate a sample $X = \{x^{(k)}\}$ from some simple distribution.
  for example, $x_i^{(k)} = 0$ — "cold"trajectories
- construct new random trajectory for each $x \in X$ $y^{(k)} = x^{(k)} + \sqrt{\tau} dh$

$$d \sim \mathbb{U}[-1, 1] \qquad h \in \mathbb{R}^N$$

- form a new sample $X^1$: we change $x^{(k)} \in X$ on $y^{(k)}$ with transition probability $\pi(y, x) = \min\left(\frac{P[y]}{P[x]}, 1\right)$
- form a new sample $X^2$ based on $X^1$ in the same way.
  Continue the process.
  $X^1, ..., X^N$— Markov chain with final distribution $P(x)$

# Disadvantages of the technique

- takes a lot of time
- unable to take symmetries into account

# Exactly soluble model: harmonic oscillator

$$H = \frac{p^2}{2} + \frac{x^2}{2} \tag{16}$$

$$S(x) = \frac{1}{2}(Ax, x) \tag{17}$$

$$A = \frac{1}{\tau}\left((2+\tau^2)I - T - T^\dagger\right) \tag{18}$$

$$T = \begin{pmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & 1\dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & 1 \\ 1 & \dots & \dots & 0 \end{pmatrix} \tag{19}$$

$$x = wz \qquad z \sim \mathcal{N}(0,1) \qquad w^T A w = I \tag{20}$$

# Orthogonal transformation

$$w = On \qquad O^{\dagger} T O = diag(...) \tag{21}$$

$$v_0 = \frac{1}{\sqrt{N}}(1, 1, ..., 1)^T \qquad v_{N/2} = \frac{1}{\sqrt{N}}(1, -1, 1, ..., -1)^T \tag{22}$$

$$u_k = \sqrt{\frac{2}{N}} \begin{pmatrix} 1 \\ \cos\left(\frac{2\pi k}{N}\right) \\ \cos\left(\frac{4\pi k}{N}\right) \\ ... \\ \cos\left(\frac{2(N-1)\pi k}{N}\right) \end{pmatrix} \qquad w_k = \sqrt{\frac{2}{N}} \begin{pmatrix} 0 \\ \sin\left(\frac{2\pi k}{N}\right) \\ \sin\left(\frac{4\pi k}{N}\right) \\ ... \\ \sin\left(\frac{2(N-1)\pi k}{N}\right) \end{pmatrix} \tag{23}$$

$$O = ||v_0, v_{N/2}, u_1, w_1, ..., u_{N/2-1}, w_{N/2-1}|| \tag{24}$$

# Minimization task

$$x = wz \qquad P_w(x) = r(z)\,|\det w| \tag{25}$$

$$r(z) = \frac{1}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2}||z||^2\right) \tag{26}$$

$$P(x) = \frac{e^{-\frac{1}{2}(Ax,x)}}{Z} \tag{27}$$

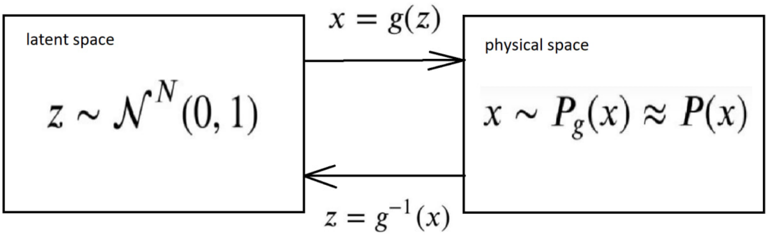
$$D_{KL}(p|q) = \int dx\, p(x) \ln \frac{p(x)}{q(x)} \tag{28}$$

$$D_{KL}(p|q) \geq 0 \qquad D_{KL}(p|q) = 0 \Longleftrightarrow p(x) = q(x) \tag{29}$$

$$L[w] = D_{KL}(P_g|P) - \ln Z = \frac{1}{2} tr\left(w^T A w\right) - \ln|\det w| \tag{30}$$

$$\nabla_w L = Aw - (w^T)^{-1} \qquad \nabla_w L = 0 \Longleftrightarrow w^T A w = I \tag{31}$$

# Normalizing flow




$$P_g(x) = r(z)\left|\det\frac{\partial g}{\partial z}\right| \qquad P(x) = \frac{e^{-S(x)}}{Z} \tag{32}$$

$$L[w] = D_{KL}(P_g|P) - \ln Z = \frac{1}{M}\sum_{k=1}^{M} S(g(z_k|w)) - \ln\left|\det\frac{\partial g(z_k|w)}{\partial z}\right| \tag{33}$$

# Normalizing flow

The map $g$ is a composition of affine transformations

$$g = A_n \circ ... \circ A_1 \tag{34}$$

We devide $z$ on two parts

$$z = u + v \tag{35}$$

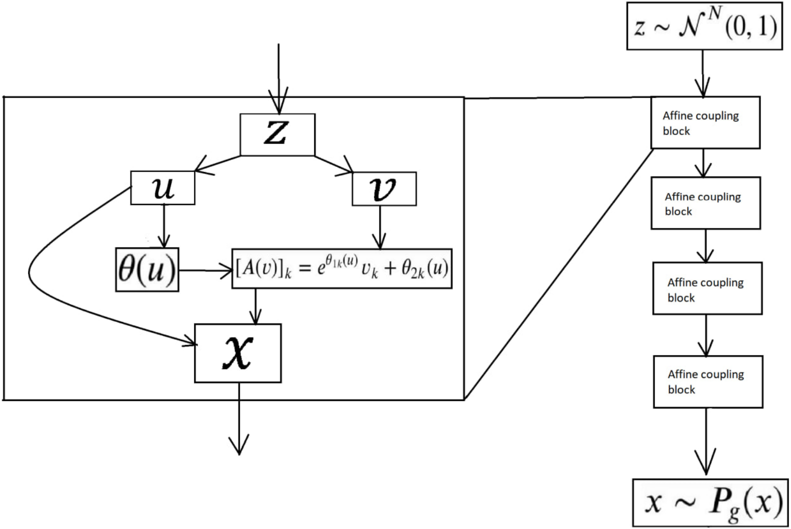
For example, $u$ contains coordinats of $z$ with even numbers, and $v$ – the odd one.

$$A(u) = u \qquad [A(v)]_k = e^{\theta_{1k}(u)} v_k + \theta_{2k}(u) \tag{36}$$

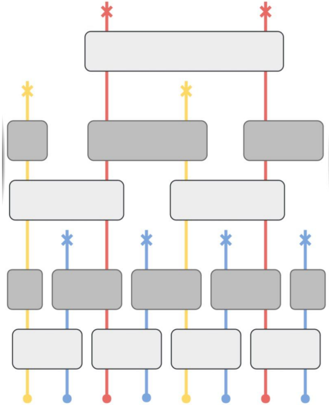
$$\theta : \mathbb{R}^{N/2} \to \mathbb{R}^N \tag{37}$$

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, Balaji Lakshminarayanan "Normalizing Flows for Probabilistic Modeling and Inference" Journal of Machine Learning Research, 22(57):1-64, 2021

# Normalizing flow



# Multiscale architecture



Shuo-Hui Li, Lei Wang "Neural network renormalization group" Phys. Rev. Lett. 121, 260601 (2018) arXiv:1802.02840 — Ising model

# Shift symmetry

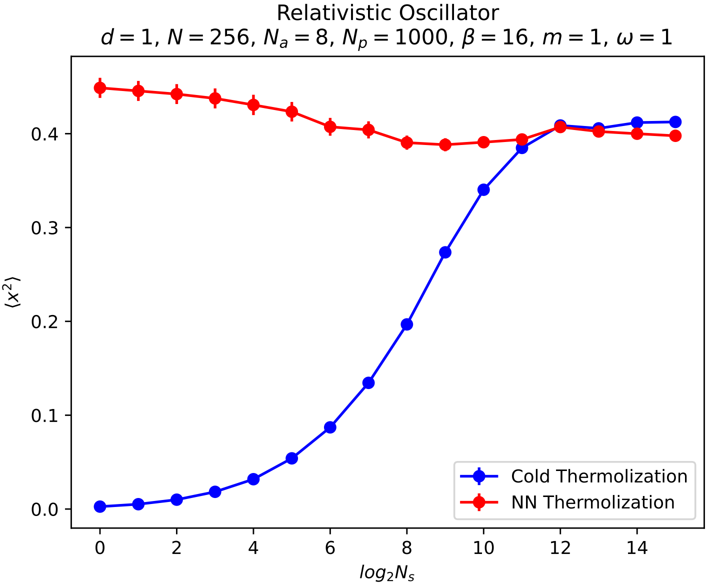
$$y = g(z) \tag{38}$$
$$x = Oy \tag{39}$$

- $y_i$ transforms on irreducible (on $\mathbb{R}$) representations of group $\mathbb{Z}_N$

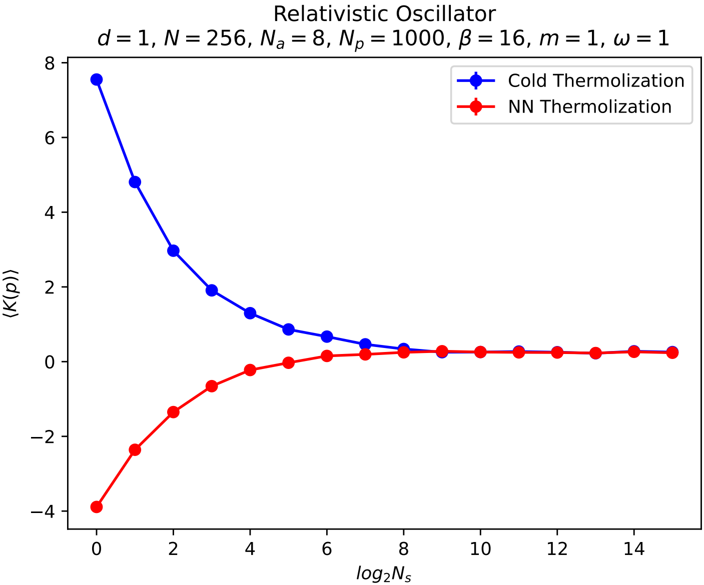
# Some papers

- M. S. Albergo, G. Kanwar, and P. E. Shanahan "Flow-based generative models for Markov chain Monte Carlo in lattice field theory"Phys. Rev. D 100, 034515 — $\phi^4$
- Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, Phiala E. Shanahan "Introduction to Normalizing Flows for Lattice Field Theory" arXiv:2101.08176 (preprint) — QCD, Gauge equivariant coupling layers
- Isay Katsman, Aaron Lou, Derek Lim, Qingxuan Jiang "Equivariant Manifold Flows" arXiv:2107.08596 [stat.ML] (preprint)

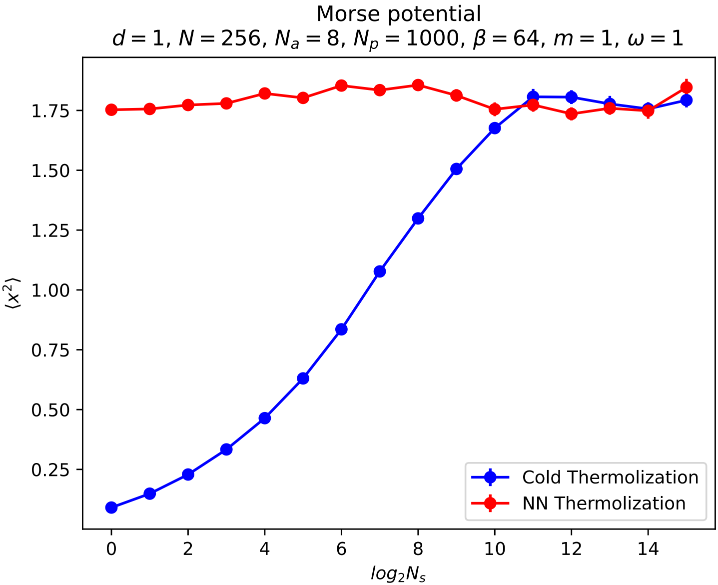
$$H = \sqrt{p^2 + m^2} - m + \frac{m\omega^2 x^2}{2} \tag{40}$$



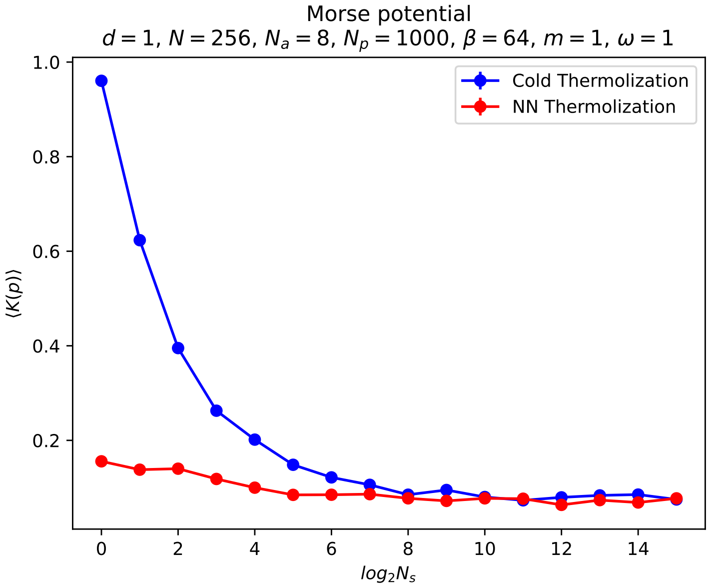
$$H = \sqrt{p^2 + m^2} - m + \frac{m\omega^2 x^2}{2} \tag{41}$$



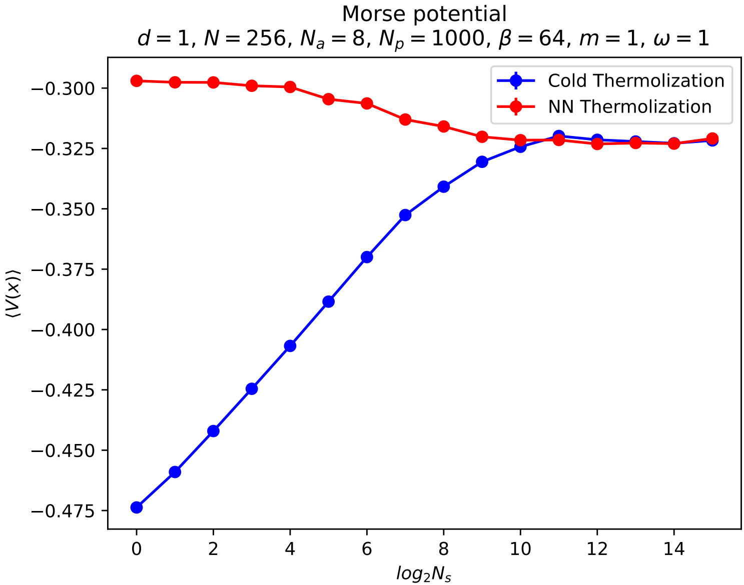
$$H = \frac{p^2}{2} + \frac{1}{2}\left(\left[e^{-\omega x} - 1\right]^2 - 1\right) \tag{42}$$



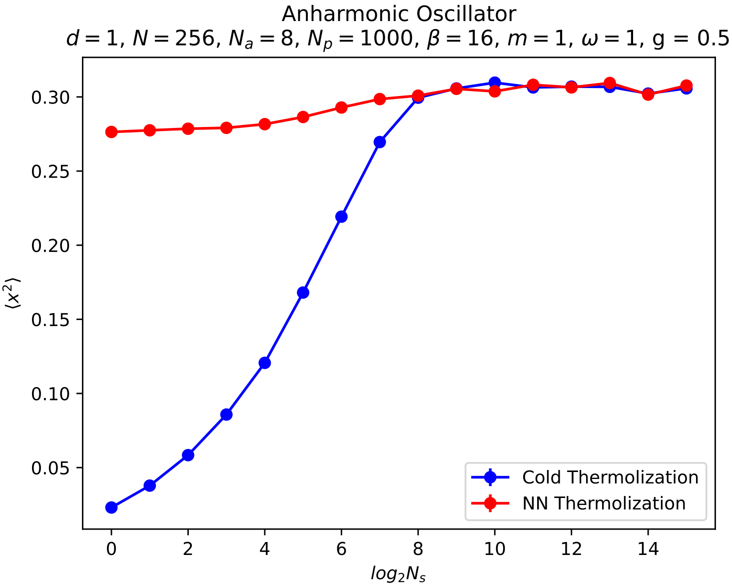
$$H = \frac{p^2}{2} + \frac{1}{2}\left(\left[e^{-\omega x} - 1\right]^2 - 1\right) \tag{43}$$



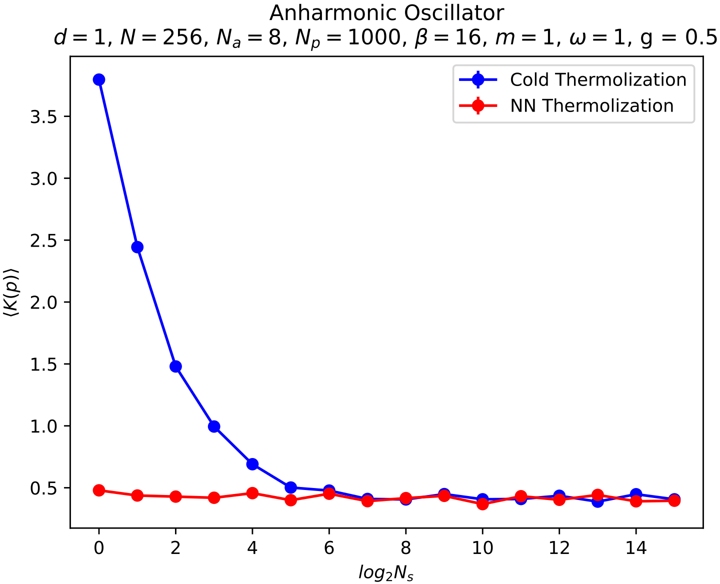
$$H = \frac{p^2}{2} + \frac{1}{2}\left(\left[e^{-\omega x} - 1\right]^2 - 1\right) \tag{44}$$



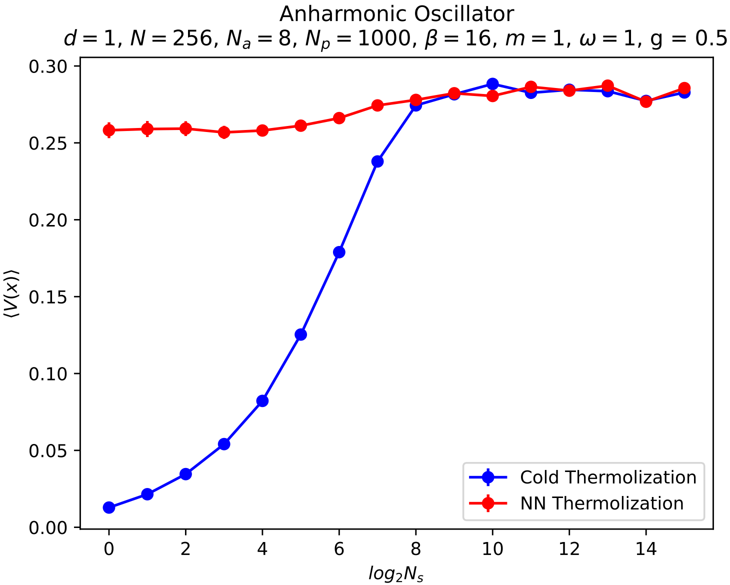
$$H = \frac{p^2}{2} + \frac{x^2}{2} + gx^4 \tag{45}$$



$$H = \frac{p^2}{2} + \frac{x^2}{2} + gx^4 \tag{46}$$

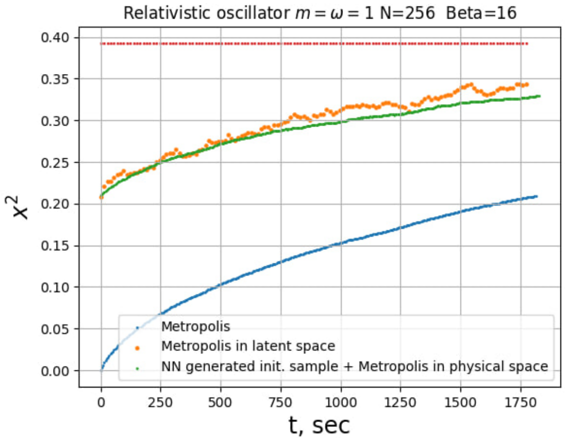


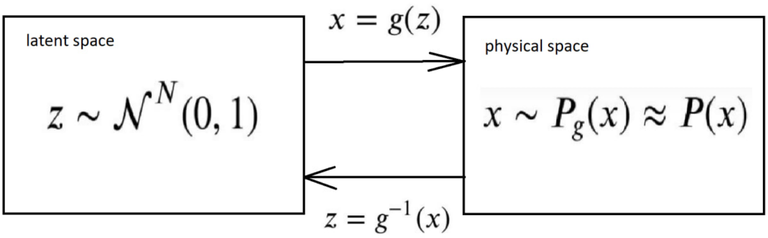
$$H = \frac{p^2}{2} + \frac{x^2}{2} + gx^4 \quad (47)$$



# MCMC in latent space

$$H = \sqrt{p^2 + m^2} - m + \frac{m\omega^2 x^2}{2} \quad (48)$$




latent space
$z \sim \mathcal{N}^N(0,1)$
$x = g(z)$
$z = g^{-1}(x)$
physical space
$x \sim P_g(x) \approx P(x)$

# Conclusion

- The use of neural networks makes it possible to speed up the calculation of functional integrals several times.
- The approach is universal: acceleration is observed for different models. This will allow it to be used for a wide range of tasks.
- The symmetry of the problem is taken into account, which may be especially important for applications to the theory of gauge fields.
- The artificial intelligence algorithms used are quite simple. It is expected to significantly improve the results by applying more sophisticated methods.

The research is supported by Basis Foundation. grant: 23-2-9-18-1