Application of neural networks for computing path integrals in quantum theory

Artyom Vasiliev[1], Alexandr Ivanov[1],
Dmitry Salnikov[1,2] and Vsevolod Chistiakov[1]

salnikov@inr.ru

[1]Lomonosov Moscow State University
[2]Institute for Nuclear Research of the Russian Academy of Science

The work of DS and VC is supported by «BASIS» Foundation grants:
№ 23-2-2-40-1 and № 23-2-9-18-1.

June 19, 2024

Introduction

- Path integral in imaginary time:

$$\langle O \rangle_\beta = \frac{1}{Z} \int D[x(t)]\, O[x(t)]\, e^{-S_E[x(t)]}, \quad x(0) = x(\beta). \tag{1}$$

- Gibbs average:

$$\langle O \rangle_\beta = \frac{1}{Z} \mathrm{Tr}\left[e^{-\beta H}\, O\right] = \frac{1}{Z} \sum_n \langle n|O|n\rangle e^{-\beta E_n}. \tag{2}$$

- Low temperature limit: $\lim\limits_{\beta\to\infty} \langle O \rangle_\beta = \langle 0|O|0\rangle$.
- Theory on the lattice:

$$D[x(t)] = \prod_n dx_n \equiv d^n x,\ \ S_E[x(t)] = S_E(x_1, ..., x_n) \equiv S_E(x). \tag{3}$$

- QFT generalization: $t \to x^\mu$, $x(t) \to \phi(x)$.

Monte Carlo calculation

- Average over a distribution:

$$\langle O \rangle_\beta = \int d^n x \, O(x) \, p(x) \simeq \frac{1}{N} \sum_{j=1}^{N} O\left(x^{(j)}\right), \tag{4}$$

where

$$\{x^{(j)}\} \sim p(x) = \frac{1}{Z} \, e^{-S_E(x)}. \tag{5}$$

- Path integral calculation = generation of the samples $\{x^{(j)}\}$ with target density distribution function.

Markov chain Monte Carlo (Metropolis)

- Starting with a sample of the «cold» trajectories:

$$X_0 = \{x^{(j)}\},\ x_i^{(j)} = 0. \tag{6}$$

- Random modification of the trajectories:

$$\tilde{x}_i^{(j)} = x_i^{(j)} + \sigma(\tau) u_i^{(j)}, \quad u_i^{(j)} \sim U[-1, 1]. \tag{7}$$

- Replacement the trajectories (j) in the sample X_0 with probabilities

$$\pi(y^{(j)}, x^{(j)}) = \min\left\{\frac{p(y^{(j)})}{p(x^{(j)})}, 1\right\} \tag{8}$$

  and obtain the new sample X_1.
- A sample chain $X_0 \to X_1 \to ... \to X_m$ has target density distribution function $p(x)$ in the limit.

Markov chain Monte Carlo (Metropolis)

- Disadvantages:
  - High computing costs and time are required;
  - Unable to take into account the physical symmetries.

Alternative approach

- Alternative approach: construction a map $x = g(z)$, $z = g^{-1}(x)$, where z is a set of random variables with a certain density distribution $\pi(z)$.
- Then target distribution $p(\cdot)$ and $g(\cdot)$ are connected as follows:

$$p(x) = \pi(z) \left| \det \frac{\partial z_i}{\partial x_j} \right| = \pi(g^{-1}(x)) \left| \det \frac{dg^{-1}}{dx} \right|. \tag{9}$$

- Neural Network as the map:

$$x^{(j)} = g(z^{(j)}; w), \tag{10}$$

where $z^{(j)} \sim N^n(0,1)$ and sample $\{x^{(j)}\}$ has target density distribution $p(x)$.

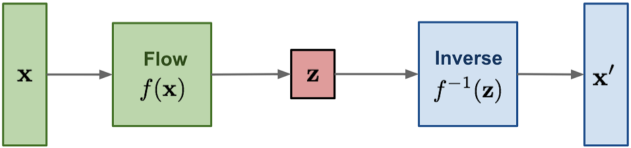


Figure 1 – Normalizing Flow scheme

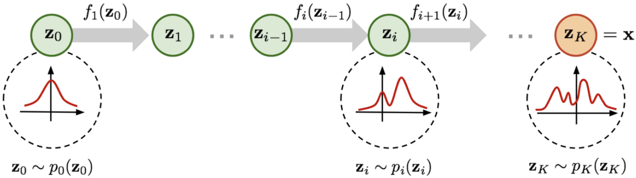


Figure 2 – Normalizing Flow architecture

Normalizing Flows

- The map g is a composition of affine transformations:

$$g = A_n \circ ... \circ A_1 \tag{11}$$

- We divide z on two parts: $z = u + v$, where, for example, u contains coordinates with even numbers, and v with the odd one.

$$A(u) = u, \quad [A(v)]_k = e^{\theta_{1k}(u)} v_k + \theta_{2k}(u), \quad \theta : \mathbb{R}^{n/2} \to \mathbb{R}^n. \tag{12}$$

- Loss function reads as follows:

$$\mathrm{Loss}(w) = D_{\mathrm{DL}}(p_g \,||\, p) - \ln Z = \int d^n x \, p_g(x)[\ln p_g(x) + S(x)]. \tag{13}$$

- Orthogonal transform $x = \mathcal{O} g(z)$ was applied to account the shift symmetry of the theory.

Models

- Euclidean action:

$$S_E(x_1, ..., x_n) = \tau \sum_{i=1}^{n} \left[K(x_i - x_{i-1}) + V(x_i)\right], \quad x_0 = x_n, \tag{14}$$

where $K(\cdot)$ and $V(\cdot)$ are kinetic and potential energies terms.

- Non-relativistic model:

$$H_{\text{kin}} = \frac{p^2}{2m}, \quad K(\xi) = \frac{m\xi^2}{2}. \tag{15}$$

- Relativistic model:

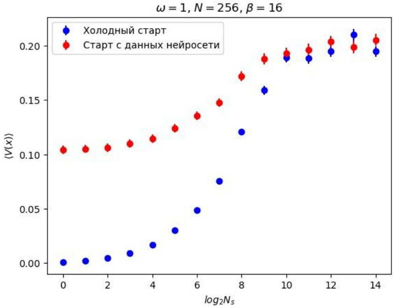
$$H_{\text{kin}} = \sqrt{p^2 + m^2} - m, \quad K(\xi) = \frac{1}{\tau} \ln\left[\frac{m\tau K_1(m\sqrt{\tau^2 + \xi^2})}{\pi\sqrt{\tau^2 + \xi^2}}\right]. \tag{16}$$

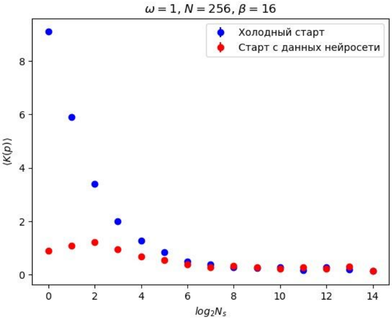
- Ultra-relativistic regime $m \to 0$:

$$H_{\text{kin}} = |p|, \quad K(\xi) = -\frac{1}{\tau} \ln\left[\pi(\tau^2 + \xi^2)/\tau\right]. \tag{17}$$

Relativistic Oscillator

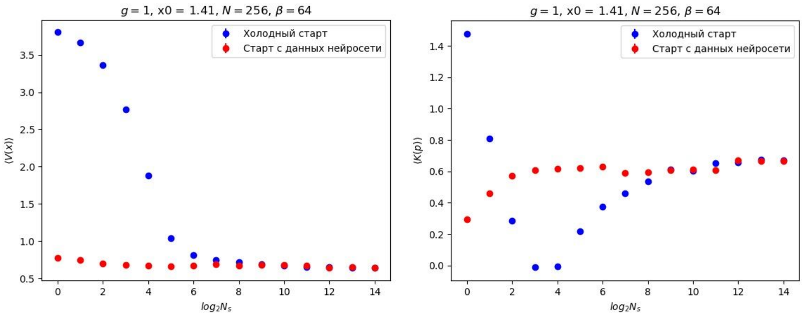
$$H = \sqrt{p^2 + m^2} - m + \frac{m\omega^2 x^2}{2} \tag{18}$$





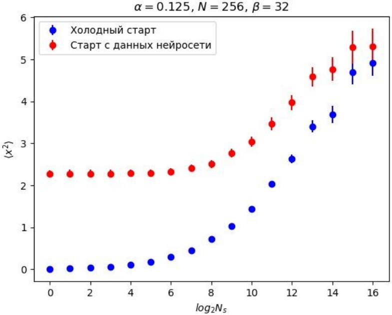
Relativistic double well

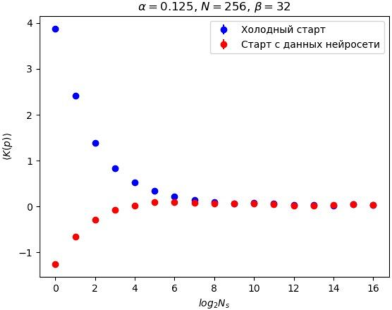
$$H = \sqrt{p^2 + m^2} - m + g(x^2 - x_0^2)^2 \tag{19}$$



Relativistic Morse

$$H = \sqrt{p^2 + m^2} - m + \frac{1}{2}\left[\left(e^{-\alpha x} - 1\right)^2 - 1\right] \tag{20}$$





Conclusions

- 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.

Thank you!