Learning to Predict Quantum Dynamics

Ryan Hsiang[1]

[1]*Department of Electrical Engineering, National Taiwan University*

In this paper, we provide a brief overview of the different research directions and approaches used to predict the dynamics of quantum systems. We formulate the learning problem and methods outlined in relevant research papers using traditional ML, deep learning, and quantum machine learning. We describe the implementation details and experiments of different ML algorithms on the 1D Heisenberg spin chain. These results demonstrate that learning-based methods can approximate quantum dynamics with high fidelity while offering substantial improvements in scalability and inference speed.

I. INTRODUCTION

Predicting the time evolution of quantum systems is a central challenge in many areas of physics and quantum technologies. Traditional approaches often rely on solving the Schrödinger equation or simulating open system dynamics, which can become computationally intractable for large or strongly correlated systems. However, Data-driven methods has emerged as a powerful alternative to simulate and predict quantum dynamics. Simulations on a quantum computer are restricted to a limited number of qubits and substantial error rates due to decoherence and operational imperfections. On the other hand, ML models can predict dynamics orders of magnitude faster than numerically integrating time-dependent equations, and many works have demonstrated the potential of applying machine learning to various dynamics prediction problems.

We investigate three distinct machine learning frameworks for quantum dynamics prediction: Fourier Neural Operators (FNOs) [8], Machine learning on classical shadows [3], and the Resource-Efficient Fast-Forwarding (REFF) algorithm [2]. We examine the model's ability to generalize to time-scale well beyond the training data and its scalability with system size. We compare these learning-based approaches against traditional methods such as Trotterized time evolution and classical deep learning architectures like the U-Net. Experiments show that these methods consistently outperform traditional models and simulation methods in fidelity and loss, showing superior generalizability and robustness.

II. BACKGROUND

A. The Heisenberg Spin Chain

An important model in quantum many-body physics is the 1D Heisenberg chain, which consists of n sites, each containing a spin-1/2 particle, the physical realization of a qubit. This is a one-dimensional model of magnetism and spin-1/2 particles that have a spin-spin interaction. In some metals and crystals where this is some one-dimensional isotropy, these spin chains actually appear and describe the dominant physical behavior.

For quantum mechanical reasons, the dominant coupling between two dipoles may cause nearest neighbors to have the lowest energy when they are aligned. Under this assumption, we consider only magnetic interactions between adjacent dipoles, such that the Hamiltonian can be written in the form,

$$H = \sum_{i=1}^{n} \frac{1}{4}(J_x \sigma_i^x \sigma_{i+1}^x + J_y \sigma_i^y \sigma_{i+1}^y + J_z \sigma_i^z \sigma_{i+1}^z) + \frac{1}{2} h \sigma_i^z. \tag{1}$$

where J_x, J_y, J_z are the coupling constant for two-qubit spin interactions, and h is the single-qubit driving field. The Pauli matrix σ_i^a is defined as $\sigma_i^a = I^{\otimes i-1} \otimes \sigma^a \otimes I^{\otimes n-i}$. In addition, we impose the boundary conditions $\sigma_{n+1}^a = \sigma_1^a$. There are some special cases of (1),

- The Ising model, with $J_x = J_y = 0$,

$$H = \sum_{i=1}^{n} \frac{1}{4} J_z \sigma_i^z \sigma_{i+1}^z + \frac{1}{2} h \sigma_i^x. \tag{2}$$

- The XY model, with $J_z = 0$,

$$H = \sum_{i=1}^{n} \frac{1}{4}(J_x \sigma_i^x \sigma_{i+1}^x + J_y \sigma_i^y \sigma_{i+1}^y) + \frac{1}{2} h \sigma_i^x. \tag{3}$$

For simplicity, we let all non-zero coupling constants to be 1. Importantly, the 1D Heisenberg chain can be exactly solved[7] by analytical methods, making the generation of training data on a classical computer possible.

B. Neural Operators

Neural operators[5] are a class of machine learning models designed to learn mappings between infinite-dimensional function spaces, making them particularly well-suited for solving parametric partial differential equations, such as the Navier-Stokes equation in fluid dynamics or the Scrödinger equation in quantum mechanics,

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t). \tag{4}$$

The neural operator, similar to a conventional neural network, is formulated as an iterative architecture $v_0, v_1, ..., v_T$, where v_j is a sequence of functions. The iterative update has the form,

$$v_{t+1}(x) = \sigma\left(Wv_t(x) + \left(\mathcal{K}_t v_t\right)(x)\right). \tag{5}$$

where W is a linear transformation and σ is a non-linear activation function. The kernel integral transformation $\mathcal{K}(a;\phi)$ is defined as,

$$(\mathcal{K}_t v_t)(x) = \int_{D_t} \kappa^{(t)}(x,y)v_t(y)\,dy, \quad \forall x \in D_{t+1}. \tag{6}$$

where $D_t \subset \mathbb{R}^d$ is a bounded domain, and κ_t is implemented as a neural network, where gradient updates can be performed. Note the similarity of this architecture to a conventional neural network:

$$x_{t+1} = \sigma\left(Wx_t + b\right). \tag{7}$$

Now, instead of using vectors as input, we use functions as input—typically represented by their discretized evaluations on a spatial grid.

[6] introduced a new neural operator framework called the *Fourier neural operator* (FNO), which replaces the kernel integral operator $\mathcal{K}$ by a convolution operator defined in the Fourier space. Let $\mathcal{F}$ denote the Fourier transform of a function f and $\mathcal{F}^{-1}$ its inverse, then by letting $\kappa(x,y) = \kappa(x-y)$ and applying the convolution theorem, we have

$$(\mathcal{K}v_t)(x) = \mathcal{F}^{-1}(\mathcal{F}(\kappa)\cdot\mathcal{F}(v_t))(x) \tag{8}$$

From (8), we can define the *Fourier integral operator* R that directly parameterize κ in the Fourier space.

$$R = \mathcal{F}(\kappa). \tag{9}$$

Experiments on the Navier-Stokes equation showed that FNO is the only ML architecture to successfully model turbulent flows with zero-shot super-resolution and outperforms existing methods on Darcy Flow and Burgers' equation.

C. Trotterization

Trotterization is a method for simulating the time evolution of a complex Hamiltonian by a simplified quantum circuit. Consider the time evolution of some quantum state $|\psi(t)\rangle$,

$$|\psi(t)\rangle = e^{-iHt}\,|\psi(0)\rangle\,. \tag{10}$$

We expand the Hamiltonian with respect to the Pauli basis,

$$H = \sum_{P\in\{I,X,Y,Z\}^{\otimes n}} \alpha_P P. \tag{11}$$

In general, the exponential of the sum of two operators can not be split if the operators are non-commuting, i.e.,

$$e^{-i(A+B)t} \neq e^{-iAt}e^{-iBt}. \tag{12}$$

The key idea of Trotterization is to approximate the exponential of the Hamiltonian, which is generally difficult to implement directly, by a product of exponentials of the individual Pauli terms.

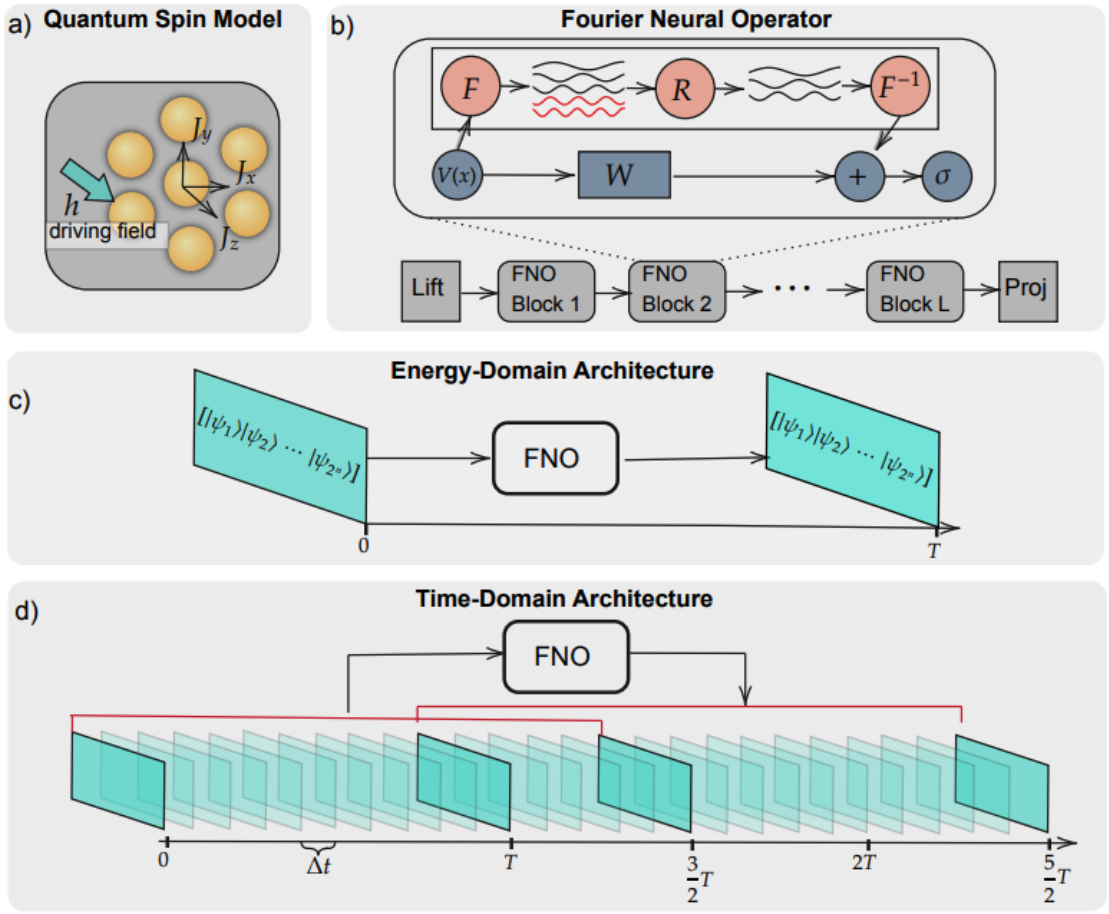
$$|\psi(t)\rangle = \prod_P \left(e^{-i\alpha_P P t/r}\right)^r |\psi(0)\rangle\,. \tag{13}$$

That is, we apply a gate r times to approximate the full circuit. However, this also leads to deeper circuits, which, in practice, leads to more error accumulation.

III. LEARNING DYNAMICS USING THE COMPLETE WAVE FUNCTION

Inspired by the results of [6], [8] applies the Fourier neural operator framework to predicting quantum wavefunctions. Given an input state $|\psi\rangle$ at time $t = 0$ or states over a certain time interval, we look to predict $\psi(x, t)$ at a future time. The performance of the ML model will be evaluated with the fidelity between the prediction and the ground truth.

$$F(\psi_{\text{pred}}, \psi_{\text{truth}}) = |\langle\psi_{\text{pred}}|\psi_{\text{truth}}\rangle|^2.$$



[8] proposes two distinct architectures for learning the complete wavefunction:

A. Energy-domain Architecture

We first decompose the state at $t = 0$ into its Hamiltonian eigenbasis, and sort each basis state by its energy level.

$$|\psi\rangle = \sum_{i=1}^{2^n} c_i\,|\phi_i\rangle$$

where $|\phi_i\rangle$ are the energy eigenstates for each respective energy level. This is equivalent to performing a Fourier transform over $\psi(x,0)$ due to the Planck-Einstein relation $E = h\nu$. For each energy level, we concatenate a positional embedding to help the model learn the relation between the state and the energy. The input is then fed into the ML model and outputs $\psi(x,T)$. In [8], T is chosen to be π. In addition, we filter out high-energy modes to focus on slow, long-term transitions.

The drawback of this architecture is that it is not discretization-agnostic, i.e., it cannot directly output intermediate times or finer grids. Thus, for scenarios requiring more fine-grained predictions, we consider an alternate architecture.

B. Time-Domain Architecture

The time-domain architecture considers states over a time interval instead of the initial condition as input. We discretize the state within the interval, e.g., $[0, 3T/2]$ with an arbitrary width ΔT. We also concatenate the positional embedding for each time. The output, similar to the input, is the wavefunction evolved over a discretized time interval, say $[T, 5T/2]$. This architecture demonstrates the FNO's capability to achieve zero-shot super-resolution predictions.

C. Challenges

The main challenge in this approach is scalability, since it requires 2^n terms to describe an n-qubit system. To solve this problem, other works[1, 4] have proposed methods to learn a partial representation of the unknown state polynomial in the number of qubits that allows accurate prediction of many of its important properties. We will discuss this in detail in the following section.

IV. LEARNING DYNAMICS USING THE CLASSICAL SHADOW

Prior works on predicting unknown quantum states include [1] and [4]. [1] introduced the *shadow tomography* problem: Given a number of identical copies of an unknown n-qubit quantum state ρ, as well as two-outcome measurements $\Pi_1, ...\Pi_M$, how to estimate ρ such that the probability of the outcome associated with Π_i occurs is within error ε? A key insight from [1] is that while a complete classical description of a quantum state is in principle possible via an exponential number of experiments, it is not necessary in most scenarios. To accurately predict many properties of a quantum system, $O(n\varepsilon^{-4}\log^4 M)$ of state copies suffices. That is, we can learn the behavior of an arbitrary quantum state by measuring only a polynomial number of qubits. Building upon [1], [4] introduced an improved protocol that learns the *classical shadow*, a minimal classical representation of an unknown quantum state. Specifically, we outline the procedure of learning the classical shadow of ρ as follows,

1. Apply one of the three Pauli observables X, Y, Z, chosen randomly, to ρ then perform a measurement on the computational basis.

$$\rho \to U\rho U^\dagger. \tag{14}$$

2. After obtaining the measurement outcome $|\hat{b}\rangle$, collect the classical snapshot of ρ,

$$\hat{\rho} = \mathcal{M}^{-1}(U^\dagger|\hat{b}\rangle\langle\hat{b}|U) \tag{15}$$

where $\mathcal{M}$ is a quantum channel defined as,

$$\mathcal{M}(\rho) = \mathbb{E}\left[U^\dagger|\hat{b}\rangle\langle\hat{b}|U\right] \tag{16}$$

3. Repeat this procedure N times to obtain N independent samples of ρ:

$$S(\rho, N) = \{\hat{\rho}_1, ..., \hat{\rho}_N\}. \tag{17}$$

The array is called the *classical shadow* of ρ.

Using the classical shadow of sufficient size, we can predict many properties of the quantum state efficiently, such as fidelity, entanglement entropy, Hamiltonian, etc.

Another relevant learning task is predicting an arbitrary and unknown *observable* O. We now consider a distribution $\mathcal{D}$ on n-qubit quantum states. The problem involves finding a function $h(\rho)$ that predicts the expected value of the observable O on ρ with small error,

$$\mathop{\mathbb{E}}_{\rho\sim\mathcal{D}}\left[h(\rho)-\mathrm{Tr}(O\rho)\right]^2\leq\epsilon. \tag{18}$$

In practice, we do not have the expected value of O readily available. Instead, we use multiple measurements of O to obtain an accurate estimate of the expected value. We expand the observable in terms of the Pauli operators,

$$O=\sum_{P\in\{I,X,Y,Z\}^{\otimes n}}\alpha_P P. \tag{19}$$

We can approximate the sum over P by a truncated sum to include

$$O^{(k)}=\sum_{P\in\{I,X,Y,Z\}^{\otimes n}:|P|\leq k}\alpha_P P. \tag{20}$$

That is, we only include operators P that act nontrivially on no more than k qubits. To learn O with mean squared error ϵ, it suffices to learn $O^{(k)}$ with $k=\mathcal{O}(\log(1/\epsilon))$.

From the previous two tasks, we can construct a procedure to learn to predict arbitrary quantum processes. Given an arbitrary and unknown quantum process $\mathcal{E}$ and a family of observables $\{O_i\}$. We now look to find target functions that satisfy

$$\mathop{\mathbb{E}}_{\rho\sim\mathcal{D}}\left[h(\rho,O_i)-\mathrm{Tr}(O_i\mathcal{E}(\rho))\right]^2\leq\epsilon. \tag{21}$$

This task can be reduced to learning unknown states and learning unknown observables, we let $\mathcal{E}(\rho)$ be the unknown state, after we learn this state using the procedure outlined above. We can then predict the expected value $\mathrm{Tr}(O_i\mathcal{E}(\rho))$ with accuracy. Now we rewrite the expected value in the Heisenberg picture,

$$\mathrm{Tr}(O_i\mathcal{E}(\rho))=\mathrm{Tr}(\mathcal{E}^\dagger(O_i)\rho) \tag{22}$$

This is equivalent to learning an unknown observable with $O=\mathcal{E}^\dagger(O_i)$. We use $\{\rho_\ell,\mathrm{Tr}(\mathcal{E}^\dagger(O_i)\rho_\ell)\}$ obtained by learning $\mathcal{E}(\rho_\ell)$ as training data to learn $\mathcal{E}^\dagger(O_i)$, thus learning the unknown process.

Next, we introduce the learning algorithm described in [3]. We first prepare a classical dataset consisting of randomly prepared product states $|\psi^{(\mathrm{in})}\rangle=\otimes_{i=1}^n|s_i^{(\mathrm{in})}\rangle$, where $|s_i^{(\mathrm{in})}\rangle\in\{|0\rangle,|1\rangle,|+\rangle,|-\rangle,|y+\rangle,|y-\rangle\}$, then we pass it through $\mathcal{E}$ and perform random Pauli measurements to produce the measurement outcome $|\psi^{(\mathrm{out})}\rangle=\otimes_{i=1}^n|s_i^{(\mathrm{out})}\rangle$.

$$S_N(\mathcal{E})=\left\{|\psi_\ell^{(\mathrm{in})}\rangle,|\psi_\ell^{(\mathrm{out})}\rangle\right\}_{\ell=1}^N. \tag{23}$$

Using this classical dataset and given an observable O, we predict $\mathrm{Tr}(O\mathcal{E}(\rho_\ell))$ with

$$h(O)=\frac{1}{N}\sum_{\ell=1}^N\mathrm{Tr}\left(O\otimes_{i=1}^n\left(3|s_{\ell,i}^{(\mathrm{out})}\rangle\langle s_{\ell,i}^{(\mathrm{out})}|-I\right)\right) \tag{24}$$

where $\otimes_{i=1}^n 3|s_{\ell,i}^{(\mathrm{out})}\rangle\langle s_{\ell,i}^{(\mathrm{out})}|-I$ is an unbiased estimator of $\mathcal{E}(\rho_\ell)$. Thus we are able to transform the classical dataset into

$$\left\{\rho_\ell=|\psi_\ell^{(\mathrm{in})}\rangle\langle\psi_\ell^{(\mathrm{in})}|,\quad\mathrm{Tr}(O\mathcal{E}(\rho_\ell))\right\}. \tag{25}$$

Using (22) and (20), we truncate the observable $\mathcal{E}^\dagger(O)$ into its weight-k truncation.

$$\mathcal{E}^\dagger(O)^{(k)}=\sum_{P\in\{I,X,Y,Z\}^{\otimes n}:|P|\leq k}\alpha_P P. \tag{26}$$

After which, we apply linear regression with ℓ_1 regularization on (25),

$$\min_{\alpha_P}\frac{1}{N}\sum_{\ell=1}^N\left|y_\ell-\sum_P\alpha_P\,\mathrm{Tr}(P|\psi_\ell\rangle\langle\psi_\ell|)\right|^2+a\sum_P|\alpha_P|. \tag{27}$$

We then recover an approximate model of the observable

$$\sum_P\alpha_P P. \tag{28}$$

V. LEARNING DYNAMICS USING QUANTUM MACHINE LEARNING

In this section, we discuss the quantum machine learning (QML) framework introduced in [2]. Standard methods for dynamical simulation like Trotterization grow the circuit depth in proportion to the simulation time, ultimately running into the decoherence time of the NISQ device. Fast-forwarding methods for long-time simulations on NISQ devices is thus crucial. This approach uses quantum machine learning, which learns the dynamics of quantum systems using approximate quantum circuits. Specifically, this work introduces the Resource-Efficient Fast Forwarding algorithm (REFF), which learns a fixed-depth quantum circuit that allows for fast-forwarding. Concretely, we would like to learn a time-dependent quantum neural network (QNN) $V_t(\alpha)$ that approximates,

$$\mathrm{Tr}[V_t(\alpha)\rho V_t(\alpha)^\dagger O] \approx \mathrm{Tr}[e^{-iHt}\rho e^{iHt}O]. \tag{29}$$

where the QNN has the following form,

$$V_t(\theta,\gamma) = W(\theta)D_t(\gamma)W^\dagger(\theta). \tag{30}$$

$W(\theta)$ is a time-independent unitary, and $D_t(\gamma)$ is a time-dependent unitary that is diagonal in the standard basis. Similar to (23), we use training data consisting of input-output states, where the output states are the input states evolved over a time Δt. Training over this time interval, we hope that the trained QNN V_t can approximate the target unitary U_t such that the long-time simulation U_T of the quantum state can be approximated by the fixed-depth quantum circuit V_T. Different from [8], since fidelity suffers from barren plateaus — regions where gradients vanish exponentially with system size, we use a local version of the cost of the form,

$$\frac{1}{N}\sum_{\ell=1}^{N}\mathrm{Tr}[V^\dagger|\psi_\ell^{(\mathrm{out})}\rangle\langle\psi_\ell^{(\mathrm{out})}|VO_\ell], \tag{31}$$

$$O_\ell = I - \frac{1}{n}\sum_{i=1}^{n}|\psi_\ell^{(i)}\rangle\langle\psi_\ell^{(i)}| \otimes I_{\bar{i}}. \tag{32}$$

where $\bar{i}$ denotes the set of all qubits except for i. That is, the cost is vanishing if and only if $V = e^{-iHt}$.

VI. EXPERIMENTS

Numerical experiments[3] on the 50-qubit 1D XY chains with external magnetic field show that the ML model's prediction agrees well with the ground truth over a wide range of evolution time t.

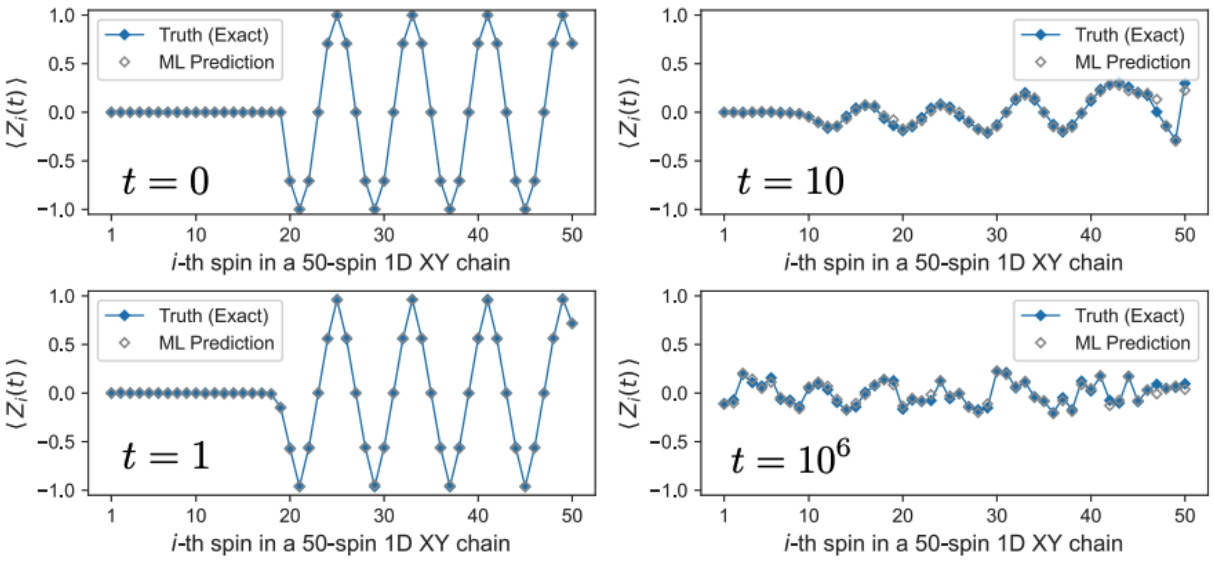


Figure 1: Results of training the ML model of [3] on the 1D XY chain

Experiments[8] using the FNO and the energy-domain architecture on 4 and 8-qubit Heisenberg chains showed that training conducted over various time intervals, such as $[0, T]$, $[T, 2T]$, and $[2T, 3T]$ rather than a single time interval is crucial, as training on a single interval will often lead to overfitting. Using multiple training intervals, the model is able to achieve high fidelity across unseen states.

Experiments using the time-domain architecture is compared to a standard U-Net. The results show that FNO consistently outperforms U-Net on different input states. Importantly, both results demonstrate FNO's ability to generalize to time intervals well beyond the training data. The model is able to achieve a 6.71× speedup compared to exact unitary-based methods with only negligible fidelity reduction.

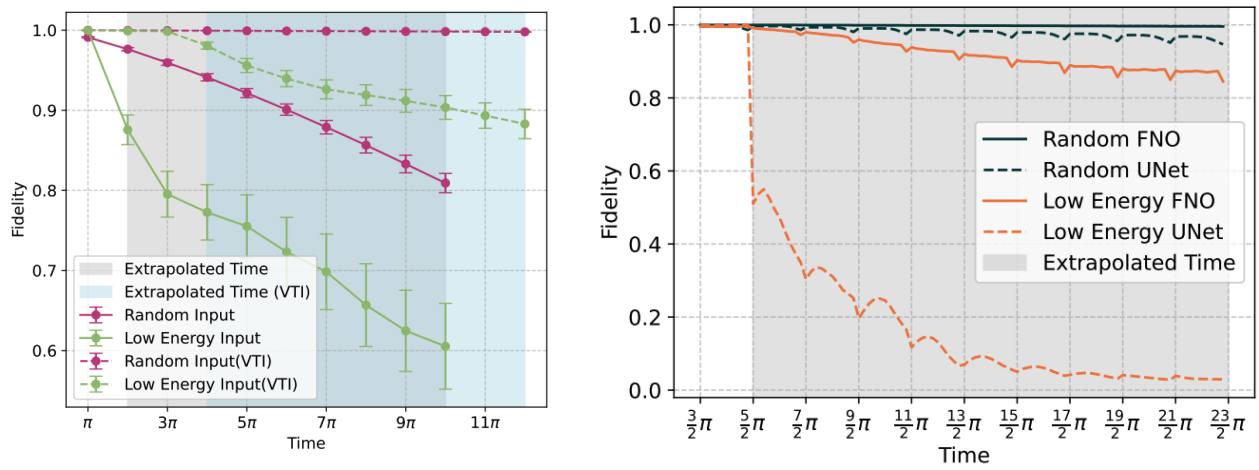


Figure 2: Results of training the FNO on the energy-domain architecture (left) and the time-domain architecture (right)

[2] studies the suitability of the REFF on a 2-qubit XY chain. The training results of REFF are compared with Trotterization, which shows that the circuit outperforms Trotterization in both fidelity and fast-forwarding. The method achieves on average 0.8 fidelity for 94 time steps, a 23.8× improvement over the standard Trotter methods, which has a fidelity of less than 0.8 after 4 time steps.

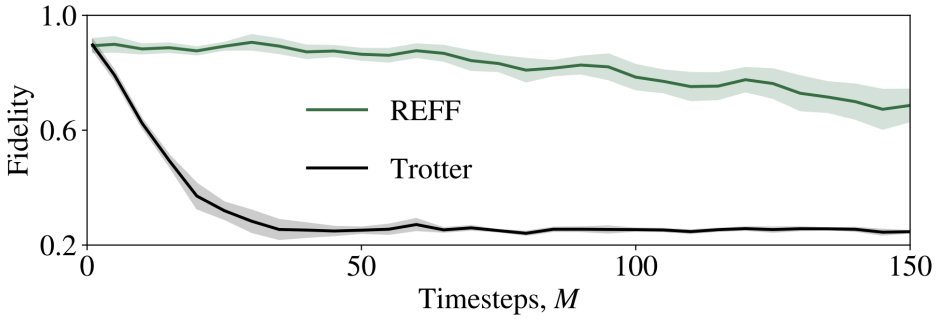


Figure 3: Results of the REFF algorithm compared to Trotterization

VII. CONCLUSION

In this work, We discussed three main ML frameworks: Fourier neural operators, machine learning on classical shadows, and the Resource-Efficient Fast-Forwarding algorithm. We discuss the experimental results of each setup on the 1D Heisenberg spin chain and compare their performance with traditional methods such as Trotterization and the U-Net. Each algorithm demonstrated the model's ability to generalize beyond the training data to longer time horizon not achievable by traditional simulators without decoherence while drastically reducing simulation time. These results highlight the growing potential of machine learning as a scalable tool for quantum simulation, especially in regimes where conventional methods are limited by

computational cost or quantum hardware constraints.

---

[1] Scott Aaronson. Shadow tomography of quantum states, 2018.

[2] Joe Gibbs, Zoë Holmes, Matthias C. Caro, Nicholas Ezzell, Hsin-Yuan Huang, Lukasz Cincio, Andrew T. Sornborger, and Patrick J. Coles. Dynamical simulation via quantum machine learning with provable generalization. *Physical Review Research*, 6(1), March 2024.

[3] Hsin-Yuan Huang, Sitan Chen, and John Preskill. Learning to predict arbitrary quantum processes. *PRX Quantum*, 4:040337, Dec 2023.

[4] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, June 2020.

[5] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(1):1–140, 2023.

[6] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021.

[7] Elliott Lieb, Theodore Schultz, and Daniel Mattis. Two soluble models of an antiferromagnetic chain. *Annals of Physics*, 16(3):407–466, 1961.

[8] Freya Shah, Taylor L. Patti, Julius Berner, Bahareh Tolooshams, Jean Kossaifi, and Anima Anandkumar. Fourier neural operators for learning dynamics in quantum spin systems, 2024.