Deep Stochastic Mechanics

Elena Orlova, Aleksei Ustimenko, Ruoxi Jiang, Peter Y. Lu, and Rebecca Willett

ICML 2024

Problem Statement

The time-**dependent** Schrödinger equation (SE) for $0 < t \leq T$ and $\forall x \in \mathbb{R}^d$:

$$i\hbar\partial_t\psi(x,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\psi(x,t),$$

$$\psi(x,0) = \psi_0(x)\,.$$

$\psi : \mathbb{R}^d \times [0,T] \to \mathbb{C}$

wave function (unknown)

known constants

known potential function
(describes physics)

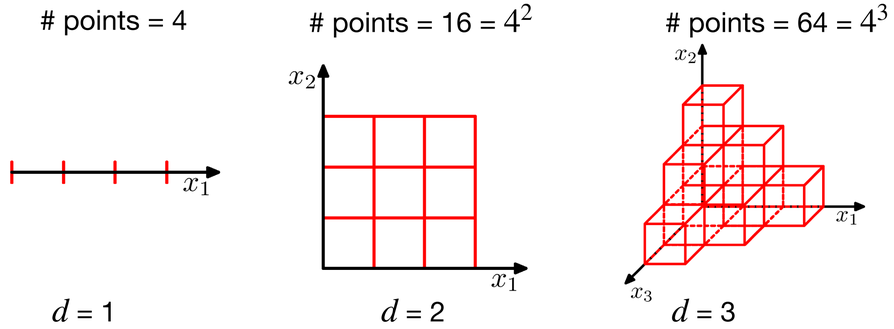
The **probability density** of finding a particle at position x at time t is

$$\rho(x,t) = |\psi(x,t)|^2.$$

GOAL: given $\psi_0(x)$, draw samples from $|\psi(x,t)|^2$ for $t \in (0,T]$

Solving Schrödinger Equation

- **Classical numerical solvers** and **Physics-informed neural networks (PINN)**: solve SE on a grid of points → **grid size grows exponentially** with the number of particles d



- **Variational Monte Carlo (VMC) methods**: reformulate solving SE as minimization of an energy functional
  - Great NNs for time-**in**dependent SE (FermiNet, PauliNet)
  - **t-VMC**: parametrizes ψ with parameters θ (ansatz), and solves the corresponding optimization problem to find θ
    - **Requires carefully designed ansatzes** to work well, but struggles with complex parametric forms (e.g. NNs)
    - Does expensive **MCMC sampling for every time step**

$$
\begin{aligned}
E(\theta) &= \int \Psi_\theta(x)^* H \Psi_\theta(x) \mathrm{d}x \\
&= \int |\Psi_\theta(x)|^2 \frac{H\Psi_\theta(x)}{\Psi_\theta(x)} \mathrm{d}x \\
&= \mathbb{E}_{x\sim|\Psi_\theta(x)|^2} \left[\frac{H\Psi_\theta(x)}{\Psi_\theta(x)} \right]
\end{aligned}
$$

Raissi, Perdikaris, & Karniadakis, 2019
Pfau, Spencer, Matthews, & Foulkes, 2020

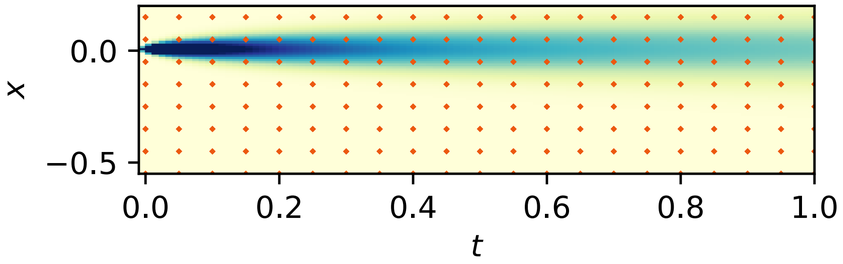
Our Approach

Based on stochastic formulation of quantum mechanics, our approach:

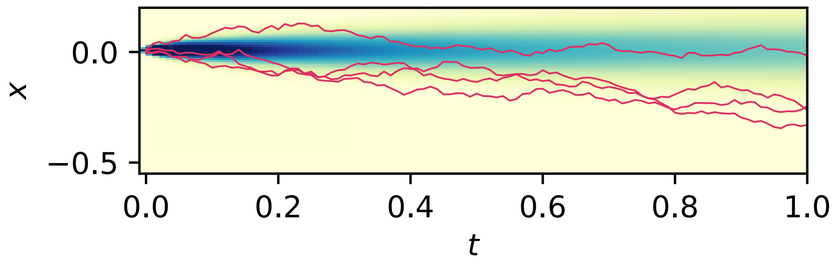
- **Parameterizes the gradients of the wave functions** rather than the density, analogous to score-based diffusion model.

- **Efficiently samples** from the density $\rho(x,t) = |\psi(x,t)|^2$ using an SDE, **side-stepping the curse of dimensionality** and **avoiding expensive MCMC sampling**.

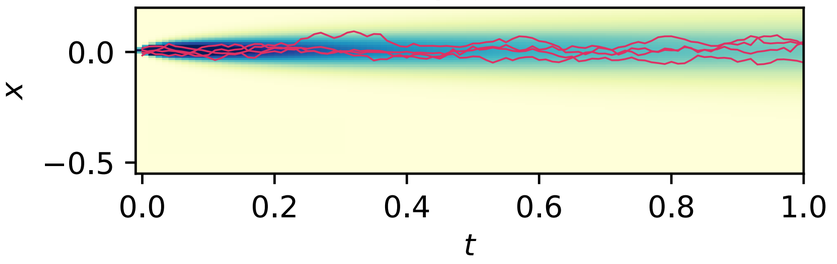
Uniform collocation points



DSM samples from $|\psi(x,t)|^2$, early epochs



DSM samples from $|\psi(x,t)|^2$, final epochs



Nelson's Stochastic Mechanics

Wave function:

$$\psi(x,t) = \sqrt{\rho(x,t)}\, e^{iS(x,t)}$$

Schrödinger equation:

$$i\hbar\partial_t\psi(x,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\psi(x,t),$$
$$\psi(x,0) = \psi_0(x)\,.$$

Sampling (MCMC) from $|\psi(\,\cdot\,,t)|^2$.

Equivalence

(if ψ satisfies SE)

Velocities:

current velocity: $$v(x,t) = \frac{\hbar}{m}\nabla S(x,t),$$

osmotic velocity: $$u(x,t) = \frac{\hbar}{2m}\nabla \log \rho(x,t)\,.$$

Madelung equations:

$$\partial_t u = -\nabla\langle u, \nabla u\rangle - \frac{\hbar}{2m}\nabla\left(\operatorname{div} v\right),$$

$$\partial_t v = -\frac{1}{m}\nabla V + \langle u, \nabla u\rangle - \langle v, \nabla v\rangle + \frac{\hbar}{2m}\nabla\left(\operatorname{div} u\right)$$

Diffusion process $X(t) \in \mathbb{R}^d$:

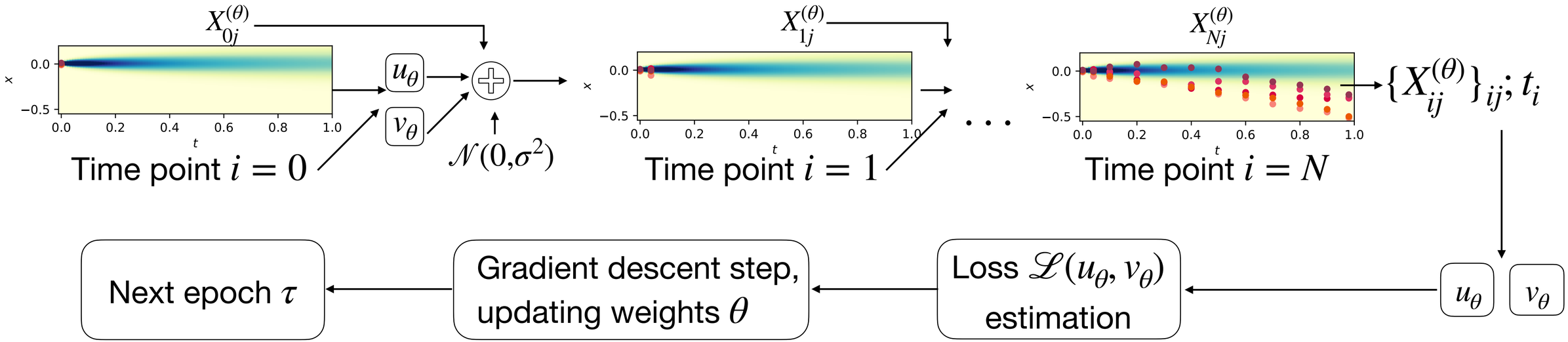
$$\mathrm{d}X(t) = \Big(v\big(X(t),t\big) + u\big(X(t),t\big)\Big)\mathrm{d}t + \sqrt{\frac{\hbar}{m}}\mathrm{d}W,$$

$$X(0) \sim |\psi_0|^2$$

Efficient sampling: $X(t) \sim |\psi(\,\cdot\,,t)|^2$ **for all** t.

Nelson, 1966

Deep Stochastic Mechanics (DSM)



Parametrize u and v via NNs, yielding a new process $X^\theta(t) \in \mathbb{R}^d$ that approximates $X(t)$.
How to train it?

- Define **loss $\mathscr{L}$ from Madelung equations** and **initial conditions** on u_θ and v_θ at $t = 0$
- **Generate trajectories** $X_{i+1}^\theta = X_i^\theta + \left(v_\theta(X_i^\theta, t_i) + u_\theta(X_i^\theta, t_i)\right)\epsilon + \xi,\ \xi \sim \mathcal{N}(0,\sigma^2)$**, evaluate** $\mathscr{L}$ and **update the models' weights,** repeat for every epoch

Experiments: Two Interacting Bosons

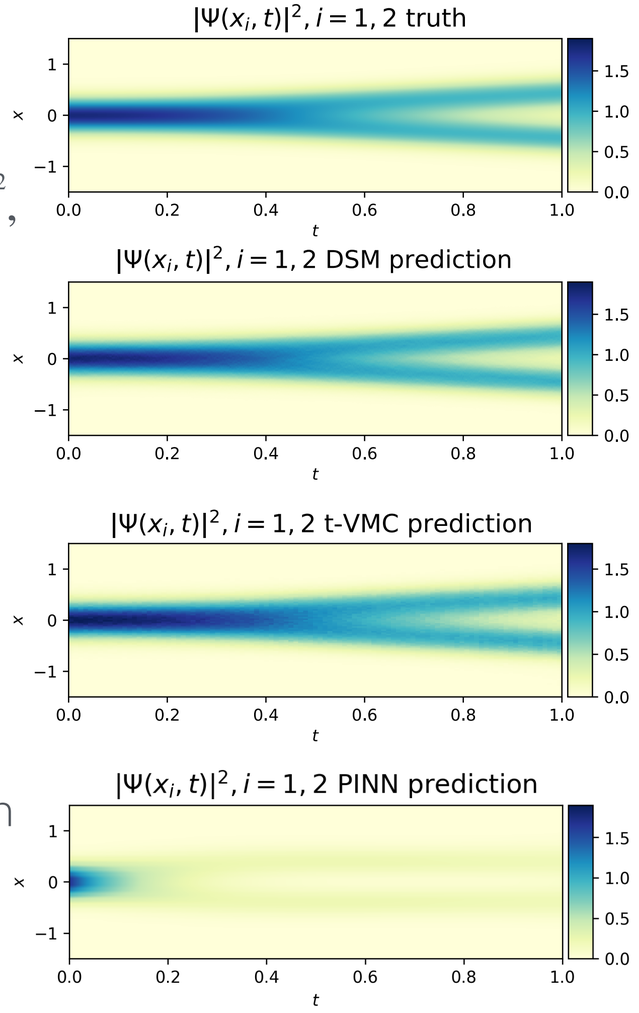
Interacting bosons in a harmonic potential:

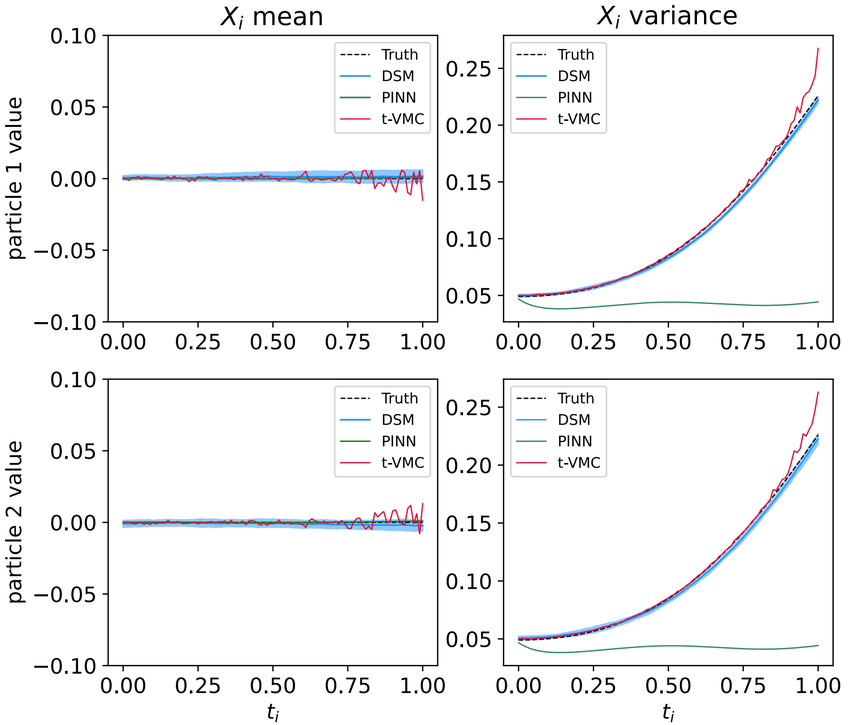
$$V(x,t) = \sum_i \frac{1}{2} m\omega^2 x_i^2 + \frac{1}{2} g \sum_{i,j} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i - x_j)^2/2\sigma^2},$$

$$\psi(x,0) = e^{-\omega^2 x^2/(2\hbar)}.$$

DSM and t-VMC perform reasonably well. What about higher dimensions?

- DSM/PINN: a feed-forward linear model with skip connections and tanh activations.
- t-VMC ansatz: Hermite polynomials with two-body interaction terms.

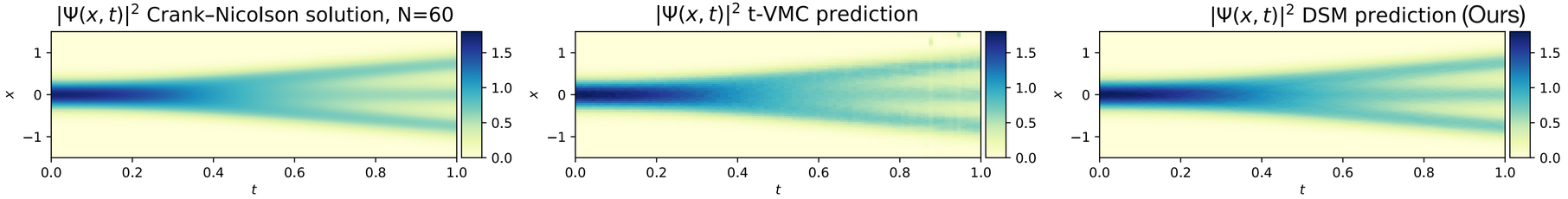




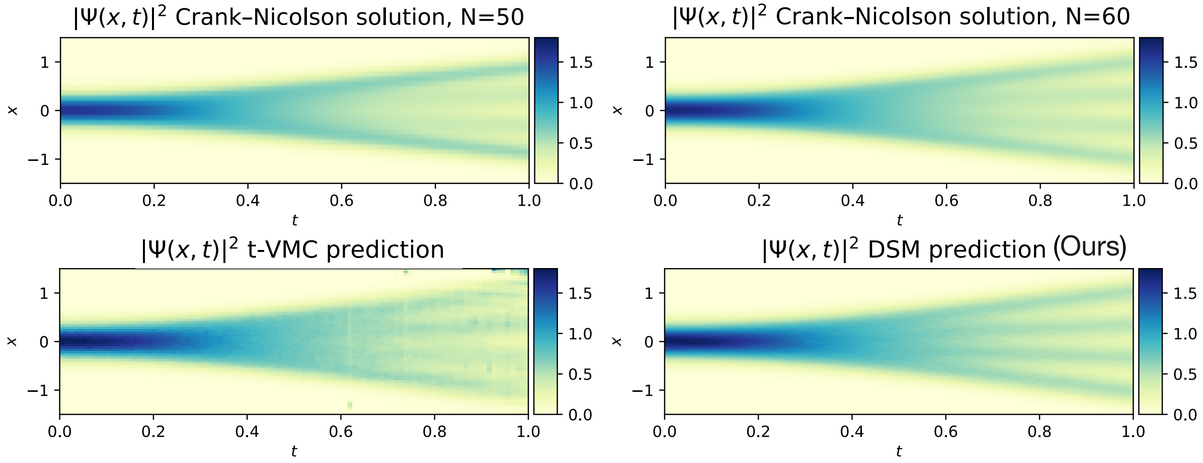
Results: More Interacting Bosons

$d = 3,4,5$ interacting bosons

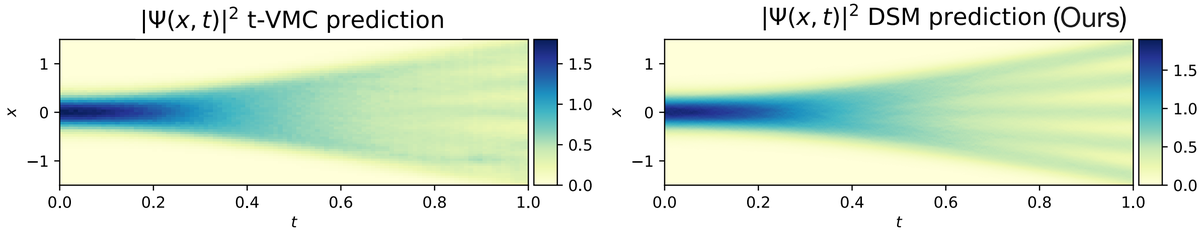
- **t-VMC performance deteriorates for larger d** (likely due the lack of higher-order interactions in the ansatz) while our method follows the ground truth
- NN-based parameterization of t-VMC did not yield satisfactory results



a) Three particles



b) Four particles



c) Five particles

** N is the number of discretization points in every spatial dimension.*

Results: Scaling to Many Particles

Computational complexity

Time *and* ***memory*** *usage of the* ***Crank-Nicolson method***

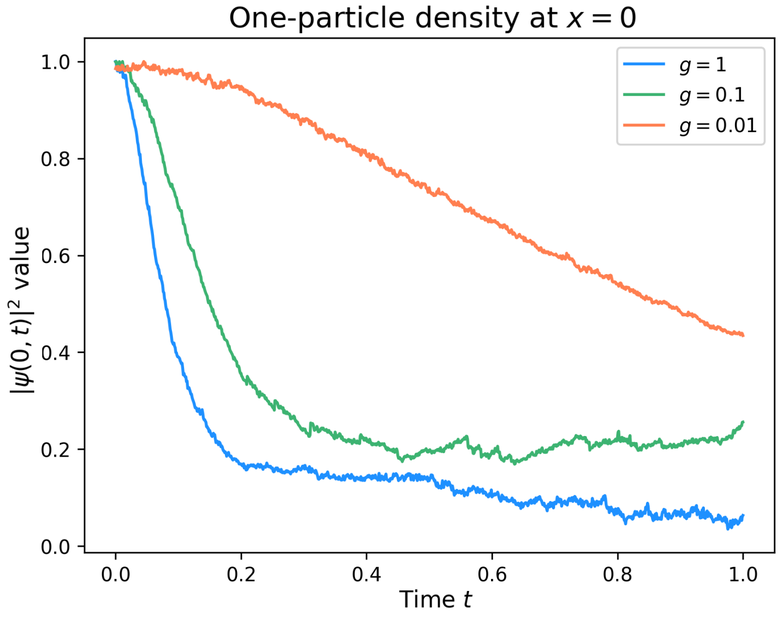
	$d=2$	$d=3$	$d=4$
Time (s)	0.75	35.61	2363
Memory (Gb)	7.4	10.6	214

Time *and* ***memory*** *usage of* ***our method***

	$d=2$	$d=3$	$d=4$	$d=5$
Training time (s)	1770	3618	5850	9240
Time per epoch (s/ep)	0.52	1.09	1.16	1.24
Memory (Gb)	17.0	22.5	28.0	33.5

DSM memory usage and time per epoch grow ~linearly in d, in contrast to the classical numerical solver

100 interacting particles



No ground truth. DSM results are qualitatively reasonable for different interacting strength g.
(higher for a stable system with small g;
lower for moving apart particles with higher g)

Conclusions

- Developed the **new efficient computational method** for simulating quantum dynamics based on **Nelson's stochastic mechanics**
  - Adaptive to latent low-dimensional support of density: efficient sampling and avoiding the curse of dimensionality
- **Theoretical guarantees** (a bound between processes X and X^θ) for our DSM method are given in our paper
- The experiments show **better performance** of our method compared to the numerical solvers/PINNs/t-VMC both in terms of prediction quality and computation time

Thank you for your attention!

https://elena-orlova.github.io/dsm/