

ACCEPTED MANUSCRIPT • OPEN ACCESS

Quantum-Enhanced Inference of Conditional Future Probabilities with Reduced Memory Cost

To cite this article before publication: Jianjun Chen *et al* 2026 *Quantum Sci. Technol.* in press <https://doi.org/10.1088/2058-9565/aeae6d>

Manuscript version: Accepted Manuscript

Accepted Manuscript is “the version of the article accepted for publication including all changes made as a result of the peer review process, and which may also include the addition to the article by IOP Publishing of a header, an article ID, a cover sheet and/or an ‘Accepted Manuscript’ watermark, but excluding any other editing, typesetting or other changes made by IOP Publishing and/or its licensors”

This Accepted Manuscript is © 2026 The Author(s). Published by IOP Publishing Ltd.



As the Version of Record of this article is going to be / has been published on a gold open access basis under a CC BY 4.0 licence, this Accepted Manuscript is available for reuse under a CC BY 4.0 licence immediately.

Everyone is permitted to use all or part of the original content in this article, provided that they adhere to all the terms of the licence <https://creativecommons.org/licenses/by/4.0>

Although reasonable endeavours have been taken to obtain all necessary permissions from third parties to include their copyrighted content within this article, their full citation and copyright line may not be present in this Accepted Manuscript version. Before using any content from this article, please refer to the Version of Record on IOPscience once published for full citation and copyright details, as permissions may be required. All third party content is fully copyright protected and is not published on a gold open access basis under a CC BY licence, unless that is specifically stated in the figure caption in the Version of Record.

View the [article online](#) for updates and enhancements.

Quantum-Enhanced Inference of Conditional Future Probabilities with Reduced Memory Cost

Jianjun Chen,^{1,2,*} Akshat Gupta,^{3,4,†} Chengran Yang,^{1,2,‡} Ariel Neufeld,⁵ Jayne Thompson,^{6,2,§} Thomas J. Elliott,^{3,7,4} and Mile Gu^{1,2,¶}

¹Nanyang Quantum Hub, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371

²Centre for Quantum Technologies, Nanyang Technological University, Singapore 639798

³Department of Physics & Astronomy, University of Manchester, Manchester M13 9PL, United Kingdom

⁴Centre for Quantum Science and Engineering, University of Manchester, Manchester M13 9PL, United Kingdom

⁵Division of Mathematical Sciences, School of Physical and

Mathematical Sciences, Nanyang Technological University, Singapore

⁶College of Computing and Data Science, Nanyang Technological University, Singapore 639798

⁷Department of Mathematics, University of Manchester, Manchester M13 9PL, United Kingdom

(Dated: August 19, 2026)

Many potential future events have devastating consequences, from the onset of earthquakes to financial black swan events. Accurate predictions of their probability of occurrence are vital. Yet as the stochastic systems underlying such events become increasingly complex, these probabilities can depend on ever-growing amounts of past data, which require ever more memory to track. The accuracy of our estimates is ultimately limited by both computational and memory constraints; the former to reduce sampling error, the latter to avoid sampling bias. Traditional quantum algorithms – such as amplitude estimation – address only the former, and become untenable especially in near-future quantum devices where quantum memory is a premium. Here, we offer quantum-algorithmic means to reduce systematic bias while maintaining a quadratic speed-up. Our results thus advance the timeline for the practical deployment of quantum technologies, and illustrate their additional benefits when simulating processes of growing complexity.

Given a past sequence of seismic observations, is the probability of a future earthquake 1 or 1.1 in a million? Accurately estimating the conditional probability p of certain future events in stochastic systems can be of vital importance. Conventional means to estimate such p through sampling possible futures are ultimately constrained by the central limit theorem. To produce estimates of p with additive error ϵ , one must draw $\mathcal{O}(1/\epsilon^2)$ independent samples. Here, quantum amplitude estimation (QAE) promises a quadratic reprieve, reducing that cost to $\mathcal{O}(1/\epsilon)$ [1]. This prospect has fuelled a surge of interest in quantum-enhanced risk analysis and provides a key use-case study often cited in the emerging field of quantum finance [2, 3].

Yet, realistic deployment of QAE faces a major hurdle: state preparation. QAE methods to estimate such outcome probabilities rely critically on a quantum circuit $\mathcal{A}$ that maps $|0\rangle$ to its quantum sample $|\phi_Z\rangle = \sum_z \sqrt{p_z} |z\rangle$. Consider a process that emits binary symbols x_t governed by random variable X_t at discrete times. A joint distribution over L time-steps already has 2^L possible outcomes, and thus loading their amplitudes by brute force is generally intractable [4, 5].

A more feasible approach is to first identify a classical model of the process, then translate this into a quantum circuit gate-by-gate and qubit-by-qubit. However, qubits are much more expensive to control than bits. Meanwhile, classical models of complex processes can be extremely memory intensive – to the point that they motivated a plethora of classical memory reduction tech-

niques [6, 7]. Indeed, there are fundamental theoretical limits on how many bits we must track about a process to faithfully infer its future [8, 9]. Drop below this, and prediction bias is unavoidable. With quantum memory being much less readily available, one may anticipate that any reduction in error from QAE will be outweighed by such bias, presenting a major hurdle for practical quantum advantage.

Here, we ask: *Could we simultaneously obtain memory and sampling advantages?* We answer this in the affirmative by leveraging recent advances in quantum-enhanced dimensional reduction [10]. We illustrate a means for inferring future likelihood that exhibits both advantages: (1) Under memory constraints, they can infer future likelihoods of complex stochastic processes with significantly reduced bias, (2) they inherit the quadratic computational advantage of Quantum-enhanced Monte Carlo methods. Taken together, these advances help accelerate the timeline in which quantum computers can deliver a practical advantage.

FRAMEWORK

Problem Statement. Consider a dynamical system that emits observable outputs $x_t \in \mathcal{X}$ at each discrete time step t . Each x_t is governed by a corresponding random variable X_t , such that the system's observable statistics are captured by the bi-infinite sequence $\vec{X} = \dots X_{-2}X_{-1}X_0X_1X_2\dots$. We focus on stationary

stochastic processes, in which the joint distribution of any consecutive L outputs $x_{0:L} := x_0 x_1 \dots x_{L-1}$ is invariant under time translations. This stationarity allows us to set $t = 0$ as the present without loss of generality. Each realisation of the process has a specific past $\overleftarrow{x} = x_{-\infty:0}$ and a corresponding conditional future $\overrightarrow{x} = x_{0:\infty}$, governed by the distribution $P(\overrightarrow{X} | \overleftarrow{x})$.

Our objective is to estimate the likelihood of a target future event up to some time horizon L , formalised as the subset of futures that satisfy some binary property f . Mathematically, we define f as a binary function such that $x_{0:L}$ satisfies the property if and only if $f(x_{0:L}) = 1$. Our goal is then to estimate the conditional probability p that a sequence $x_{0:L}$ distributed according to $P(X_{0:L} | \overleftarrow{x})$ satisfies f , that is, to compute $p := P[f(x_{0:L}) = 1 | \overleftarrow{x}] := P_f | \overleftarrow{x}$.

Classical Baseline. In classical settings, a standard approach to estimating the probability of a target event is statistical sampling using a predictive model $\mathcal{M}$. Formally, such a predictive model operates by encoding each past $\overleftarrow{x}$ into a memory state $s_i \in \mathcal{S}$. At each time step, the dynamics of the model are then specified by transition probabilities $T_{ij}^x = P(s_j, x | s_i)$, describing the probability that it emits x and transitions to state s_j when in state s_i . A model $\mathcal{M}$ is then said to faithfully represent a particular process P if iterating this procedure for L time steps results in a sequence drawn from the conditional distribution $P(X_{0:L} | \overleftarrow{x})$. Such models can be distilled from recurrent neural networks and transformer models, or alternatively constructed analytically based on an underlying understanding of the process we wish to sample.

Here, we refer to the process of generating one sample from $P(X_{0:L} | \overleftarrow{x})$ as one simulation. By running this simulation N times, the proportion of samples that satisfies f produces an estimate $\hat{p}$ of p . The performance of any such estimation algorithm can then be benchmarked by the root mean squared error (RMSE):

$$\epsilon = \sqrt{\mathbb{E}[(p - \hat{p})^2]}, \quad (1)$$

where the expectation is taken over the distribution of $\hat{p}$.

There are then two main resource concerns. The first is the number of times, N , that we need to simulate the process. This scales as $1/\epsilon^2$ to reach a designated error threshold ϵ [11–13]. The second is the number of distinct memory states $D = |\mathcal{S}|$ tracked by the model, which can grow rapidly in modern sequence-modelling settings.

Quantum Amplitude Estimation (QAE) offers a quadratic reduction in the number of simulations needed. It assumes that we begin with $\mathcal{A}$, a state-preparation circuit that generates a quantum sample of $X_{0:L}$, i.e., $\sum_{x_{0:L}} \sqrt{P(x_{0:L} | \overleftarrow{x})} |x_{0:L}\rangle$. We can then define the Grover operator

$$\mathbf{Q} = -\mathcal{A}S_0\mathcal{A}^\dagger S_\chi, \quad (2)$$

where, $S_0 = \mathbb{I} - 2|0\rangle\langle 0|$ is a reflection operator about the all-zero state, and S_χ is the oracle that marks the target subspace, i.e., $S_\chi = \mathbb{I} - 2\sum_{x_{0:L}: f(x_{0:L})=1} |x_{0:L}\rangle\langle x_{0:L}|$, where $f(x_{0:L}) = 1$ if $x_{0:L}$ satisfies the property f and 0 otherwise [14]. Standard QAE uses the Grover operator $\mathbf{Q}$ within a phase estimation subroutine, such that the number of times we need to execute $\mathcal{A}$ scales only as $1/\epsilon$ [1]. Subsequently, the limitations of present-day quantum technology motivated shallow circuit variants that avoid phase estimation via classical post-processing [15–18].

While QAE provides a quadratic reduction in sample complexity, it offers little relief regarding the cost of generating each sample (i.e., executing $\mathcal{A}$). Given that the number of outcomes of $X_{0:L}$ scales exponentially with L , direct synthesis of $\mathcal{A}$ generally requires an exponential number of quantum gates or ancilla qubits [4, 5]. A standard workaround is to leverage an existing classical model $\mathcal{M}$. In principle, if $\mathcal{M}$ can generate classical samples efficiently, then we can use it to construct an efficient $\mathcal{A}$ with similar gate complexity, provided we are willing to commit at least one qubit for each bit. In practice, this often remains impractical. Classical bits are cheap, but qubits are not. Tracking a dozen floating-point numbers to 32-bit precision is trivial on today’s computers, yet would quickly saturate the capabilities of a near-future quantum device.

Model Reduction and Distortion. Given the above constraints, model reduction is paramount. The simpler our model $\mathcal{M}$, the easier it will be to implement $\mathcal{A}$. This motivates the question: what is the minimal memory – in terms of memory dimension D (that is, how many distinct configurations $\mathcal{M}$ must hold) – required to sample $P(x_{0:L} | \overleftarrow{x})$ faithfully? By identifying such a representation, we can encode the past into D distinct states, $\mathcal{S} = \{s_k\}_{k=1}^D$, using only $O(\log_2 D)$ bits of memory.

This question is extensively studied in complexity science. Every stochastic process $\mathcal{P}$ possesses an intrinsic measure of complexity given by D_c , which lower-bounds the memory dimension of $\mathcal{M}$. Its logarithm, $d_c = \log_2(D_c)$, is known as the *topological state complexity* of $\mathcal{P}$ [9] (see Appendices A and B for details). The corresponding memory states $\{s_k\}$ are known as *causal states*, capturing the minimal belief states an agent must retain in memory about the past for optimal prediction of the future – and arising naturally within contemporary transformer and recurrent neural networks [19].

Yet, even when we can build such minimal models, their memory requirements may be immense or unbounded – especially when future behaviour is sensitive to continuous parameters or involves long-range temporal correlations [20, 21]. This necessitates approximate models, such as clustering and finite-precision encodings [6, 22–24], in which the memory dimension is constrained to some desired limit $\tilde{D} < D_c$. This memory

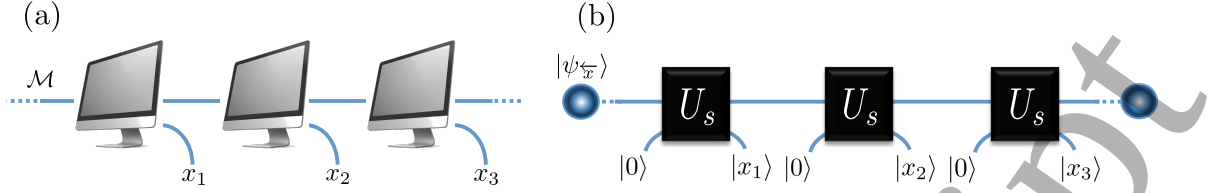


FIG. 1: **Classical vs. Quantum Models.** (a) A classical model of a stochastic process operates by tracking relevant past information in its memory states $\{s_j\}$, encoded within the memory system M . At each time step, a classical model in state s_j emits a symbol x and transitions to some state s_k with probability T_{jk}^x . (b) A quantum model consists of a quantum memory initialised in a state $|s_j\rangle$, based on the past observations $\bar{x}$. At each time step t , the unitary U_s entangles the memory register with a fresh ancillary system, imprinting the output $|x_t\rangle$ onto the ancilla and simultaneously updating the memory. The output $|x_t\rangle$ can be measured immediately to generate a sequence for the stochastic process defined by the unitary U_s or left coherent in the quantum circuit for further processing.

reduction results in distortion, such that our approximate model estimates $\hat{p}$ by sampling from some induced distribution $\tilde{P}(X_{0:L} | \bar{x})$ that approximates $\mathcal{P}$. The resulting errors in estimating p then comprise (1) random contributions ϵ_{rand} , which reflect errors due to a finite number of samples, and (2) systematic bias ϵ_b , arising from inherent distortions in the model. Crucially, the latter does not diminish with N , and as $N \rightarrow \infty$, our total error satisfies $\epsilon \rightarrow |\tilde{P}_{f|\bar{x}} - P_{f|\bar{x}}|$, which does not approach 0. We therefore have a fundamental trade-off between reducing memory cost and reducing predictive error.

APPROACH

Algorithm Outline. Our aim is to further reduce the resources required to synthesise a suitable state-preparation operator $\mathcal{A}$ below what is possible through conventional bit-to-qubit translation. Our approach begins with a target process $\mathcal{P}$ that has topological state complexity $\log_2 D_c$. A memory-minimal classical model $\mathcal{M}_c$ that generates predictions with zero bias then has D_c memory states. Meanwhile, enforcing a target memory constraint $\tilde{D} < D_c$ would naturally force the use of a biased model, and any $\mathcal{A}$ obtained by translating such a model would inherit similar bias. Our approach instead exploits the fact that quantum models with a fixed memory dimension $\tilde{D}$ are more expressive [10, 20], allowing us to design an efficient implementation of $\mathcal{A}$ whose bias is significantly lower than that of the baseline approach.

Let $\mathcal{M}_c$ have memory states $\{s_j\}$ and transition elements T_{jk}^x . We can then define an associated quantum model $\mathcal{M}_q$ that associates each s_j with a corresponding quantum memory state $|s_j\rangle$ [25]. Thus, if $\mathcal{M}_c$ encodes $\bar{x}$ in state s_j within memory M , its associated quantum model encodes $\bar{x}$ in $|s_j\rangle$ within a quantum memory M_q . At each time step t , the quantum model operates by introducing a fresh ancilla initialised in $|0\rangle$ and enacting a

unitary U_s satisfying

$$U_s |s_i\rangle |0\rangle = \sum_{j=0}^{|S|} \sum_{x \in \mathcal{X}} \sqrt{T_{i,j}^x} |s_j\rangle |x\rangle, \quad (3)$$

which couples its memory to the ancilla (see Figure 1). Here, there is no requirement that the set $\{|s_j\rangle\}$ be linearly independent [25]. As such, the associated memory dimension of $\mathcal{M}_q$, denoted D_q , can be strictly less than D_c (See Appendix C). When this occurs, it immediately enables a memory advantage in the zero-bias regime. More generally, when given a target memory $\tilde{D} < D_c$, we have a means to design quantum models whose conditional future statistics more closely approximate $\mathcal{P}$ (see Algorithm 4 in Appendix D 4).

We then leverage the resulting quantum model to construct a Grover operator. First, we need to define its components: the state-preparation operator $\mathcal{A}$ and the reflection operators $S_{\mathcal{X}}$ and S_0 . For a future sequence of length L , i.e. $x_{0:L} = x_0 x_1 \dots x_{L-1}$, the quantum circuit will consist of a memory register of size $n = \lceil \log_2 \tilde{D} \rceil$ qubits and L output registers, each of size $\lceil \log_2 |\mathcal{X}| \rceil$ qubits. This initial circuit can be represented as $|0\rangle_{\psi} |0\rangle_0 |0\rangle_1 \dots |0\rangle_{L-1}$, where $|0\rangle_{\psi}$ is the memory register and $|0\rangle_i$ is the output register to store the i -th index of the sequence $x_{0:L}$, each initialized in the computational zero state.

To construct the state-preparation operator, we first initialise the memory register in the appropriate $|s_j\rangle$ consistent with the observed $\bar{x}$. This can always be done via some unitary U_{ini} (see Appendix D 3). Subsequently, for each symbol of the sequence, the unitary U_s is applied to the memory register and a new output register initially in the zero state. This sequence of operations defines the state-preparation operator $\mathcal{A}$ as

$$\mathcal{A} = U_s^{(\psi, L-1)} \dots U_s^{(\psi, 1)} U_s^{(\psi, 0)} U_{\text{ini}}, \quad (4)$$

where $U_s^{(\psi, i)}$ denotes the application of U_s on the memory register and the i -th output register. Application of

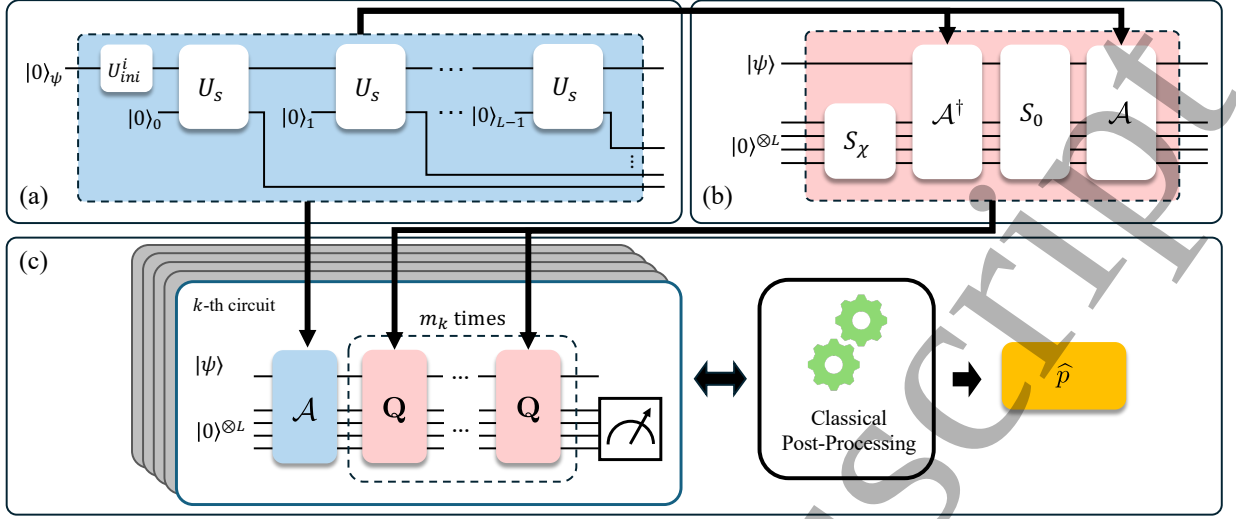


FIG. 2: **Algorithm Schematic.** Our method for memory-reduced conditional probability estimation tackles the data-loading problem through the use of a recurrent quantum circuit $\mathcal{A}$ (a). Through L repeated applications of U_s on a suitably initialised memory system and L ancilla qubits, the circuit is able to generate a quantum superposition of all possible futures up to a time horizon of L . Provided the memory system (top line) is properly accounted for, we can build a suitable Grover iterator $\mathbf{Q}$ (b). We can then construct a family of quantum circuits N_k , each involving m_k repeated applications of $\mathbf{Q}$. The resulting measurement statistics from these circuits then allow us to estimate $\hat{p}$ (c). For cases where f is an indicator function that flags target strings, S_χ can be constructed from a multi-controlled single-qubit gate. The reflection operator S_0 can also be constructed using a multi-controlled single-qubit gate.

$\mathcal{A}$ results in $\sum_{x_{0:L}} \sqrt{P(x_{0:L}|\tilde{x})} |x_{0:L}\rangle |s_{\tilde{x}'}\rangle$ where $\tilde{x}' = \tilde{x} x_{0:L}$. It thus does not strictly prepare the q-sample state, but rather the q-sample entangled with a memory state for generating samples beyond time $t = L$.

Algorithm 1 (MRQPE)

Memory-reduced Quantum Probability Estimation

Input:

- $\mathcal{M}_c = (\mathcal{S}, T_{ij}^x)$: Classical model of a target process $\mathcal{P}$.
- f : binary function that marks the target subspace.
- L : length of future sequence.
- $x_{-r:0}$: past sequence of length r .
- $\tilde{D}$: desired number of states.

Output: Probability estimate $\hat{p}$

- 1: $\{K^x\}, U_s \leftarrow \text{QUANTUMMODEL}(\{T^x\})$ (See Appendix D 2 Algorithm 2)
- 2: **if** $\tilde{D} < |\mathcal{S}|$ **then**
- 3: $\{K^x\}, U_s \leftarrow \text{TRUNCATE}(\{K^x\}, \tilde{D})$ (See Appendix D 4 Algorithm 4)
- 4: $U_{ini} \leftarrow \text{ENCODEPAST}(x_{-r:0}, \{K^x\})$ (See Appendix D 3 Algorithm 3)
- 5: Construct operator $\mathcal{A}$ from U_{ini} and U_s (Eq. (4))
- 6: Construct oracle S_χ to implement binary function f :

$$f(|x_0 x_1 \dots x_{L-1}\rangle) = \begin{cases} -|x_{0:L}\rangle & \text{if } f(x_{0:L}) = 1 \\ |x_{0:L}\rangle & \text{otherwise} \end{cases}$$
- 7: Construct Grover operator $\mathbf{Q} = -\mathcal{A}S_0\mathcal{A}^\dagger S_\chi$
- 8: Estimate $\hat{p} \leftarrow \text{QAE}(\mathbf{Q}, \mathcal{A})$
- 9: **return** $\hat{p}$

In Figure 2, we illustrate that provided this memory is properly accounted for, the principles of QAE still apply. Specifically, in constructing the Grover operator, the oracle gate S_χ is applied only to the output registers as per standard QAE. However, the rest of the components, i.e., the operator $\mathcal{A}$, its reversal $\mathcal{A}^\dagger$ and the reflection operator S_0 are applied to both memory and output registers.

An outline of the resulting end-to-end algorithm is documented in Algorithm 1. This algorithm takes as input (i) a classical model $\mathcal{M}_c$ of the process $\mathcal{P}$, (ii) the binary indicator function f , (iii) the number of future time-steps we wish to sample L , (iv) the past r outputs observed, and (v) the available memory dimension $\tilde{D}$. The algorithm then outputs an estimate of $P_{f|\tilde{x}}$ with both reduced variance and bias. Here, the subroutine **QuantumModel** begins by transforming a supplied $\mathcal{M}_c$ to a quantum model that replicates its statistical behaviour exactly. The **Truncate** subroutine then provides a means to reduce the memory of this model to dimension $\tilde{D}$. We can then connect these models with a variant of quantum amplitude estimation, denoted by QAE [1, 16–18], that can leverage such quantum models to estimate probabilities of future events up to some future time-horizon L (See Figure 2).

Complexity. As two-qubit operations are generally costly to implement, we focus on the number of CNOT gates required to realise the Grover operator. For a model of memory dimension D , each U_s acts on $\lceil \log_2 |D| \rceil$ mem-

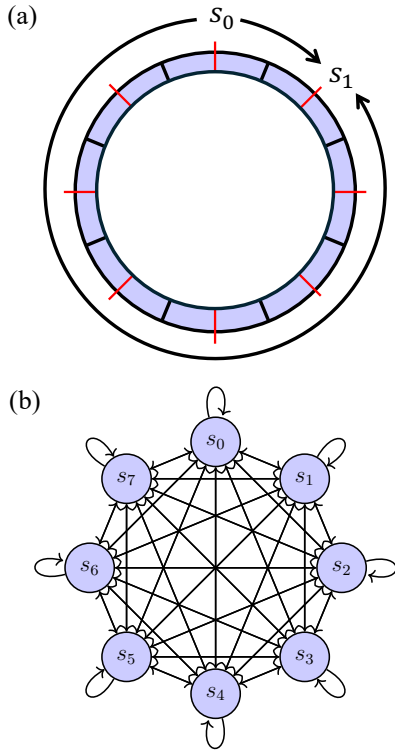


FIG. 3: **Cyclic random walk and its discretisation.**

(a) We can discretise a cyclic random walk by dividing the unit interval into equal segments, each represented by its midpoint $s_j = j/N$, $j = 0, 1, \dots, N-1$ (red markers). Between any two states, there exist two possible transition pathways — one clockwise and one anticlockwise. (b) Given a continuous position X_t , the discrete state of the system at time t is defined as the closest segment midpoint s_j , denoted $S_t = s(X_t)$. The transition probability from s_k to s_j is then $P(s_k|s_j) = P(X^{t+1} \in \mathcal{I}_k | X^t = x_j)$, where $\mathcal{I}_k = \{x : |x - \frac{k}{N}| < \frac{1}{2N}\}$ denotes the interval associated with the segment s_k . Here, we consider the 8-state discretisation as the target process (see Appendix G for more details).

ory qubits and $\lceil \log_2 |\mathcal{X}| \rceil$ output qubits. Thus, simulating L time steps yields a total CNOT gate complexity for implementing $\mathcal{A}$ of $\mathcal{O}(L \cdot D^2 \cdot |\mathcal{X}|^2)$. In the scenario of zero bias, a quantum model would have CNOT gate complexity of $\mathcal{O}(L \cdot D_q^2 \cdot |\mathcal{X}|^2)$. In contrast, if we were to directly translate the classical model of memory dimension D_c into a quantum circuit, then the memory would be of dimension D_c , resulting in a CNOT gate complexity of $\mathcal{O}(L \cdot D_c^2 \cdot |\mathcal{X}|^2)$ (see Appendix E for more details).

This distinction can be very significant. Even when starting with a classical model that achieves a theoretical minimal memory dimension D_c , D_q can be significantly lower. There exists families of $\mathcal{P}$ where D_q remains bounded while D_c scales without bound [20, 26, 27]. In the illustrative example below, we see that setting

$\tilde{D} < D_q$ can result in quantum models whose predictions are significantly more accurate than classical counterparts.

ILLUSTRATIVE EXAMPLE

We illustrate the efficacy of our methods in a number of settings. Here, we outline the case of a particle undergoing a stochastic random walk on a ring. In the Appendix H, we include several other examples exhibiting similar advantage, including non-Markovian processes such as the discretised autoregressive models (Appendix H 1).

Consider now a particle on a ring [28]. Let $x_t \in [0, 1)$ denote the position of the particle at time step t on a ring. At each time step,

$$x_{t+1} = \text{Frac}(x_t + y), \quad (5)$$

where the random variable $y \sim P(Y)$, describes the particle's stochastic movement in space, and $\text{Frac}(z) = z - \lfloor z \rfloor$ maps this to the topology of the ring. Here, we take $P(Y)$ to be a zero-mean Gaussian with variance $\sigma^2 = 0.01$. The resulting sequence $\vec{X} = \dots X_{-2}X_{-1}X_0X_1X_2\dots$ forms a continuous-valued discrete-time stochastic process, which models ring-based diffusion dynamics that can appear in diverse contexts such as ring networks and phase noise [29–31].

Exact simulation of this process would require tracking $[0, 1)$ to perfect precision. Thus, in digital simulation, the standard practice is to consider it satisfactory to track x_t some to k bits of precision [32]. We take this as our target process. For proof of concept, we first take $k = 3$, and will later return to discuss how our results generalise for higher values of K . The target process then involves dividing the ring into $N = 2^k = 8$ equal segments (see Figure 3), and thus represents a 8-state process.

We ask: if memory is constrained to four-state models ($\tilde{D} = 4$); how well could we infer conditional futures on the target 8-state process? We compare the performance between (a) baseline classical Monte-Carlo methods using the optimal classical model of dimension $\tilde{D}$ (as measured by KL divergence rate [33]) and (b) our dimension-reduced quantum probability estimator (Algorithm 1).

Figure 4a illustrates the dual advantage of our quantum methods in mitigating sampling error and bias in the context of estimating the probability of observing the output sequence $x_{0:3} = 100$ conditioned on a particular past (Here $x_{-3:0} = 000$). In addition to the standard quadratic-speed-up due to reduced sampling complexity (Figure 4b), the quantum model exhibits significantly less bias – converging to values significantly closer to the true probability. To demonstrate the typicality of the bias advantage, we then consider the set of all 36 events whose actual probability of occurrence conditioned on the same

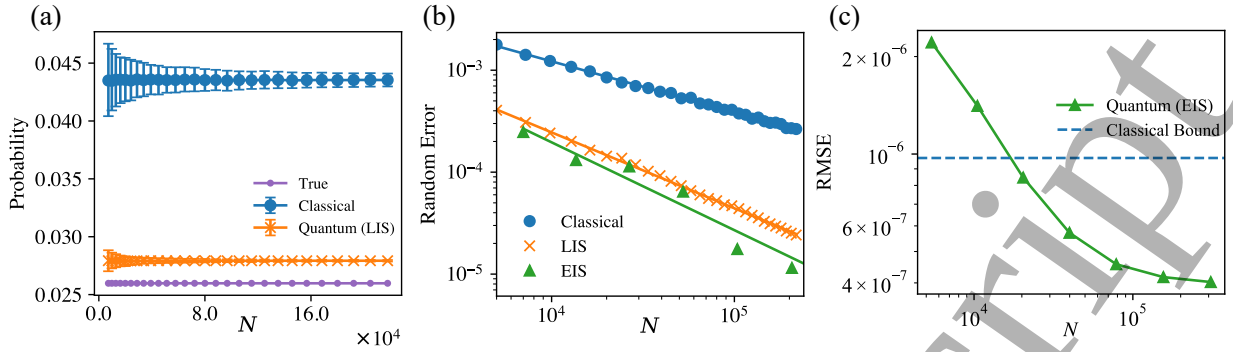


FIG. 4: Illustration of quantum advantage on a cyclic random walk. Here we compare the performance of classical Monte Carlo methods to our memory-reduced quantum probability estimator (MRQPE) when simulating a 8-segment cyclic random walk when memory dimension is constrained to 4. (a) illustrates the estimates of the probability $p = P(x_{0:3} = 100 | x_{-3:0} = 000)$ using classical Monte Carlo (blue), MRQPE with linear (LIS) scheduling (see Appendix F for details on LIS and EIS scheduling) as a function of the number of simulations N , with the true value of p plotted in comparison (purple). The error bars represent the error bars corresponding to the 90th and 10th percentiles over 1000 independent trials. (b) illustrates the advantageous scaling in random error for MRQPE using LIS (orange) and EIS schedules (green), when compared to classical baseline Monte Carlo methods (blue). The numerical values of the fitting parameters and the corresponding correlation coefficient r^2 can be found in table I in Appendix H 5. (c) illustrate the average RMSE of our MRQPE method (with memory dimension $\tilde{D} = 4$) when estimating events with probabilities from 10^{-9} to 10^{-8} over 100 independent trials (green) as a function of N . This is compared against the classical bound (blue), representing the RMSE of classical Monte-Carlo methods with identical memory constraints, in the limit of infinite samples (i.e., $N \rightarrow \infty$). More details on the simulation hyperparameters are given in Appendix G 3.

past lies between 10^{-9} and 10^{-8} (see Appendix G 4 for details). The resulting bias in our classical baseline 4-state model results in an RMSE of $\sim 10^{-6}$ when averaged across all 36 events – even in the limit of infinite samples (Figure 4c). In contrast, the average RMS error of our quantum models dips below this bound at around $N \sim 2 \times 10^4$ samples. Such an advantage would be immediately beneficial for rare-event classification, where we wished to determine that the probability of a target adverse outcome is less than some $\epsilon \ll 1$.

This quantum advantage is not limited to a particular choice of precision k for the target process. Figure 5 documents the analogue of Figure 4c for the case where the target process aims to simulate the dynamics of x_t to $k = 4$ bits of precision, and thus requires a memory dimension of $N = 16$ states to model exactly. We then continue to constrain both the classical baseline and quantum models to a memory dimension of 4. We see that the average RMSE of the classical models on the rare events (with probabilities between 10^{-9} and 10^{-8}) remains bounded below by 8×10^{-4} in the limit of infinite samples. Meanwhile, a 4-level quantum model can undercut this classical bound with approximately 3000 samples – more than a factor of 6 fewer than required for the $k = 3$ baseline. Indeed, in Appendix G 6, we outline why we expect this quantum advantage to persist for increasing values of k as we approach the continuum limit.

DISCUSSION

Quantum amplitude estimation promises quadratic speed-ups in estimating the likelihood of future events, but its practical application faces a significant hurdle: an efficient *quantum sampler* to generate such future distributions in quantum superposition. Here, we provide a means to mitigate this issue when inferring future probabilities of general stochastic processes. Our resulting quantum-enhanced future inference algorithm exhibits (i) the typical speed-up of quantum amplitude estimation from a quadratic improvement in how many times we need to simulate the process and (ii) further improved memory-distortion trade-offs that enable reduced errors from finite-memory constraints. The two advantages synergize - providing potential quantum advantage beyond quadratic speed-up. Our procedure is broadly applicable. It requires as input only a classical predictive model of a process. Such a model could be obtained in a number of established methods, from base analytical models (e.g. Markov machines, random walks) to machine-learned models (e.g. RNNs, Transformers, together with classical model reduction techniques).

Here, we have primarily focused on memory as a key limiting factor for the first generation of universal quantum computing devices, as we anticipate that such devices will be the first to support practical-scale quantum Monte Carlo methods. However, proof-of-principle ex-

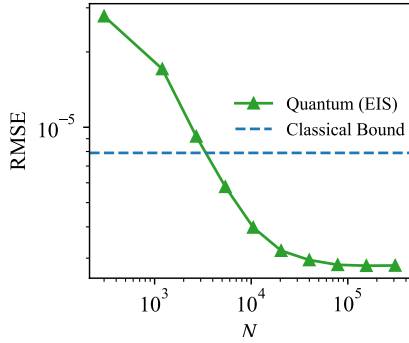


FIG. 5: **Quantum advantage when simulating a 16-state cyclic random walk.** Here, we consider the average RMSE error when estimating all events with probabilities from 10^{-9} to 10^{-8} for the 16-state cyclic random walk (see Appendix G 4 for details). As in the 8-state case, both classical baseline and quantum models are set to have memory dimension of 4. As in Figure 4c, the blue line then denotes the the lower bound on RMSE achieved using classical models — even if we allowed unlimited samples. Meanwhile the green line plots the RMSE error of our quantum models as function of number of simulations. We see that quantum models can achieve RMSE below classical limits using only around 3000 samples.

perimental demonstrations may be possible on nearer-term noisy quantum devices. Indeed, dimensionally reduced single-qubit quantum models have already been experimentally realised [34], including those exhibiting provable accuracy advantage [27]. Meanwhile, our preliminary analysis indicates that dimensional reduction could provide further benefits in such noisy settings. In particular, the size of our unitary U_s depends on the available quantum memory: the larger U_s is, the more elementary gates are required for its implementation, with each gate introducing additional noise. Thus, the additional bias introduced by dimensionally reducing the memory may be more than offset by the benefits of requiring fewer gates. In Appendix G 5, we perform an analysis assuming noise levels comparable to those of current Quantum processors when sampling from the target 8-state cyclic random walk considered above. Our results show that the overall bias in estimating conditional future probabilities generally *decreases* when we use a 2-qubit dimensionally reduced quantum model to simulate the target process, compared with the exact 3-qubit model.

There are also several promising future directions. One is to incorporate classical importance sampling techniques. There, sophisticated methods developed allow the potential to pre-process a classical model to produce a variant that samples more often from atypical sequences

— and thus accelerate estimating their probability of occurrences [35, 36]. Quantum processing turns out particularly effective in reducing the complexity of such variants, where resulting models can have unbounded memory advantage [37]. Incorporating such pre-processing techniques could further enhance our quantum algorithm’s capability of estimating the likelihoods of rare events. A second is further circuit optimisation. We note that the gate complexities quoted for our quantum models are presently worst-case scenarios. In practice, there is significant freedom in choosing the interaction unitary: as the stochastic model constrains only its action when the output register is initialised in $|0\rangle$ [25]; a freedom that may enable further reduction in circuit complexity [38]. Meanwhile, integration of our methodologies with existing transformer and RNN models has exciting prospects. Here, the exponential growth in memory costs of present AI models has motivated model quantization — a classical technique where continuous properties tracked by the AI model are represented by low-bit approximations [39]. Quantum models have shown provably unbounded memory advantage in such settings [40], presenting an ideal case study for replacing classically ‘quantized’ models with truly quantum counterparts.

FUNDING

This work is supported by the National Research Foundation of Singapore through the NRF Investigatorship Program (Award No. NRF-NRFI09-0010), the National Quantum Office, hosted in A*STAR, under its Centre for Quantum Technologies Funding Initiative (S24Q2d0009) and the Singapore Ministry of Education Tier 1 Grant RT4/23 and RG91/25. A.G. is supported by EPSRC DTP studentship EP/W524347/1 2932638. T.J.E. is supported by the University of Manchester Dame Kathleen Ollerenshaw Fellowship. C.Y. is funded by Schmidt Sciences, LLC.

* Corresponding author; chen1554@e.ntu.edu.sg

† akshat.gupta-4@postgrad.manchester.ac.uk

‡ yangchengran92@gmail.com

§ jayne.thompson@ntu.edu.sg

¶ Corresponding author; mgu@quantumcomplexity.org

- [1] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemporary Mathematics* **305**, 53 (2002).
- [2] S. Woerner and D. J. Egger, Quantum risk analysis, *npj Quantum Information* **5**, 15 (2019).
- [3] P. Rebentrost, B. Gupt, and T. R. Bromley, Quantum computational finance: Monte carlo pricing of financial derivatives, *Phys. Rev. A* **98**, 022321 (2018).
- [4] X. Sun, G. Tian, S. Yang, P. Yuan, and S. Zhang, Asymptotically optimal circuit depth for quantum state prepa-

- ration and general unitary synthesis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **42**, 3301 (2023).
- [5] X.-M. Zhang, T. Li, and X. Yuan, Quantum state preparation with optimal circuit depth: Implementations and applications, *Phys. Rev. Lett.* **129**, 230504 (2022).
 - [6] Y. Cheng, D. Wang, P. Zhou, and Z. Tao, A survey of model compression and acceleration for deep neural networks, *ArXiv abs/1710.09282* (2017).
 - [7] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, A Survey of Quantization Methods for Efficient Neural Network Inference, in *Low-Power Computer Vision* (Chapman and Hall/CRC) 1st ed., pp. 291–326.
 - [8] J. P. Crutchfield, Between order and chaos, *Nature Physics* **8**, 17 (2012).
 - [9] J. P. Crutchfield and K. Young, Inferring statistical complexity, *Phys. Rev. Lett.* **63**, 105 (1989).
 - [10] C. Yang, M. Florido-Llinàs, M. Gu, and T. J. Elliott, Dimension reduction in quantum sampling of stochastic processes, *npj Quantum Information* **11**, 34 (2025).
 - [11] F. Cérou, A. Guyader, and M. Rousset, Adaptive multi-level splitting: Historical perspective and recent results, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**, 043108 (2019).
 - [12] J. A. Bucklew, *Introduction to Rare Event Simulation*, Springer Series in Statistics (Springer, New York, NY, 2004).
 - [13] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Applications of Mathematics : Stochastic Modelling and Applied Probability No. 53 (Springer, 2010).
 - [14] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996) pp. 212–219.
 - [15] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, *Quantum Information Processing* **19**, 75 (2020).
 - [16] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, *npj Quantum Information* **7**, 52 (2021).
 - [17] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, Low depth algorithms for quantum amplitude estimation, *Quantum* **6**, 745 (2022).
 - [18] S. Aaronson and P. Rall, Quantum approximate counting, simplified, in *2020 Symposium on Simplicity in Algorithms (SOSA)* (2020) pp. 24–32.
 - [19] A. S. Shai, S. E. Marzen, L. Teixeira, A. G. Oldenziel, and P. M. Riechers, Transformers represent belief state geometry in their residual stream, *ArXiv abs/2405.15943* (2024).
 - [20] T. J. Elliott, C. Yang, F. C. Binder, A. J. P. Garner, J. Thompson, and M. Gu, Extreme dimensionality reduction with quantum modeling, *Phys. Rev. Lett.* **125**, 260501 (2020).
 - [21] S. E. Marzen and J. P. Crutchfield, Informational and Causal Architecture of Discrete-Time Renewal Processes, *Entropy* **17**, 4891 (2015).
 - [22] Y. Choi, M. El-Khamy, and J. Lee, Towards the limit of network quantization, *ArXiv abs/1612.01543* (2016).
 - [23] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, Compressing deep convolutional networks using vector quantization, *ArXiv abs/1412.6115* (2014).
 - [24] S. E. Marzen and S. DeDeo, The evolution of lossy compression, *Journal of The Royal Society Interface* **14**, 20170166 (2017).
 - [25] F. C. Binder, J. Thompson, and M. Gu, Practical unitary simulator for non-markovian complex processes, *Phys. Rev. Lett.* **120**, 240502 (2018).
 - [26] J. Thompson, A. J. P. Garner, J. R. Mahoney, J. P. Crutchfield, V. Vedral, and M. Gu, Causal asymmetry in a quantum world, *Phys. Rev. X* **8**, 031013 (2018).
 - [27] K.-D. Wu, C. Yang, R.-D. He, M. Gu, G.-Y. Xiang, C.-F. Li, G.-C. Guo, and T. J. Elliott, Implementing quantum dimensionality reduction for non-Markovian stochastic simulation, *Nature Communications* **14**, 2624 (2023).
 - [28] A. J. P. Garner, Q. Liu, J. Thompson, V. Vedral, and M. Gu, Provably unbounded memory advantage in stochastic simulation using quantum mechanics, *New Journal of Physics* **19**, 103009 (2017).
 - [29] M. Zhong, K. Shen, and J. Seiferas, Non-uniform random membership management in peer-to-peer networks, in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, Vol. 2 (IEEE, 2005) pp. 1151–1161.
 - [30] A. Kutschireiter, M. A. Basnak, R. I. Wilson, and J. Drugowitsch, Bayesian inference in ring attractor networks, *Proceedings of the National Academy of Sciences* **120**, e2210622120 (2023).
 - [31] A. Piemontese, G. Colavolpe, and T. Eriksson, Discrete-Time Models and Performance of Phase Noise Channels, *IEEE Open Journal of the Communications Society* **5**, 2211 (2024).
 - [32] R. Gray and D. Neuhoff, Quantization, *IEEE Transactions on Information Theory* **44**, 2325 (1998).
 - [33] C. Yang, F. C. Binder, M. Gu, and T. J. Elliott, Measures of distinguishability between stochastic processes, *Phys. Rev. E* **101**, 062137 (2020).
 - [34] F. Ghafari, N. Tischler, J. Thompson, M. Gu, L. K. Shalm, V. B. Verma, S. W. Nam, R. B. Patel, H. M. Wiseman, and G. J. Pryde, Dimensional quantum memory advantage in the simulation of stochastic processes, *Phys. Rev. X* **9**, 041013 (2019).
 - [35] F. Bouchet, J. Rolland, and J. Wouters, Rare event sampling methods, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**, 080402 (2019).
 - [36] S. T. Tokdar and R. E. Kass, Importance sampling: a review, *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 54 (2010).
 - [37] C. Aghamohammadi, S. P. Loomis, J. R. Mahoney, and J. P. Crutchfield, Extreme quantum memory advantage for rare-event sampling, *Phys. Rev. X* **8**, 011025 (2018).
 - [38] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, Quantum circuits for isometries, *Phys. Rev. A* **93**, 032318 (2016).
 - [39] M. Z. Alom, A. T. Moody, N. Maruyama, B. C. Van Essen, and T. M. Taha, Effective quantization approaches for recurrent neural networks, in *2018 international joint conference on neural networks (IJCNN)* (IEEE, 2018) pp. 1–8.
 - [40] T. J. Elliott, M. Gu, A. J. P. Garner, and J. Thompson, Quantum adaptive agents with efficient long-term memories, *Phys. Rev. X* **12**, 011007 (2022).
 - [41] J. P. Crutchfield and D. P. Feldman, Regularities unseen, randomness observed: Levels of entropy convergence, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **13**, 25 (2003).

- [42] C. R. Shalizi and J. P. Crutchfield, Computational mechanics: Pattern and prediction, structure and simplicity, *Journal of Statistical Physics* **104**, 817 (2001), [arXiv:9907176 \[cond-mat\]](#).
- [43] M. Gu, K. Wiesner, E. Rieper, and V. Vedral, Quantum mechanics can reduce the complexity of classical models, *Nature Communications* **3**, 762 (2012).
- [44] J. P. Crutchfield, C. J. Ellison, and J. R. Mahoney, Time's barbed arrow: Irreversibility, crypticity, and stored information, *Phys. Rev. Lett.* **103**, 094101 (2009).
- [45] J. Thompson, P. M. Riechers, A. J. P. Garner, T. J. Elliott, and M. Gu, Energetic advantages for quantum agents in online execution of complex strategies, *Phys. Rev. Lett.* **135**, 160402 (2025).
- [46] G. H. Golub and C. F. Van Loan, *Matrix computations* (JHU press, 2013).
- [47] N. J. Higham, Computing the polar decomposition—with applications, *SIAM Journal on Scientific and Statistical Computing* **7**, 1160 (1986).
- [48] C. Yang, A. J. P. Garner, F. Liu, N. Tischler, J. Thompson, M.-H. Yung, M. Gu, and O. Dahlsten, Provably superior accuracy in quantum stochastic modeling, *Phys. Rev. A* **108**, 022411 (2023).
- [49] C. Yang, F. C. Binder, V. Narasimhachar, and M. Gu, Matrix product states for quantum stochastic modeling, *Phys. Rev. Lett.* **121**, 260602 (2018).
- [50] B. Vanhecke, M. Van Damme, J. Haegeman, L. Vanderstraeten, and F. Verstraete, Tangent-space methods for truncating uniform mps, *SciPost Phys. Core* **4**, 004 (2021).
- [51] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Quantum circuits with uniformly controlled one-qubit gates, *Phys. Rev. A* **71**, 052330 (2005).
- [52] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, *Phys. Rev. A* **83**, 032302 (2011).
- [53] M. Möttönen and J. Vartiainen, Decompositions of general quantum gates, in *Trends in Quantum Computing Research*, edited by e. {by S.Shannon} (Nova Science Publishers, United States, 2006).
- [54] A. M. Krol, A. Sarkar, I. Ashraf, Z. Al-Ars, and K. Bertels, Efficient decomposition of unitary matrices in quantum circuit compilers, *Applied Sciences* **12**, 10.3390/app12020759 (2022).
- [55] V. Shende, S. Bullock, and I. Markov, Synthesis of quantum-logic circuits, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25**, 1000–1010 (2006).
- [56] R. Vale, T. M. D. Azevedo, I. C. S. Araújo, I. F. Araujo, and A. J. da Silva, Circuit decomposition of multiconrolled special unitary single-qubit gates, *Trans. Comp.-Aided Des. Integ. Cir. Sys.* **43**, 802–811 (2024).
- [57] A. J. da Silva and D. K. Park, Linear-depth quantum circuits for multiqubit controlled gates, *Phys. Rev. A* **106**, 042602 (2022).
- [58] Quantinuum, Quantinuum Helios System Specifications, <https://www.quantinuum.com/products-solutions/quantinuum-systems/helios> (2026), accessed 6 August 2026.
- [59] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications* (Springer, 2017).

SUPPLEMENTARY MATERIALS

Appendix A: ϵ -machine

The ϵ -machine of a stochastic process $\mathcal{P}$ is the provably memory-optimal predictive model of the process [41]. Consider a bi-infinite sequence $\dots X_{-2}X_{-1}X_0X_1X_2\dots$, partitioned into the past $\overleftarrow{X} = \dots X_{-2}X_{-1}$ and the future $\overrightarrow{X} = X_0X_1X_2\dots$. The ϵ -machine groups all the past $\overleftarrow{x}$ with identical futures into equivalent classes termed *causal states*, $\{s_i\}$, defined by the following equivalence relation:

$$\overleftarrow{x} \sim_{\epsilon} \overleftarrow{x'} \text{ if and only if } P(\overrightarrow{X}|\overleftarrow{x}) = P(\overrightarrow{X}|\overleftarrow{x'}). \quad (\text{A1})$$

Given a stationary stochastic process with corresponding probability distribution $P(\overleftarrow{X}, \overrightarrow{X})$, a set of causal states, $\mathcal{S} = \{s_i\}, i = 1, \dots, d$ for some finite d , can be determined, which will constitute the state space of the ϵ -machine. The future probability conditioned on any causal state s_i is identical to conditioning on any past $\overleftarrow{x} \in s_i$, i.e.

$$P(\overrightarrow{X}|s_i) = P(\overrightarrow{X}|\overleftarrow{x}_k), \forall x_k \in s_i, \quad i = 1, \dots, d \quad (\text{A2})$$

At each time step, the ϵ -machine operates according to a set of predefined transition probabilities $T_{kj}^{(x)} = P(x, s_j|s_k)$, representing the probability that the system will output an alphabet $x \in \mathcal{X}$ and transition from causal state s_k to state s_j . These ϵ -machines are also *unifilar*: given the initial causal state that the ϵ -machine occupies and output x_t , the subsequent causal state that the ϵ -machine transitions to is determined with certainty.

Appendix B: Complexity measures for ϵ -machines and quantum models

There are two common measures of complexity associated with the ϵ -machine. The first is the *statistical complexity* C_{μ} , defined as

$$C_{\mu} := H[\mathcal{S}] = - \sum_{k=1}^{|\mathcal{S}|} \pi_k \log_2 \pi_k, \quad (\text{B1})$$

where $H[\cdot]$ denotes the Shannon entropy, and π_k is the stationary probability of causal state s_k in the causal state distribution π [9]. This quantity represents the process's entropic memory, or the minimal memory required to accurately simulate or predict the process's future, and is lower-bounded by the mutual information between the past and the future, i.e. $\mathbf{E} = I(\overleftarrow{X}; \overrightarrow{X}) \leq C_{\mu}$ [42]. The second complexity measure is the *topological complexity* d_c , defined as:

$$d_c := \log_2 |\mathcal{S}|, \quad (\text{B2})$$

which quantifies the minimum number of bits required to store the causal states in a single-shot setting [9]. Correspondingly, a quantum model admits analogous complexity measures. The *quantum statistical memory* C_q is defined by the von Neumann entropy:

$$C_q := -\text{Tr}[\rho \log_2 \rho], \quad (\text{B3})$$

where $\rho = \sum_k \pi_k |s_k\rangle\langle s_k|$ is the stationary mixed state over quantum causal states [43]. The *quantum topological memory* is defined as

$$d_q := \log_2 [\text{rank}(\rho)]. \quad (\text{B4})$$

Appendix C: Source of quantum dimensional compression

The key relation that motivates potential quantum-enhanced dimension reduction is the strict inequality $C_{\mu} > \mathbf{E}$, which holds for many stochastic processes [42]. Here, $\mathbf{E} = I(\overleftarrow{X}; \overrightarrow{X})$ represents how much information is shared between the past and future. Naively, we would therefore expect that the minimal memory a model needs about the

past of a process to predict its future would also be $\mathbf{E}$. The strict inequality, though, indicates that this is not always possible. Even the provably optimal classical model must sometimes store wasted information — information in its memory that can never be recovered from the future. Thus, observing the entire future of a process may not allow us to retrodict precisely which state the ϵ -machine was in at present:

$$C_\mu - \mathbf{E} = H(S | \vec{X}). \quad (\text{C1})$$

This discrepancy is known in the complexity mechanics literature as *crypticity* [44]. Intuitively, one way to understand how it arises is to consider a classical ϵ -machine in which two causal states have distinct conditional future distributions that nevertheless overlap. A classical machine must still allocate memory to distinguish between these states in order to know which conditional future distribution to sample from. However, seeing the entire future may not allow us to retrodict which state the machine started in, and thus not all the information allocated to storing this distinction is reflected in the future. Such a pair of causal states has more recently been referred to as a *causally wasteful pair* and leads to unavoidable heat dissipation when such models are run [45].

Quantum models can reduce this waste because they have the additional freedom to use non-orthogonal memory states. Under the mapping $s_i \mapsto |s_i\rangle$, the model dynamics take the form

$$U_s |s_i\rangle |0\rangle = \sum_{x \in \mathcal{X}} \sum_k \sqrt{T_{ik}^x} |s_k\rangle |x\rangle. \quad (\text{C2})$$

Whenever $C_\mu > \mathbf{E}$, at least one pair of causal states cannot be perfectly distinguished by observing the future outputs. This construction then guarantees that their corresponding quantum memory states are mutually non-orthogonal, implying a strict reduction in entropic memory, such that $C_q < C_\mu$. An intuition on how these savings comes about can be gained by noting that the unitary leaves the output register and the memory entangled. Measuring the output register reproduces the required transition probabilities while collapsing the memory into the appropriate state for the next time step. As such, the quantum machine never needs to know the output symbol to make the appropriate memory transitions.

It is important, however, to distinguish this entropic memory reduction from a reduction in the exact memory dimension. The exact quantum memory dimension and its corresponding topological memory are

$$D_q = \text{rank}(\rho), \quad d_q = \log_2 D_q, \quad (\text{C3})$$

respectively. Non-orthogonality can lower C_q without necessarily lowering D_q , since a set of non-orthogonal states may still span the full Hilbert space. Nevertheless, when the spectrum of ρ decays rapidly, most of its probability weight is concentrated within a much smaller subspace. In the canonical iMPS representation of the process, this spectral concentration is reflected in the corresponding bond Schmidt spectrum. It therefore becomes possible to approximate the process using an iMPS of reduced bond dimension $\tilde{D}$, while discarding only a small spectral weight. As shown by Yang et al. [10], this observation provides a controlled route to lossy quantum dimensional compression. For a reduced bond dimension $\tilde{D}$, with corresponding memory cost $\tilde{d} = \log_2 \tilde{D}$, the resulting distortion can be bounded in terms of $C_q/\tilde{d}$, as detailed in Appendix D 4.

Appendix D: Algorithm subroutines

1. Polar Decomposition

Polar Decomposition is used extensively in the construction of the unitary operators used in the algorithms. This section provides a brief explanation of the method. By Singular Value Decomposition (SVD), any $m \times n$ matrix A can be decomposed as follows:

$$A = U \Sigma V^\dagger, \quad (\text{D1})$$

where $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ are orthonormal matrices, and $\Sigma = \text{diag}(\sigma_0, \sigma_1, \dots) \in \mathbb{C}^{m \times n}$ [46]. U is referred to as the *left singular matrix*, V as the *right singular matrix*, and the elements of Σ , i.e. $\{\sigma_0, \sigma_1, \dots\}$, as the *singular values*. Polar decomposition extends on SVD. Restricting A to square matrices, we define two matrices

$$P = V \Sigma V^\dagger \quad (\text{D2})$$

$$Q = U V^\dagger \quad (\text{D3})$$

where Q (the *orthogonal polar factor*) is unitary and P (the *symmetric polar factor*) is symmetric and positive semidefinite such that

$$A = QP \quad (\text{D4})$$

The polar decomposition of any non-singular matrix is also unique [46]. The polar decomposition is very closely related to the *orthogonal Procrustes problem*, which is stated as follows: Given two data matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times n}$,

$$\text{minimize } \|A - BQ\|_F, \text{ subjected to } Q^\dagger Q = I_n \quad (\text{D5})$$

where $\|\cdot\|_F$ is the *Frobenius norm* and calculated as

$$\|Y\|_F^2 = \sum_i \sum_j y_{ij}^2 \quad (\text{D6})$$

If the Frobenius norm is used as the distance measure, by setting $B = I_n$, the solution to the problem gives $Q = UV^\dagger$, which is exactly the orthogonal polar factor of polar decomposition. This means that the solution of the polar decomposition can be found by finding the closest orthogonal matrix Q from A [47]. Therefore, Q fulfils the following minimisation:

$$\min \|A - Q\|_F \quad (\text{D7})$$

Applying this to our problem, if we populate any columns of A with orthogonal vectors, and leave the other columns with zeros, the populated columns will be copied over into Q by the nature of the minimisation of the Frobenius norm. Thus, we will get a populated unitary matrix Q with the desired columns after matrix decomposition. Polar decomposition is used extensively here in the construction of the unitaries used in the quantum model.

2. Constructing the U_s Operator

An approach to constructing the unitary U_s involves using Kraus operators. These operators, denoted as $K^x, x \in \mathcal{X}$, act only on the quantum memory and satisfy the completeness relation:

$$\sum_{x \in \mathcal{X}} (K^x)^\dagger K^x = \mathbb{I} \quad (\text{D8})$$

For a set of output alphabets $\mathcal{X}$, we can define a set of Kraus operators $\{K^x\}, x \in \mathcal{X}$ that describe the evolution of the quantum memory following each interaction with the ancillary qubit through the unitary matrix [48]:

$$K^x := \langle x | U_s | 0 \rangle. \quad (\text{D9})$$

Specifically,

$$K_{ij}^x = \langle i | \langle x | U_s | j \rangle | 0 \rangle. \quad (\text{D10})$$

These Kraus operators can be obtained from the transition matrices $\{T^x\}$ through constructing a Matrix Product State (MPS) with site matrices $\sqrt{T_{ij}^x}$. The Kraus operators can be obtained through the left canonical form of this MPS [49].

With the definition of the Kraus operators of the process, we can define several columns of the unitary operator U_s , with the remaining matrix elements being populated, for example, through a Gram-Schmidt process. These steps define the unitary construction subroutine:

Algorithm 2 Construction of quantum models

```

1: procedure QUANTUMMODEL( $\{T^x\}$ )
2:    $\{K^x\} \leftarrow \text{GETKRAUSOPS}(\{T^x\})$ 
3:    $U_s \leftarrow \text{CREATEUSTEP}(\{K^x\})$ 
4:   return  $\{K^x\}, U_s$ 
5:
6: procedure GETKRAUSOPS( $\{T^x\}$ )
7:   Create MPS  $A$  with site matrices  $\sqrt{T_{ij}^x}$ 
8:    $\{K^x\} \leftarrow$  Left canonical form of MPS  $A$ .
9:   return  $\{K^x\}$ 
10:
11: procedure CREATEUSTEP( $\{K^x\}$ )
12:    $n_s \leftarrow$  dimensions of  $K^x$ 
13:    $n_x \leftarrow$  number of elements in  $\{K^x\}$ 
14:    $m \leftarrow 2^{\lceil \log_2 |S| \rceil}$  ▷ round to a power of 2
15:    $a \leftarrow 2^{\lceil \log_2 |\mathcal{X}| \rceil}$ 
16:    $U \leftarrow$  array of size  $(m \times a, m \times a)$ 
17:   for  $x = 1$  to  $n_x$  do
18:     for  $i = 1$  to  $n_s$  do
19:       for  $j = 1$  to  $n_s$  do
20:          $U[x + i \cdot a, j \cdot a] \leftarrow K^x[i, j]$ 
21:    $U_s \leftarrow$  Polar decomposition of  $U$ 
22:   return  $U_s$ 

```

Further details on this construction method can be found in [49].

3. Constructing the set of initialiser matrices

The initialiser matrices, $\{U_{\text{ini}}^i\}, i = \{0, \dots, D-1\}$ where D is the number of quantum causal states, are used to initialise the memory register to the correct initial quantum causal state at the start of the simulation. Similar to the construction of the U_s operator, there are two ways to construct the initialiser matrices.

a. Using Causal States

The first method to create the initialiser matrices involves defining the set of causal states in the computational basis, as described in [25]. To create the operators necessary for the quantum circuit, we establish a set of square matrices $A_i, i = 0, \dots, D-1$ of size $n \times n, n = 2^{\lceil \log_2 D \rceil}$ with the following structure:

$$A_i = \begin{bmatrix} | & 0 & \dots & 0 \\ |s_i\rangle & 0 & \dots & 0 \\ | & 0 & \dots & 0 \end{bmatrix}, \quad (\text{D11})$$

where the first column is defined by the quantum causal state. Polar decomposition can be used to obtain the initialising unitary matrix U_{ini} of the same size as follows:

$$A_i = U_{\text{ini}} P, \quad (\text{D12})$$

where P is some positive semidefinite Hermitian matrix. Through this method, we obtain a set of initialiser matrices $U_{\text{ini}}^i, i = 0, \dots, D-1$, where each causal state $|s_i\rangle$ is mapped to a unique initialiser matrix U_{ini}^i .

b. Using Kraus Operators

An alternative method for generating these matrices involves the use of Kraus operators associated with the process, which can be obtained from the transition matrices as in section C.2. The unifilar property of the ϵ -machine allows for the unique determination of the initial quantum causal state given a past sequence that is equal to or longer than the Markov order of the process. If a Markov order r of the process and a specific past $x_{-r:0} = x_{-r} \dots x_{-2} x_{-1}$ is

given, this past sequence will, by definition, be an element of a particular quantum causal state $|s_i\rangle$. This quantum causal state $|s_i\rangle$ can be produced through the repeated application of the respective Kraus operators:

$$|s_i\rangle \propto K^{x-1} K^{x-2} \dots K^{x-r} |\phi\rangle, \quad (\text{D13})$$

where $|\phi\rangle$ is in any arbitrary state. A suitable set of pasts $\{\bar{x}\}$ will allow the construction of the full set of causal states, which can be converted into initialiser matrices using the aforementioned method. The following is the subroutine `encodePast` which encodes the past into the unitary operator U_{ini} .

Algorithm 3 Subroutine to encode the past into U_{ini}

```

procedure ENCODEPAST( $x_{-r:0}, \{K^x\}$ )
   $m \leftarrow \lceil \log_2 |\mathcal{S}| \rceil$ 
   $\mathbf{x} \leftarrow$  leading eigenvector of one of the Kraus operators  $\{K^x\}$ 
  for  $i$  from  $-r$  to  $-1$  do
     $\mathbf{x} \leftarrow K^{x_i} \mathbf{x}$ 
  Let  $A$  be a zero matrix of size  $2^m \times 2^m$ 
   $A \leftarrow [\mathbf{x}, \mathbf{0}, \dots, \mathbf{0}]$ 
   $U_{\text{ini}} \leftarrow$  Polar decomposition of  $A$ 
  return  $U_{\text{ini}}$ 

```

4. Dimension Reduction of Quantum Models

In general, the topological complexity of a process, d_q , upper-bounds its statistical complexity C_q . If the gap between d_q and C_q is sufficiently large, it allows for a substantial reduction in the quantum model memory dimensions without introducing significant errors. The process can be represented as an infinite Matrix Product State (iMPS) [49]. We denote the bond dimension of the original iMPS by D and the target bond dimension of the truncated iMPS by $\tilde{D} \leq D$. As shown in [10], for a given truncated dimension $\tilde{D}$, the error between the approximated quantum process $|\tilde{P}\rangle$ and the true process $|P\rangle$ is upper-bounded by $O\left(\frac{C_q}{\tilde{d}}\right)$, where $\tilde{d} = \log_2(\tilde{D})$, and the error is defined as the Quantum Fidelity Divergence Rate (QFDR) [50]. A straightforward truncation can be done by keeping only the $\tilde{D}$ largest Schmidt coefficients across each bond of the iMPS and discarding the rest. The resulting reduced iMPS produces the approximate model as described in [10]. The method adopted in this work is the improved version [10], where an ansatz iMPS $\tilde{A}$ of bond dimension $\tilde{D}$ is initialised and then iteratively optimized. This method yields a more stable and accurate reduced iMPS, which is used to compute the final truncated model.

Algorithm 4 Subroutine to truncate the quantum model

```

procedure TRUNCATE( $\{K^x\}, \tilde{D}$ )
  Represent  $\{K^x\}$  as an iMPS  $A$ 
  Initialise an ansatz iMPS  $\tilde{A}$  of bond dimension  $\tilde{D}$ 
  Iteratively optimize  $\tilde{A}$  to maximise its overlap with target iMPS  $A$  using algorithm in [10].
  Extract reduced Kraus operators  $\{\tilde{K}^x\}$  from the left canonical form of  $\tilde{A}$ 
   $U_s \leftarrow \text{CREATEUSTEP}(\{\tilde{K}^x\})$ 
  return  $\{\tilde{K}^x\}, U_s$ 

```

The main computational bottleneck in `Truncate` lies in the iterative optimization. The most computationally intensive step consists of computing the leading eigenvalues and associated left and right eigenvectors. If dense diagonalisation is used to compute the eigenvalues and eigenvectors, and T iterations are required before reaching the desired error tolerance, the complexity of the algorithm is

$$\mathcal{O}\left(T(D\tilde{D})^3\right). \quad (\text{D14})$$

If a matrix-free approach is used, the complexity of this step can be reduced to

$$\mathcal{O}\left[TK|\mathcal{X}|\left(D^2\tilde{D} + D\tilde{D}^2\right)\right], \quad (\text{D15})$$

where K is the number of transfer-operator applications per variational iteration. After convergence, the isometry defined by the iMPS must be completed to a unitary acting on a space of dimension $|\mathcal{X}|\tilde{D}$. Using a standard dense Gram–Schmidt or QR completion introduces an additional one-off worst-case cost of

$$\mathcal{O}\left(|\mathcal{X}|\tilde{D}^3\right). \quad (\text{D16})$$

These complexity estimates are conditional on the iteration counts T and K , which depend on the truncation threshold, initialization, and spectral properties of the transfer operators. For fixed T and K , the arithmetic cost is polynomial in D , $\tilde{D}$, and $|\mathcal{X}|$.

5. Dimensional Reduction of Classical Models

To compare the performance of the dimension reduced quantum model, we construct an equivalent dimensionally reduced classical model. Given a future length L that we wish to estimate, and the reduced dimension $\tilde{D}$, the optimal dimension-reduced classical model is defined as the reduced model of dimension $\tilde{D}$ which minimises the Kullback–Leibler divergence rate with the original exact classical ϵ -machine [33]. This reduced model is found by a brute-force search over all possible clusterings of the causal states in the original ϵ -machine to $\tilde{D}$ dimensions.

Appendix E: Gate Complexity Analysis

For a quantum model with $D = |\mathcal{S}|$ hidden states and an alphabet size of $|\mathcal{X}|$, the memory register requires $\lceil \log_2 D \rceil$ qubits, while $\lceil \log_2 |\mathcal{X}| \rceil$ qubits are needed to store the model's output at each time step. The key component of the algorithm is the operator $\mathcal{A}$, which consists of the U_{ini} and U_s operators. The operator U_{ini} prepares the memory register in the appropriate quantum causal state, and its CNOT cost is, in general, upper bounded by $\mathcal{O}(2^{\lceil \log_2 D \rceil})$ [51, 52]. The operator U_s acts on both the memory register and the output register, involving a total of $\lceil \log_2 D \rceil + \lceil \log_2 |\mathcal{X}| \rceil$ qubits. Its general implementation requires at most $\mathcal{O}(4^{\lceil \log_2 D \rceil + \lceil \log_2 |\mathcal{X}| \rceil})$ CNOT gates [53–55]. Since U_s needs to be applied L times to simulate a sequence of L time steps, the total CNOT gate complexity of implementing $\mathcal{A}$ is

$$\begin{aligned} \text{CNOT}(\mathcal{A}) &\sim \mathcal{O}(L \cdot 4^{\lceil \log_2 D \rceil + \lceil \log_2 |\mathcal{X}| \rceil} + 2^{\lceil \log_2 D \rceil}) \\ &\sim \mathcal{O}(L \cdot 4^{\lceil \log_2 D \rceil + \lceil \log_2 |\mathcal{X}| \rceil}) \\ &\sim \mathcal{O}(L \cdot D^2 \cdot |\mathcal{X}|^2). \end{aligned}$$

If the quantum model is direct map from classical models using orthogonal encoding, then $|\mathcal{S}| = D_c$, and we have

$$\text{CNOT}(\mathcal{A}) \sim \mathcal{O}(L \cdot D_c^2 \cdot |\mathcal{X}|^2). \quad (\text{E1})$$

If the quantum model is created using a non-orthogonal encoding, then $D = D_q \leq D_c$, and we have

$$\text{CNOT}(\mathcal{A}) \sim \mathcal{O}(L \cdot D_q^2 \cdot |\mathcal{X}|^2). \quad (\text{E2})$$

Finally, if the quantum model is further compressed to $\tilde{D} < D_q$, then we have

$$\text{CNOT}(\mathcal{A}) \sim \mathcal{O}(L \cdot \tilde{D}^2 \cdot |\mathcal{X}|^2). \quad (\text{E3})$$

For the CNOT complexity of the Grover operator, it can only be defined with respect to the complexity of the binary function f . For a simple search protocol, the oracle S_χ and the operator S_0 can both be implemented using multicontrolled- Z gates, which only requires a linear number of CNOT gates with respect to the number of qubits [56, 57]. As a result, their CNOT costs are $\mathcal{O}(L \cdot \lceil \log_2 |\mathcal{X}| \rceil)$ and $\mathcal{O}(L \cdot \lceil \log_2 |\mathcal{X}| \rceil + \lceil \log_2 \tilde{D} \rceil)$, respectively. Overall, the total CNOT cost of implementing the Grover operator $\mathbf{Q}$ for a search function is

$$\text{CNOT}(\mathbf{Q}) \sim \mathcal{O}(L \cdot \tilde{D}^2 \cdot |\mathcal{X}|^2). \quad (\text{E4})$$

Assuming that the complexity of constructing S_χ and S_0 is simpler than constructing $\mathcal{A}$, the final CNOT complexity of the circuit is dominated by the complexity of $\mathcal{A}$. Under this assumption, using the conventional quantum amplitude

estimation, to reach an absolute additive error of ϵ , the number of samples required is $O(\frac{1}{\epsilon} \log \frac{1}{\delta})$, where δ is the failure rate of the algorithm [1]. Thus, to reach a certain additive error ϵ with a failure rate of δ , the number of Grover operations is then

$$N \sim O\left(L \cdot \tilde{D}^2 \cdot |\mathcal{X}|^2 \frac{1}{\epsilon} \log \frac{1}{\delta}\right) \quad (\text{E5})$$

Appendix F: Maximum Likelihood Amplitude Estimation

Maximum Likelihood Amplitude Estimation (MLAE) represents a family of shallow-circuit variants of standard quantum amplitude estimation. As with standard amplitude estimation, each MLAE algorithm takes in as input the algorithm operator $\mathcal{A}$ and the Grover operator $\mathbf{Q}$ and aims to output the probability estimate $\hat{p}$ of the subspace as defined by the embedded binary function f . Unique to the MLAE algorithm, it also takes in a *schedule*, which is a set of circuits defined by $\{(m_k, N_k)\}_k$, where m_k is the number of Grover operations applied in the k -th circuit, and N_k is the number of measurement shots for the k -th circuit. Both N_k and m_k are implicit functions of the overall sample complexity N . The measurement results are passed into a maximum likelihood estimation subroutine to obtain the probability estimate $\hat{p}$ of the desired sequence. Given the probability of obtaining the desired sequence $p = \sin^2 \theta$, for some unknown parameter $\theta \in [0, \pi/2]$, for each of the circuit and its measurement results $\{h_k, N_k - h_k\}$, where h_k is the number of times the target sequence is measured, we have a likelihood function of the following form:

$$L_k(h_k; \theta) \propto [\sin^2(\theta_k)]^{h_k} [\cos^2(\theta_k)]^{N_k - h_k}, \quad 0 \leq k \leq M \quad (\text{F1})$$

where $\theta_k = (2m_k + 1)\theta$. The unified likelihood function then is defined by

$$L(\mathbf{h}; \theta) = \prod_{k=0}^M L_k(h_k; \theta). \quad (\text{F2})$$

A maximum likelihood estimation is then conducted to estimate θ :

$$\hat{\theta} = \arg \max_{\theta \in [0, \pi/2]} \ln L(\mathbf{h}; \theta). \quad (\text{F3})$$

This gives the probability estimate $\hat{p}$:

$$\hat{p} = \sin^2 \hat{\theta}. \quad (\text{F4})$$

The two schedules used in this paper is the Linear Increment Schedule (LIS) and Exponential Increment Schedule (EIS), which is defined as follows:

- LIS: $m_0 = 0, m_1 = 1, m_2 = 2, \dots, m_k = k, N_k = \text{constant over all circuits.}$
- EIS: $m_0 = 0, m_1 = 1, m_2 = 2, m_3 = 4, \dots, m_k = 2^{k-1}, N_k = \text{constant over all circuits.}$

For a given sample complexity N and schedule, the corresponding values of m_k and N_k can be determined. The exact relationship between N, m_k and N_k is defined by equation (G5).

Appendix G: Cyclic random walk

1. Discretization of the cyclic random walk

For the cyclic random walk [28], we discretize the cycle into 8 equal intervals: $\mathcal{I}_k = \{x : |x - \frac{k}{8}| < \frac{1}{16}\}$, which give rise to 8 distinct states, $\{s_k\}_{k=0}^7$, as illustrated in Figure G1(a). The emission at each transition corresponds to the index of the resulting state. For example, a transition from s_i to s_j emits j , forming the alphabet $\mathcal{X} = \{0, 1, 2, \dots, 7\}$. Due to the cyclic nature of the cyclic walk, there are multiple distinct paths by which the particle may travel from one state to another, potentially looping around the circle multiple times. However, since the Gaussian distribution used for the example has a zero mean with variance $\sigma^2 = 0.01$, the probability of multi-cycle transitions is vanishingly

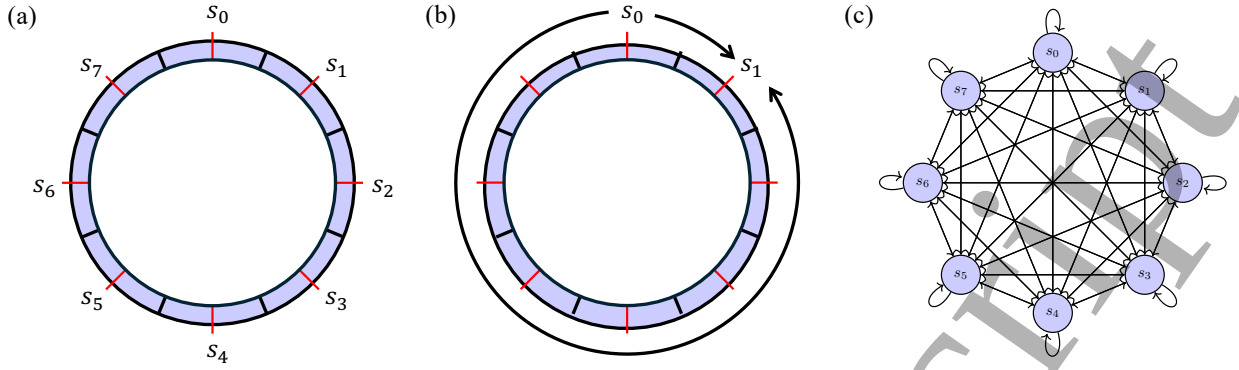


FIG. G1: (a) Illustration of the discretization of the interval $[0, 1]$ into eight equal parts, with the causal states s_k representing the midpoint of each interval. (b) Example of the two possible paths considered for the transition from state s_0 to s_1 . (c) A state machine representation of the 8-state discrete cyclic random walk. Each edge represents a transition from state s_i to s_j , with an emission of symbol j . The probability of transition is given in equation (G2).

small. Thus, we restrict our attention to transitions within a single cycle. Accordingly, each state transition from s_i to s_j is associated with only two possible paths: one clockwise and one anticlockwise around the ring. Furthermore, by symmetry, the transition probability between any two states depends only on their relative displacement along the ring. Therefore, for each state transition from s_i to s_j , the transition probability is given by

$$P(x = j, s_j | s_i) = P(s_j | s_i) = P\left(\mathcal{I}_j = \left\{ \left| z - \frac{j}{8} \right| < \frac{1}{16} \right\} \middle| z = \frac{i}{8} \right) \quad (\text{G1})$$

$$= \alpha \left[\Phi\left(d_{ij} + \frac{1}{16}\right) - \Phi\left(d_{ij} - \frac{1}{16}\right) \right] + \left[\Phi\left(d_{ij} - 1 + \frac{1}{16}\right) - \Phi\left(d_{ij} - 1 - \frac{1}{16}\right) \right], \quad (\text{G2})$$

where

$$d_{ij} = \frac{(j - i) \bmod 8}{8} \in \left\{ 0, \frac{1}{8}, \dots, \frac{7}{8} \right\}, \quad (\text{G3})$$

and $\alpha = 1/[\Phi(15/16) - \Phi(-17/16)]$ is the re-normalisation constant to account for the discarding of multi-cycle paths, and $\Phi(x)$ is the cumulative distribution function of the Gaussian distribution.

2. Dimension reduced 4-state cyclic random walk

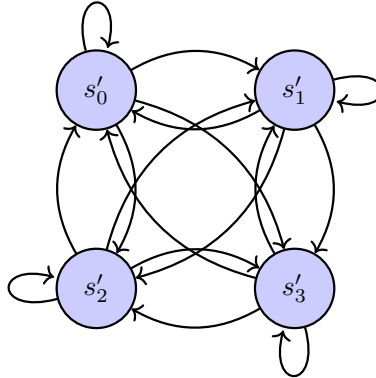


FIG. G2: Illustration of the 4-state dimension reduced cyclic random walk. The transition probability and corresponding emissions are defined by equation (G4).

The optimal classical dimension reduced 4-state cyclic random walk clusters adjacent states together in the following groups: $s'_0 = \{s_0, s_1\}$, $s'_1 = \{s_2, s_3\}$, $s'_2 = \{s_4, s_5\}$, $s'_3 = \{s_6, s_7\}$. This gives the 4-state machine as shown in Figure G2. The state transition from s'_i to s'_j is then

$$P(x = k, s'_j | s'_i) = P(x = k, s_k | s_{2i}) + P(x = k, s_k | s_{2i+1}). \quad (\text{G4})$$

In general, the reduced quantum model exhibits lower bias compared to the reduced classical model. For certain specific events, the classical reduced model demonstrates smaller bias, but these cases are in the minority. Among the 36 events with probability between 10^{-9} to 10^{-8} used for numerical simulation, only 5 events show the quantum model producing higher bias than the classical model.

3. Simulation Hyperparameters for Cyclic Random Walk

For the numerical simulations on the error scaling in Figure 4(a) and (b), the length of the future outcomes is set to $L = 3$, i.e. $x_{0:3} = x_0 x_1 x_2$. The simulations are performed using the Exponential and Linear Increment Schedule (EIS and LIS, respectively), and the results are benchmarked against classical estimators. To quantify the estimation error, we compute the RMSE ϵ (equation (1)), which is then fitted with a power-law model of the form $\epsilon = bN^a$, where $a, b \in \mathbb{R}$ are the fitting parameters, and N is the number of applications of the operator $\mathcal{A}$ (each Grover operator consist of 2 $\mathcal{A}$ operations). To calculate the sample complexity, N is defined as the number of applications of the algorithm operator $\mathcal{A}$, which, for each schedule, has the following form:

$$N = \sum_{k=0}^M N_k (2m_k + 1). \quad (\text{G5})$$

This is used for comparison with classical Monte Carlo methods. The quality of the fit is evaluated via the correlation coefficient r . The best-fit parameters and the corresponding correlation coefficient r^2 values are provided in Appendix H5 (Table I). For Figure 4(a) and (b), the number of shots N_k for EIS and LIS is set to 200. Each numerical experiment is repeated 1000 times for the calculation of the RMSE. For Figure 4(c), among all future outcomes with $L = 3$, events with probabilities between 10^{-8} and 10^{-9} (corresponding to 36 different events, details are in Appendix G4) were chosen as the target. The EIS schedule is used with $N_k = 300$ shots, and 100 independent trials were conducted to compute the RMSE. The mean RMSE of all the events are computed for the graph. Similarly, the mean classical error bound is computed from the classical error bounds of all the events.

4. List of futures used in cyclic random walk example

For the computation of Figure 4(c), starting from the past $x_{-3:0} = 000$, and estimating three steps into the future, 36 different future events have an occurrence probability between 10^{-9} to 10^{-8} . These events are listed as follows:

$$\begin{aligned} x_{0:3} = \{ & 037, 041, 047, 051, 140, 150, 152, 162, 326, 337, 340, 370, \\ & 376, 377, 410, 411, 412, 430, 436, 441, 447, 450, 452, 470, \\ & 476, 477, 510, 511, 512, 540, 551, 562, 726, 730, 736, 740 \} \end{aligned} \quad (\text{G6})$$

For the computation of Figure 5, we utilize the same past of $x_{-3:0} = 000$, and estimate the probabilities two steps into the future, which corresponds to 12 different future have an occurrence probability between 10^{-9} to 10^{-8} . These events are listed as follows:

$$\begin{aligned} x_{0:2} = \{ & [5, 13], [6, 13], [6, 15], [7, 1], \\ & [7, 13], [8, 3], [8, 13], [9, 3], \\ & [9, 15], [10, 1], [10, 3], [11, 3] \} \end{aligned} \quad (\text{G7})$$

5. Improving Simulation Accuracy through Dimensional Reduction on Noisy Processors

On an ideal quantum hardware, dimensional reduction trades a small, controlled truncation bias for a significant reduction in the required qubit count and circuit complexity. On noisy processors, however, this resource reduction

provides an additional practical advantage: decreasing the size of the unitary required to simulate the process may also substantially reduce the gate count and gate error accumulation. The corresponding reduction in hardware-induced error can outweigh the truncation bias, allowing the truncated model to achieve a lower estimation error than the untruncated model.

In this section, we illustrate an example where a quantum computer with a noise profile comparable to that of contemporary NISQ devices can perform better when using our dimensional-reduction method. We utilize the same eight-state cyclic random walk as the reference model, and the corresponding four-state memory-reduced model. As demonstrated in the paper, the memory reduction introduces only a small amount of bias. We consider a quantum device with a CNOT depolarizing error rate of 7.9×10^{-4} and a single-qubit gate depolarizing error rate of 2.5×10^{-5} , chosen to be comparable to the performance of the Quantinuum Helios processor [58]. We take $P(Y)$ to be a zero-mean Gaussian distribution with variance $\sigma^2 = 0.01$.

Using these parameters, we benchmark the ability of the two models to estimate the probability of the one-step future sequence $x_{0:1} = 1$, conditioned on the past $\bar{x} = 000$. The true probability is ≈ 0.235589 . For each model, we draw $N_{\text{shots}} = 10,000$ samples, and repeat the experiment across 10 independent trials before computing the mean. The untruncated model yields a mean estimate of $\hat{p} = 0.1372(28)$, whereas the truncated model yields $\hat{p} = 0.1929(26)$. The corresponding absolute errors are 0.098 and 0.043, respectively, amounting to a reduction in estimation error of more than 50%.

We extend this analysis to all eight possible one-step outputs. Averaged over these outputs, the untruncated model yields a mean absolute error of 0.126, whereas the truncated model yields a mean absolute error of 0.0679. Thus, the approximately 50% reduction in estimation error persists across the full output alphabet. Finally, we perform a parameter sweep over different instances of the cyclic random walk by varying σ from 0.1 to 0.3. As shown in Figure G3, the reduction in estimation error persists throughout the tested parameter range.

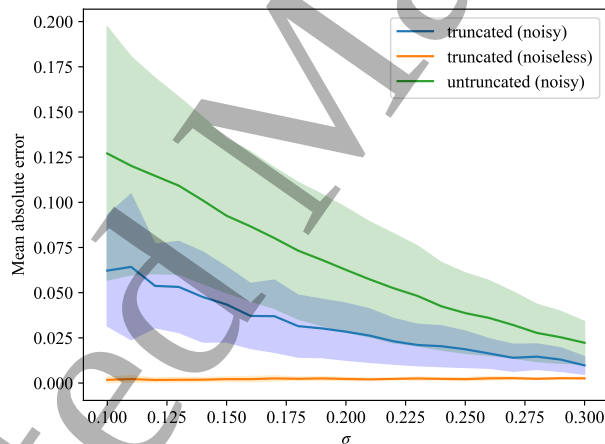


FIG. G3: Mean absolute errors of the noisy truncated and untruncated models for different values of σ , with the noiseless truncated model shown as a baseline. The truncated model has a lower mean absolute error than the untruncated model throughout the tested parameter range.

6. Finer Discretization and the Continuum Limit

Our main analysis has focused on a target process obtained by discretizing the cyclic random walk to $k = 3$ bits of precision, corresponding to $N = 2^k = 8$ discrete states. This choice was made for proof-of-principle purposes. As the discretization is made finer by increasing k , the resulting target process approximates the underlying continuous-valued cyclic random walk ever more closely. It is therefore natural to ask whether the benefits of quantum memory reduction persist as the discretization distortion is reduced.

To formalize this, let $P^{(k)}$ denote the target stochastic process obtained from the k -bit discretization of the cyclic random walk, with $N = 2^k$ states. An exact classical model of $P^{(k)}$ therefore requires k bits of memory. Meanwhile,

let $\tilde{P}^{(k)}$ denote an approximation generated by a dimensionally reduced quantum model with memory dimension

$$\tilde{D} = 2^n, \quad (\text{G8})$$

such that only $n < k$ memory qubits are required. The iMPS framework underlying our dimensional-reduction method provides a useful theoretical bound on the distortion introduced by this reduction. Let $|P^{(k)}\rangle$ and $|\tilde{P}^{(k)}\rangle$ denote the corresponding quantum samples. Their distortion can be quantified by the quantum fidelity divergence rate (QFDR),

$$R_F(|P^{(k)}\rangle, |\tilde{P}^{(k)}\rangle) := -\lim_{L \rightarrow \infty} \frac{1}{2L} \log_2 F(|P_L^{(k)}\rangle, |\tilde{P}_L^{(k)}\rangle), \quad (\text{G9})$$

where

$$F(|\psi\rangle, |\phi\rangle) = |\langle\psi|\phi\rangle|. \quad (\text{G10})$$

Here, $|P_L^{(k)}\rangle$ and $|\tilde{P}_L^{(k)}\rangle$ denote the respective length- L quantum samples. A low QFDR also implies a low statistical fidelity divergence rate between the corresponding stochastic processes. Let

$$C_q^{(k)} = -\text{Tr}[\rho^{(k)} \log_2 \rho^{(k)}] \quad (\text{G11})$$

denote the quantum statistical complexity of $P^{(k)}$, where $\rho^{(k)}$ is the stationary state of the quantum memory. Ref. [10] shows that, when the available quantum memory is restricted to dimension $\tilde{D} = 2^n$, there exists a reduced quantum model satisfying

$$R_F(|P^{(k)}\rangle, |\tilde{P}^{(k)}\rangle) \leq \mathcal{O}\left(\frac{C_q^{(k)}}{n}\right) = \mathcal{O}\left(\frac{C_q^{(k)}}{\log_2 \tilde{D}}\right). \quad (\text{G12})$$

Thus, when the quantum statistical complexity is much smaller than the number of memory qubits available, significant dimensional reduction can be achieved while introducing only small distortion.

This observation is particularly relevant to the cyclic random walk considered here. For cyclic random walks subject to Gaussian noise of fixed nonzero width, the quantum statistical complexity remains bounded as the discretization becomes progressively finer [28]. That is, there exists some finite constant C_* such that

$$C_q^{(k)} \leq C_* \quad (\text{G13})$$

for arbitrarily large k , even though an exact classical representation requires 2^k memory states, or k bits. We emphasize that this boundedness concerns the *entropic* quantum memory $C_q^{(k)}$; the Hilbert-space dimension required for an exact quantum model need not itself remain bounded. It is precisely this separation between information content and Hilbert-space dimension that makes quantum dimensional reduction useful.

We can now consider increasing the memory available to the reduced quantum model much more slowly than the precision of the target process. For example, choose

$$n(k) = \lceil \sqrt{k} \rceil, \quad \tilde{D}(k) = 2^{n(k)}. \quad (\text{G14})$$

The reduced quantum model then uses an asymptotically vanishing fraction of the memory qubits required to represent the k -bit target process exactly,

$$\frac{n(k)}{k} \rightarrow 0 \quad \text{as} \quad k \rightarrow \infty. \quad (\text{G15})$$

At the same time, Eq. (G12) gives

$$R_F(|P^{(k)}\rangle, |\tilde{P}^{(k)}\rangle) \leq \mathcal{O}\left(\frac{C_*}{\sqrt{k}}\right) \rightarrow 0. \quad (\text{G16})$$

Thus, the additional distortion introduced by quantum memory reduction can vanish even while the number of memory qubits used by the reduced model grows parametrically more slowly than the precision k of the target process.

Meanwhile, increasing k also makes the target process $P^{(k)}$ an increasingly fine discretization of the underlying continuous random walk [28]. We can therefore simultaneously reduce both sources of approximation: the discretization

distortion between the underlying continuous process and $P^{(k)}$, and the memory-reduction distortion between $P^{(k)}$ and $\tilde{P}^{(k)}$. Importantly, this can be achieved while using far fewer than k qubits of quantum memory. Thus, finer discretization does not force the distortion introduced by quantum memory reduction to increase. Together with our numerical results for both the $k = 3$ and $k = 4$ target processes, where the reduced quantum models outperform classical baseline models subject to the same memory constraints, this provides strong evidence that the quantum advantage observed here is not specific to the particular discretization precision used in our original proof-of-principle example.

Appendix H: Additional examples

1. Autoregressive Model

The autoregressive (AR) model is a simple model frequently used to represent time series processes, including security prices in finance [59]. Consider the autoregressive model with order 2, AR(2), where the output x_t is time t obeys

$$X_t = \kappa_1 X_{t-1} + \kappa_2 X_{t-2} + \xi_t, \quad (\text{H1})$$

$\kappa_1, \kappa_2 \in \mathbb{R}$ are the set of weights on the previous time steps, and ξ_t is the white noise at time t . This gives a process with Markov order 2. Since the original AR(2) model takes values in $\mathbb{R}$, we first discretise it such that $X_t \in \{0, 1\}$. This generates the set of transition matrices, which are then used to construct the quantum model. The ϵ -machine for the discretised process is shown in Figure H4, and the discretisation procedure is provided in Section I.

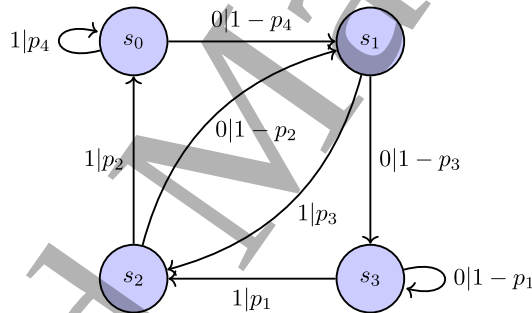


FIG. H4: ϵ -machine of the discretised AR(2) process. The value of $p_i, i \in \{1, 2, 3, 4\}$ are parameterised by κ_1 and κ_2 . The exact relationships are shown in Section I.

We utilised our algorithm to estimate the probability of a specific event in this process. The parameters κ_1 and κ_2 are set to 0.5 and 0.4, respectively. Starting from state s_0 , the event chosen is $x_{0:4} = 1001$. with probability $P(x_{0:4} = 1001|s_0) \approx 0.00842$. The results are displayed in Figure H5, where a quantum speedup over the classical algorithm is observed.

Furthermore, we apply the method in [10] to reduce the dimension of the discretised AR(2) model from four states to three states. This creates an approximate model that will no longer be able to capture the full statistics of the process. We compare both the effect of truncation on the accuracy of probability estimation for both the quantum and classical models. The truncated quantum model was used to estimate the probability of events with probability less than 10^{-2} , starting from a past $x_{-3:0} = 000$. This corresponds to 3 different events: $x_{0:4} = \{1001, 1100, 1101\}$. For the classical comparison, a search through all possible classical three-state machine is conducted as per [48], and a classical bound is computed. The results (Figure H5c) show the average estimation error of these 3 events compared to the average classical bound. The error of the quantum estimate falls below the average classical limit even with just $N \sim 10^2$ samples, highlighting the benefit of the quantum dimension reduction.

2. Perturbed Coin

The classical ϵ -machine for the perturbed coin process is depicted in Figure H6.

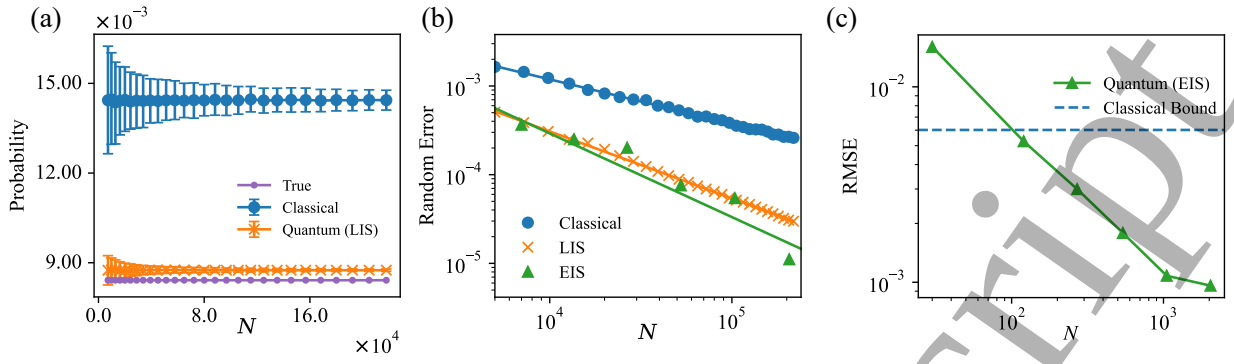


FIG. H5: (a) Trend of the convergence of the mean estimate of the algorithm over 1000 independent trials. The convergence of the classical algorithm is plotted for comparison, with the error bar corresponding to the 95th and 5th percentile. (b) Error trend of the different schedules as N increases. The numerical values of the fitting parameters and the corresponding correlation coefficients r^2 are given in Table I in the Appendix. (c) Average error for estimating the probability of three events $x_{0:4} = \{1001, 1100, 1101\}$, conditioned on $x_{-3:-1} = 000$ for the dimension reduced 3-state quantum model for different number of samples, compared with the classical bound for a classical dimension reduced 3-state machine.

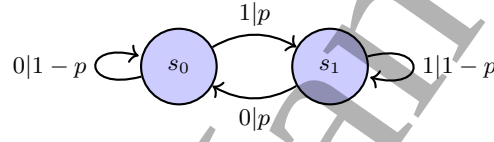


FIG. H6: ϵ -machine of the perturbed coin process. The process has two causal states, and it is characterised by the transition probability p between each causal state.

The algorithm is utilized to estimate the future sequence $x_{0:4} = 0000$ from the initial state s_0 . With the transition probability set at $p = 0.9$, this results in a theoretical probability of $P(0000|s_0) = 0.0001$. The convergence of the error is shown in Figure H7.

3. Nemo Process

The Nemo process comprises three causal states, necessitating two qubits to store the memory in the quantum model. The first causal state, s_0 , has a probability p of emitting a 0 and remaining in the same state, and a probability of $1 - p$ of emitting a 1 and transitioning to the second causal state, s_1 . The state s_1 transitions to s_2 with a probability of 1, emitting a 1 in the process. Lastly, from s_2 , there is an equal probability of emitting a 1 or 0 before the system transitions back to s_0 . The ϵ -machine is depicted in Figure H8.

The algorithm was applied to estimate the probability of the future sequence $x_{0:4} = 0000$ in the Nemo process, with the quantum model initialized in the quantum causal state s_0 . With the parameter $p = 0.1$, the true probability of this future sequence is $P(x_{0:4} = 0000|s_0) = 0.0001$. The results are presented in Figure H9.

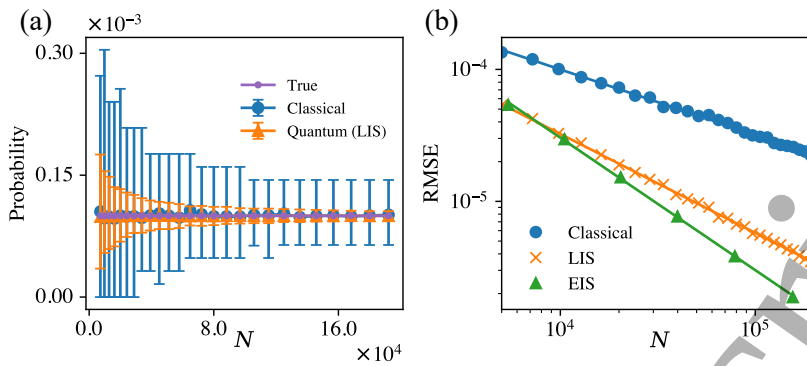


FIG. H7: (a) Trend of convergence for the mean estimate of the algorithm over 1000 samples for the perturbed coin process, with the transition probability $p = 0.9$ and the initial state set to s_0 . For comparison, the convergence of the classical algorithm is plotted. The error bars represent the 95th and 5th percentiles of probability estimates in the sample of 1000. (b) Error trend of different schedules as the number of shots increases.

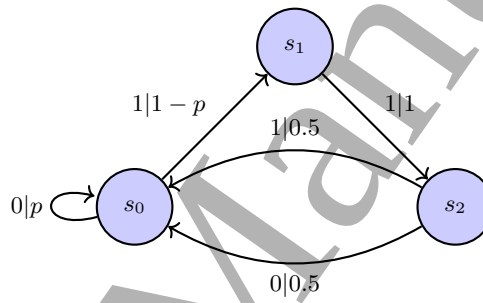


FIG. H8: ϵ -machine of the Nemo process. The single variable in this process is p , which determines the probability of the system staying in the state s_0 .

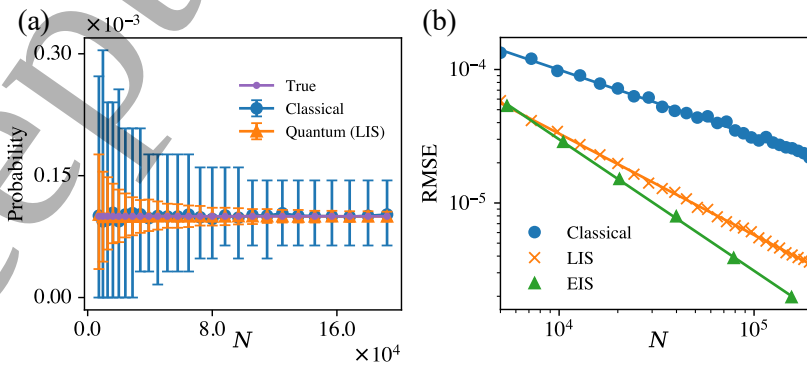


FIG. H9: (a) Convergence trend of the mean estimate of the algorithm over 1000 samples. For comparison, the convergence of the classical algorithm is plotted. The error bar represents the 95th and 5th percentiles of the probability estimates across the 1000 samples. (b) Error trend of different schedules as the number of shots increases.

4. Dual Poisson process

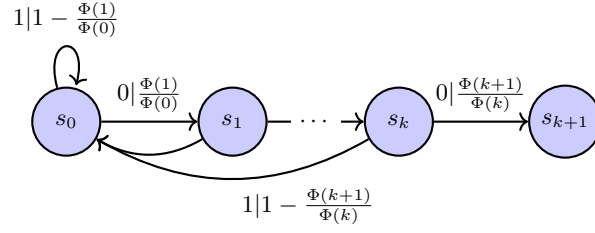


FIG. H10: ϵ -machine of a dual Poisson process. The system begins in state s_0 and transitions to a new state whenever a 0 is emitted, but it resets to s_0 when a 1 is emitted.

The dual Poisson process [20] is a highly non-Markovian process that is parameterised by $q, r_1, r_2, 0 \leq q, r_1, r_2 \leq 1$, which defines the so-called *survival probability* $\Phi(k)$ as follows:

$$\Phi(k) = qr_1^k + (1-q)r_2^k \quad (\text{H2})$$

The process's ϵ -machine is depicted in Figure H10. It was shown in [20] that the ϵ -machine to generate a contiguous string of k 0s requires an infinite number of causal states as k goes to infinity. This contrasts sharply with the corresponding quantum model, which only requires one qubit to simulate this process [20].

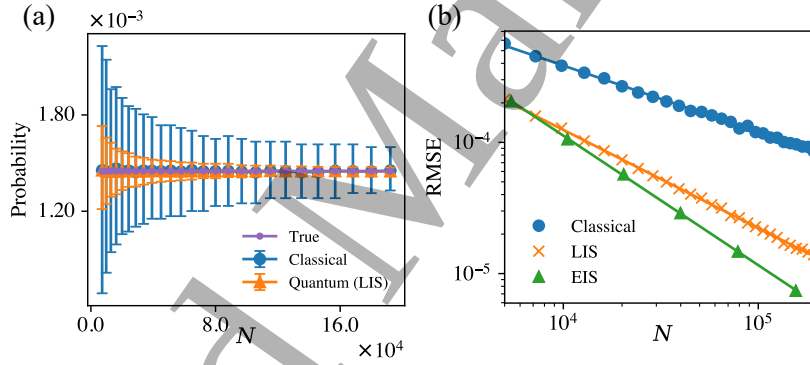


FIG. H11: (a) Convergence trend of the mean estimate of the algorithm over 1000 samples. For comparison, the convergence of the classical algorithm is plotted. The error bar represents the 95th and 5th percentiles of the probability estimates across the 1000 samples. (b) Error trend of different schedules as the number of shots increases.

We employ our quantum probability estimation algorithm to estimate the probability of the sequence $x_{0:4} = 0000$, from the initial state s_0 . With parameters $q = 0.1, r_1 = 0.1$ and $r_2 = 0.2$, the theoretical probability is $P(x_{0:4} = 0000|s_0) = 0.00145$. The findings are shown in Figure H11. The quantum algorithm exhibits a speedup over the classical method, with the quantum algorithm's standard error decreasing faster with increasing Grover iterations than with the corresponding number of shots in the classical algorithm. The observed speedup factors align with the predicted error scaling.

5. Table of numerical results

Schedule	Classical		LIS		EIS	
Process	a	r^2	a	r^2	a	r^2
Perturbed Coin	-0.4946 ± 0.0055	0.9969	-0.7498 ± 0.0042	0.9992	-1.004 ± 0.011	0.9995
Nemo Process	-0.4962 ± 0.0067	0.9955	-0.7591 ± 0.0044	0.9991	-0.9847 ± 0.0097	0.9996
Dual Poisson	-0.5066 ± 0.0043	0.9982	-0.7489 ± 0.0032	0.9996	-0.9871 ± 0.0057	0.9999
AR(2) (exact)	-0.4906 ± 0.0053	0.9970	-0.7476 ± 0.0040	0.9993	-0.875 ± 0.064	0.9792
AR(2) (truncated)	-0.4978 ± 0.0034	0.9987	-0.7553 ± 0.0032	0.9995	-0.944 ± 0.077	0.9560
Cyclic Random Walk (exact)	-0.5037 ± 0.0045	0.9979	-0.7529 ± 0.0034	0.9995	-0.995 ± 0.014	0.9994
Cyclic Random Walk (truncated)	-0.5024 ± 0.0038	0.9985	-0.7682 ± 0.0086	0.9967	-1.085 ± 0.076	0.9762
Average	0.4988		0.7545		0.982	

TABLE I: Table of the index of the best-fit error scaling for all the processes, with the corresponding r^2 correlation value to indicate fit. The exact reference for AR(2) and cyclic random walk are the discretized version of the continuous process (the 4-state machine for AR(2) and 8-state machine for cyclic random walk). Here, the errors denote the estimated standard deviation of the parameter estimate, computed using the function `curve_fit` from the `scipy.optimize` package in Python.

Appendix I: Discretization of AR(2) process

The AR(2) process over real values is defined as:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \epsilon_t, \quad (I1)$$

where $\phi_1, \phi_2 \in \mathbb{R}$ and

$$\epsilon_t \sim N(0, \sigma_\epsilon^2) \quad (I2)$$

for some $\sigma_\epsilon^2 > 0$. The output space is discretized into two different output symbols:

$$X_t^{(D)} = \begin{cases} -1 & \text{if } X_t < 0 \\ 1 & \text{if } X_t \geq 0, \end{cases} \quad (I3)$$

Since the process has a Markov order of 2, the full set of states is given as $\{(-1)(-1), (-1)(1), (1)(-1), (1)(1)\}$. The output symbols from each of these states can be computed exactly:

$$\begin{aligned} (-1)(-1) : X_t &= \phi_1 \cdot -1 + \phi_2 \cdot -1 + \epsilon_t \\ &= \epsilon_t - \phi_1 - \phi_2 \end{aligned} \quad (I4)$$

$$\begin{aligned} (-1)(1) : X_t &= \phi_1 \cdot 1 + \phi_2 \cdot -1 + \epsilon_t \\ &= \epsilon_t + \phi_1 - \phi_2 \end{aligned} \quad (I5)$$

$$\begin{aligned} (1)(-1) : X_t &= \phi_1 \cdot -1 + \phi_2 \cdot 1 + \epsilon_t \\ &= \epsilon_t - \phi_1 + \phi_2 \end{aligned} \quad (I6)$$

$$(1)(1) : X_t = \epsilon_t + \phi_1 + \phi_2, \quad (I7)$$

We define:

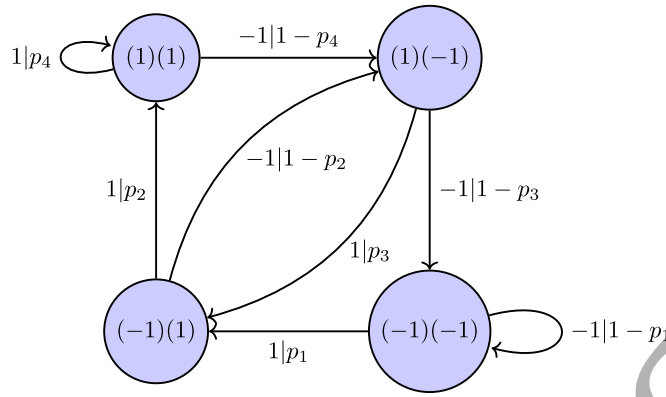
$$p_1 = P(\epsilon_t > \phi_1 + \phi_2) \quad (I8)$$

$$p_2 = P(\epsilon_t > -\phi_1 + \phi_2) \quad (I9)$$

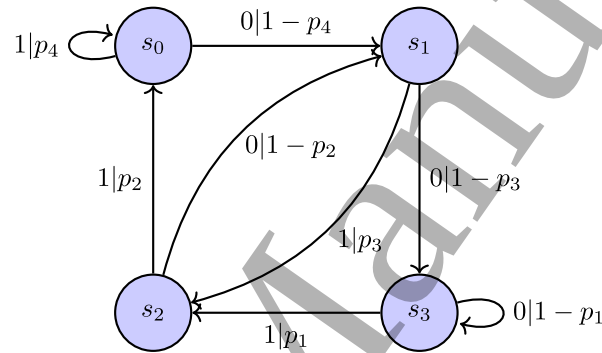
$$p_3 = P(\epsilon_t > \phi_1 - \phi_2) \quad (I10)$$

$$p_4 = P(\epsilon_t > -\phi_1 - \phi_2), \quad (I11)$$

With this, the definition of the discretization process defined in equation (I3) and the dynamics of the undiscretized process defined by (I4) – (I7) gives the following hidden Markov model:



We can then rewrite each state with the causal state notation s_i , for $i = 0, 1, 2, 3$, and map the output values from $\{-1, 1\}$ to $\{0, 1\}$ for ease of coding:



Thus, we have the following transition matrices

$$T^0 = \begin{bmatrix} 0 & 1-p_4 & 0 & 0 \\ 0 & 0 & 0 & 1-p_3 \\ 0 & 1-p_2 & 0 & 0 \\ 0 & 0 & 0 & 1-p_1 \end{bmatrix}, \quad T^1 = \begin{bmatrix} p_4 & 0 & 0 & 0 \\ 0 & 0 & p_3 & 0 \\ p_2 & 0 & 0 & 0 \\ 0 & 0 & p_1 & 0 \end{bmatrix} \quad (\text{I12})$$

where for $i, j \in \{0, 1, 2, 3\}, k \in \{0, 1\}$, T_{ij}^k corresponds to $P(x_t = k, s_t = s_j | s_{t-1} = s_i)$.