Quantum Encoding of Structured Data with Matrix Product States

Josh Green* [1] and Jingbo Wang [2]

The University of Western Australia, Perth, WA 6009, Australia

February 25, 2025

[0]josh.green@uwa.edu.au

[0]jingbo.wang@uwa.edu.au

Abstract

Quantum computing faces a fundamental challenge: the amplitude encoding of an arbitrary n-qubit state vector generally requires $\Omega(2^n)$ gate operations. We can, however, form dimensionality-reduced representations of quantum states using matrix product states (MPS), providing a promising pathway to the efficient amplitude encoding of states with limited entanglement entropy. In this paper, we explore the capabilities of MPS representations to encode a wide range of functions and images using $\mathcal{O}(n)$-depth circuits without any ancilla qubits, computed with the so-called Matrix Product Disentangler algorithm with tensor network optimisation. We find that MPS-based state preparation enables the efficient encoding of functions up to low-degree piecewise polynomials with accuracy exceeding 99.99% accuracy. We also showcase a novel approach to encoding structured image data based on MPS approximations of the discrete wavelet transform (DWT) representation, which is shown to prepare a 128×128 ChestMNIST image on 14 qubits with fidelity exceeding 99.1% on a circuit with a total depth of just 425 single-qubit rotation and two-qubit CNOT gates.

1. Introduction

Quantum computation is the synergy of physics and computer science, and it represents a paradigm through its potential to solve problems deemed inefficient or intractable by classical computers. A crucial component of many quantum applications is state preparation, which involves designing quantum circuits to encode classical information into quantum amplitudes. Quantum state preparation is an imperative subroutine for machine learning [1, 2], the HHL algorithm [3–5], quantum chemistry simulation [6–10], computational finance [11], Monte Carlo simulation [12,13], and among others. These applications represent some of the most extensively studied and impactful areas in quantum computation.

The core incongruity of quantum state preparation is that the unitary transformation to prepare an arbitrary n-qubit state has $\mathcal{O}(2^n)$ real degrees of freedom, which cannot be accurately decomposed into $\mathcal{O}(\text{poly}(n))$ one- and two-qubit quantum gates in general [14–17]. Therefore, the design of efficient quantum circuits for the state preparation of any arbitrary target state requires the assumption that the target state has some form of "polynomial-sized" intrinsic structure that can be effectively exploited. Moreover, many of the foundational techniques for state preparation – such as the Grover-Rudolph algorithm – rely on resource-intensive circuits for quantum arithmetic [18, 19], or on variational quantum circuits that are prone to barren plateaus during training [20, 21].

Matrix product states (MPS) allow the efficient and close approximation of quantum states with limited entanglement entropy, specifically states with entanglement entropy scaling as an area law [22]. The MPS format can be effectively used as a dimensionality reduction technique, enabling an efficient classical representation of approximate quantum states. This has led to a myriad of proposals for MPS techniques in quantum state preparation [23–30]. However, the error introduced in approximating a quantum state with an MPS lacks rigorous error bounds in general. For this reason, it remains theoretically unclear to what extent MPS-based state preparation can be utilised for practically useful quantum computational applications.

We address this open question by exploring the capabilities and limitations of approximate MPS-based state preparation schemes for states representing structure data, such as discretised functions and images. We adopt the classically efficient "matrix product disentangler" (MPD) algorithm introduced by Shi-Ju Ran in 2020 [25]. This approach approximately prepares MPS with a layer-by-layer $\mathcal{O}(nL)$-depth circuit without any ancilla qubits, where n is the number of qubits and L is the number of circuit layers. However, it lacks guarantees in its ability to converge to a high-quality solution as the number of layers is increased. We added an efficient global tensor network optimisation (TNO) scheme similar to those introduced in [26, 29] with parameters initialised by the output of the MPD algorithm. The TNO is computed classically using polynomially scaling resources, and we denote this algorithm using MPD+TNO.

The first chief objective of this article is to delineate the efficacy of the MPD and MPD+TNO algorithms in preparing states representative of discretised functions. We empirically demonstrate the viability of approximating a broad class of discretised functions using MPS representations, including Gaussian, root, logarithmic, and a range of piecewise polynomials. We find that the MPD and MPD+TNO algorithms can encode functions up to piecewise Chebyshev polynomials with approximately 10 intervals with fidelity exceeding 99.99% using linear-depth circuits. Such states follow an area law of entanglement entropy, and these results scale to large qubit counts. While the state preparation of functions can be applied to load boundary conditions to solve PDEs, this work serves as a core benchmark for the MPD and MPD+TNO algorithms, which helps to delineate their limitations in a context where high-fidelity encodings are essential.

The second core contribution of this work is a novel method for encoding structured image data. This approach aims to encode the 2-dimensional discrete wavelet transform (DWT) representation of images via the MPD and MPD+TNO state preparation techniques. We demonstrate the effectiveness of the MPD algorithm for this approach by encoding the DWT representation of a 128×128 ChestMNIST image on 14 qubits. The performance of the MPD algorithm is underlined by a 99.6% fidelity achieved using 300 circuit layers, which we show can be further improved using deeper circuits. Crucially, however, MPD+TNO enables significant reductions to circuit depth. For instance, the MPD+TNO algorithm with just 50 circuit layers achieves an encoding with 99.3% fidelity, comparable to the MPD algorithm with over 200 layers. With just 10 layers, the MPD+TNO algorithm achieves a 98.2% fidelity, far exceeding the 94.8% fidelity performance of the MPD algorithm alone.

We find that the TNO is a paramount improvement over the MPD algorithm alone in instances where the MPD algorithm converges slowly, enabling higher fidelity states with shallower depth circuits. While TNO comes with an increased classical overhead in computing the optimisation, this trade-off to reduce the required quantum resources makes the MPD+TNO algorithm is pertinent for near-term implementation on existing quantum devices. Enabling the approximate encoding of discretised functions and image data with linear-depth circuits consisting of one- and two-qubit gates, without any ancilla qubits, we identify that MPS-based encoding is a highly promising candidate for near-term approximate state preparation, while more sophisticated encoding methods will be essential for meaningful quantum advantage in an extended horizon.

2. Background and Motivation

This section outlines the conventional methods and confounding challenges of designing general and efficient quantum circuits for state preparation. This serves as a condensed review and critique of the foundational literature in quantum state preparation, which motivates the adoption and attraction of MPS-based encoding techniques in the proceeding sections.

2.1 Amplitude Encoding and Quantum State Preparation

Let $\mathbf{a} = (a_0, \ldots, a_{N-1})^T \in \mathbb{C}^N$ be a vector of $N = 2^n$ complex numbers, where n denotes the number of qubits in the system. State preparation aims to encode this vector into the amplitudes of a quantum state wavefunction. The state $|\psi\rangle$, known as the target state, is given by:

$$|\psi\rangle = \frac{1}{\|\mathbf{a}\|_2} \sum_{i=0}^{N-1} a_i |i\rangle \ , \tag{1}$$

where $\|\mathbf{a}\|_2$, the Euclidean norm of $\mathbf{a}$, is the normalisation coefficient. $|\psi\rangle$ is prepared by:

$$U_S |0\rangle^{\otimes n} = |\psi'\rangle \ , \tag{2}$$

where $U_S \in U(N)$ is the so-called state preparation operator, $|0\rangle^{\otimes n}$ is the initial state, and $|\psi'\rangle$ is an approximation of the target state. We aim to maximise the fidelity $F = |\langle\psi|\psi'\rangle|^2$ with at most $\mathcal{O}(\text{poly}(n))$ circuit depth.

The amplitude encoding of 2^n unstructured points requires a one- and two-qubit gate complexity of $\Omega(2^n)$ [14]. This stems from the fact that the most general unitary operation mapping the zero state $|0\rangle^{\otimes n}$ to another quantum state has $2^{n+1} - 2$ real degrees of freedom, excluding the degrees of freedom associated with the global phase and normalisation constant [15–17]. Considering that there is no currently feasible means of maintaining coherence long enough for practical QRAM [31], the issue of inefficient quantum state preparation is exacerbated by the requirement to encode the classical information for every measurement of the output register. For example, it has been argued that the exponential speed-up for qPCA [32] is derived from state preparation assumptions of QRAM, and there is no viable means to efficiently encode a dense vector $\boldsymbol{b}$ to solve the linear system $A\boldsymbol{x} = \boldsymbol{b}$ without predominating the overall complexity of the HHL algorithm [3].

This highlights critical limitations for a wide range of quantum computing applications, including quantum machine learning, quantum image processing, and quantum signal processing. These fields often rely on efficient quantum state preparation to encode significant classical data into quantum systems. Without scalable and practical solutions to these challenges, the computational overhead required for data encoding could negate the potential advantages of quantum algorithms. Addressing these limitations is therefore essential to unlocking the full potential of quantum technologies in solving real-world problems.

In recent years, a workaround to the exponential circuit depth of arbitrary quantum state preparation has gained traction, namely the space-time trade-off approach [33–37]. This approach exploits quantum circuits that effectively *parallelises* $\mathcal{O}(2^n)$ operations using $\mathcal{O}(2^n)$

ancilla qubits, achieving an optimal linear circuit depth of $\mathcal{O}(n)$ [33,34]. However, this method necessitates exponentially large quantum computers capable of arbitrarily long-range, non-local gate operations, which is unrealistic. Decomposing this circuit into local operations retrieves an exponential asymptotic scaling of circuit depth.

We are ultimately compelled to assume some form of innate *structure* in the target state that can be exploited for efficient state preparation. The first form of structure to consider is the assumption that the target vector is m-sparse, containing at most m non-zero elements, where $m \ll 2^n$. Numerous efficient quantum algorithms have been developed specifically to handle the preparation of sparse states [38–42]. However, this approach offers a trivial resolution to the dimensionality burden through the restrictive assumption of a sub-exponential number of non-zero elements in the target state vector. This motivates a consequential question: What kind of non-trivial structures can be leveraged? Furthermore, how complex can these structures be while allowing for efficient state preparation? As delineated in Figure 1, we aspire to characterise aspects of the transition period that may enable efficient quantum state preparation. As the target state becomes increasingly complex, it will start to resemble unstructured data, which will then forbid an efficient and accurate $\mathcal{O}(\text{poly}(n))$ quantum circuit to encode it.
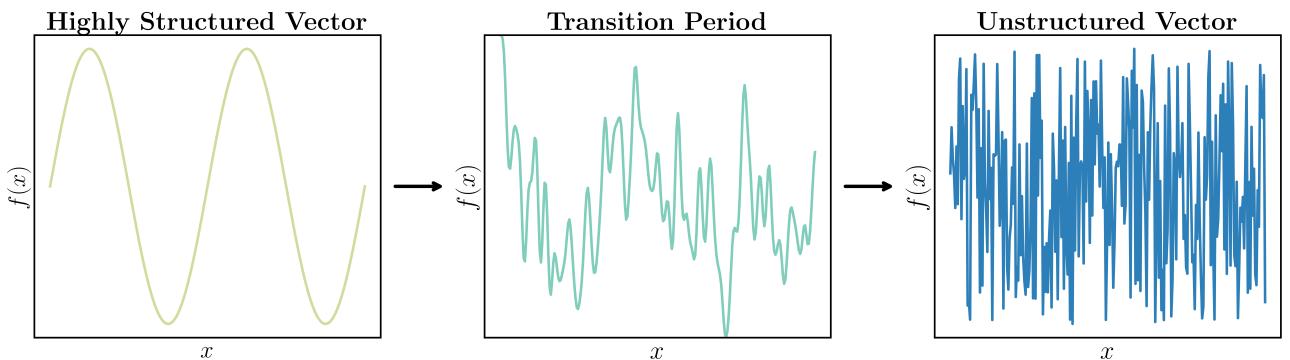

Figure 1: The transition from a highly structured smooth function target vector with low entanglement entropy to fitting a smooth function to an arbitrary unstructured state vector.

2.3 Existing Approaches to State Preparation

The Grover-Rudolph Algorithm

A foundational quantum state preparation algorithm is the Grover-Rudolph algorithm, first proposed by Zalka in 1998 [43] and independently rediscovered by its namesake in 2002 [18]. The Grover-Rudolph algorithm is a coarse-graining scheme designed to prepare log-concave probability distributions. We begin with a discretisation of the target state across $m < n$ qubits,

$$|\psi_m\rangle = \sum_{i=0}^{2^m-1} \sqrt{p_i^{(m)}} \, |i\rangle_m \tag{3}$$

where $p(x)$ is log-concave probability distribution uniformly discretised across 2^m point. Next we compute a set of rotation angles $\boldsymbol{\theta} = \{\theta_i\}_{i=0}^{2^m}$, in quantum parallel, using k ancilla qubits,

$$U_\theta : \sqrt{p_i^{(m)}} \, |i\rangle_m \, |0...0\rangle_k \longrightarrow \sqrt{p_i^{(m)}} \, |i\rangle_m \, |\theta_i\rangle_k \tag{4}$$

where U_θ is a unitary oracle, $\theta_i := 2\arccos\sqrt{f(i)}$ is a k-bit approximation of the ith rotation angle, and

$$f(i) = \frac{\int_{x_L^i}^{(x_R^i - x_L^i)/2} p(x)\,dx}{\int_{x_L^i}^{x_R} p(x)\,dx} \tag{5}$$

is the probability that, given x lies in the ith region, it also lies in the left half of this region. Here x_L^i and x_R^i are the left and right boundaries of region i. Then, applying a controlled-R_y rotation to an ancilla qubit yields the following state:

$$|\psi'_{m+1}\rangle = \sum_{i=0}^{2^m - 1} \sqrt{p_i^{(m)}} \, |i\rangle_m |\theta_i\rangle_k \left(\cos\theta_i \, |0\rangle + \sin\theta_i \, |1\rangle\right)_1 \tag{6}$$

Uncomputing with $U_\theta^\dagger$ disentangles the first register into $|0\rangle^{\otimes k}$, which can be reused for subsequent computations. Now absorbing the state of the ancilla qubit into the second register, we are left with the state:

$$|\psi_{m+1}\rangle = \sum_{i=0}^{2^{m+1}-1} \sqrt{p_i^{(m+1)}} \, |i\rangle \tag{7}$$

which is an $(m+1)$-qubit discretisation of the function with half the discretisation error, as depicted in Figure 2. Fundamentally, we can begin this process with a trivial 1-qubit discretisation of the function. Repeating the process $n-1$ times will yield the target state $|\psi_n\rangle$ to an exponentially small error of $\mathcal{O}(2^{-n})$.
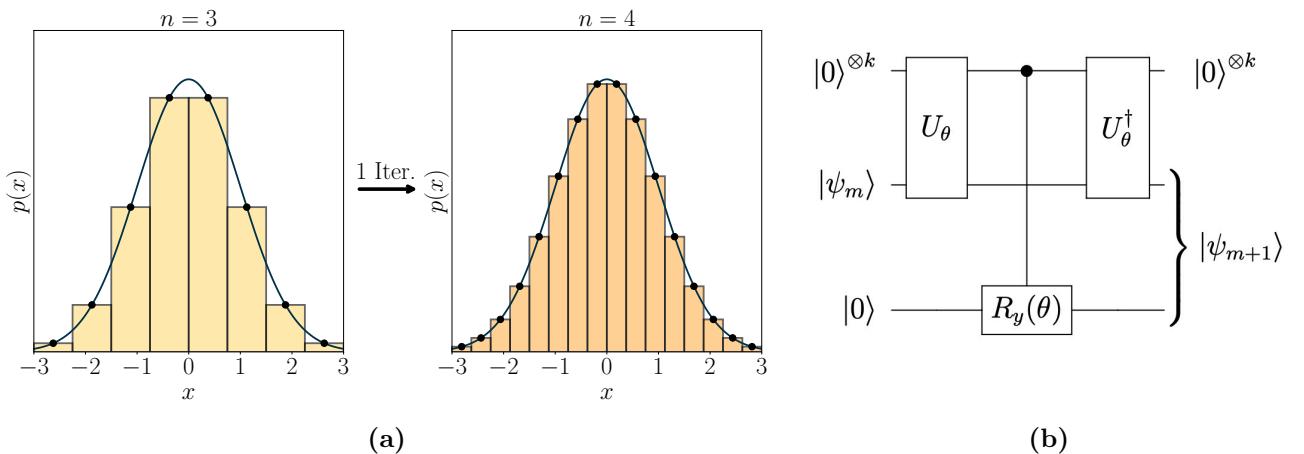

Figure 2: The Grover-Rudolph Algorithm: (a) The resolution of the target state doubles with each Grover-Rudolph iteration and (b) The circuit for the mth iteration of the algorithm.

If the target probability distribution has well-defined integrals, then numerical integration can be avoided in the computation of the rotation angles in Equation 5. This enables a more direct arithmetic approach to computing the rotation angles in parallel. Designing circuits for quantum arithmetic is non-trivial, partly due to the requirement that the computation be reversible. This can result in deep circuits with 100's of ancilla qubits [44]. Moreover, the design of circuits for each specific function is complicated, often requiring some form of optimisation to minimise resource costs. This alludes to a critical underlying issue with the widespread assumption that arbitrary functions can be applied to the basis states of quantum computers:

While theoretically efficient, the practical implementation of function-based transformations in the computational basis is limited by the lack of pre-existing circuitry for complex operations.

Generally, the target probability distribution will not have a well-defined analytic integral. This implies that the integrals in Equation 5 must be computed using some form of numerical integration, such as classical Monte Carlo simulation in quantum parallel. The error in such an approach scales as $\mathcal{O}(\frac{1}{\sqrt{N}})$ where N is the number of samples, an additional error, which eliminates the quadratic speedup of quantum Monte Carlo integration when the Grover-Rudolph algorithm is used for state preparation [45]. This is a severe limitation for applications in quantum finance, many of which depend on the quadratic speedup of quantum Monte Carlo integration [46]. In fact, this quadratic speedup is expected to be lost in *any* state preparation regime utilising classical Monte Carlo simulation in its computation. The circuitry to implement such a routine is not readily available and could be expected to accumulate significant resource costs. Hence, the Grover-Rudolph algorithm is only feasibly achievable when the rotation angles can be easily computed in parallel and does not currently appear to be an effective option for near-term quantum state preparation.

Digital-to-Analog Encoding

Digital-to-analog encoding refers to loading relevant state vector information into the computational basis of the quantum system (digital) before applying a unitary transformation to transfer this information into the amplitudes of the system (analog) [47–51]. This is the oracle unitary:

$$U_{D \to A} : |j\rangle_n |a_j\rangle_k \to a_j |j\rangle_n \tag{8}$$

where a_j is a k-bit approximation of the jth element of the normalised target vector **a**. The foundational algorithm to implement $U_{D \to A}$ was introduced by Grover in 2000 [47] with an asymptotic complexity of $\mathcal{O}(\sqrt{N})$, which is a quadratic speed up over arbitrary state preparation scaling as $\mathcal{O}(N)$. Much like the Grover-Rudolph algorithm, Grover's approach requires the computation of $\arcsin(x)$ via quantum arithmetic in computing the rotation angles of the $U_{D \to A}$ oracle. Using a non-arithmetic comparison operator in its place was proven to reduce the gate complexity by $\sim 10^2$ with the addition of $\mathcal{O}(n)$ ancilla qubits [49]. The gate complexity and ancilla requirement can be further decreased if we assume knowledge of the average bit-weight of the target vector [50].

These encoding schemes are commonly referred to as 'black-box' schemes, owing to the assumed existence of an amplitude-loading oracle:

$$U_A : |j\rangle_n |0\rangle_k \to |j\rangle_n |a_j\rangle_k \tag{9}$$

or some closely related oracle that prepares the k-bit-approximated vector elements in the computational basis. The lack of specification of the gate implementation of this oracle could be useful for quantum machine learning tasks where the inputs are derived directly from a quantum process and are not known classically [52]. Alternatively, U_A can be implemented efficiently if **a** is sparse [39]. In general, however, an efficient, practical implementation of U_A requires the assumption that the target vector is a discretised function, in which case the

amplitude-loading oracle can be reformulated as:

$$U_A : |x_j\rangle_n |0\rangle_k \rightarrow |x_j\rangle_n |f(x_j)\rangle_k \tag{10}$$

where $f(x_j)$ represents a k-bit approximation of $f : \mathbb{R} \rightarrow \mathbb{R}$ evaluated at the point x_j.

Variational Algorithms for State Preparation

Machine learning-based variational quantum algorithms (VQAs) have been widely applied for state preparation of probability distributions [53] and quantum chemistry [54–58]. VQAs train a parameterised quantum circuit (PQC) to extremise a problem-specific cost function. While VQAs generate (NISQ-friendly) shallow circuits with minimal ancilla qubits [59,60], they suffer from critical trainability problems derived from the near-ubiquitous barren plateau phenomena and local minima in the optimisation landscape [21,61]. The barren plateau is formally defined by the condition:

$$\text{Var}\left(C(\boldsymbol{\theta})\right) \leq b^n \quad \text{for} \quad 0 < b < 1. \tag{11}$$

This implies the exponential suppression of the variance in the cost function as the number of qubits n in the system grows. In turn, cost function gradients decay exponentially, resulting in severe inefficiency in minimising $C(\theta)$ for large n [2, 20, 21]. This is not to say that the variance in the cost function is zero everywhere: the barren plateaus typically emerge because of a severe concentration of cost function variance in an exponentially small region of the parameter landscape, which is called the narrow gorge [20]. The remaining parameter landscape is featureless and flat.

Barren plateaus can be avoided in special cases where a *local* cost function is utilised, and the circuit depth of the PQC is sufficiently shallow, i.e. less than $\mathcal{O}(\text{poly}(\log(n)))$ [21]. The shallow depth of these circuits constrains their expressibility, i.e. they can only express states belonging to a small subset of the state space. It has recently been suggested that VQAs with such limited expressibility are likely *classically efficient* to train in any case, eliminating the potential quantum advantage of VQAs [62]. This is not a severe restriction for applying VQAs to state preparation since the quantum advantage is anticipated to emerge in the subsequent quantum algorithm rather than from state preparation itself. Nonetheless, VQA-based state preparation is heavily constrained in terms of what it can achieve.

3. Matrix Product State Encoding

In the previous section, we outlined that state preparation techniques dependent on deep VQAs without efficient pre-training generally suffer from barren plateaus. We also noted that the assumption of shallow-depth circuits for quantum arithmetic may require many ancilla qubits and is likely not a permissible approach in the near term. In contrast, MPS-based state preparation techniques have recently become popularised as a resource-efficient approach that avoids the pitfalls of VQAs and quantum arithmetic [25–30]. This section introduces the MPS format alongside the MPD and MPD+TNO algorithms.

Method	Depth	Ancillas	Avoids Arithmetic	Scope
Space Efficient [14]	$\mathcal{O}(2^n)$	0	✓	Any quantum state. Exponential depth.
Time Efficient [33]	$\mathcal{O}(n)$	$\mathcal{O}(2^n)$	✓	Any quantum state. Exponential ancillas.
Grover-Rudolph [18]	$\mathcal{O}(\text{poly}(n))$	*Varies**	✗	Functions with $\mathcal{O}(\text{poly}(n))$ arithmetic circuits.
Grover Black-Box [47]	$\mathcal{O}(\sqrt{2^n})^\dagger$	*Varies**	✗	Functions with $\mathcal{O}(\text{poly}(n))$ arithmetic circuits.
Variational Quantum Algorithms	*Varies*	*Limited*	✓	Prone to barren plateaus and local minima.

Table 1: A summary of (non-MPS) quantum state preparation methodologies. *The number of ancilla qubits is generally predominated by the ancilla qubits required by quantum arithmetic, which can be substantial [44] †Worst case rounds of amplitude amplification.

3.1 Formulation

The matrix product state (MPS) or tensor train (TT) format of tensor networks is a factorisation of a tensor with N indices into a chain-like product of three-index tensors [22]. The MPS of an n-qubit system with open boundary conditions is given by:

$$|\psi\rangle = \sum_{i_1,...,i_n=\{0,1\}} A^{[1]i_1}_{\alpha_1} A^{[2]i_2}_{\alpha_1,\alpha_2}...A^{[n-1]i_{n-1}}_{\alpha_{n-2},\alpha_{n-1}} A^{[n]i_n}_{\alpha_{n-1}} |i_0 i_1...i_n\rangle \tag{12}$$

where $A^{[k]i_k}_{\alpha_{k-1},\alpha_k}$ is the MPS core of dimension $(\alpha_{k-1}, 2, \alpha_k)$ corresponding to the kth qubit, α_{k-1} is the left virtual index, α_k is the right virtual index, and i_k is the physical index. The left and right virtual indices connect each MPS core to its neighbours, with the left virtual index of $A^{[0]}$ and the right virtual index of $A^{[n]}$ set to 1 by the convention of open boundary conditions [63]. The bond dimension of the MPS is defined as $\chi := \max_k \alpha_k$ (that is, the bond dimension χ of an MPS is defined as the maximum of the dimensions of all virtual indices). An intuitive way to conceptualise the MPS construction is to imagine a one-dimensional spin-lattice consisting of n sites, where the virtual indices carry information related to the entanglement between sites, as illustrated in Figure 3.
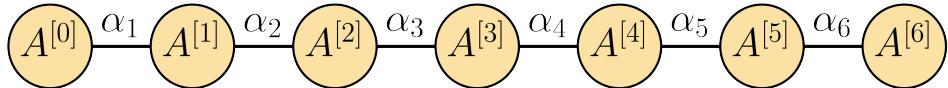

Figure 3: A toy model visualisation of the virtual indices α_j connecting $n = 7$ MPS cores $A^{[j]}$ with open boundary conditions. Each core corresponds to a qubit, and α_j represents correlations.

The open boundary MPS representation of a quantum state is non-unique. We can reduce

the gauge redundancy in the representation by expressing the MPS in left-canonical form:

$$\sum_{i_1,\alpha_1} A^{[1]i_1}_{\alpha_1}(A^{[1]i_1}_{\alpha_1})^\dagger = 1,$$

$$\sum_{i_k,\alpha_k} A^{[k]i_k}_{\alpha_{k-1},\alpha_k}(A^{[k]i_k}_{\alpha_{k-1},\alpha_k})^\dagger = \delta_{\alpha_{k-1}\alpha'_{k-1}}, \tag{13}$$

$$\sum_{i_{n-1},\alpha_{n-1}} A^{[n]i_n}_{\alpha_{n-1}}(A^{[n]i_n}_{\alpha_{n-1}})^\dagger = \delta_{\alpha_{n-1}\alpha'_{n-1}}$$

which defines the orthonormality constraints of the MPS.

3.2 Approximating Quantum States

An n-qubit MPS of bond dimension χ contains n tensors of dimension at most $(\chi, 2, \chi)$. Hence, at most $2n\chi^2$ elements are required to specify the MPS. The exact specification of an arbitrary n-qubit state requires a bond dimension $\chi = \mathcal{O}(2^{n/2})$. We can optimally approximate states, however, using fixed-χ representations, improving the classical efficiency of the representation at the expense of an approximation error.

Consider a bipartition of a pure n-qubit quantum state $|\Psi_{AB}\rangle$ into subsystems A and B containing j and $n-j$ sites, respectively. By reshaping $A^{[j-1]}$ and $A^{[j]}$ into a single matrix $M_{\mu\nu}$, we can compute the singular value decomposition (SVD) and truncate the smallest s values. This truncates the virtual index α_j to $\alpha'_j = \alpha_j - s$. By the Eckart-Young-Mirsky theorem [64,65], this is an optimal approximation of the quantum state. The Frobenius norm error introduced by this truncation is given by,

$$\epsilon^{(F)}_{T,k} = \sqrt{\sum_{i=\alpha'_j+1}^{\alpha_j} \sigma_i^2} \tag{14}$$

where σ_i denotes the singular values. In other words, the truncation error is the L^2-norm of the s singular values discarded. By iterating across all $n-1$ possible bipartitions of the state, we can form an optimal fixed-χ approximation of $|\Psi_{AB}\rangle$ satisfying $\forall j, \alpha'_j \leq \chi$.

The maximum entanglement entropy across two entropy S across any possible bipartition of an n-qubit state represented by an MPS is bounded by,

$$S \leq \log_2 \chi \tag{15}$$

where S is the von Neumann entropy across the bipartition and χ is the bond dimension of the MPS [22]. Equation 15 bounds the entanglement entropy that a fixed-χ MPS can express.

The MPS is classically efficient if $\chi \leq \mathcal{O}(\text{poly}(n))$, meaning that efficient MPS can precisely express states with entanglement entropy scaling poly-logarithmically in n. We are specifically interested in states that obey an area law of entanglement entropy, satisfying $S = \mathcal{O}(1)$ for one-dimensional states. Area-law states are characterised by entanglement entropy that grows with the boundary of the partition rather than the volume. This means that area-law states have an approximation error independent of n such that $\chi = \mathcal{O}(1)$. Area-law states are characterised

by exponentially decaying singular values, meaning that fixed-χ MPS approximations of such states can be efficient, accurate, and scalable [22].

Algorithm 1 MPS From State Vector [66]

Require: n-qubit state $|\psi\rangle$, optional bond dimension χ
Ensure: MPS tensors $\{A^{[1]}, \ldots, A^{[n]}\}$ with bond dimension χ
 1: Express $|\psi\rangle = \sum_{i_1,\ldots,i_n} \psi(i_1,\ldots,i_n) |i_1,\ldots,i_n\rangle$
 2: Reshape ψ to matrix $B(i_1; i_2,\ldots,i_n)$ of size $2 \times 2^{n-1}$
 3: Perform SVD: $B = U\Sigma V^\dagger$
 4: Set $A^{[1]}_{i_1,\alpha_1} \leftarrow U(i_1,\alpha_1)$
 5: Reshape $V^\dagger$ to $B(\alpha_1, i_2; i_3,\ldots,i_n)$
 6: **for** $k = 2$ to $n-1$ **do**
 7:   SVD: $B = U\Sigma V^\dagger$
 8:   Set $A^{[k]}_{\alpha_{k-1},i_k,\alpha_k} \leftarrow U(\alpha_{k-1}, i_k, \alpha_k)$
 9:   Reshape $V^\dagger$ to $B(\alpha_k, i_{k+1}; i_{k+2},\ldots,i_n)$
10: **end for**
11: Set $A^{[n]}_{\alpha_{n-1},i_n} \leftarrow B(\alpha_{n-1}, i_n)$
12: **if** fixed bond dimension χ **then**
13:   Truncate singular values to top χ at each SVD step
14: **end if**
15: **return** $\{A^{[1]}, \ldots, A^{[n]}\}$

3.3 Matrix Product Disentangler (MPD) Algorithm

An MPS with n sites and bond dimension χ can be exactly prepared with a single layer of (at most) $(\log_2 \chi + 1)$-qubit unitaries [23–25], i.e. elements of the unitary group $U(2\chi)$. Any q-qubit gate can be decomposed into a circuit of depth $\mathcal{O}(4^q)$ one- and two-qubit gates using the Quantum Shannon Decomposition [67]. Using $q = \log_2 \chi + 1$, we have that the circuit depth of the exact MPS preparation scheme scales as $\mathcal{O}(n\chi_{\max}^2)$ where big-O notation may hide significant scale factors in the decomposition process. Hence, we are motivated to adopt a more approximate MPS encoding scheme that avoids the polynomial dependence of circuit depth on χ, such as the MPD algorithm.

The matrix product disentangler (MPD) algorithm introduced by Ran in 2020 [25] approximately maps an n site MPS using L sequential layers of 2-qubit $U(4)$ unitaries, as depicted in Figure 4. Each layer is computed iteratively using an analytic method detailed in Algorithm 1. The layers are computed in reverse, beginning with the original MPS $|\psi\rangle = |\psi_1\rangle$ (i.e., $|\psi_1\rangle$ is initialised to the original MPS). At the kth iteration, the MPS $|\psi_k\rangle$ is approximated by a $\chi = 2$ MPS denoted by $|\tilde{\psi}_k\rangle$. Then, the kth MPD unitary $\hat{U}_k$ is computed such that $\hat{U}_k |\tilde{\psi}_k\rangle \approx |0\rangle^{\otimes n}$, before the MPS is updated using $|\psi_{k+1}\rangle = \hat{U}_k |\psi_k\rangle$. Note that we can reduce computational costs by setting a maximum bond dimension $\chi_{\max}$ for $|\psi_k\rangle$ for all $k = 1, ..., L$. Since $\chi_{\text{exact}} = \mathcal{O}(2^{n/2})$, setting $\chi_{\max} < \chi_{\text{exact}}$ is essential for large n. The unitary $\hat{U}_k$ is called the MPD unitary because it approximately disentangles $|\psi_k\rangle$ to $|\psi_{k+1}\rangle$. Each MPD unitary is a $\chi = 2$ approximation, which can be exactly mapped into a single layer of $U(4)$ gates [23]. After repeating the disentangling process until $k = L$, we are left with L MPD unitaries that can each be collectively mapped onto L layers of $U(4)$ gates similar to the circuit in Figure 4.

The MPD unitaries act such that $\hat{U}_1\hat{U}_2\cdots\hat{U}_L|\psi\rangle \approx |0\rangle^{\otimes n}$ where $|\psi\rangle$ is the target state. It follows that $\hat{U}_L^\dagger\hat{U}_{L-1}^\dagger\cdots\hat{U}_1^\dagger|0\rangle^{\otimes n} = |\psi'\rangle \approx |\psi\rangle$. Hence, we are interested in applying a sequence of inverse MPD unitaries $\{\hat{U}_k^\dagger\}_{k=1}^L$ to the ground state, which enables the target state to be approximately prepared. At iteration k, the quality of the MPD unitary depends on the $\chi = 2$ approximation of $|\psi_k\rangle$, meaning that the algorithm's performance depends on these approximations. If the $\chi = 2$ approximation is very poor, then the MPD unitary $\hat{U}_k$ barely disentangles $|\psi_k\rangle$ and the additional circuit layer produced by the MPD algorithm does not contribute to the quality of the prepared state. In the extreme worst case, the MPD algorithm prepares the $\chi = 2$ approximation of $|\psi\rangle$, and subsequent layers do not contribute. This is consistent with the findings in [29] that the MPD approach exhibits diminishing gains in fidelity with increasing layers, and it is not evident that the state preparation error $\epsilon_S \to 0$ as $L \to \infty$. The quality of the $\chi = 2$ approximations corresponds to the decay rate of the singular values in the MPD, meaning that the MPD algorithm will perform best in preparing states with rapidly decaying singular values. A key pitfall, however, is that no rigorous performance guarantees exist.
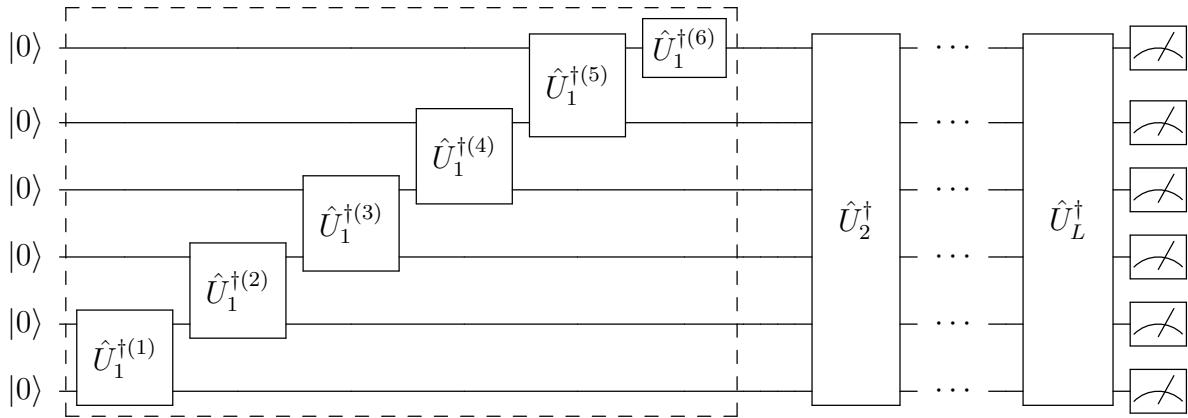

Figure 4: The Staircase Circuit for the MPD and MPD+TNO Algorithms

---

Algorithm 2 Analytic Matrix Product Disentangler (MPD) Algorithm [25]

Require: Target MPS $|\psi_1\rangle$ with bond dimension $\chi_{\max}$, number of layers L
Ensure: Set of unitary layers $\{\hat{U}_k^\dagger\}_{k=1}^L$
 1: Initialize $k \leftarrow 1$
 2: **while** $k \leq L$ **do**
 3:    Approximate $|\psi_k\rangle$ by $|\tilde{\psi}_k\rangle$ with bond dimension $\tilde{\chi} = 2$
 4:    Compute the MPD $\hat{U}_k$ such that $\hat{U}_k|\tilde{\psi}_k\rangle \approx |0\rangle^{\otimes n}$
 5:    Approximately disentangle $|\psi_k\rangle$ to $|\psi_{k+1}\rangle$ using $|\psi_{k+1}\rangle = \hat{U}_k|\psi_k\rangle$
 6:    Update $k \leftarrow k + 1$
 7: **end while**
 8: **return** The set $\{\hat{U}_k^\dagger\}_{k=1}^{L-}$
 9: **Note:** The circuit $U_S := \hat{U}_L^\dagger\hat{U}_{L-1}^\dagger\cdots\hat{U}_1^\dagger$ applied to the zero state $|0\rangle^{\otimes n}$ approximately reconstructs the input MPS $|\psi_1\rangle$

3.4 Tensor Network Optimisation (TNO)

We aim to investigate the efficacy and applications of the MPD algorithm combined with Tensor Network Optimisation (TNO), which we denote as the MPD+TNO algorithm. To begin with, we compute the MPD algorithm with L circuit layers. The output of the MPD algorithm is a quantum circuit consisting of L layers of $U(4)$ unitaries (see Figure 4) that prepare an approximation of the target state. This quantum circuit is decomposed into arbitrary single-qubit rotation gates and CNOT gates. Each single-qubit rotation is parameterised by 3 Euler angles, and the total number of parameters in the circuit scales as $\mathcal{O}(nL)$ (with n being the number of qubits). The parameterised quantum circuit is represented as a tensor network to facilitate efficient contraction and differentiation.

The cost function is defined as,

$$L(\theta) = 1 - |\langle \psi_{\mathrm{mps}} | \psi_{\mathrm{circuit}}(\theta) \rangle|^2 \tag{16}$$

where ψ_{mps} is the target state represented as an MPS with bond dimension χ_{max} and expressed in left-canonical form, and $\psi_{\mathrm{circuit}}(\theta)$ is the parameterised quantum circuit expressed as a tensor network. We can compute this cost function efficiently by leveraging efficient tensor contractions between the parameterised circuit and the target MPS (i.e., it circumvents the exponential cost associated with full state-vector simulation).

Gradient descent is computed via the widely employed Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B), which is a quasi-Newton method that approximates the Hessian matrix H (the second derivative of $L(\theta)$ w.r.t. θ). The parameters at iteration $t+1$ are updated as follows,

$$\theta^{(t=1)} = \theta^{(t)} - \eta H_t \nabla L(\theta^{(t)}) \tag{17}$$

where θ is the set of parameters, η is the learning rate, and H_t is the approximate Hessian matrix at iteration t. The gradients are computed with automatic differentiation.

The efficiency of this scheme depends on the degree of initial overlap $|\langle \psi_{\mathrm{mps}} | \psi_{\mathrm{MPD\ circuit}} \rangle|$. This can be thought of as initialising the parameters near the global minimum, reducing the number of iterations required to converge and avoiding suboptimal local minima [68]. The resulting quantum circuit can be retrieved via efficient contractions of the tensor network [22, 69]. Potentially, alternative approaches to gradient descent and optimisation strategy could be considered. For instance, it was shown by Rudolph et al. [26] that iteratively optimising each additional layer (rather than all the layers at once) can improve training efficiency in some instances. This alternative approach comes with a greater (quadratic) dependence of complexity on the number of iterations T scaling as $\mathcal{O}(nT^2 L \chi^3)$.

An alternative TNO strategy developed by Melnikov et al. [70] was also considered. This scheme employs the hardware efficient ansatz (HEA) consisting of layers of single-qubit rotations and CNOT gates, optimised via Riemannian optimisation over the Stiefel manifold. This approach initialises the circuit parameters by training subsets of the quantum circuit represented as MPSs, before recombining them and training on the full quantum circuit. However, this algorithm's performance was unsatisfactory, owing to the relatively low overlap with the

global minimum obtained from this initialisation strategy. In contrast, using the MPD algorithm to initialise parameters before TNO often led to significant overlap between the initial state and the global minimum, significantly increasing training efficiency. This illustrates the importance of high-quality parameter initialisation for training efficiency in MPS-based optimisation strategies.

3.5 Complexity Analysis

We consider the complexity analysis of the MPD and MPD+TNO methods, where n denotes the number of qubits, L denotes the number of layers, T denotes the number of optimisation iterations, and $\chi_{\max}$ denotes the maximum bond dimension used to approximate the target state. We consider the complexity of the following components of each algorithm:

- The representation of a function via cross-approximation has time complexity $\mathcal{O}(n\chi_{\max}^2)$ [71].

- The contraction of a quantum circuit in tensor product form is dominated by a complexity linear in the number of qubits and circuit layers $\mathcal{O}(nL)$. Note that the dimensions arising from this contraction are much larger than reasonable values of $\chi_{\max}$, meaning that the dependence of circuit contraction on $\chi_{\max}$ is negligible [70].

- The memory requirements of the MPS scale as $\mathcal{O}(n\chi_{\max}^2)$. The cost of bringing the MPS into canonical form is $\mathcal{O}(n\chi_{\max}^3)$ [63].

- It follows that the MPD algorithm scales as $\mathcal{O}(nL\chi_{\max}^3)$ and the MPD+TNO algorithm as $\mathcal{O}(nL\chi_{\max}^3 * T)$ where T is the number of iterations required to converge to a solution.

- We must also consider the cost of decomposing arbitrary $U(q)$ unitaries into elementary gates. The Quantum Shannon Decomposition approach to this problem encompasses a $\mathcal{O}(8^q)$ classical overhead and a circuit depth $\mathcal{O}(4^q)$ for each q-qubit gate [72]. In the special case of $q = 2$, any $U(4)$ unitary can be decomposed into *at most* 3 CNOT gates and 15 elementary single-qubit operations, which is optimal [73]. Hence, the big-O notation in the MPD and MPD+TNO algorithms hides only modest scale factors. Accounting for elementary gate decomposition into single-qubit rotations and CNOT gates, the elementary circuit depth in the exact method of [23] is $\mathcal{O}(n \cdot 4^{\log_2(\chi_{\max}+1)}) = \mathcal{O}(n \cdot 4\chi_{\max}^2) = \mathcal{O}(n\chi_{\max}^2)$.

4. MPS Preparation of Functions

This section explores the viability of the MPD and MPD+TNO algorithms for preparing states representing a vast array of discretised elementary and polynomial functions. We ultimately find the MPS approach exceptionally efficient in preparing high-fidelity approximations of functions up to low-degree piecewise polynomials with linear-depth circuits and efficient classical preprocessing costs.

Method	Circuit Type	Time	Depth	Ancillas	Comments
Exact [23, 66]	Single Layer of $U(2\chi_{\max})$ Staircase	$\mathcal{O}(nL\chi_{\max}^3)$	$\mathcal{O}(n\chi_{\max}^2)^*$	1	*After decomposition
MPD [25]	L Layers of $U(4)$ Staircase	$\mathcal{O}(nL\chi_{\max}^3)$	$\mathcal{O}(nL)$	0	Modest hidden scale factors
Melnikov et al. TNO [70]	L Layers of Hardware Efficient Ansatz	$\mathcal{O}(nTL\chi_{\max}^3)$	$\mathcal{O}(nL)$	0	MPS subsets trained to initialise parameters
MPD+TNO [26, 29]	L Layers of $U(4)$ Staircase	$\mathcal{O}(nTL\chi_{\max}^3)$	$\mathcal{O}(nL)$	0	MPD algorithm used to initialise parameters
MPD+Layer-by-Layer TNO [26]	L Layers of $U(4)$ Staircase	$\mathcal{O}(nT^2L\chi_{\max}^3)$	$\mathcal{O}(nL)$	0	Full MPS optimised with each new layer

Table 2: A comparison of the complexity of MPS encoding schemes.

4.1 Previous Work and Gaps

MPS encoding has been identified as a highly NISQ-friendly framework for state preparation [25–29, 74]. Holmes & Matsuura (2020) [75], motivated by the bound on entanglement entropy for discretised functions [76], explored applications of the MPS framework for the state preparation of smooth, differentiable functions. In particular, [75] proposed the computation of the MPS corresponding to a piecewise polynomial approximation of the target function before the MPD algorithm is used to construct the quantum circuit. [75] ultimately demonstrated the efficacy of this scheme for encoding $\chi = 2$ approximations of the Gaussian, Lognormal, and Lorentzian family of functions. Additionally, Iaconis et al. (2024) [77] employed the MPD algorithm to encode Gaussian functions in the *probabilities* of the state. However, it remains an open question about how this approach can scale to functions necessitating larger bond dimension approximations.

Recently, TNO methods for MPS preparation have emerged as viable encoding schemes [26, 29, 70]. While some benchmarking of these methods has been studied [26], their *practical* capabilities, especially in the context of the state preparation of functions, have been explored to a limited extent. Hence, the practical application of functions is chosen within this paper as a framework in which the capabilities of the MPD algorithm and the MPD+TNO scheme can be quantified.

Another critical research gap in the practical decomposition of MPS into quantum states is the circuit depth required to accurately prepare a given fixed-χ MPS. This largely stems from the distinctive individual singular value decay profile characterising any target state. The theoretical lower-bound on the exact representation of the class of translation-invariant normal MPS (i.e. MPS where all cores are equal and the singular values decay exponentially) is $\Omega(\log n)$ [27]. The theoretical lower bounds can sometimes be violated with access to measurements [27, 30].

MPS of constant bond dimension can be exactly prepared with a unitary circuit of depth $\mathcal{O}(n)$ [23, 24, 78]. This follows the exact decomposition method of Schön (2005) [23], which maps

the MPS into a circuit of arbitrary $(\log_2 \chi + 1)$-qubit unitaries. To be precise, each virtual index α can be mapped to an $(\log_2 \alpha + 1)$-qubit unitary, which forms a single-layer sequential circuit similar to Figure 4 except with larger multi-qubit unitaries. The theoretical lower-bound on the unitary decomposition of a q-qubit unitary into one- and two-qubit gates is $\Omega(\frac{1}{4}4^q)$ CNOT gates [67], and the Quantum Shannon Decomposition can achieve a scaling of $\mathcal{O}(4^q)$ circuit depth [72]. This implies that the decomposition of the circuit in [23] results in a depth of $\mathcal{O}(n \cdot 4^{\log_2 \chi + 1}) = \mathcal{O}(4n\chi^2)$. However, for

However, this approach carries the burden of a $\mathcal{O}(2^n)$ classical overhead in forming the circuit initially [66], which motivates an approximate encoding scheme [25]. Nonetheless, this implies that a circuit of depth $\mathcal{O}(4n\chi^2)$ can exactly encode the MPS.

4.2 MPS Representations of Discretised Functions

States Representing Discretised Functions

We consider target states of the form given by,

$$|\psi\rangle = \frac{1}{||f||_2} \sum_{j=0}^{N-1} f(x_j) |j\rangle \tag{18}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a univariate real-valued function defined over the domain $D = [a, b]$, $x_j = a + j\frac{b-a}{N-1}$ for a uniform discretisation, $N = 2^n$ is the dimension of the n-qubit state vector $|\psi\rangle$, and $||f||_2$ is the Euclidean norm.

We first investigate the capabilities of MPS structures to represent functions with small bond dimensions accurately. The limiting factor in MPS state preparation is the bond dimension that can be prepared by the approximate quantum circuit encoding. Hence, we must ask what states can be represented by small bond dimensions without significant errors. We measure this by the truncation error, defined as,

$$\epsilon_T = 1 - |\langle\psi|\tilde{\psi}\rangle|^2 \tag{19}$$

where $|\psi\rangle$ represents the exact target state and $|\tilde{\psi}\rangle$ represents the approximated target state achieved by optimally truncating the bond dimension to $\chi < \chi_{\text{exact}}$. In other words, the truncation error is the infidelity of the optimally approximated target state of a specified bond dimension χ.

The discretisation error of a real-valued and smooth function across a uniform grid of 2^n points in the domain $D = [a, b] \subset \mathbb{R}$ scales as $\mathcal{O}(2^{-n})$. Moreover, the maximum amount of von Neumann entropy (across any bipartition of the quantum system) that can be introduced when adding one qubit to a state representing a real-valued and smooth discrete function is bounded by,

$$\Delta S \leq \frac{(b-a)\sqrt{||f'||_\infty}}{2^{n/2-1}} = \mathcal{O}(2^{-n/2}) \tag{20}$$

where $||f'||_\infty = \max_{x \in [a,b]} |f'(x)|$ and $f(x)$ is discretised over the domain $[a, b]$ [76]. Assuming that $f(x)$ is Lipschitz continuous (it has bounded derivatives almost everywhere), then the

upper bound is dominated by the exponential decay with system size n. This implies that functions representative of discrete functions generally have limited entanglement and weak correlations, which typically grow with the maximum derivative of the function in the relevant domain. Consequently, we expect many discrete functions to have limited entanglement entropy, making them suitable for efficient and accurate MPS representations. Additionally, consider fitting a polynomial to an arbitrary, random state vector. The maximum derivative of such a function would grow unbounded as qubits are added to the system, allowing the state to accumulate a near maximum amount of entanglement entropy and exhibit long-range solid correlations. Hence, this framework quantifies the transition from limited entanglement entropy in straightforward discrete functions to large entanglement entropy expected of random, arbitrary state vectors.

Bond Dimension Considerations

There is an explicit MPS construction for polynomials $f(x) = \sum_{j=0}^{d} a_j x^j$ of degree d with a bounded $\chi \leq d + 1$ [79]. [75] extended this construction to degree-d piecewise polynomials defined across 2^I subdomains (i.e. I piecewise intervals), which accumulates a maximum bond dimension of $\chi = 2^I(d + 1)$. These provide upper bounds on the exact χ required to represent these states. By considering the exact polynomial required to represent a given function, we can determine the maximum bond dimension to express it precisely. However, in many cases, a function can be well-approximated by a polynomial of a much lower degree. Additionally, the MPS of functions with periodic behaviour, like sinusoidal functions, can permit near-exact MPS representations with low bond dimensions [79]. The transition in the fidelity of the prepared state with increasing bond dimension can be visualised in the simple example of a Gaussian function in Figure 5.
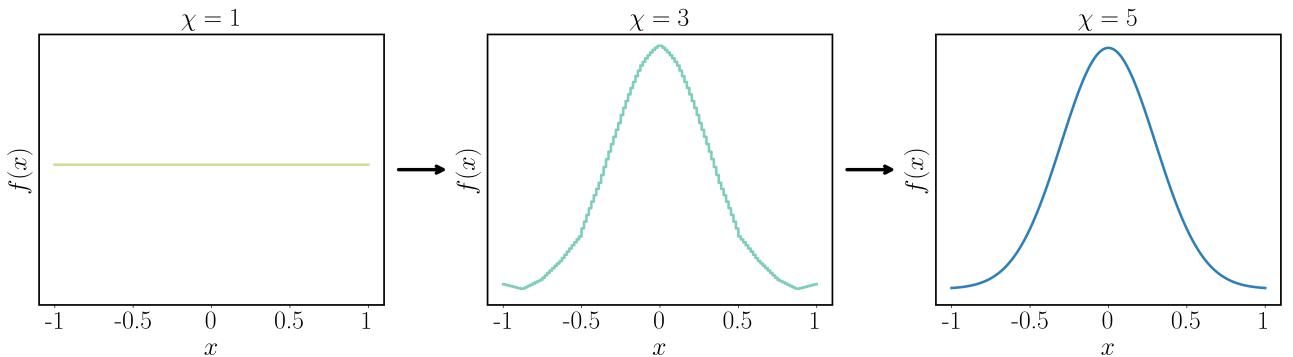

Figure 5: The visual improvement in the fidelity of the MPS representation of the discretised Gaussian function $f(x) \sim N(0, 0.3^2)$ with increasing bond dimension ($n = 12$).

Elementary Functions

First, we investigated the approximate representation of discretised elementary functions, such as straightforward probability distributions. We began by reproducing the findings of [75] that MPS structures can efficiently represent Gaussian functions $f(x) \sim N(0, \sigma^2)$. Rather than assessing the $\chi = 2$ representation of piecewise polynomials (fit to Gaussian functions), as is done in [75], this study delineated the *exact* truncation error (ϵ_T) of a fixed-χ MPS across a

range of χ values. Results for Gaussian functions of varying σ are presented in Figure 6(a). The truncation error is universally larger at each χ as σ decreases. This aligned with the expectation that the increased maximum gradient in these functions raises the bound on entanglement entropy scaling in Equation 20. This effect was referred to as a "squeezing" of the function in [75]. By Equation 15, the bond dimension required to capture the increased entanglement entropy must also increase. Equivalently, the singular values in the SVD across any bipartition decay more slowly, resulting in a larger error when truncating the bond dimension.
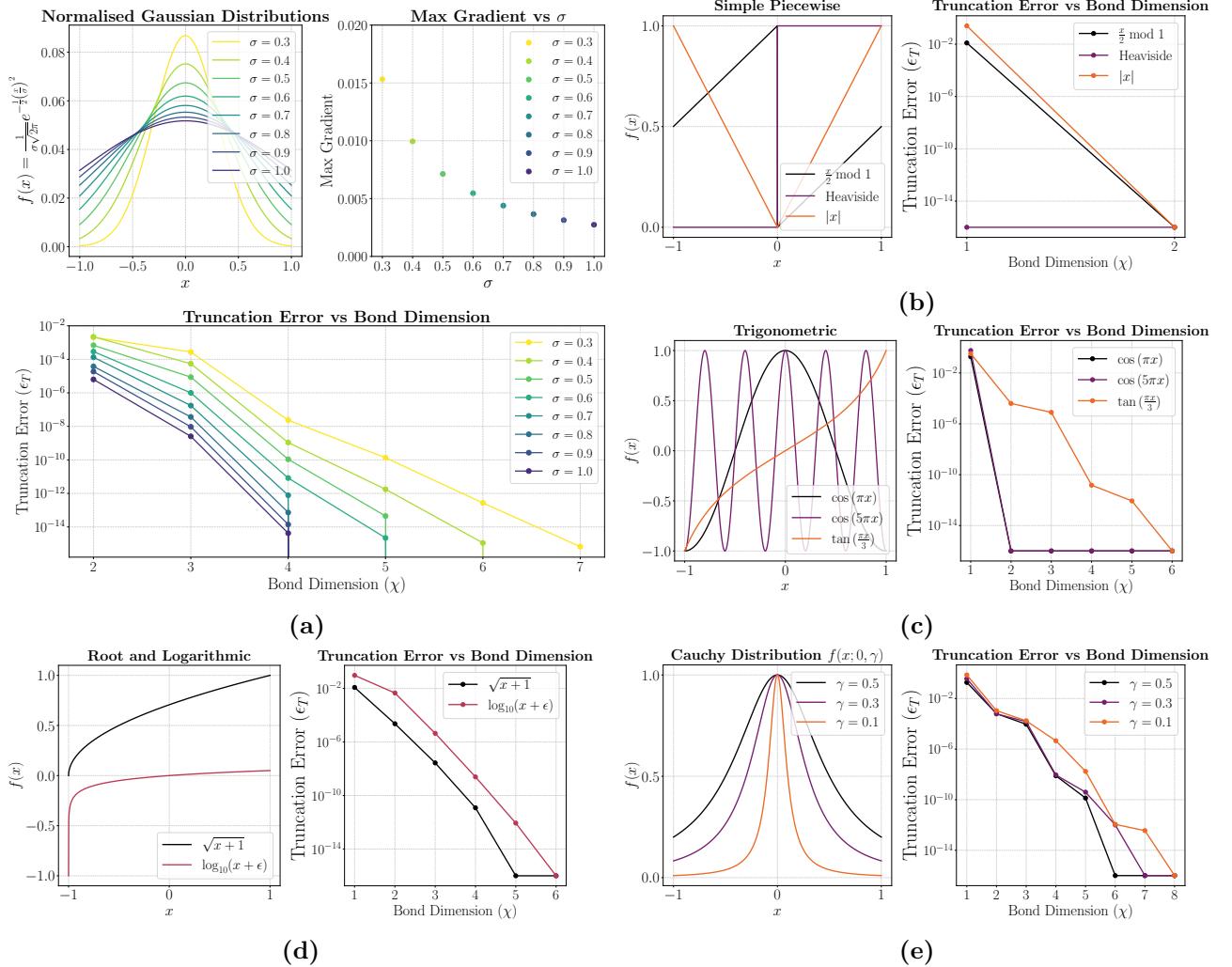

Figure 6: Truncation Error Analysis for Fixed-χ Elementary Functions. The truncation error for $n = 16$-qubit fixed-χ MPS representations is computed using Algorithm 1. The range of elementary functions explored includes (a) The Gaussian family, (b) Simple, linear piecewise functions, (c) Basic trigonometric functions, (d) Root and logarithmic functions, and (e) The Cauchy family. *$\epsilon = 1e - 6$ for the logarithmic function.

Figure 6(b)-(e) extended this work to a myriad of discretised functions defined on the domain $D = [-1, 1]$. This included simple piecewise linear functions, such as the Heaviside step function and $f(x) = |x|$, basic trigonometric functions, the root function $f(x) = \sqrt{x + 1}$, the logarithmic function $f(x) = \log_{10}(x + 1 + \epsilon)$ ($\epsilon = 1e - 6$), and the Cauchy distribution for varying γ parameters. The Heaviside step function is a separable state with an exact $\chi = 1$ representation. The linear and sinusoidal functions were demonstrated to have an exact $\chi = 2$ representation, which aligns with the exact bond dimension of the *explicit* MPS construction

of these states provided in [79]. Functions exhibiting increasingly irregular behaviour, with sharp gradients and asymmetries, were anticipated to necessitate a larger bond dimension χ. Nonetheless, all explored functions have a near-exact MPS representation with a bond dimension at most $\chi \sim 8$, owing to the predominantly smooth behaviour of the functions. This is despite the significant discontinuity in the modulo function $f(x) = \frac{x}{2} \mod 1$ and the singularities at $x = -1$ in the root and logarithmic functions, illustrating that these states have bounded entanglement entropy despite their irregular characteristics. It was also demonstrated that high-quality approximations of each state correspond to a much lower-than-exact bond dimension, exchanging increased efficiency for decreased accuracy in the MPS representation. This is compelling evidence that the MPS structures are adept at accurately representing quantum states representative of discretised functions, exploiting limited entanglement entropy to overcome the dimensionality burden through an efficiently compressed representation of the states.

The truncation error at varying 'singular value thresholds' refers to truncating all singular values below a minimum threshold to approximate the MPS. All truncation error results are computed for $n = 16$ qubits. However, the singular value decays are computed for $n = 8$ to help visualise the varying singular value decay rates. These results help to establish a visual intuition behind the capability of efficient MPS to accurately encode functions being dependent on the decay rate of the singular values in the SVD across each virtual index.

Polynomial Functions

The core results are presented in Figure 7 for a class of polynomials of varying functional complexity. This includes standard polynomials, cubic splines, discontinuous piecewise functions, and piecewise Chebyshev polynomials. The polynomial degree d and number of piecewise intervals I are varied widely, neatly capturing the transition between highly structured (low-degree, low-interval) and unstructured (high-degree, high-interval) target vectors. All results were consistent with the theoretical upper bounds derived in [75, 79]. As anticipated, the truncation error was positively correlated with d and I in nearly all cases. One exception was the class of $I = 10$ discontinuous piecewise functions in Figure 7(d), in which the truncation error is essentially independent of d. In such states, the degree of discontinuity in the functions is the dominant contribution to entanglement entropy, and the polynomial degree is only a minor contribution. Similar effects were observed in Figure 7(c), though the role of polynomial degree is comparatively more substantial since there are fewer discontinuities in the target vectors.

The class of moderate to low degree polynomial functions with limited piecewise intervals can ultimately be approximated with fixed-χ MPS to a reasonable level of accuracy. This is an encouraging result that MPS is a viable pathway to efficiently representing sufficiently structured information, especially in machine learning contexts where a small degree of noise in the prepared state is permissible and can even offer a natural form of regularisation [80]. As the information becomes more complex (i.e. described by a high-degree, high-interval Chebyshev polynomial), truncating the bond dimension can induce a consequential degree of error, and an accurate representation necessitates a large bond dimension representation. This presents a critical challenge in the design of efficient quantum circuits to encode these states accurately.
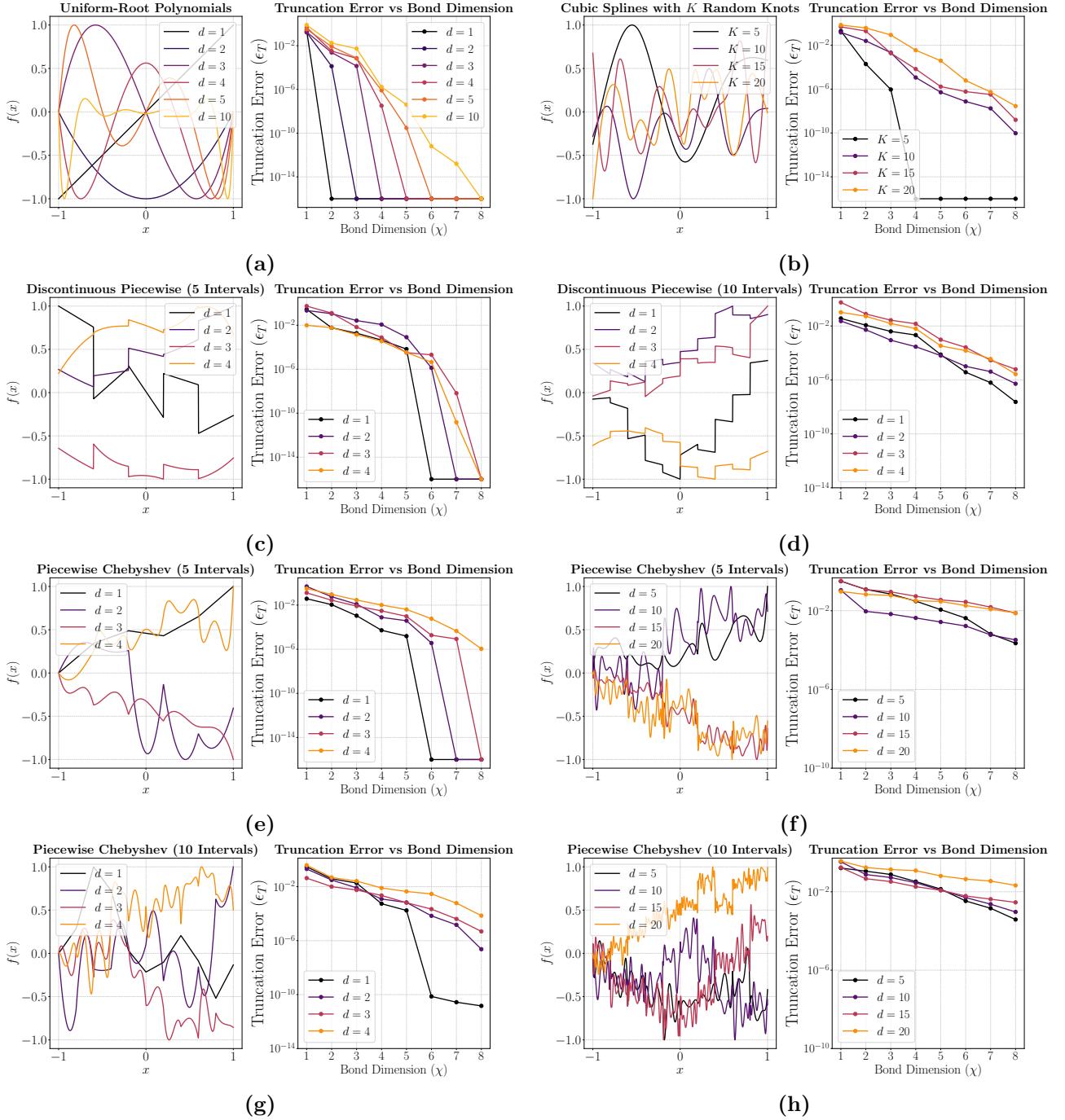
Figure 7: Truncation Error Analysis for Fixed-χ Polynomial Functions. The generated functions are only included for illustrative purposes so that the reader can gain an intuitive of the general level of complexity in each state. The MPS truncation error (ϵ_T) vs χ ($n = 16$) is plotted for (a) Polynomials with uniformly distributed roots, (b) Cubic splines connecting K randomly generated (x, y) points called knots, (c) Discontinuous piecewise polynomials (5 Intervals), (d) Discontinuous piecewise polynomials (10 Intervals), (e) Low-degree piecewise Chebyshev (5 Intervals), (f) High-degree piecewise Chebyshev (5 Intervals), (g) Low-degree piecewise Chebyshev (10 Intervals), (h) High-degree piecewise Chebyshev (10 Intervals). All polynomials are generated across $N = 65536$ points in the domain [-1,1]. All piecewise polynomials have uniformly distributed piecewise intervals and are generated using randomisation in the coefficients.

4.4 MPS Encoding Results

As a central point of comparison, the TNO strategy based on the scheme introduced by [70] was tested by encoding the sine function on 12 qubits across 5 layers of the hardware efficient ansatz (HEA) with up to 500 iterations. The sine function was found to be encoded with fidelity $\sim 10^{-4}$. While this is still a high-quality state, the sine function has an exact $\chi = 2$ representation. This means that the MPD algorithm [25] can prepare this state to machine precision with a single layer of arbitrary $U(4)$ gates with an extremely minimal computational cost. Crucially, the HEA-based TNO uses a random initialisation of parameters, whereas the staircase-based MPD+TNO explored in this paper has the advantage of using the MPD algorithm to initialise parameters. This comparative limitation severely limits the capability of the HEA-based scheme to train efficiently, leaving it exposed to barren plateaus and local minimum effects in its navigation towards the global minimum. This result was established for a point of comparison to the MPD+TNO algorithm to illustrate the significant role of parameter initialisation in determining training efficiency.
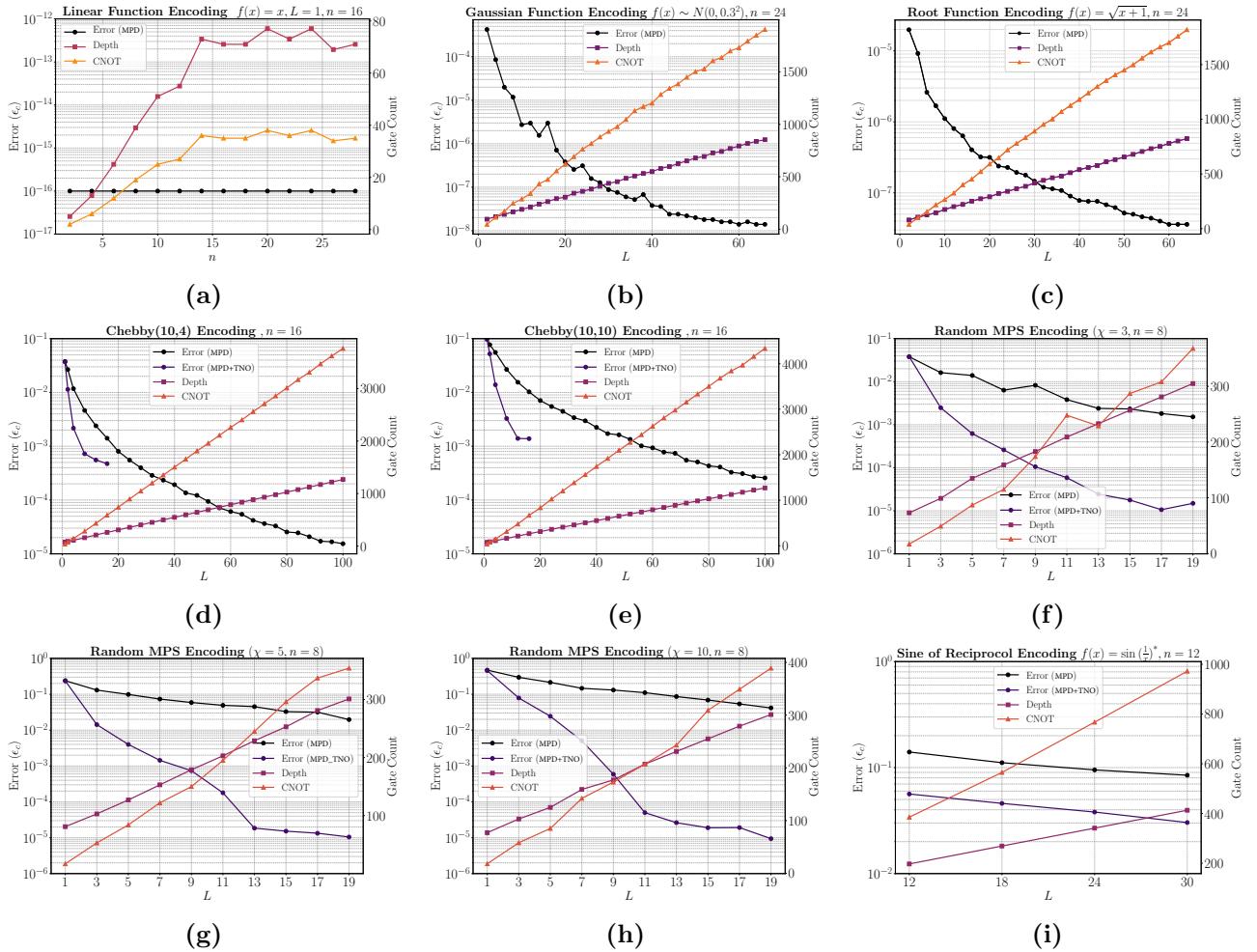

Figure 8: Benchmarks of the MPD and MPD+TNO Algorithms. $Cheby(I, d)$ denotes a randomly generated degree-d Chebyshev polynomial with I piecewise intervals. All functions were generated in the domain $D = [-1, 1]$. The CNOT and circuit depth gate counts were retrieved using the Qiskit decomposition of the arbitrary $U(4)$ unitaries into elementary quantum gates. 500 iterations were used for the MPD+TNO algorithm.

The quantum circuits produced by the MPD and MPD+TNO algorithms were simulated

for a range of target states, and the results are presented in Figure 8. We do not constrain $\chi_{\max}$ in the MPD algorithm in all trials. However, in practice, setting $\chi_{\max}$ such that the target state $|\psi\rangle$ can be approximated with sufficiently high fidelity will increase computational efficiency, without significantly reducing circuit performance (unless the target state cannot be well-approximated by an MPS with truncated bond dimension). We summarise the results with four key points:

1. The MPD scheme provides efficient, low-depth, and accurate circuits that prepare discretised functions with accurate low-bond dimension representations. This included a single-layer exact encoding of the linear line function in Figure 8(a), the Gaussian $f(x) \sim N(0, 0.3^2)$ in Figure 8(b), the root function $f(x) = \sqrt{x+1}$ in Figure 8(c), and low-degree piecewise polynomials in Figure 8(d)-(e). The fidelity of the encoded target state can be tuned by adjusting the number of layers L. However, the fidelity that can be achieved inevitably degrades as a larger bond dimension is required to represent the state accurately. Nonetheless, the MPD encoding scheme was numerically stable and efficient in classically generating quantum circuits up to a modest number of layers. The generated circuits are NISQ-friendly with CNOT gate complexity and circuit depth shown to scale linearly with L, aligning with the expected $\mathcal{O}(nL)$ circuit depth scaling.

2. The MPD algorithm generalises poorly beyond the context of the state preparation of low-bond dimension functions. This is because the MPD algorithm's accuracy and rate of convergence depend on the fidelity of the $\chi = 2$ approximation at each iteration. For example, the state vector discretising $f(x) = \sqrt{x+1}$ has a $\chi = 2$ approximation with fidelity of $F = 0.99998 \pm 0.00001$. On the other hand, truncating a random $\chi = 5$ MPS to $\chi = 2$ has a fidelity of $F = 0.36 \pm 0.11$ (10 trials). This is because random MPS tend to have fairly uniform singular values compared to the exponentially decaying singular values for well-behaved smooth functions. The performance of the MPD algorithm was inadequate in preparing sufficiently high-quality encodings of all random MPS explored, as well as the irregularly behaving $f(x) = \sin\left(\frac{1}{x}\right)$ function in Figure 8(i). In these states, the fidelity improves only marginally with increasing layers. This is consistent with earlier observations made by [26, 29] that the analytic MPD algorithm exhibits diminishing marginal gains in fidelity at each subsequent layer. It is important to note that the classical computational cost grows exponentially with L, making it computationally intractable to compute arbitrarily deep circuits to capitalise on these marginal gains.

3. The MPD+TNO scheme significantly improves the fidelity of the prepared target states. This effect is especially pronounced when the analytic MPD scheme performs poorly, as illustrated in the random MPS in Figure 8(e)-(h). It was also shown to improve the fidelity of the piecewise Chebyshev functions Figure 8(d)-(e) at low circuit depths (this was not extended to larger numbers of layers due to the high qubit count contributing to significant computational costs). This effect can be explained by the observation that that expressible χ grows as $\chi = \mathcal{O}(2^x)$ [25] with the exact encoding of a fixed-χ MPS requiring $\mathcal{O}(n\chi^2)$ circuit depth in general [23]. In many instances, deeper quantum circuits from the MPD algorithm lead to significantly higher fidelity approximations of the target state. The numerical results support this finding, demonstrating that the MPD+TNO scheme can significantly improve performance. This enables an improvement in the fidelity of the target state without increasing the circuit

depth, albeit at the expense of an additional classical cost in optimising the tensor network representation of the quantum circuit.

4. The MPD+TNO scheme suffers from trainability problems as the number of qubits increases. The optimisation is generally efficient when the initialised parameters overlap significantly with the global minimum. However, the algorithm's overall performance degrades as the dimension of the parameter landscape increases, corresponding to deeper circuits with a greater number of qubits. Alternative approaches to gradient descent and optimisation strategy could potentially be considered to improve trainability for deeper circuits. For instance, it was shown by Rudolph et al. [26] that iteratively optimising layer-by-layer (rather than all the layers at once) can improve training efficiency in some instances. This alternative approach comes with a greater (quadratic) dependence of complexity on the number of iterations scaling as $\mathcal{O}(nT^2L\chi_{\max}^3)$ [26]. Alternatively, training efficiency could be significantly reduced by truncating $\chi_{\max}$ to a reasonable target level, especially when n is large.

5. MPS Preparation of Images

5.1 The Discrete Wavelet Transform of Images

The Discrete Wavelet Transform (DWT) decomposes a signal or image into a set of wavelet coefficients. In our approach, the two-dimensional DWT is employed to compress an image $I(x, y)$ by transforming it into its frequency components, thereby separating the image into components that capture both coarse and fine details. The 2D DWT is implemented via separable filtering where one-dimensional low-pass and high-pass filters are applied along the rows and columns of the image. This process results in four subband images: one approximation (low-frequency) component and three detail (high-frequency) components, typically denoted as Approximation (LL), Horizontal Detail (LH), Vertical Detail (HL), and Diagonal Detail (HH). After applying the low-pass and high-pass filters along each dimension, the output is downsampled by a factor of 2 in each dimension such that each decomposition level corresponds to a subband of $\frac{P}{2} \times \frac{Q}{2}$ coefficients. For further details on the implementation of the DWT, see [81].

We utilise the PyWavelet 2D multilevel decomposition function `wavedec2` to compute the 2D DWT. For an image $I : P \times Q$, the `wavedec2` returns a list $[c_{LL}, (c_{LH}, c_{HL}, c_{HH})]$ where c_{LL} is an array of shape $(\frac{P}{2}, \frac{Q}{2})$ containing the coefficients of the LL subband (and similarly for the other coefficients). The total number of coefficients is $P \cdot Q$, and the time complexity of this process is $\mathcal{O}(PQ)$. This is an expensive pre-processing cost for extremely large images. However, as we will see, this pre-processing cost enables us to shift resources from quantum to classical computation, which is highly advantageous if the quantum component of the computation must be repeated many times (e.g. for a polynomial number of samples) and while quantum processors remain in the NISQ phase of development.

For quantum state preparation, we flatten the 2D DWT representation into a one-dimensional array of $P \cdot Q$ coefficients, where the first $\frac{PQ}{4}$ elements correspond to the flattened LL subband, the next $\frac{PQ}{4}$ elements correspond to the flattened LH subband, and so on.

5.2 MPS-Based Quantum Image Encoding

This subsection details how MPS-based encoding frameworks can be utilised to approximately encode the 2D DWT representation of images into the amplitudes of a quantum state.

I. Wavelet Transform for Image Compression

The target image $I = I(x, y)$ of size $P \times Q$ is compressed via the 2D DWT, decomposing it into the LL, LH, HL, and HH frequency subbands. This representation is flattened into a vector $v \in \mathbb{R}^N$ where $N = PQ$. If v is not a power of 2, it can be padded with zeros. v serves as the $n = \log N$-qubit target state that we aim to prepare.

II. Construction of the Target MPS

The MPS of a fixed bond dimension $\chi_{\max}$ can be constructed from this state vector using Algorithm 1. The size of $\chi_{\max}$ determines the degree of additional MPS compression. The wavelet coefficients are highly localised and concentrated in the low-frequency components (LL subbands), while high-frequency components (LH, HL, HH) are sparse. Hence, capturing the LL subcomponent of the target state will enable the core image details to be approximately encoded, while capturing higher-frequency components will be essential to capturing finer image structures.

A low bond dimension MPS is more likely to capture the LL correlations while discarding the high-frequency ones, while a large bond dimension MPS is required to capture the shorter-range intricate image details. This explains why an MPS with a truncated bond dimension may still well-approximate the flattened DWT representation of the image.

III. MPS Encoding Results and Validation

We employ the MPD and MPD+TNO algorithms to assess their efficacy in encoding the 2D DWT representation of image data. We use a sample image from sample 128×128 image from ChestMNIST sourced from the MedMNIST database for machine learning [82, 83]. We begin by computing the flattened DWT coefficients of the image before constructing its (exact) MPS representation. For this study, we aim to encode the exact MPS such that the fidelity of the encoded image is directly dependent on the capabilities of the MPD algorithm for a specified number of circuit layers L.

First, the target image was encoded using the MPD scheme on $n = 14$ qubits. This was highly efficient to compute for $L = 100$ to $L = 600$ layers. By computing the output of the MPD algorithm as a Qiskit circuit, we can find the exact circuit depth by decomposing this circuit into a circuit consisting of exclusively single-qubit rotations and CNOT gates. In the case of $L = 300$ layers, we achieve a fidelity of 99.61% with a circuit depth of 3665 gates. The MPD algorithm expectedly exhibits marginally decreasing increases in fidelity with increasing layers, with a maximum fidelity of 99.84% achieved with $L = 600$ layers.

Next, the MPD+TNO algorithm was employed for up to $L = 100$ variational circuit layers and trained for up to 400 iterations. The MPD+TNO scheme was utilised to achieve shallower-depth image encodings. For example, using the MPD+TNO scheme with $L = 20$ layers achieved

a fidelity of 98.62% after 600 iterations with a circuit depth of only 305 gates. The achieved fidelity was highly comparable with the non-optimised MPD algorithm for $L = 100$ layers and a circuit depth of 1265 gates. Similarly, the MPD+TNO with $L = 100$ layers achieved a fidelity of 99.54%, comparable to the MPD algorithm's performance with $L = 300$ layers. The addition of optimisation ultimately leads to shallower circuit depths at the expense of increased computational costs associated with the optimisation process. All layer-by-layer results are provided in Figure 9.
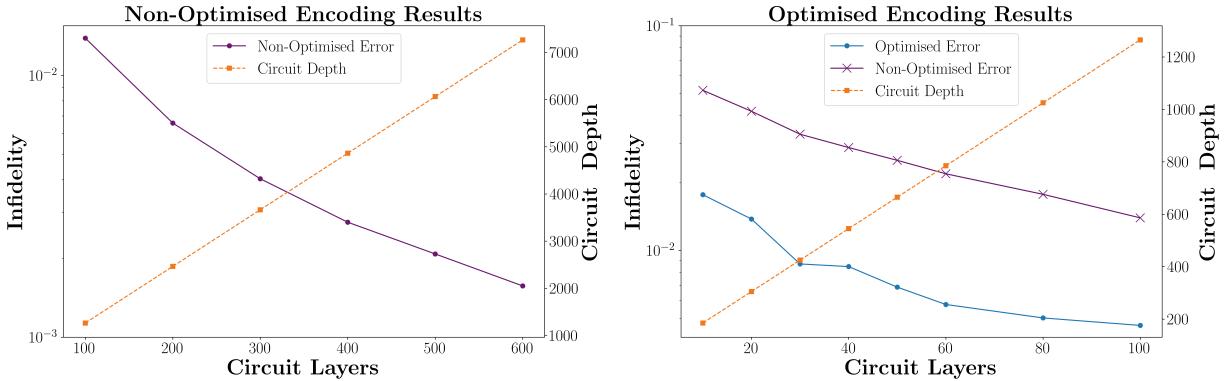

Figure 9: Image encoding results using the (non-optimised) MPD and (optimised) MPD+TNO algorithms for various circuit layers. The utilisation of each circuit layer is far superior when TNO is used.

To validate the success of this protocol, the encoded vector computed by the MPD algorithm with $L = 300$ layers is compared to the flattened target wavelet vector in Figure 10, which is zoomed in to show the first 4096 elements (corresponding to the LL subband). This helps to provide a visual intuition behind the sparsity in the DWT representation and the concentration of amplitudes in the low-frequency domain of the encoded image. We can reshape the encoded state into its original 2D DWT structure and apply the inverse DWT to recover the encoded image to visualise the encoding quality, which is provided in Figure 11.
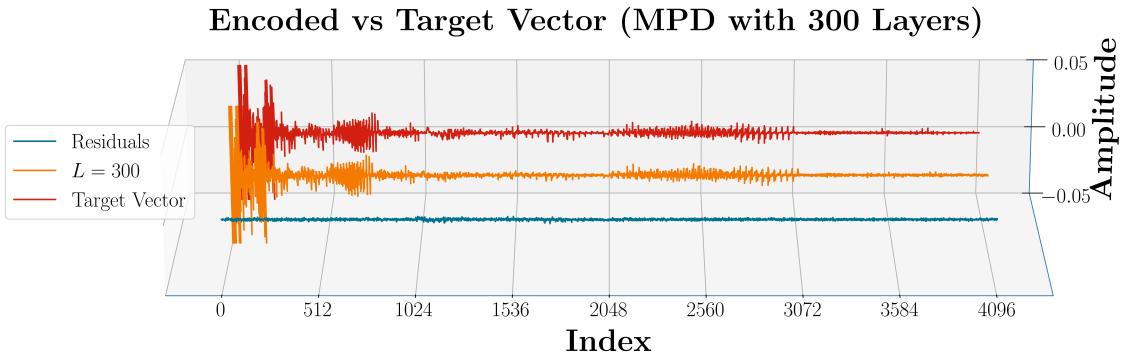

Figure 10: Visualising the first 4096 elements of the (normalise) encoded and target states, representing the LL-subband of the flattened discrete wavelet transform of a 128×128 ChestMNIST image. The encoded state vector corresponds to the output of the MPD algorithm with $L = 300$ layers, comparable to 100 layers of MPD+TNO. The residuals are computed as the difference between the encoded and target states.
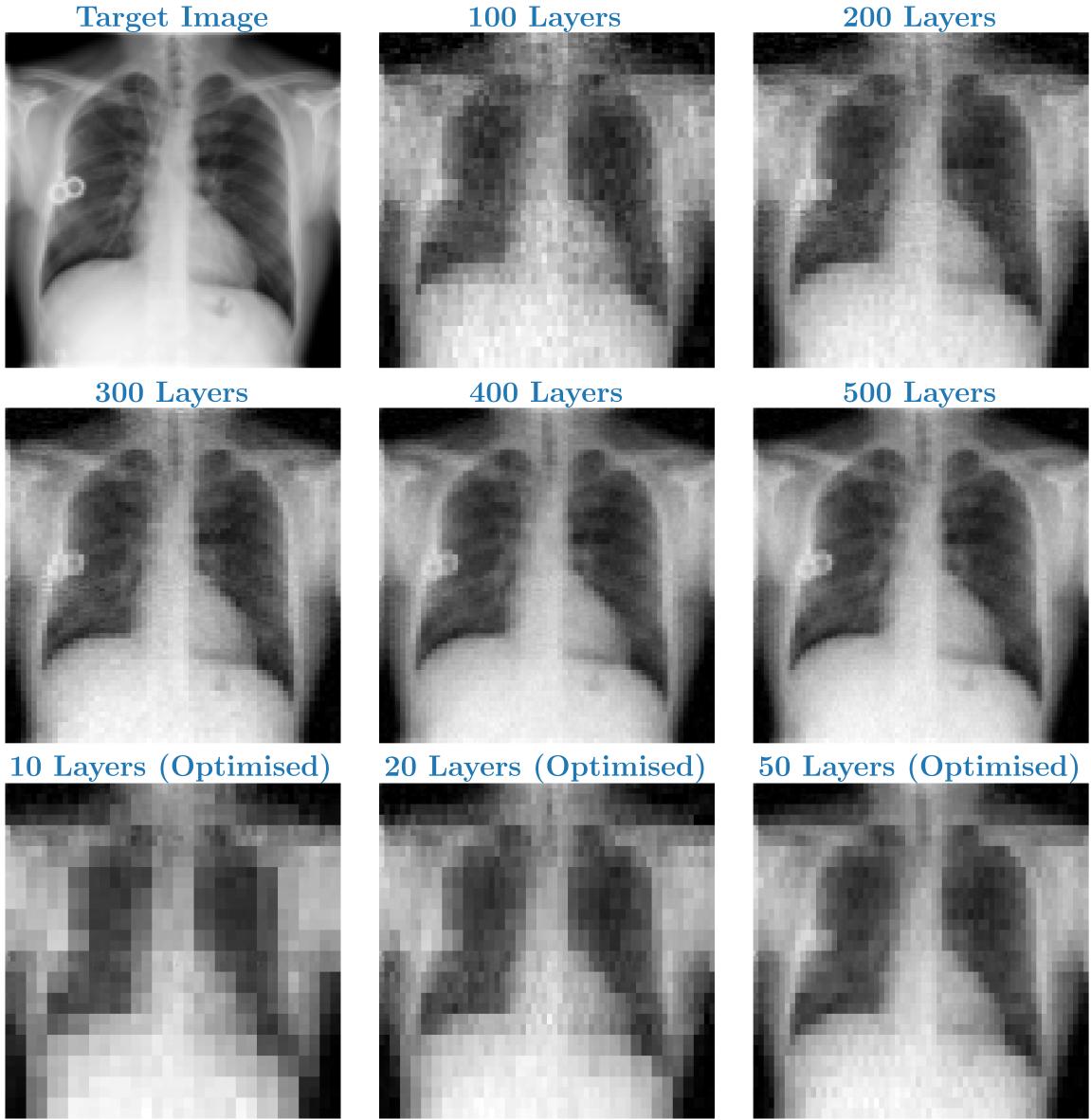
Figure 11: The target image (a 128×128 ChestMNIST image) alongside the reconstructed image encoded by the MPD and MPD+TNO algorithms for various circuit layers. All MPD+TNO (Optimise) results were computed using 400 iterations.

IV. Analysis and Extensions

This section introduced a novel approach to preparing MPS representative of the DWT representation of image data. This is just one example of a practically feasible approach to MPS-based image encoding, and there are likely many alternative approaches that one might consider. We chose the DWT approach based on its superior performance compared to (i) computing the MPS directly from black-and-white pixel data, and (ii) encoding the Discrete Fourier Transform (DFT) representation of image data during initial testing. We propose that the viability of the DWT-based approach is likely due to the concentration of core image data in the LL subband, which can be well-encapsulated by an MPS with a truncated bond dimension.

A core limitation of this approach is that the computation of the DWT uses all pixel values (i.e., it scales as $\mathcal{O}(PQ)$ for an image of dimensions $P \times Q$) and therefore accumulates a significant pre-processing cost for very large images. In addition, the MPS is constructed with

a top-down approach (i.e., the full MPS is computed before it is truncated), which has a $\mathcal{O}(2^n)$ cost. Hence, this approach may benefit from more approximate methods of constructing the target state. However, the key advantage is that this pre-processing cost enables the image data to be approximately encoded in much shallower depth circuits. This is advantageous since the (classical) pre-processing computations must only be computed once. In contrast, the quantum state preparation circuit must be run every time the data is processed or sampled (assuming no QRAM).

While the circuit depth scales as $\mathcal{O}(nL)$, further testing is required to explore how the fidelity of the prepared image can be sustained as the number of qubits increases. An initial test showed that a (different) 224×224 ChestMNIST image achieved a 99.14% fidelity encoding with 800 layers on $n = 16$ qubits, suggesting that the number of layers required for a high-fidelity encoding increases with the number of qubits. However, the exact circuit depth required to match the encoding performance of the above $n = 14$ example is unclear. While this framework is well-suited to NISQ devices and can be readily implemented on existing quantum devices, it also necessitates that the subsequent quantum algorithm can process image data in its DWT form unless a quantum equivalent of the inverse DWT can be applied. As such, this approach could be very useful in testing toy models of quantum algorithms, while more sophisticated methods of approximate image encoding are likely required to pursue meaningful quantum advantage.

6. Discussion and Conclusion

While the 2^n-dimensional Hilbert space represents an immense resource for computation, it imposes a dimensionality burden in the context of quantum state preparation. In turn, the absence of any general and efficient algorithm for arbitrary state preparation represents a severe bottleneck for quantum machine learning, linear algebra, and many other significant algorithms. Resultingly, we confined our scope to the design of general algorithms for the state preparation of structured states. Motivated by the avoidance of resource-intensive circuits for quantum arithmetic, lack of ancillary qubits, and in-built dimensionality reduction, we adopted an MPS-based framework for the approximate state preparation of structured data centred around the preparation of discretised functions and images. We utilised the MPD and MPD+TNO algorithms as the focal points of this investigation.

We ultimately illustrated the viability of these algorithms for the efficient state preparation of a broad class of discretised functions, including Gaussian, trigonometric, and smooth piecewise polynomial functions. We could also encapsulate irregular behaviour, such as simple discontinuities in piecewise polynomials and steep gradients exemplified by the root and logarithmic functions, with high accuracy. This research extended previous work by Holmes & Matsuura [75] into the MPS-based preparation of smooth, differentiable functions while incorporating TNO to achieve significantly improved performance over the MPD algorithm alone. The inclusion of TNO enables high-fidelity encodings with far shallower circuits. However, this reduction in quantum resources comes at the expense of an increase in classical overhead through the optimisation process.

Many of the prepared simple functions have immediate applications. For example, we can

exactly prepare the 'hockey-stick' function $f(x) = \max 0, x - K$ for some constant K, which is used as the boundary condition when solving the Black-Scholes PDE [84]. Since this state can be prepared exactly with a single layer of $U(4)$ gates, this is likely the most resource-efficient approach to preparing similarly straightforward functions. As the target state becomes more complex, much deeper circuits are required to achieve high-quality encodings. For linear algebra problems, where very high accuracy is essential, we identify that the MPD and MPD+TNO approaches can encode up to low-degree piecewise polynomials with approximately 10 intervals with fidelity exceeding 99.99% using shallow, linear-depth circuits. More complex structures can be approximately encoded if a more significant error in the encoded state is permissible.

The second core investigation of this article extended the application of the MPD and MPD+TNO algorithms to the amplitude encoding of image data. We introduced a novel approach based on encoding the approximate MPS representation of the discrete wavelet transform of image data. This novel proposal demonstrated significant promise, underlined by ¿99.5% fidelity encodings with 300 layers of the MPD algorithm or 100 optimised layers using MPD+TNO on a 14-qubit black-and-white 128×128 ChestMNIST image. By adopting TNO, we achieved extremely shallow-depth approximations of the image with fidelities of 98.2% and 98.6% with just 10 and 20 circuit layers, respectively. This protocol enables the MPS to capture the approximation component of the DWT representation with shallow-depth circuits, and increasingly finer details of the image can be captured by increasing the number of circuit layers. While the DWT protocol exhibited empirical success, future research may consider alternatives to the DWT representation of images. In any case, this approach can be readily implemented on NISQ devices due to the shallowness of the computed quantum circuits, the locality of quantum gates, and the lack of ancillary qubits.

While the MPS-based algorithms explored in this article are clear candidates for NISQ-era state preparation, pursuing meaningful quantum advantage may necessitate preparing more complex and precise representations of functions, images, or other (more complex) structured data. As the target state becomes increasingly more complex and less structured, the entanglement entropy and the required bond dimension necessary to represent the state accurately must increase. Correspondingly, the performance of the MPD and MPD+TNO algorithms will worsen, and more sophisticated state preparation algorithms will be needed. Another promising non-arithmetic, non-variational state preparation algorithm, which was not discussed thoroughly in this article, is the Quantum Eigenvalue Transform (QET) [19, 74, 85]. QET-based methods produce $\mathcal{O}(nd)$-depth circuits that prepare polynomials of degree d with $\mathcal{O}(n)$ ancillary qubits. A primary advantage of QET preparation rather than the MPS approach is better control over the error of the produced state [74]. However, the $\mathcal{O}(nd)$-depth circuits for the QET method hide substantial constant factors and require non-local gate operations, making this approach better suited to early fault-tolerant quantum computing.

Interestingly, circuits preparing MPS can be incorporated into QET-based preparation frameworks. Specifically, the (QET-based) non-linear transformation of amplitudes methodology introduced in [85] enables polynomial transformations to be applied to the amplitudes of any state that can be prepared efficiently (see Lemma 9 in [85]). This was the foundation of the state preparation algorithm introduced in [74], which introduced a clever discrete Hadamard-Walsh transform approach to preparing the linear function $f(x) = x$, before polyno-

mial transformations are applied to this state. We have shown, however, that the linear function state representing $f(x) = x$ can be prepared *exactly* with a single layer of $U(4)$ unitaries using the MPD algorithm. While both encoding strategies employ linear-depth circuits, encoding the linear function via the MPD algorithm uses only local operations, which may make circuits easier to implement. This is an extension that could be considered by future research. Additionally, future research may consider more sophisticated image encoding techniques compatible with the QET format.

In conclusion, the MPD and MPD+TNO algorithms are exceptionally well-suited to near-term NISQ devices with applications ranging from the state preparation of boundary conditions for PDEs, probability distributions for quantum Monte Carlo, and images for quantum image processing and classification. Alongside the extension to QET-based state preparation, further exploration into the TNO methodology, such as alternative methods of gradient descent and local minima-avoiding numerical strategies, is a key area of future research. Notably, these state preparation techniques can be entirely efficiently simulated using classical devices, meaning that quantum advantage cannot be derived directly from assumptions of state preparation. The pressing question remains: *for what applications does this permit a quantum advantage?* In any case, the explored algorithms are highly versatile and well-suited to the current stage of quantum development and can be readily implemented on existing quantum devices for further experimentation and testing.

0.1 Acknowledgements

Bibliography

[1] David Peral-García, Juan Cruz-Benito, and Francisco José García-Peñalvo. Systematic literature review: Quantum machine learning and its applications. *Computer Science Review*, 51:100619, 2024.

[2] M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, September 2022.

[3] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[4] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017.

[5] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, 118(35):e2026805118, 2021.

[6] Yulong Dong, Lin Lin, and Yu Tong. Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *PRX Quantum*, 3:040305, Oct 2022.

[7] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, Jan 2017.

[8] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.

[9] Mario Motta and Julia E. Rice. Emerging quantum computing algorithms for quantum chemistry. *WIREs Computational Molecular Science*, 12(3):e1580, 2022.

[10] H. Shang, Y. Fan, L. Shen, et al. Towards practical and massively parallel quantum computing emulation for quantum chemistry. *npj Quantum Information*, 9:33, 2023.

[11] S. Woerner and D. J. Egger. Quantum risk analysis. *npj Quantum Information*, 5:15, 2019.

[12] Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, September 2015.

[13] Almudena Carrera Vazquez and Stefan Woerner. Efficient state preparation for quantum amplitude estimation. *Phys. Rev. Appl.*, 15:034027, Mar 2021.

[14] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. Transformation of quantum states using uniformly controlled rotations. *Quantum Info. Comput.*, 5(6):467–473, September 2005.

[15] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. Quantum circuits for general multiqubit gates. *Phys. Rev. Lett.*, 93:130502, Sep 2004.

[16] Y.G. Chen and J.B. Wang. Qcompiler: Quantum compilation with the csd method. *Computer Physics Communications*, 184(3):853–865, 2013.

[17] T. Loke and J.B. Wang. Optqc v1.3: An (updated) optimized parallel quantum compiler. *Computer Physics Communications*, 207:531–532, 2016.

[18] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions, 2002.

[19] Sam McArdle, András Gilyén, and Mario Berta. Quantum state preparation without coherent arithmetic, 2022.

[20] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and M. Cerezo. A review of barren plateaus in variational quantum computing, 2024.

[21] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12:1791, March 2021.

[22] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[23] C. Schön, E. Solano, F. Verstraete, J. I. Cirac, and M. M. Wolf. Sequential generation of entangled multiqubit states. *Phys. Rev. Lett.*, 95:110503, Sep 2005.

[24] C. Schön, K. Hammerer, M. M. Wolf, J. I. Cirac, and E. Solano. Sequential generation of matrix-product states in cavity qed. *Phys. Rev. A*, 75:032311, Mar 2007.

[25] Shi-Ju Ran. Encoding of matrix product states into quantum circuits of one- and two-qubit gates. *Phys. Rev. A*, 101:032310, Mar 2020.

[26] Manuel S Rudolph, Jing Chen, Jacob Miller, Atithi Acharya, and Alejandro Perdomo-Ortiz. Decomposition of matrix product states into shallow quantum circuits. *Quantum Science and Technology*, 9(1):015012, Nov 2023.

[27] Daniel Malz, Georgios Styliaris, Zhi-Yuan Wei, and J. Ignacio Cirac. Preparation of matrix product states with log-depth quantum circuits. *Phys. Rev. Lett.*, 132:040404, Jan 2024.

[28] Mohsin Iqbal, David Muñoz Ramo, and Henrik Dreyer. Preentangling quantum algorithms – the density matrix renormalization group-assisted quantum canonical transformation, 2022.

[29] M. Ben-Dov, D. Shnaiderov, A. Makmal, et al. Approximate encoding of quantum states using shallow circuits. *npj Quantum Information*, 10:65, 2024.

[30] Kevin C. Smith, Abid Khan, Bryan K. Clark, S.M. Girvin, and Tzu-Chieh Wei. Constant-depth preparation of matrix product states with adaptive quantum circuits. *PRX Quantum*, 5:030344, Sep 2024.

[31] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.

[32] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Phys*, 10:631–633, 2014.

[33] Kaiwen Gui, Alexander M. Dalzell, Alessandro Achille, Martin Suchara, and Frederic T. Chong. Spacetime-efficient low-depth quantum state preparation with applications. *Quantum*, 8:1257, February 2024.

[34] Xiao-Ming Zhang, Man-Hong Yung, and Xiao Yuan. Low-depth quantum state preparation. *Phys. Rev. Res.*, 3:043200, Dec 2021.

[35] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum state preparation with optimal circuit depth: Implementations and applications. *Phys. Rev. Lett.*, 129:230504, Nov 2022.

[36] Israel F. Araujo, Daniel K. Park, Francesco Petruccione, and Adenilton J. da Silva. A divide-and-conquer algorithm for quantum state preparation. *Scientific Reports*, page 6329, March 2021.

[37] Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum circuits for isometries. *Phys. Rev. A*, 93:032318, Mar 2016.

[38] Debora Ramacciotti, Andreea I. Lefterovici, and Antonio F. Rotundo. Simple quantum algorithm to efficiently prepare sparse states. *Phys. Rev. A*, 110:032609, Sep 2024.

[39] Emanuel Malvetti, Raban Iten, and Roger Colbeck. Quantum circuits for sparse isometries. *Quantum*, 5:412, March 2021.

[40] Niels Gleinig and Torsten Hoefler. An efficient algorithm for sparse quantum state preparation. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 433–438, 2021.

[41] César Feniou, Olivier Adjoua, Baptiste Claudon, Julien Zylberman, Emmanuel Giner, and Jean-Philip Piquemal. Sparse quantum state preparation for strongly correlated systems. *The Journal of Physical Chemistry Letters*, 15(11):3197–3205, March 2024.

[42] Tiago M. L. de Veras, Leon D. da Silva, and Adenilton J. da Silva. Double sparse quantum state preparation. *Quantum Information Processing*, 21(6):204, June 2022.

[43] Christof Zalka. Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):313–322, 1998.

[44] Thomas Häner, Martin Roetteler, and Krysta M. Svore. Optimizing quantum circuits for arithmetic, 2018.

[45] Steven Herbert. No quantum speedup with grover-rudolph state preparation for quantum monte carlo integration. *Phys. Rev. E*, 103:063302, Jun 2021.

[46] Patrick Rebentrost, Brajesh Gupt, and Thomas R. Bromley. Quantum computational finance: Monte carlo pricing of financial derivatives. *Phys. Rev. A*, 98:022321, Aug 2018.

[47] Lov K. Grover. Synthesis of quantum superpositions by quantum computation. *Phys. Rev. Lett.*, 85:1334–1337, Aug 2000.

[48] Kosuke Mitarai, Masahiro Kitagawa, and Keisuke Fujii. Quantum analog-digital conversion. *Phys. Rev. A*, 99:012301, Jan 2019.

[49] Yuval R. Sanders, Guang Hao Low, Artur Scherer, and Dominic W. Berry. Black-box quantum state preparation without arithmetic. *Phys. Rev. Lett.*, 122:020502, Jan 2019.

[50] Johannes Bausch. Fast black-box quantum state preparation. *Quantum*, 6:773, August 2022.

[51] Lorenzo Laneve. Robust black-box quantum-state preparation via quantum signal processing, 2023.

[52] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, and Jarrod R. McClean. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, June 2022.

[53] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1), November 2019.

[54] Viacheslav V. Kuzmin and Pietro Silvi. Variational quantum state preparation via quantum data buses. *Quantum*, 4:290, July 2020.

[55] Bryan T. Gard, Liang Zhu, George S. Barron, Nicholas J. Mayhall, Sophia E. Economou, and Edwin Barnes. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Information*, 6(1):10, 2020.

[56] Nikita Astrakhantsev, Guglielmo Mazzola, Ivano Tavernelli, and Giuseppe Carleo. Phenomenological theory of variational quantum ground-state preparation. *Phys. Rev. Res.*, 5:033225, Sep 2023.

[57] Youle Wang, Guangxi Li, and Xin Wang. Variational quantum gibbs state preparation with a truncated taylor series. *Phys. Rev. Appl.*, 16:054035, Nov 2021.

[58] Mirko Consiglio, Jacopo Settino, Andrea Giordano, Carlo Mastroianni, Francesco Plastina, Salvatore Lorenzo, Sabrina Maniscalco, John Goold, and Tony J. G. Apollaro. Variational gibbs state preparation on noisy intermediate-scale quantum devices. *Phys. Rev. A*, 110:012445, Jul 2024.

[59] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.*, 94:015004, Feb 2022.

[60] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, August 2021.

[61] E. R. Anschuetz and B. T. Kiani. Quantum variational algorithms are swamped with traps. *Nature Communications*, 13:7760, 2022.

[62] M. Cerezo, Martin Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana, Manuel S. Rudolph, Pablo Bermejo, Aroosa Ijaz, Supanut Thanasilp, Eric R. Anschuetz, and Zoë Holmes. Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing, 2024.

[63] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[64] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. This paper introduces the matrix approximation theorem, now known as the Eckart-Young theorem.

[65] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

[66] Prithvi Gundlapalli and Junyi Lee. Deterministic and entanglement-efficient preparation of amplitude-encoded quantum registers. *Phys. Rev. Appl.*, 18:024013, Aug 2022.

[67] Vivek V. Shende, Igor L. Markov, and Stephen S. Bullock. Minimal universal two-qubit controlled-not-based circuits. *Phys. Rev. A*, 69:062321, Jun 2004.

[68] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214, 2019.

[69] Johnnie Gray and Garnet Kin-Lic Chan. Hyperoptimized approximate contraction of tensor networks with arbitrary geometry. *Phys. Rev. X*, 14:011009, Jan 2024.

[70] Ar Melnikov, Alena Termanova, S Dolgov, Florian Neukart, and M Perelshtein. Quantum state preparation using tensor networks. *Quantum Science and Technology*, 8(3):035027, 06 2023.

[71] Ivan Oseledets and Eugene Tyrtyshnikov. Tt-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

[72] Anna M. Krol, Aritra Sarkar, Imran Ashraf, Zaid Al-Ars, and Koen Bertels. Efficient decomposition of unitary matrices in quantum circuit compilers. *Applied Sciences*, 12(2), 2022.

[73] Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Phys. Rev. A*, 69:032315, Mar 2004.

[74] Javier Gonzalez-Conde, Thomas W. Watts, Pablo Rodriguez-Grasa, and Mikel Sanz. Efficient quantum amplitude encoding of polynomial functions. *Quantum*, 8:1297, March 2024.

[75] Adam Holmes and A. Y. Matsuura. Efficient Quantum Circuits for Accurate State Preparation of Smooth, Differentiable Functions . In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 169–179, Los Alamitos, CA, USA, October 2020. IEEE Computer Society.

[76] Juan José García-Ripoll. Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations. *Quantum*, 5:431, April 2021.

[77] Jason Iaconis, Sonika Johri, and Elton Yechao Zhu. Quantum state preparation of normal distributions using matrix product states. *npj Quantum Information*, 10(1):15, 2024.

[78] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Info. Comput.*, 7(5):401–430, July 2007.

[79] I.V. Oseledets. Constructive representation of functions in low-rank tensor formats. *Constructive Approximation*, 37(1):1–18, 2013.

[80] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(110):3371–3408, 2010.

[81] Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, Inc., USA, 3rd edition, 2008.

[82] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. *ChestX-ray: Hospital-Scale Chest X-ray Database and Benchmarks on Weakly Supervised Classification and Localization of Common Thorax Diseases*, pages 369–392. Springer International Publishing, Cham, 2019.

[83] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10:41, 01 2023.

[84] Yusen Wu, Jingbo B. Wang, and Yuying Li. Quantum computing for option portfolio analysis. *arXiv:2406.00486 (2024)*.

[85] Arthur G. Rattew and Patrick Rebentrost. Non-linear transformations of quantum amplitudes: Exponential improvement, generalization, and applications. *arXiv:2309.09839 (2023)*.