

Article

Not peer-reviewed version

A Quantum-Inspired Machine Learning Based on the Heisenberg Uncertainty

[Phuong-Nam Nguyen](#) *

Posted Date: 31 July 2025

doi: 10.20944/preprints202507.2607.v1

Keywords: Physics-informed Machine Learning; Neural Networks; Quantum Mechanics; Mathematical Modeling; Simulation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

A Quantum-Inspired Machine Learning Based on the Heisenberg Uncertainty

Phuong-Nam Nguyen ^{1,2} ¹ The Laboratory of Quantum Algorithms and Advanced Analytics (QAAA Lab)² Faculty of Information Technology, College of Technology, National Economics University, Hanoi, 100000, Vietnam

* Correspondence: namnp@neu.edu.vn; Tel.: +(84)388666801

Abstract

A quantum-inspired machine learning framework based on the Heisenberg uncertainty principle is presented in this article. First, we will give the geometric interpretation of the Heisenberg uncertainty principle. Second, the derived uncertainty relation is used to construct a machine learning model, named Heisenberg Bases. The computational experiments demonstrate that the proposed model's performance is comparable to that of the conventional FNN, GRN, RNN, and Transformer, as shown in three case studies. Moreover, a neural-based cell automata created from the Heisenberg bases is introduced and demonstrated in two use cases.

Keywords: Physics-informed Machine Learning, Neural Networks, Quantum Mechanics, Mathematical Modeling, Simulation

1. Introduction

Quantum Computing is a cutting-edge field that leverages the principles of quantum mechanics to perform computations far beyond the capabilities of classical computers. Quantum computers utilize qubits, which can exist in multiple states simultaneously (superposition) and are interconnected through quantum entanglement. This allows for parallel processing at an unprecedented scale, offering the potential to revolutionize fields like cryptography, optimization, and material science.

Physics-Informed Machine Learning (PIML) integrates physical laws and principles into machine learning models to enhance predictive capabilities and improve generalization. By incorporating constraints from fields like quantum mechanics, fluid dynamics, or thermodynamics, PIML ensures that models adhere to known physical phenomena, making them more robust and interpretable in applications such as scientific computing, engineering, and environmental modeling.

The rapidly evolving field of quantum machine intelligence can—somewhat humorously—be summarized into three major directions. The first involves implementing classical machine learning algorithms on quantum computers, aiming to harness quantum advantages such as parallelism and entanglement. The second focuses on hybrid approaches, where classical algorithms assist quantum processes. In my prior work, I developed a circuit architecture search algorithm called QES, which uses graphical representations to efficiently search for optimal quantum entangling layouts [1]. The third direction emphasizes the development of native quantum algorithms tailored for quantum systems. By exploiting the probabilistic nature of the quantum wavefunction, I proposed a Bayesian quantum neural network (BayesianQNN) for classification tasks [2], and later introduced a game-theoretic framework known as the Duality Game for modeling dynamical interactions [3]. This framework was further extended to address the problem of biomarker identification in medical applications. A key challenge in this area lies in data encoding for quantum systems, which led to further contributions focused on efficient quantum data representation [4,5].

The main theme of this article is Quantum-inspired machine learning (QIML), a paradigm where the physical laws of quantum mechanics constrain models. QIML differs from QML as the former

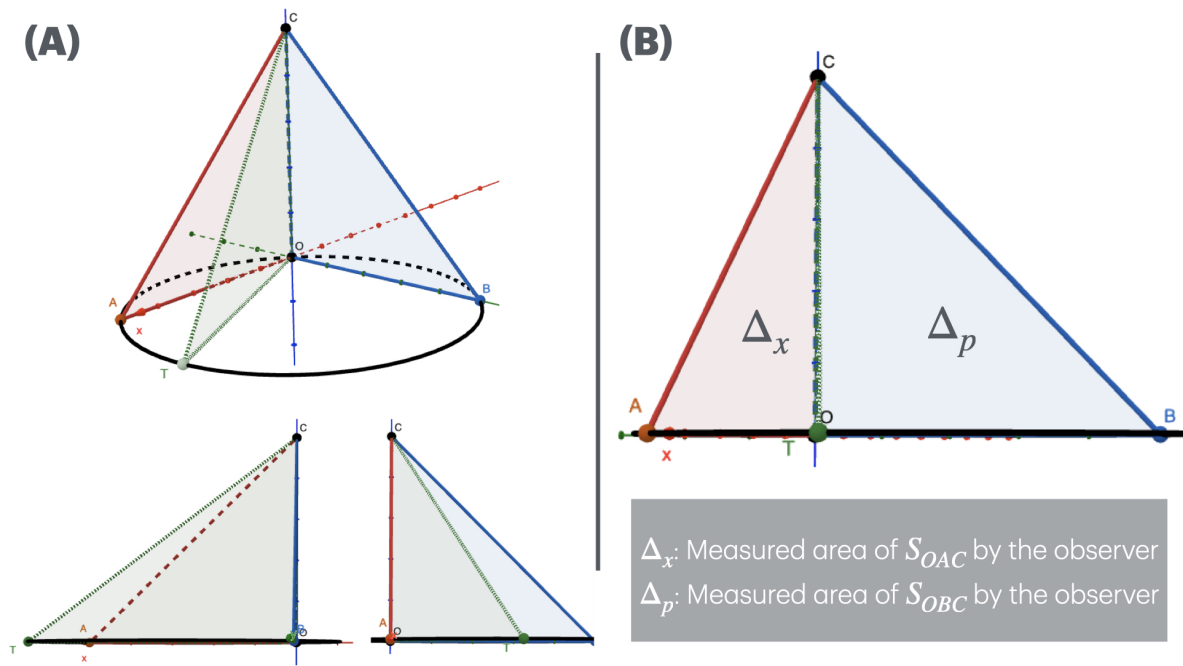


Figure 1. Geometric Interpretation of the Heisenberg Uncertainty

is a classical neural network inspired by quantum mechanics, while the latter is a machine learning algorithm on quantum computers. PIML differs from classical NN since the model weights (degrees of freedom) are constrained to represent the physics of quantum systems.

Our contribution is three-fold:

1. We will propose a new PIML framework that develops machine intelligence grounded in quantum physics;
2. We will develop AI models inspired by the Heisenberg uncertainty principle, referred to as HeisenbergBases;
3. The proposed model is powerful for representation learning, demonstrating superiority over classical neural network architectures, as numerically validated in three case studies.

The setting for numerical demonstration is designed for low-limited datasets, i.e., a small number of observations. The proposed model is generally compatible with or better than classical neural architectures such as FNN, GRN, RNN, and Transformer models.

This article is organized as follows:

- Section 2 introduces our proposed framework of QIML based on the Heisenberg uncertainty principle of quantum mechanics. We will start with the geometric interpretation of this uncertainty principle in Section 2.1. Then, we will give an implementation of AI models using our derivatives in Section 2.4.
- Section 3 reports the performance of our model compared with FNN, GRN, RNN, and Transformer models in three case studies: (1) quantum state learning, (2) estimation of temperature-solubility in *Al* and *Cu*, (3) spectral classification of materials. We also introduce how to create a cell automaton from the Heisenberg basis model.
- Section 5 discusses the link of our derivation to the original Heisenberg picture. Then, we will emphasize how our proposed model differs from the classical NN. Finally, several future directions are discussed.

2. The Proposed Framework

2.1. Geometric Interpretation of the Heisenberg Uncertainty

Given a pyramid in $Oxyz$ coordinates with edge length $OA = OB = OC = u$. Imagine an observer moves along the side AB looking straight into the origin O , and their position on AB is a point T (Figure 1(A), top panel). The observation plane is OCT . In this setting, the observer can measure the area of the two triangles

$$\begin{aligned} S_{OAC} &= \frac{1}{2} OA * OC = 0.5u \times OB \\ S_{OBC} &= \frac{1}{2} OB * OC = 0.5u \times OA \end{aligned} \quad (1)$$

The perceived area by the observers changes due to the coordinate of T . For example in the bottom panel of Figure 1(A):

- $T \equiv A$, the observer fully see the plane Oyz and so the area of Δ_p is fully measured. When the observer is at point A , they have a complete view of triangle OBC , meaning that S_{OBC} is fully measured while S_{OAC} is not.
- $T \equiv B$, the observer fully see the plane Oxz and so the area of Δ_x is fully measured. When the observer is at point B , they fully see triangle OAC , meaning S_{OAC} is fully measured, but S_{OBC} is not.
- When T moving from A to B , the area of Δ_x increases while that of Δ_p decreases.

The observer cannot simultaneously fully measure Δ_p and Δ_x . As the observer moves from A to B , the measured area of S_{OAC} increases while that of S_{OBC} decreases, much like the uncertainty relationship: a more precise measurement of one triangle corresponds to a less precise measurement of the other. This mirrors the idea of uncertainty in quantum systems, where the precise measurement of one variable (e.g., position) leads to increased uncertainty in the measurement of its complementary variable (e.g., momentum). In the geometrical analogy, the observer's inability to fully measure both areas simultaneously represents the inherent trade-off between precision in complementary measurements.

2.2. Derivation of Parameterized Uncertainty

Let $\mathcal{M}$ be the manifold parameterized by (x, y, t) denoted as $M(x, y, t)$. We consider $M(x, p, t)$ to be linked with the Heisenberg uncertainty. We denote

- $\phi_{AC} = \phi(x, t)$ as a function obtained by project M onto Oxz .
- $\psi_{BC} = \psi(p, t)$ as a function obtained by project M onto Oyz .

The area under the curve $\phi(x, t)$ and $\psi(p, t)$ are the area of Δ_x and Δ_p ; respectively, measured by the observers (Figure 1(B)). We have

$$\begin{aligned} \Delta_x &= \int_{\Omega_x} \phi(x, t) dt \\ \Delta_p &= \int_{\Omega_y} \psi(p, t) dt \end{aligned} \quad (2)$$

When $OA = OB$, the upper bound of the integrals is the same, which implies $\Omega_x = \Omega_y = \Omega$. We can rewrite

$$\Delta_x + \Delta_p = \int_{\Omega} \phi(x, t) dt + \int_{\Omega} \psi(p, t) dt \quad (3)$$

Let

$$\begin{aligned} \phi(x, t) &= x^2(t) \kappa(x|p, t) \\ \psi(p, t) &= p^2(t) \kappa(p|x, t) \end{aligned} \quad (4)$$

where $\kappa(x, p, t)$ is the multivariate distribution of (x, p, t) . The conditional probability of x (position) and p (momentum) being measured are $\kappa(x|p, t)$ and $\kappa(p|x, t)$, respectively. Plugging-in Eq. 3, we have

$$\begin{aligned}\eta &:= \Delta_x + \Delta_p \\ \eta &= \int_{\Omega} x^2(t) \kappa(x|p, t) dt + \int_{\Omega} p^2(t) \kappa(p|x, t) dt\end{aligned}\quad (5)$$

Here, we assume that $x(t)$ and $p(t)$ are random variables that change w.r.t time t . Besides, we set

$$\begin{aligned}(A_1) : \kappa(x|t) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2(t)}{2}\right) \\ (A_2) : \kappa(p|t) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{p^2(t)}{2}\right) \\ (A_3) : \kappa(t) &= \frac{1}{u} \quad \text{for } t \in [0, u]\end{aligned}\quad (6)$$

(A_1) and (A_2) assume that x and p are random variables from $\Phi(0, 1)$ and (A_3) assumes t is from $U[0, u]$. If $x(t)$ and $p(t)$ are independent, their joint distribution is

$$\kappa(x, p|t) = \frac{1}{2\pi} \exp\left(-\frac{x^2(t)}{2}\right) \exp\left(-\frac{p^2(t)}{2}\right) \quad (7)$$

Moreover, x and p are depend on t by relations $x(t)$ and $p(t)$. Using the Bayesian rule, we obtain

$$\kappa(x, p|t) = \frac{\kappa(x, p, t)}{\kappa(t)} = u \kappa(x, p, t) \quad (8)$$

or

$$\kappa(x, p, t) = \frac{1}{2u\pi} \exp\left(-\frac{x^2(t)}{2}\right) \exp\left(-\frac{p^2(t)}{2}\right). \quad (9)$$

Similarly, we have

$$\begin{aligned}\kappa(x, t) &= \kappa(x|t) \kappa(t) = \frac{1}{u\sqrt{2\pi}} \exp\left(-\frac{x^2(t)}{2}\right) \\ &= \frac{\gamma}{u} A(x, t) \\ \kappa(p, t) &= \kappa(p|t) \kappa(t) = \frac{1}{u\sqrt{2\pi}} \exp\left(-\frac{p^2(t)}{2}\right) \\ &= \frac{\gamma}{u} A(p, t)\end{aligned}\quad (10)$$

where $\gamma = 1/\sqrt{2\pi}$ and $A(o_1, o_2) = \exp\left(-\frac{o_1(o_2)^2}{2}\right)$. The conditional probability in Eq. 5 is

$$\begin{aligned}\kappa(x|p, t) &= \kappa(x, p, t) \kappa(p, t) = \frac{\gamma^3}{u^2} A(x, t) A^2(p, t) \\ \kappa(p|x, t) &= \kappa(x, p, t) \kappa(x, t) = \frac{\gamma^3}{u^2} A^2(x, t) A(p, t)\end{aligned}\quad (11)$$

We have

$$\begin{aligned}
 \Delta_x &= \int_0^u x^2(t) \frac{\gamma^3}{u^2} A(x, t) A^2(p, t) \\
 &= \frac{\gamma^3}{u^2} \int_0^u [A(x, t) A(p, t)] A(p, t) x^2(t) \\
 \Delta_p &= \int_0^u p^2(t) \frac{\gamma^3}{u^2} A^2(x, t) A(p, t) \\
 &= \frac{\gamma^3}{u^2} \int_0^u [A(x, t) A(p, t)] A(x, t) p^2(t) \\
 A(x, t) A(p, t) &= 2u\pi\kappa(x, p, t) \\
 \Rightarrow \Delta_x &= \frac{2\pi\gamma}{u} \int_0^u \kappa(x, p, t) A(p, t) x^2(t) dt \\
 \Rightarrow \Delta_p &= \frac{2\pi\gamma}{u} \int_0^u \kappa(x, p, t) A(x, t) p^2(t) dt
 \end{aligned} \tag{12}$$

Simplifying the term Δ_x and Δ_p , we have

$$\begin{aligned}
 \Delta_x &= \frac{\sqrt{2\pi}}{u} \int_0^u \kappa(x, p, t) A(p, t) x^2(t) dt \\
 \Delta_p &= \frac{\sqrt{2\pi}}{u} \int_0^u \kappa(x, p, t) A(x, t) p^2(t) dt
 \end{aligned} \tag{13}$$

Since u is just the unit of $Oxyz$ (recalled that $OA = OB = OC = u$), we can set $u = 1$ and the term

$$\begin{aligned}
 \Delta_x \Delta_p &= 2\pi \int_0^u \kappa(x, p, t) B_x dt \cdot \int_0^u \kappa(x, p, t) B_p dt \\
 B_x(t) &= A(p, t) x^2(t) \\
 B_p(t) &= A(x, t) p^2(t)
 \end{aligned} \tag{14}$$

Clearly, B_x and B_p are random variables, assumed that B_x and B_p from the same distribution $\kappa(x, p, t)$; then, the integral returns the expectation of E_x and E_p , says

$$\begin{aligned}
 \mathbb{E}(B_x) &= \int_0^u \kappa(x, p, t) B_x(t) dt \\
 \mathbb{E}(B_p) &= \int_0^u \kappa(x, p, t) B_p(t) dt
 \end{aligned} \tag{15}$$

As a result, we have

$$\Delta_x \Delta_p = 2\pi \mathbb{E}(B_x) \cdot \mathbb{E}(B_p). \tag{16}$$

and

$$\begin{aligned}
 \mathbb{E}(B_x) &= \int_0^u \frac{1}{2u\pi} \exp\left(-\frac{x^2(t) + p^2(t)}{2}\right) A(p, t) x^2(t) dt \\
 \mathbb{E}(B_p) &= \int_0^u \frac{1}{2u\pi} \exp\left(-\frac{x^2(t) + p^2(t)}{2}\right) A(x, t) p^2(t) dt
 \end{aligned} \tag{17}$$

which directly implies

$$\begin{aligned}
 \mathbb{E}(B_x) &= \int_0^u \frac{1}{2u\pi} \exp\left(-\frac{x^2(t) + 2p^2(t)}{2}\right) x^2(t) dt \\
 \mathbb{E}(B_p) &= \int_0^u \frac{1}{2u\pi} \exp\left(-\frac{2x^2(t) + p^2(t)}{2}\right) p^2(t) dt
 \end{aligned} \tag{18}$$

and

$$\begin{aligned}
 \Delta_x \Delta_p &= \frac{1}{u} \mathbb{E}(B_x) \cdot \mathbb{E}(B_p) \\
 &= \int_0^u C_1(x(t), p(t)) x^2(t) dt \cdot \int_0^u C_2(x(t), p(t)) p^2(t) dt \\
 C_1(x(t), p(t)) &:= \exp\left(-\frac{x^2(t) + 2p^2(t)}{2}\right), \\
 C_2(x(t), p(t)) &:= \exp\left(-\frac{2x^2(t) + p^2(t)}{2}\right)
 \end{aligned} \tag{19}$$

Applying the Cauchy-Schwarz inequality, we have

$$\Delta_x \Delta_p \geq \int_0^u x(t) p(t) \sqrt{C_1 C_2} dt, \tag{20}$$

where

$$\begin{aligned}
 C_1 C_2 &= \exp\left(-\frac{x^2(t) + 2p^2(t) + 2x^2(t) + p^2(t)}{2}\right) \\
 &= \exp\left(-\frac{3x^2(t) + 3p^2(t)}{2}\right) \\
 &= \exp\left(-\frac{3(x^2(t) + p^2(t))}{2}\right)
 \end{aligned} \tag{21}$$

and

$$\sqrt{C_1 C_2} = \exp\left(-\frac{3(x^2(t) + p^2(t))}{4}\right). \tag{22}$$

The normalization of $\sqrt{C_1 C_2}$ is computed by

- Compute the integral of $\exp\left(-\frac{3(x^2 + p^2)}{4}\right)$ over all x and p :

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{3(x^2 + p^2)}{4}\right) dx dp$$

This integral can be separated into two Gaussian integrals:

$$\left(\int_{-\infty}^{\infty} \exp\left(-\frac{3x^2}{4}\right) dx\right) \times \left(\int_{-\infty}^{\infty} \exp\left(-\frac{3p^2}{4}\right) dp\right)$$

Each integral is a Gaussian integral of the form:

$$\int_{-\infty}^{\infty} \exp(-\alpha x^2) dx = \sqrt{\frac{\pi}{\alpha}}$$

- Here, $\alpha = \frac{3}{4}$. Therefore:

$$\int_{-\infty}^{\infty} \exp\left(-\frac{3x^2}{4}\right) dx = \sqrt{\frac{4\pi}{3}}$$

$$\int_{-\infty}^{\infty} \exp\left(-\frac{3p^2}{4}\right) dp = \sqrt{\frac{4\pi}{3}}$$

So, the total integral is:

$$\left(\sqrt{\frac{4\pi}{3}}\right) \times \left(\sqrt{\frac{4\pi}{3}}\right) = \frac{4\pi}{3}$$

- To normalize $\sqrt{C_1 C_2}$, divide it by this integral:

$$\text{Normalized } \sqrt{C_1 C_2} = \frac{\exp\left(-\frac{3(x^2 + p^2)}{4}\right)}{\frac{4\pi}{3}}$$

Simplify the fraction:

$$\text{Normalized } \sqrt{C_1 C_2} = \frac{3}{4\pi} \exp\left(-\frac{3(x^2 + p^2)}{4}\right)$$

We denote $\text{Normalized } \sqrt{C_1 C_2} = \Psi(x, p, t)$, then

$$\exp\left(-\frac{3(x^2 + p^2)}{4}\right) = \frac{4\pi}{3} \Psi(x, p, t) \quad (23)$$

and the Eq. 20 becomes

$$\begin{aligned} \Delta_x \Delta_p &\geq \int_0^u x(t) p(t) \sqrt{C_1 C_2} dt \\ &= \frac{4\pi}{3} \int_0^u x(t) p(t) \Psi(x, p, t) dt. \end{aligned} \quad (24)$$

Of note, the multivariate probability Ψ is the density function due to the normalization, i.e., it is some joint distribution of (x, p, t) . Thus, the integral can be given as

$$\Delta_x \Delta_p \geq \frac{4\pi}{3} \mathbb{E}_{\Psi}[x(t)p(t)] \quad (25)$$

Geometric Relation Between $x(t)$ and $p(t)$: The quantities $x(t)$ and $p(t)$ can be projected onto Oxy , say the projected point of $H(x, 0, u)$ and $Q(0, p, u)$ are $P(x, 0, 0)$ and $N(p, 0, 0)$. Besides,

- $|OP| = (1 - \alpha(t))u$ and
- $|ON| = \alpha(t)u$, where $\alpha(t) \in [0, 1]$.

Therefore,

$$x(t)p(t) = u^2 \alpha(t)(1 - \alpha(t)) \quad (26)$$

$$\begin{aligned} \Delta_x \Delta_p &\geq \int_0^u u^2 \alpha(t)(1 - \alpha(t)) \sqrt{C_1 C_2} dt \\ &= u^2 \int_0^u \alpha(t)(1 - \alpha(t)) \exp\left(-\frac{3}{4}r^2\right) dt \\ r &= x^2(t) + p^2(t) \end{aligned} \quad (27)$$

Given the expressions for $x(t)$ and $p(t)$ as:

$$x(t) = (1 - \alpha(t))u, \quad p(t) = \alpha(t)u,$$

We denoted the **modified Heisenberg relation** $\Delta_x \Delta_p(t, \alpha, u)$, given as

$$\begin{aligned} \Delta_x \Delta_p(t, \alpha, u) &= u^2 \int_0^u \alpha(t)(1 - \alpha(t)) D(t, \alpha) dt, \\ D(t, \alpha) &= \exp\left(-\frac{3u^2}{4} [\alpha(t)^2 + (1 - \alpha(t))^2]\right) \end{aligned} \quad (28)$$

where:

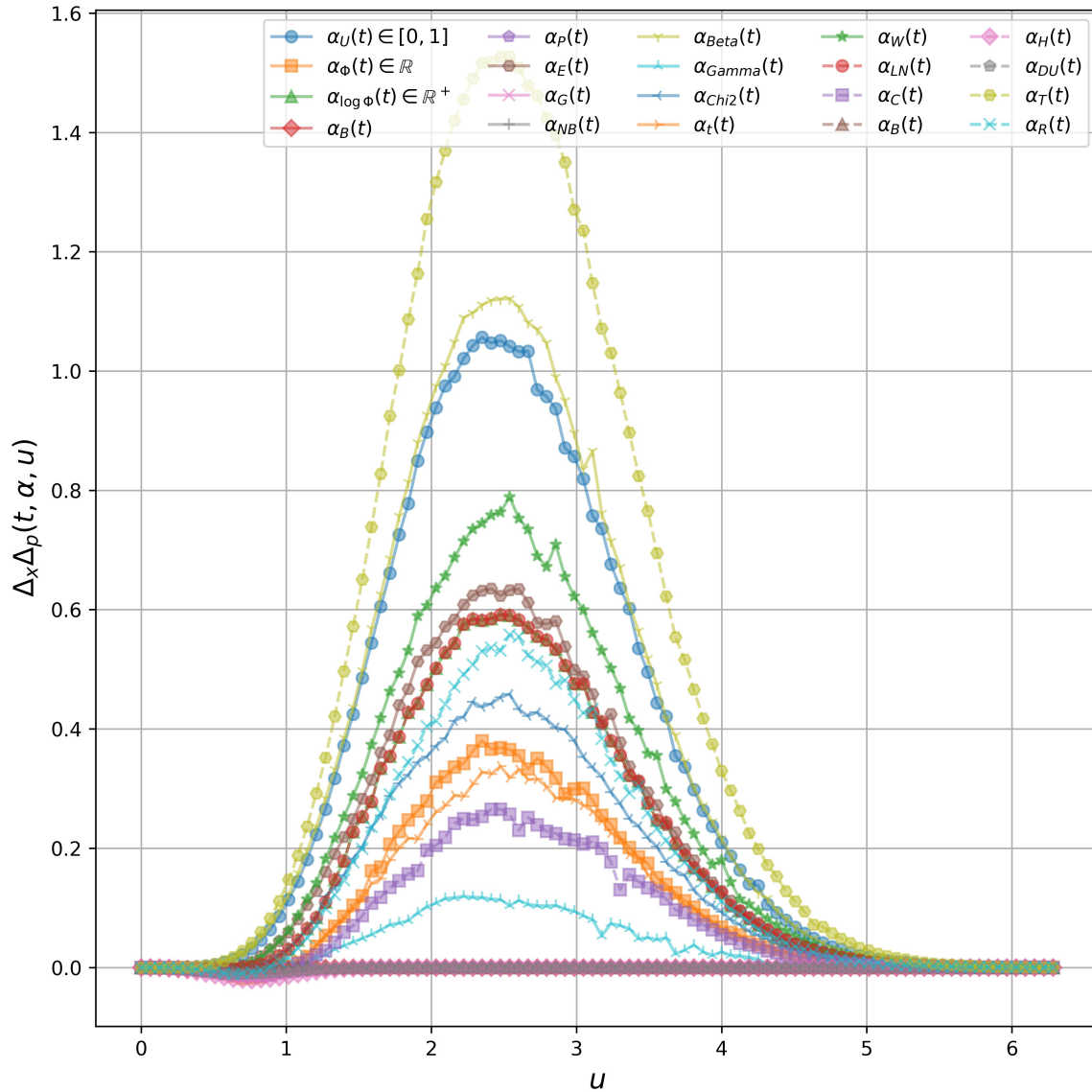


Figure 2. The value of $\Delta_x \Delta_p(t, \alpha, u)$ from our proposed probabilistic sampler

- $\alpha(t)$ is a bounded function in $[0, 1]$,
- $t \in [0, u]$

2.3. Our Postulation

In the Heisenberg uncertainty, the product $\Delta_x \Delta_p \geq \hbar/2$ is a constant lower bound. Here, we postulate that this product is a function of time t , the time-dependent action $\alpha(t)$ bounded in $[0, 1]$, and unit of measurements u . We test seven bounded function for α , given in Figure 3, including $\sin, \cos, \tanh, \exp(-t^2), \sinh, \cosh$ and $\alpha \in U[0, 1]$ (uniformly distributed). There are three main observations:

1. Not all bounded function α gives a physical sense. Specifically, using $\alpha_6(t) = \cosh(\pi t)$ result in negative $\Delta_x \Delta_p$. In fact, $\Delta_x > 0$ and $\Delta_p > 0$.
2. There exist classes of functions with positive $\Delta_x \Delta_p$. Specifically, other α functions (except α_6) satisfies the above physical constraint, so as scaled α functions by a positive scalar.
3. When α in a random variable from the uniform distribution $[0, 1]$, we find that $\Delta_x \Delta_p$ not only has lower bound $\hbar/2$ as the Heisenberg uncertainty principle but also has the upper bound, showed as α_U in Figure 3.

These findings contribute to understanding how bounded time-dependent actions influence the uncertainty product, showing that specific classes of functions for $\alpha(t)$ can maintain the physical constraints of the Heisenberg uncertainty principle while revealing the existence of an upper bound in certain cases.

2.4. The Proposed Model HeisenbergBases

This section will introduce a neural network model based on the class of functions $\Delta_x \Delta_p(t, \alpha, u)$. It is observed that the values of the Δ function depend on the generating probability distribution for $\alpha(t)$. We create the array

$$\Lambda_{N_b \times d} = [\Delta_x \Delta_p(\alpha_1), \Delta_x \Delta_p(\alpha_2), \dots, \Delta_x \Delta_p(\alpha_{N_b})], \quad (29)$$

using the same values of t and u . The summary of all generating distribution functions α is reported in Table 1.

Table 1. The generating distribution $\alpha(t)$ used in our HeisenbergBases

Distribution Name	Mathematical Eqtion
Uniform	$f(x) = 1, 0 \leq x \leq 1$
Normal	$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$
Binomial	$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$
Poisson	$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}$
Exponential	$f(x) = \frac{1}{\text{scale}} e^{-x/\text{scale}}$
Geometric	$f(x) = p(1-p)^x$
Negative Binomial	$f(x) = \binom{n+x-1}{x} p^n (1-p)^x$
Beta	$f(x) = \frac{x^{a-1} (1-x)^{b-1}}{B(a, b)}$
Gamma	$f(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-x/\theta}$
Chi-squared	$f(x) = \frac{1}{2^{k/2} \Gamma(k/2)} x^{(k/2)-1} e^{-x/2}$
Student's t	$f(x) = \frac{\Gamma\left(\frac{df+1}{2}\right)}{\sqrt{df\pi} \Gamma\left(\frac{df}{2}\right)} \left(1 + \frac{x^2}{df}\right)^{-\frac{df+1}{2}}$
Weibull	$f(x) = cx^{c-1} e^{-x^c}$
Lognormal	$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2 / (2\sigma^2)}$
Cauchy	$f(x) = \frac{1}{\pi \text{scale} \left(1 + \left(\frac{x - \text{loc}}{\text{scale}}\right)^2\right)}$
Bernoulli	$f(x) = p^x (1-p)^{1-x}$
Hypergeometric	$f(x) = \frac{\binom{n_{\text{good}}}{x} \binom{n_{\text{bad}}}{n_{\text{sample}} - x}}{\binom{n_{\text{good}} + n_{\text{bad}}}{n_{\text{sample}}}}$
Discrete Uniform	$f(x) = \frac{1}{\text{high} - \text{low} + 1}$
Triangular	$f(x) = \begin{cases} \frac{2(x - \text{left})}{(\text{right} - \text{left})(\text{mode} - \text{left})}, & \text{if } \text{left} \leq x \leq \text{mode}, \\ \frac{2(\text{right} - x)}{(\text{right} - \text{left})(\text{right} - \text{mode})}, & \text{if } \text{mode} \leq x \leq \text{right}. \end{cases}$
Rayleigh	$f(x) = \frac{x}{\text{scale}^2} e^{-x^2/(2\text{scale}^2)}$

Given an input matrix $\mathbf{X}_{N \times D} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D]$ where N is the number of samples and D is the number of features, we compute the embedding

$$\mathbf{h}_{N \times D} = \frac{1}{D} \sum_i^D \frac{1}{N_b} \sum_j^{N_b} \Delta_x \Delta_p(\alpha_j(t)) \cdot \mathbf{x}_i. \quad (30)$$

The classification layer is a common MLP $f_{\Theta}(\mathbf{h})$, which gives the prediction $\hat{\mathbf{y}} = f_{\Theta}(\mathbf{h})$. We name this neural network HeisenbergBases since it is based on our derived uncertainty constraint.

2.5. Quantum State Learning

The Wigner function for a coherent state is given by:

$$W(q, p; \lambda) = \frac{2}{\pi} \exp\left(-2\left((q - \text{Re}(\alpha))^2 + (p - \text{Im}(\lambda))^2\right)\right) \quad (31)$$

The Wigner function for the superposition of two coherent states (Schrödinger's cat state) is defined as:

$$W_{\text{cat}}(q, p; \lambda) = \frac{1}{2}(W(q, p; \lambda) + W(q, p; -\lambda)) + 2 \exp(-2(q^2 + p^2)) \cos(4p \text{Re}(\lambda)) \quad (32)$$

We use $\lambda = 1$ for the synthesized data. The performance of different models is compared using hypothesis tests with the quantum cat state learning task, reported in Fig 4.

For FNN vs. HeisenbergBases, the p-value is 0.1398, which is higher than 0.05, indicating no significant difference in performance between these models. In the comparison of GRN vs. HeisenbergBases, the p-value is 0.6504, again showing no significant difference. However, for RNN vs. HeisenbergBases, the p-value is 0.0206, which is below the 0.05 threshold, indicating that HeisenbergBases performs significantly better than the RNN model. Lastly, the Transformer vs. HeisenbergBases comparison yields a p-value of 0.3082, indicating no significant difference in their performance. HeisenbergBases shows relatively competitive performance with FNN and GRN across several loss functions. HeisenbergBases exhibits low variability (low standard deviation) compared to all other models (Fig 4(B)). It has a higher variance than GRN across most loss functions, while RNN and Transformer show lower variance than HeisenbergBases. In AppTable 2, we have enough statistical evidence to show that HeisenbergBases's performance is better than RNN and Transformer, comparable to FNN and GRN.

2.6. Applications in Material Science

BigSolDB[7] contains 54,273 solubility values, 830 unique molecules, and 138 individual solvents within the temperature range from 243.15 to 403.15K at atmospheric pressure. The dataset consists of six columns, which are explained as follows:

- **SMILES** — SMILES representation of a dissolved compound
- **T,K** — Temperature in Kelvin
- **Solubility** — Experimental solubility value (mole fraction)
- **Solvent** — Name of the solvent
- **SMILES_Solvent** — SMILES of the solvent
- **Source** — A data source for the given values

We use the temperature and solubility values for the model evaluation.

2.6.1. Estimation of Temperature-Solubility Relation

In Table 3, the performance of different models is compared using hypothesis tests on the *BigSol* dataset. For FNN vs. HeisenbergBases, the p-value is 0.9995, indicating no significant difference in performance between these models. Similarly, the comparison of GRN vs. HeisenbergBases gives a p-value of 0.9999, showing no significant difference. However, for the comparison of HeisenbergBases vs. RNN, the p-value is 1.79×10^{-5} , which is below the 0.05 threshold, indicating that the HeisenbergBases model significantly outperforms the RNN. For Transformer vs. HeisenbergBases, the p-value is 0.0004, also indicating a significant difference, with the HeisenbergBases model performing better. Overall, HeisenbergBases is the best model in this case study, significantly outperforming all others.

2.6.2. Spectral Classifications of Materials

Spectral data from aluminum, stainless steel, mild steel, copper, and wood were gathered using a spectrometer with advanced on-chip filtering[8]. This spectrometer, featuring a compact 9x9mm array of up to 8 wavelength-selective photodiodes, divides the spectrum into eight bands ranging from 400 to 1100 nm. The goal is to use ML/DL models to learn useful patterns for each material. This evaluation only considers 100 observations of *Al*- and *Cu*-diode. HeisenbergBases shows the strongest performance overall, while other models like FNN, GRN, and RNN show minimal performance differences.

In Table 4, the performance of different models is compared using hypothesis tests on the *Al-diode* dataset. For FNN vs. HeisenbergBases, the p-value is 1.0000, showing no significant difference between the models. In the comparison of GRN vs. HeisenbergBases, the p-value is ≈ 1 , indicating no significant difference. However, in comparisons where HeisenbergBases is the better model—against FNN, GRN, RNN, and Transformer—the p-values are all smaller than 2.52×10^{-6} , signifying that the HeisenbergBases model significantly outperforms each of these models.

In Table 5, the performance of different models is compared using hypothesis tests on the *Cu-diode* dataset. HeisenbergBases is compared to FNN, GRN, RNN, and Transformer; the p-values are all very small, with values of 2.63×10^{-6} , 0.0004, 1.79×10^{-5} , and 1.79×10^{-5} , respectively, indicating that HeisenbergBases significantly outperforms all these models.

Overall, the HeisenbergBases model shows significantly better performance than the FNN, GRN, RNN, and Transformer models, as indicated by the very small p-values in these comparisons. Meanwhile, the FNN, GRN, RNN, and Transformer models show no statistically significant differences in performance when compared with each other.

2.7. Cell Automata

In this section, we will demonstrate that the derived Heisenberg bases can be used as the cell decay rate in cell automata. From Figure 2, we observe that the Heisenberg bases can represent the Black body radiation, which describes the electromagnetic radiation emitted by a perfect black body. It is an idealized physical object that absorbs all incident radiation and emits energy based on temperature. Specifically, associate u as the wavelength and $\Delta_x \Delta_p(t, \alpha, u)$ as the spectral radiance. We find that the radiance decays as we increase the wavelength. We will generally use the Heisenberg bases as the decay rates for the cell's viability.

2.7.1. Dynamical Simulation for Two Competitive Creatures

This model simulates the evolution of a grid-based, 2D system in which cells can be in one of three states: Healthy, Tumor, or Empty. The Simulation proceeds over a specified number of iterations, with each cell's state updated based on its current state and the states of its neighboring cells. The model includes parameters to control the behavior of cell transitions, including decay and proliferation rates. The simulation begins with a square grid of size $M \times M$, where each cell can represent a healthy cell, a tumor cell, or a space. The system evolves over a specified number of iterations, $iter$, simulating dynamic changes in cell states. Although the total simulation time t is provided, it does not influence the current model behavior. At each time step, the grid is updated based on local interactions. The transition of each cell depends on its type and the state of its neighbors. Two key factors govern the changes: a decay factor $\delta_d = \frac{1}{N_b} \Lambda$, which controls how likely tumor cells decay into empty spaces, and a proliferation factor $\delta_p = \frac{1}{\delta_d}$, which regulates the growth of healthy cells. Healthy cells remain stable unless a rare event (with 1% chance) causes them to become tumor cells if nearby tumors exist. Tumor cells can decay into space based on the decay factor or persist under certain neighborhood conditions. Empty spaces, in turn, may transform into tumor or healthy cells if surrounded by exactly three tumor or healthy neighbors, respectively, depending on random chance and the associated growth rates. This model captures the competitive spatial dynamics between tumor spread, healthy tissue resilience, and

vacant space. As shown in Figure 5, the simulation typically trends toward a dominance of healthy cells, with space reaching a steady-state level by the end of the simulation.

Table 2. P-value of statistical test among model’s average performance in **quantum cat state** learning task

Hypothesis Test	p-value
FNN < GRN	0.0518
FNN < HesenbergBases	0.1398
FNN < RNN	0.0007*
FNN < Transformer	0.0146*
GRN < FNN	0.9552
GRN < HesenbergBases	0.6504
GRN < RNN	0.0898
GRN < Transformer	0.5542
HesenbergBases < FNN	0.8746
HesenbergBases < GRN	0.4091
HesenbergBases < RNN	0.0206*
HesenbergBases < Transformer	0.3082
RNN < FNN	0.9995
RNN < GRN	0.9340
RNN < HesenbergBases	0.9870
RNN < Transformer	0.9585
Transformer < FNN	0.9869
Transformer < GRN	0.4818
Transformer < HesenbergBases	0.7234
Transformer < RNN	0.0512

Table 3. P-value of statistical test among models’ average performance in **BigSol** dataset

Hypothesis Test	p-value
FNN < GRN	0.0232*
FNN < HesenbergBases	0.9995
FNN < RNN	0.0043*
FNN < Transformer	0.0160*
GRN < FNN	0.9798
GRN < HesenbergBases	0.9999
GRN < RNN	0.0705
GRN < Transformer	0.1248
HesenbergBases < FNN	0.0007*
HesenbergBases < GRN	1.79×10^{-5} *
HesenbergBases < RNN	1.79×10^{-5} *
HesenbergBases < Transformer	0.0004*
RNN < FNN	0.9964
RNN < GRN	0.9370
RNN < HesenbergBases	0.9999
RNN < Transformer	0.8546
Transformer < FNN	0.9878
Transformer < GRN	0.8936
Transformer < HesenbergBases	0.9997
Transformer < RNN	0.1677

Table 4. P-value of statistical test among models’ average performance in **Al-diode** dataset

Hypothesis Test	p-value
FNN < GRN	0.3916
FNN < HesenbergBases	1.0000
FNN < RNN	0.0328*
FNN < Transformer	0.0328*
GRN < FNN	0.6246
GRN < HesenbergBases	0.9999
GRN < RNN	0.0705
GRN < Transformer	0.0705
HesenbergBases < FNN	$1.55 \times 10^{-7*}$
HesenbergBases < GRN	$2.52 \times 10^{-6*}$
HesenbergBases < RNN	$2.52 \times 10^{-6*}$
HesenbergBases < Transformer	$2.52 \times 10^{-6*}$
RNN < FNN	0.9702
RNN < GRN	0.9370
RNN < HesenbergBases	0.9999
RNN < Transformer	0.9214
Transformer < FNN	0.9702
Transformer < GRN	0.9370
Transformer < HesenbergBases	0.9999
Transformer < RNN	0.0874

Table 5. P-value of statistical test among models’ average performance in **Cu-diode** dataset

Hypothesis Test	p-value
FNN < GRN	0.1552
FNN < HesenbergBases	0.9999
FNN < RNN	0.0270*
FNN < Transformer	0.0270*
GRN < FNN	0.8602
GRN < HesenbergBases	0.9997
GRN < RNN	0.1454
GRN < Transformer	0.1454
HesenbergBases < FNN	$2.63 \times 10^{-6*}$
HesenbergBases < GRN	0.0004*
HesenbergBases < RNN	$1.79 \times 10^{-5*}$
HesenbergBases < Transformer	$1.79 \times 10^{-5*}$
RNN < FNN	0.9755
RNN < GRN	0.8752
RNN < HesenbergBases	0.99999
RNN < Transformer	0.9214
Transformer < FNN	0.9755
Transformer < GRN	0.8752
Transformer < HesenbergBases	0.99999
Transformer < RNN	0.0874

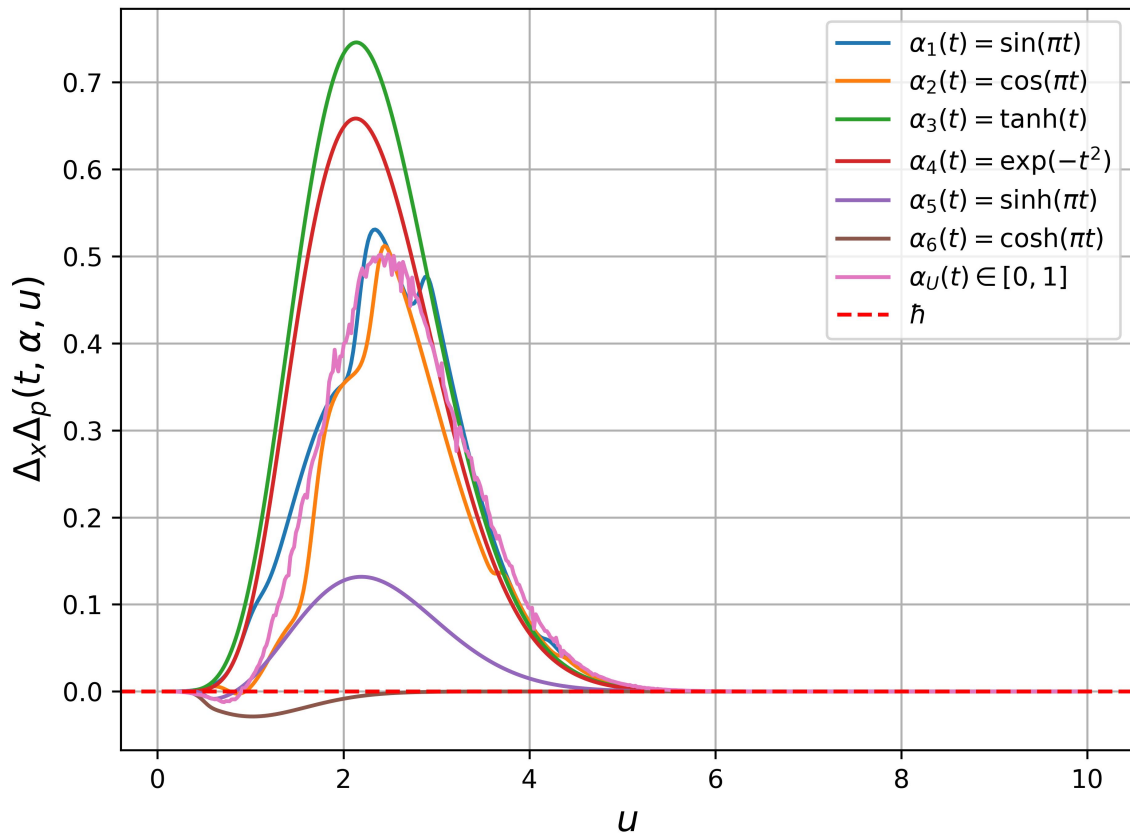


Figure 3. The value of $\Delta_x \Delta_p(t, \alpha, u)$ w.r.t unit of measurement $u \in [0, 1]$, $\alpha(t)$ is a bounded function in $[0, 1]$ and $t \in [0, u]$

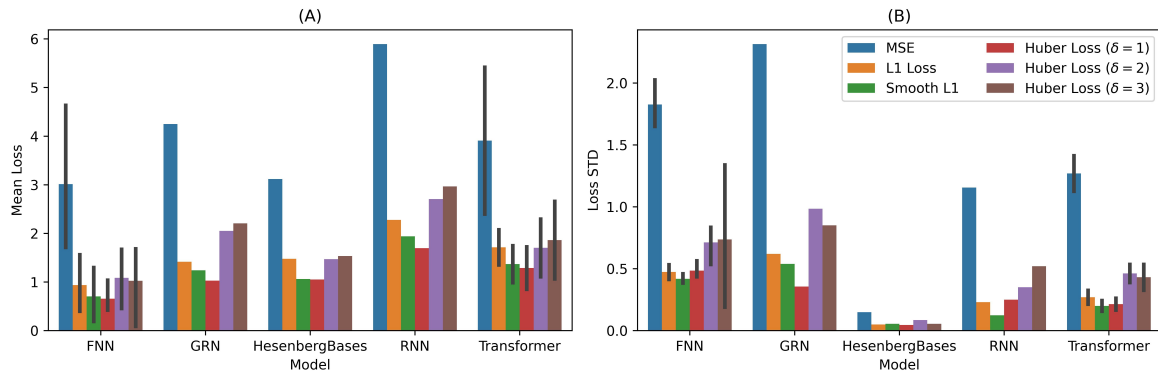


Figure 4. Evaluation in quantum state learning task. (A) mean loss, (B) standard deviation of loss.

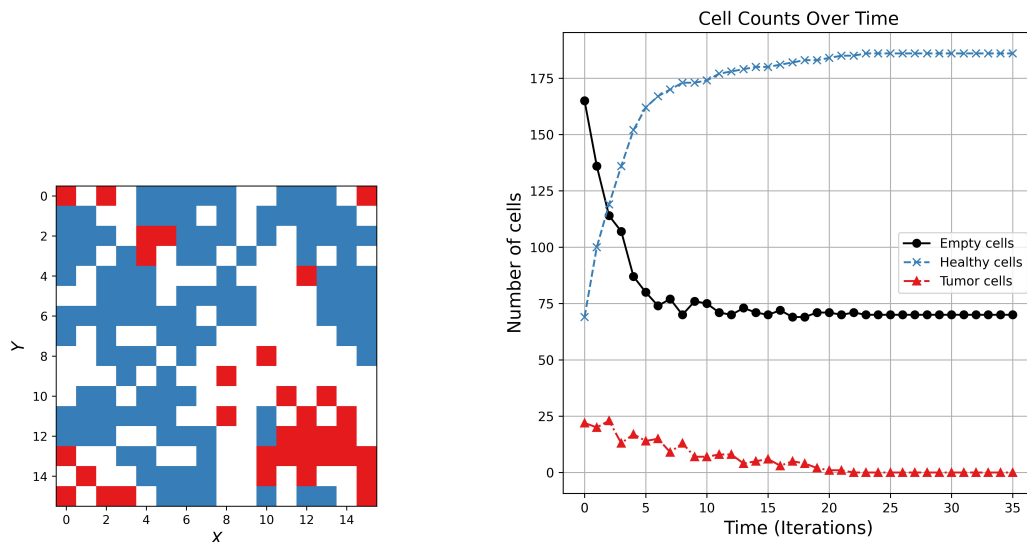


Figure 5. Cell automata model of tumor-healthy cells competition. Left: spatial distribution of tumor (red), healthy (blue), and empty cells (white). Right: cell counts over time

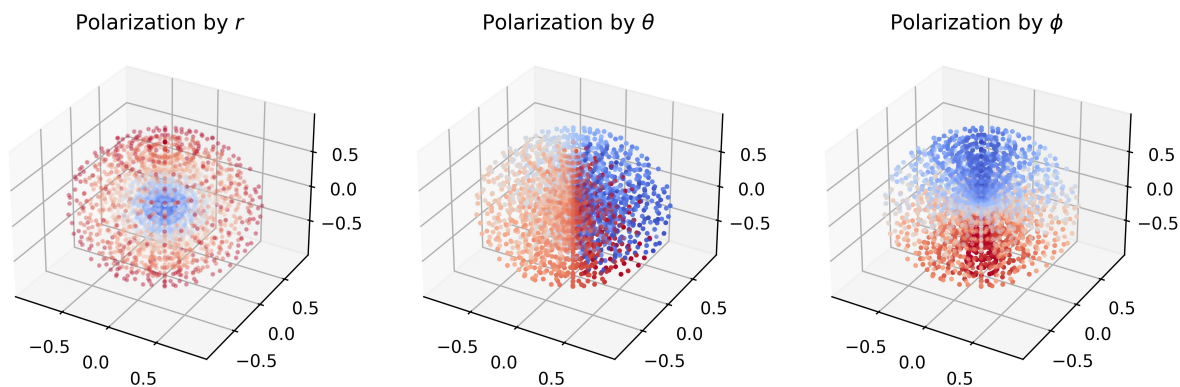


Figure 6. Simulation of hydrogen's electron orbit

3. Case Studies

4. Experiment Design

4.1. Experimental Environment

The input data is $\mathbf{X}$ for each of the following experiments. First, we split the data into two independent train and test sets (with ratio 80%, 20%), with the statistics summarized in Table 6. The number of observations is small, similar to practical scenarios. We train a model $f_{\Theta}(x)$ to reach the minimum of six objective functions $\mathcal{L}$: (1) MSE, (2) L1 Loss, (3) smooth L1, (4,5,6): Huber Loss of $\delta = 1, 2, 3$. We evaluate how well these models can reconstruct the original signals. Specifically, we aim to solve

$$\Theta^* = \arg \min_{\Theta} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{X}) \quad (33)$$

The smaller loss means the closer distance between the predicted vector $\hat{\mathbf{y}}$ and the input $\mathbf{X}$, so a good model is obtained. We build the model using `torch`, and the hyperparameter optimization is performed by `optuna`. Each neural architecture is trained using 50 epochs under 10 trials of hyperparameter optimization. We use AdamW[6] optimizer with a learning rate of 0.005. Random seeds are set for each run. All experiments were conducted with Python 3.7.0 using an Intel i9 processor (2.3 GHz, eight cores) and 16GB DDR4 RAM.

Table 6. Case studies with training and test data sizes

Index	Case Study	N_{train}	N_{test}	Ref.
1	Wigner function of Quantum Cat State	80	20	Simulated
2	Estimation of Temperature-Solubility Relation	144	37	BigSolBD[7]
3	Spectral Classifications of Materials (Al)	80	20	[8]
4	Spectral Classifications of Materials (Cu)	80	20	[8]

4.1.1. Simulation of Hydrogen’s Electron Orbit

We design an automata model for a 3D model, which simulates the electron’s position in the Hydrogen atom. First, we create a polar grid of (r, θ, ϕ) coordinates. The cell is active when the electron is present; otherwise, it is inactive. We show the simulation result in Figure 6, colored by the value r, θ, ϕ . The low values are in blue, and the high ones are in red. The update rule is given as if a random value $<$ decay rate, $\text{newcells}[i, j, k] = 1 - \text{cells}[i, j, k]$ (Flip state (0 to 1 or 1 to 0)). In Figure 6, the electron with high values of r is in the outer region, while the smaller values are closer to the proton. Besides, we can use vertical and horizontal surfaces to stratify the position by θ and ϕ .

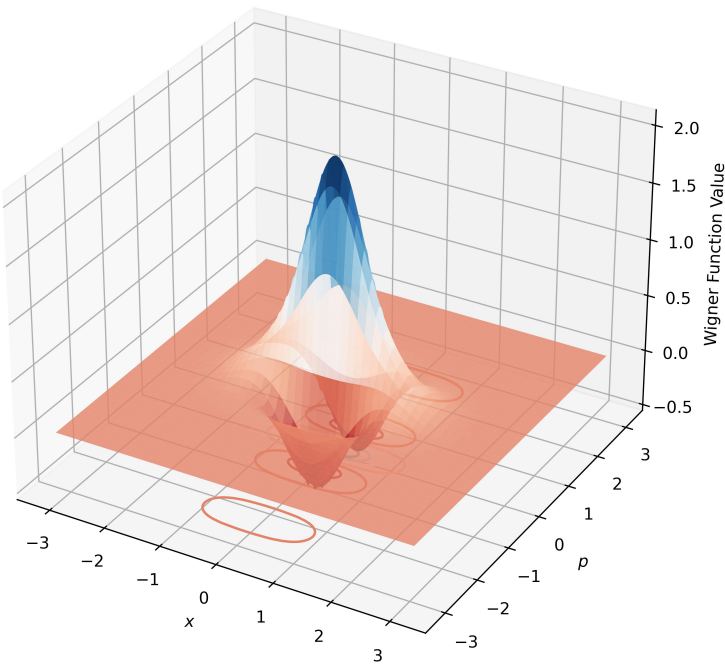


Figure 7. Generating Wigner functions of Quantified Quantum Coherent States

5. Discussion

5.1. The Link to the Heisenberg uncertainty

Back to Eq. 4, Δ_x and Δ_p are the second moment $\mathbb{E}(x^2)$ and $\mathbb{E}(p^2)$, respectively. We have

$$\begin{aligned}\sigma_X^2 &= \mathbb{E}(X^2) - \mathbb{E}(X)^2 = \Delta_x - \mu_x^2 \\ \sigma_P^2 &= \mathbb{E}(P^2) - \mathbb{E}(P)^2 = \Delta_p - \mu_p^2\end{aligned}$$

(34)

Then,

$$\sigma_X^2 \sigma_P^2 = \Delta_x \Delta_p + (\mu_x \mu_p)^2 - \Delta_x \mu_p^2 - \Delta_p \mu_x^2 \quad (35)$$

Since both Δ_x and Δ_p is assumed to be positive, we have

$$\sigma_X^2 \sigma_P^2 \geq \Delta_x \Delta_p + (\mu_x \mu_p)^2 \geq \Delta_x \Delta_p \quad (36)$$

5.2. Future Works

5.2.1. Scalability of the model

Adding more generative functions α results in more complex Heisenberg bases at the expense of higher computation time. This could lead to a better model as we provide a more general class of random functions. Parallel computing can reduce execution time and mitigate the increased computational cost.

5.2.2. Further Applications

The proposed models can be benchmarked in other learning tasks and datasets. Additionally, cellular automata can be applied to practical applications in biomedical science, physics, chemistry, and machine intelligence. Furthermore, the stochastic electron density models can be utilized for problems in atom and molecular dynamics or quantum chemistry.

5.2.3. Limitations

The proposed model's universal approximation capability remains uncertain and warrants further investigation in future work. The simulation time for the cellular automata is notably high, necessitating methods to reduce computational costs. We view this work as a preliminary effort to bridge quantum theory with modern machine intelligence. Future research should focus on exploring more intriguing quantum phenomena, such as quantum tunneling and quantum entanglement.

6. Conclusion

To this point, we have introduced a framework to construct AI models based on the parameterized uncertainty (Section 2.4). The uncertainty function is derived from our geometric interpretation in Section 2.1. Three use cases of our proposed framework are demonstrated, which concern applications in quantum computing, material science, and cell automata modeling (Section 3). Some insights into the proposal and its future extensions are also discussed in the article.

Abbreviations

Models

FNN | Feed-forward Neural Network(s)

GRN | Gated Relation Networks(s)

ML | Machine Learning

NN | Neural Networks

PIML | Physics-Informed Machine Learning

QIML | Quantum-Informed Machine Learning

QML | Quantum Machine Learning

RNN | Recurrent Neural Networks(s)

Mathematical Notations

O_{xyz} | 3D coordinate

u | unit length

$A, B, C, \dots$ | point(s)

$OA, OB, OC, AB, \dots$ | Segment(s)

$\mathcal{M}$ | Manifold

S | Area

$\phi, \psi, \kappa, \Psi, \alpha, \dots$ | function(s)

$x, p, t, \dots$ | Variable(s)

$\mathbb{E}$ | Expectation

μ | mean

σ | standard deviation

Λ | Array of bases

$\mathbf{h}$ | Latent embeddings

$\mathbf{X}, \mathbf{y}$ | Input data

Re, Im | Real and imaginary part of complex number

References

1. Nguyen, N.; Chen, K.C. Quantum embedding search for quantum machine learning. *IEEE Access* **2022**, *10*, 41444–41456.
2. Nguyen, N.; Chen, K.C. Bayesian quantum neural networks. *IEEE Access* **2022**, *10*, 54110–54122.
3. Nguyen, P.N. The duality game: a quantum algorithm for body dynamics modeling. *Quantum Information Processing* **2024**, *23*, 21.
4. Nguyen, P.N. Quantum word embedding for machine learning. *Physica Scripta* **2024**, *99*, 086004.
5. Nguyen, P.N. Quantum DNA Encoder: A Case-Study in gRNA Analysis. In Proceedings of the 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2024, pp. 232–239.
6. Loshchilov, I. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* **2017**.
7. Krasnov, L.; Mikhaylov, S.; Fedorov, M.; Sosnin, S. BigSolDB: Solubility Dataset of Compounds in Organic Solvents and Water in a Wide Range of Temperatures **2023**.
8. Bhatt, T.; Soni, R.; Upadhyay, M.; Jayswal, H.; Chaudhari, J.; Dubey, N.; Patel, A.; Sharma, S.; Makwana, A. A Spectral Dataset of different materials for metal classification **2024**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.