Thesis for The Degree of Licentiate of Engineering

Generative Models for Quantum Simulation and System Inference

Hang Zou

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2026

Generative Models for Quantum Simulation and System Inference

Hang Zou

© Hang Zou, 2026
except where otherwise stated.
All rights reserved.

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Data Science and AI
AI and Machine Learning in the Natural Sciences Lab
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Cover: The image brings together artificial intelligence, quantum computing, and the simulation of matter. A landscape beneath a neural network and a quantum circuit evokes loss landscapes in neural network training and variational quantum computing, potential energy surfaces, and probability distributions. An ethylene molecule and a two-dimensional spin lattice illustrate quantum simulation in quantum chemistry and many-body physics.

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2026.

To my family
献给我的家人

Generative Models for Quantum Simulation and System Inference

Hang Zou

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

Quantum computers promise to simulate nature beyond the reach of classical machines, yet today's devices remain limited by noise-induced errors in computation. Realizing the promise of quantum advantage demands both efficient algorithms and better-understood devices. On the algorithmic side, variational quantum algorithms offer a route to practical applications on near-term quantum hardware, but face an optimization bottleneck. On the hardware side, validating and improving devices requires characterizing their noise and other properties, which becomes demanding as systems grow.

This thesis addresses both challenges with conditional generative models, which learn probability distributions over parameters conditioned on the problem. In the forward problem of ground-state simulation, the model learns to generate circuit parameters across a family of related Hamiltonians. In the inverse problem of system characterization, it learns the posterior distribution of system parameters from data produced by polynomial-cost classical simulators. In both settings, a model trained once is reused across problem instances, so the cost of optimization and inference is amortized rather than paid anew.

The first paper presents Flow-VQE, which transfers knowledge between related molecular problems and reduces the number of circuit evaluations that variational algorithms require. The second paper introduces simulation-based quantum system inference, which unifies otherwise tailor-made characterization tasks in a single framework and paves the way toward scalable characterization of large quantum devices. These results point toward generative models as reusable tools for both quantum simulation and system inference, with the potential to accelerate the arrival of practically useful quantum computing.

Keywords

Quantum computing, Generative models, Normalizing flows, Variational quantum eigensolver, Simulation-based inference, Neural posterior estimation, Quantum parameter estimation, Quantum noise, Quantum circuits

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] **H. Zou**, M. Rahm, A. F. Kockum, and S. Olsson, Generative flow-based warm start of the variational quantum eigensolver, npj Quantum Information **12**, 5 (2025). doi: 10.1038/s41534-025-01159-x.

[**Paper II**] **H. Zou**, A. F. Kockum, M. Rahm, and S. Olsson, Simulation-based quantum system inference with neural posterior estimation, arXiv:2609.34995 (2026). Submitted, under review.

Other publications

The following publications were published during my PhD studies, or are currently in submission/under revision. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not related to the thesis.

[**a**] **H. Zou**, E. Magnusson, H. Brunander, W. Dobrautz, and M. Rahm, Multireference error mitigation for quantum computation of chemistry, Digital Discovery **4**, 2521–2533 (2025). doi: 10.1039/d5dd00202h.

[**b**] L. Boudin, I. Chollet, M.-S. Dupuy, C. Guichard, M. F. Serret, J. Tomasi, F. Vergnet, and **H. Zou**, Trotterization scheme and Pauli decomposition for the two-body Schrödinger equation, hal-05623753 (2026). Submitted, under review.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, Simon Olsson, for his guidance, support, and encouragement throughout this work. I am also deeply grateful to my co-supervisors, Martin Rahm, who also supervised my master's thesis, and Anton Frisk Kockum. I have benefited greatly from your combined insights, which have given me a broad perspective spanning AI, quantum computing, and quantum chemistry. I also thank my examiner, Devdatt Dubhashi, and Jakob Kottmann for kindly agreeing to serve as the discussion leader of this thesis.

I am grateful to my co-authors for their collaboration and support. In particular, I thank Werner for his considerable effort on the first paper of my academic career and Erika for presenting it on multiple occasions. I also enjoyed learning from Joachim, Fabrice, Laurent, and Mi-Song how mathematicians approach quantum computing, even though it took me outside my comfort zone.

I thank Rocío for her dedication in organizing many academic activities for us, and Laura for organizing the quantum algorithm meetings. I am grateful to all current and former members and friends of the AIMLeNS and AIME labs, as well as the DSAI and AQP divisions. I thank Yaochen, Selma, Flemming, Chenxiao, Mengyu, Stefano, Ross, Juan, Chris, Philipp, Kevin, Giannis, Varun, Richard, Télio, Guillaume, Marc, Tangyou, Akshay, Guangliang, and all other excellent colleagues for the discussions, help, and many lunches together.

Last, but certainly not least, I thank my friends and family, especially my parents and sister, for their unconditional support. And to my father, whose pride I carried with me every step of the way: I wish I could tell you where the journey has led me.

This work was generously supported by a joint project between the Wallenberg AI, Autonomous Systems and Software Program (WASP) and the Wallenberg Centre for Quantum Technology (WACQT), funded by the Knut and Alice Wallenberg Foundation. Computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) on Alvis and Arrhenius (projects NAISS 2025/22-480 and NAISS 2026/4-684).

Hang Zou
Gothenburg, November 2026

Contents

Abstract iii

List of Publications v

Acknowledgement vii

I Introductory Chapters 1

1 Introduction 3

2 Background 7
2.1 Quantum systems and noisy circuits 7
2.1.1 Quantum states and observables 7
2.1.2 Hamiltonians 9
2.1.3 Quantum circuits 11
2.1.4 Quantum channels 13
2.1.5 Noise models 15
2.2 Classical simulation of quantum circuits 16
2.2.1 Simulation hierarchy 17
2.2.2 Stabilizer simulation 17
2.2.3 Tensor networks 18
2.2.4 Pauli propagation 20
2.2.5 Simulators as statistical models 21
2.3 Generative modeling with normalizing flows 21
2.3.1 Generative models 22
2.3.2 Normalizing flows 23
2.4 Simulation-based inference 25
2.4.1 Approximate Bayesian computation and beyond 26
2.4.2 Neural posterior estimation 26

3 Summary of Included Papers 29
3.1 Generative flow-based warm start of the variational quantum eigensolver 29
3.2 Simulation-based quantum system inference with neural posterior estimation 30

4 Discussion and Future Work **33**

Use of Aids **35**

Bibliography **37**

II Appended Papers 51

Paper I - Generative flow-based warm start of the variational quantum eigensolver

Paper II - Simulation-based quantum system inference with neural posterior estimation

Part I

Introductory Chapters

Chapter 1

Introduction

The equations governing a physical system may fit on a T-shirt, yet working out their consequences can demand astronomical computational resources. This gap between a compact physical description and the computational effort required to explore it brings physics into direct contact with information and computation. A computer is itself a physical system, and computation is a physical process taking place within it. Information is encoded in physical states, and computational operations are implemented as transformations of those states. The laws governing these transformations therefore constrain which models of computation can be realized and what resources their operations require [1–3].

Classical computers encode *bits* in reliably distinguishable physical states, such as low and high voltage levels in digital electronic circuits, and perform logical operations on bit strings. In quantum computation, information is encoded in *qubits*, whose states can be *coherent superpositions* of the computational basis states $|0\rangle$ and $|1\rangle$. A pure state of n qubits is described by 2^n complex amplitudes, whose magnitudes and relative phases can both carry information. Such a state is *entangled* if it cannot be written as a tensor product of single-qubit states, in which case information can be encoded in correlations among the qubits. Coherence and entanglement are resources for quantum information processing [4, 5], and extending coherent control to increasingly complex entangled systems is central to exploring the "entanglement frontier" [6]. Quantum algorithms exploit these resources through controlled quantum evolution and use measurements to obtain classical outputs. This raises the question of which problems admit a quantum speedup.

With respect to computability, the standard classical and quantum models are equivalent [2, 7], so their comparison turns on the asymptotic resources required to solve a given problem [8]. For decision problems, the classes BPP and BQP consist of the languages decidable in polynomial time with bounded error by probabilistic classical and quantum algorithms, respectively. The inclusion $\mathsf{BPP} \subseteq \mathsf{BQP}$ is known, but whether it is strict remains open [8]. Shor's algorithm for integer factorization is a landmark example of a superpolynomial speedup over the best known classical algorithms [9]. Hamiltonian simulation [10], quantum linear-system solving [11], and low-rank singular-value

decomposition [12] offer potential exponential speedups, which depend on the problem structure, the input access model, and the required output [13]. The quantum Fourier transform [14], phase estimation [15], and quantum singular-value transformation [16] are common primitives in the construction of such algorithms. In addition, many quantum algorithms achieve polynomial query speedups, such as Grover's search algorithm [17], amplitude estimation [18], and the quantum-walk algorithm for element distinctness [19]. For hybrid heuristic methods, such as variational quantum algorithms [20, 21] and sample-based quantum diagonalization [22], no general speedup guarantee has yet been established. Beyond algorithmic speedups, potential quantum advantages extend to learning, sensing, cryptography, communication, and memory efficiency [23].

Despite these opportunities, a substantial gap separates today's noisy intermediate-scale quantum (NISQ) devices from fault-tolerant application-scale quantum (FASQ) computers [24]. Closing this gap requires moving beyond quantum error mitigation (QEM) [25] to quantum error correction (QEC) [26], and then from protecting a single logical qubit to performing long computations reliably across many logical qubits. Every step along this path rests on the quality of the underlying physical qubits. QEC suppresses logical errors only when physical error rates lie below threshold [27–29], and decoding performs best when informed by an accurate model of the device noise [30, 31], including crosstalk [32], correlated errors [33], and drift [34]. As physical qubit counts enter the intermediate-scale regime, however, acquiring such detailed information becomes increasingly demanding. Scalable characterization, which is essential for calibrating controls and validating device behavior, therefore poses a significant challenge [35–37].

Demonstrating practically useful quantum algorithms is equally demanding. Such demonstrations must address problems of scientific or industrial interest, reach the required accuracy, and operate within the available hardware and computational resources. Their feasibility depends on the costs of state preparation, circuit execution, repeated measurements, and classical processing. Variational quantum algorithms, which combine parameterized quantum circuits with classical optimization, are a prominent approach to computation on near-term hardware. However, finding suitable circuit parameters can itself be costly. Highly expressive circuits can exhibit exponentially small gradients [38, 39], while structures that improve trainability may also enable efficient classical estimation of the loss function [40]. A promising strategy to navigate this tension is the *warm start*, which uses prior information to choose the initial state or the initial circuit parameters [41, 42]. By starting the optimization in a favorable region of the parameter landscape, a warm start can retain the expressivity of the circuit while reducing the number of optimization steps and measurements. Effective warm starts can therefore lower the cost of variational optimization and broaden the range of problems accessible within a given experimental budget.

Artificial intelligence (AI) is changing how people live and work, with advances in natural language processing [43], image and video generation [44, 45], and robotics [46]. These advances also extend to mathematics [47] and the natural sciences, including biology [48], materials science [49], astronomy [50],

statistical physics [51], and Earth system science [52]. In scientific applications, however, data are typically far scarcer than in language or image generation, and predictions call for more rigorous validation against physical laws and experiments. Nevertheless, AI models can learn from experimental and simulated data to accelerate the prediction of properties of complex systems [48, 52], the inference of unknown parameters from data [50, 53], and the design of molecules, materials, and experiments [49, 54]. Their ability to identify patterns and approximate complex relationships in high-dimensional data makes AI a promising tool for advancing NISQ computers and for addressing the challenges of scaling toward FASQ computation [35, 36, 55–57].

This thesis investigates how generative models, which learn probability distributions from data, can address the two challenges outlined above. We cast these challenges as forward and inverse problems in quantum computing. A forward problem computes physical properties of a specified quantum model by quantum simulation, whereas an inverse problem infers the parameters of the underlying quantum system from measurement data.

Paper I addresses a forward problem, ground-state simulation with the variational quantum eigensolver (VQE) [58], and targets the cost of variational optimization. We develop *Flow-VQE*, in which a conditional normalizing flow learns a distribution over high-quality circuit parameters. The flow is trained by preference-based feedback from measured energies without quantum gradients, and the learned distribution provides warm starts that accelerate optimization on related problem instances [42].

Paper II addresses inverse problems and targets the challenge of scalable characterization. We introduce *simulation-based quantum system inference*, which pairs polynomial-cost classical simulators with neural posterior estimation to learn posterior distributions over state, Hamiltonian, and noise parameters conditioned on measurement data. Trained once on simulated data, the estimator infers the posterior for new measurement records without retraining and quantifies uncertainty within the assumed model [59]. We further show how the inferred noise models support QEM, connecting system characterization to more accurate estimates of observables.

Chapter 2

Background

2.1 Quantum systems and noisy circuits

This section introduces the quantum-computing background underlying the appended papers. We first describe quantum states, observables, and Hamiltonians, together with the statistics of finite measurements, in Sections 2.1.1 and 2.1.2. We then introduce the quantum circuit model in Section 2.1.3, which prescribes how states are prepared and transformed on a digital quantum computer. Finally, we introduce quantum channels, with emphasis on Pauli channels, in Section 2.1.4, and use them to describe noise models for quantum circuits and outline quantum error mitigation in Section 2.1.5.

2.1.1 Quantum states and observables

Quantum states

Consider a register of n qubits with Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ of dimension 2^n. The state of the register is described by a density operator ρ acting on $\mathcal{H}$, that is, a positive semidefinite operator with unit trace,

$$\rho \succeq 0, \qquad \operatorname{Tr} \rho = 1. \tag{2.1}$$

A state is *pure* if $\rho = |\psi\rangle\langle\psi|$ for some unit vector $|\psi\rangle \in \mathcal{H}$. A *mixed* state is a convex combination of pure states,

$$\rho = \sum_k p_k \, |\psi_k\rangle\langle\psi_k| \,, \quad p_k \geq 0, \quad \sum_k p_k = 1, \tag{2.2}$$

that cannot be written as a single projector. Pure and mixed states are distinguished by the purity $\operatorname{Tr}(\rho^2)$, which equals one for pure states and is strictly smaller than one for mixed states. Mixed states describe classical uncertainty about the preparation as well as entanglement with an environment, and therefore provide the natural description of states prepared on noisy hardware.

Pauli operators

The single-qubit Pauli matrices are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.3}$$

An n-qubit Pauli string is a tensor product $P = \sigma_1 \otimes \cdots \otimes \sigma_n$ with $\sigma_i \in \{I, X, Y, Z\}$, and we denote the set of all 4^n such strings by $\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n}$. Pauli strings are Hermitian and unitary, square to the identity, and are orthogonal with respect to the Hilbert–Schmidt inner product,

$$\mathrm{Tr}(PQ) = 2^n \delta_{PQ}, \quad P, Q \in \mathcal{P}_n. \tag{2.4}$$

They therefore form an orthogonal basis for the space of operators on $\mathcal{H}$, and every state admits the expansion

$$\rho = \frac{1}{2^n} \sum_{P \in \mathcal{P}_n} \mathrm{Tr}(\rho P)\, P. \tag{2.5}$$

Any two Pauli strings either commute or anticommute. We write $PQ = (-1)^{\langle P,Q \rangle} QP$, where the symplectic inner product $\langle P, Q \rangle \in \{0, 1\}$ equals zero if P and Q commute and one otherwise. The weight $|P|$ of a Pauli string is the number of qubits on which it acts nontrivially, and an operator is called *k-local* if it is a linear combination of Pauli strings of weight at most k.

Observables and finite-shot measurements

An observable is a Hermitian operator O with spectral decomposition $O = \sum_k o_k \Pi_k$, where $\{\Pi_k\}$ are orthogonal projectors. A projective measurement of O on the state ρ yields the outcome o_k with probability $\Pr(o_k) = \mathrm{Tr}(\rho \Pi_k)$ by the Born rule, so that the expectation value is

$$\langle O \rangle_\rho = \mathrm{Tr}(\rho O). \tag{2.6}$$

In practice, expectation values are estimated from a finite number N_s of repeated preparations and measurements, referred to as *shots*. For a Pauli observable P, each shot returns ± 1, with the outcome $+1$ occurring with probability $\mathrm{Tr}[\rho\,(I + P)/2] = (1 + \langle P \rangle_\rho)/2$. If the shots are independent and each prepares the same state ρ, the number N_+ of $+1$ outcomes follows a binomial distribution,

$$N_+ \sim \mathrm{Bin}\Big(N_s, \tfrac{1+\langle P \rangle_\rho}{2}\Big). \tag{2.7}$$

The estimator $\widehat{\langle P \rangle} = 2N_+/N_s - 1$ is unbiased with variance $(1 - \langle P \rangle_\rho^2)/N_s$, so that estimating $\langle P \rangle_\rho$ to additive precision ϵ requires $N_s = \mathcal{O}(\epsilon^{-2})$ shots. A general observable $O = \sum_j c_j P_j$ is estimated by measuring its Pauli components individually or in mutually commuting groups. This statistical uncertainty, known as *shot noise*, is distinct from the hardware noise discussed in Section 2.1.5.

2.1.2 Hamiltonians

A quantum system is specified by its Hamiltonian H, a Hermitian operator that, for qubit systems, can be written as $H = \sum_j h_j P_j$ with real coefficients h_j. This subsection introduces two computational tasks associated with H, ground-state estimation and time evolution, followed by the two families of Hamiltonians considered in this thesis.

Ground-state estimation

By the Rayleigh–Ritz variational principle, the ground-state energy satisfies

$$E_0 = \min_{|\psi\rangle \in \mathcal{H},\, \||\psi\rangle\|=1} \langle\psi| H |\psi\rangle \,, \tag{2.8}$$

so E_0 is a lower bound on the energy of any normalized trial state. This principle underlies the variational quantum eigensolver discussed in Section 2.1.3. Determining the ground-state energy of a *k-local* Hamiltonian to inverse-polynomial precision is QMA-complete for $k \geq 2$ [60], so efficient algorithms are not expected for arbitrary instances. Many problems of interest in physics and chemistry, however, possess additional structure, such as geometric locality, symmetries, or an efficiently preparable state with non-negligible overlap with the ground state. For such structured instances, quantum computers are still expected to provide an advantage. Given such an initial state, for instance, quantum phase estimation can estimate the ground-state energy efficiently [15]. Therefore, the difficulty shifts to finding a suitable initial state [61, 62]. Whether these structures yield an exponential advantage in practical settings, for example in quantum chemistry, remains under active investigation [61].

Time evolution

A closed system evolves under the unitary $U(t) = e^{-iHt}$, where we set $\hbar = 1$. The time-dependent expectation value of an observable can be computed in the Schrödinger picture, in which the state evolves, or equivalently in the Heisenberg picture, in which the observable evolves,

$$\langle O(t)\rangle = \mathrm{Tr}\big[U(t)\rho\, U(t)^\dagger O\big] = \mathrm{Tr}\big[\rho\, U(t)^\dagger O\, U(t)\big] \,. \tag{2.9}$$

On digital quantum computers, $U(t)$ is implemented approximately by product formulas [10]. For $H = \sum_{j=1}^{J} H_j$, the first-order Trotter formula with r steps reads

$$U(t) \approx \left(\prod_{j=1}^{J} e^{-iH_j t/r} \right)^r \,, \tag{2.10}$$

with an error that scales as $\mathcal{O}(t^2/r)$ and is controlled by the commutators $[H_j, H_k]$ [63].

Molecular electronic structure

Within the Born–Oppenheimer approximation, the nuclei are treated as fixed point charges, and the electrons move in the resulting static potential. Expanding the electronic wavefunction in a finite basis of M spin orbitals, the Hamiltonian at fixed nuclear positions takes the second-quantized form [64]

$$H = \sum_{p,q} h_{pq}\, a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs}\, a_p^\dagger a_q^\dagger a_r a_s + E_{\mathrm{nuc}}, \tag{2.11}$$

where $a_p^\dagger$ and a_p are fermionic creation and annihilation operators, and h_{pq} and h_{pqrs} are one- and two-electron integrals. The first two terms constitute the electronic Hamiltonian, which accounts for the kinetic energy of the electrons, their attraction to the nuclei, and their mutual repulsion. The last term is the nuclear repulsion energy, a constant that shifts the spectrum without affecting the eigenstates but must be included when comparing total energies across molecular geometries. The number of spin orbitals grows with the size of the molecule and of the basis set, and the cost of simulating H grows accordingly.

The Jordan–Wigner transformation [65] maps fermionic operators to qubit operators,

$$a_p^\dagger = \left(\prod_{q<p} Z_q \right) \frac{X_p - iY_p}{2}, \quad a_p = \left(\prod_{q<p} Z_q \right) \frac{X_p + iY_p}{2}, \tag{2.12}$$

where $(X_p \mp iY_p)/2 = |1\rangle\langle 0|_p$ and $|0\rangle\langle 1|_p$ change the occupation of spin orbital p, encoded in the computational basis state of qubit p. The string $\prod_{q<p} Z_q$ acts on all preceding spin orbitals and contributes a factor -1 for each occupied one, which accounts for the fermionic exchange sign. The transformation preserves the canonical anticommutation relations $\{a_p, a_q^\dagger\} = \delta_{pq}$ and $\{a_p, a_q\} = 0$ and maps the occupation-number operator to $a_p^\dagger a_p = (I - Z_p)/2$. The molecular Hamiltonian thus becomes a qubit Hamiltonian on $n = M$ qubits, containing Pauli strings of weight up to $\mathcal{O}(M)$. Symmetries of the Hamiltonian, such as fermion-parity conservation within each spin sector, can be used to remove qubits [66]. In **Paper I**, we study the ground-state problem for molecular Hamiltonians of this form.

Transverse-field Ising models

The transverse-field Ising model (TFIM) is a paradigmatic model of quantum magnetism and a standard benchmark for quantum simulation. In one dimension, a chain of N spins with nearest-neighbor coupling j and transverse field h is described by

$$H_{\mathrm{1D}} = -j \sum_{i=1}^{N-1} Z_i Z_{i+1} + h \sum_{i=1}^{N} X_i. \tag{2.13}$$

Its time evolution can be simulated with the Trotter formula of Eq. (2.10), whose circuit implementation is given in Section 2.1.3. In **Paper II**, we use

Trotterized TFIM dynamics to evaluate quantum error mitigation based on inferred noise models.

In two dimensions, we consider the TFIM with general pairwise couplings [67],

$$H_{\mathrm{2D}} = \Omega \sum_i X_i + \Delta \sum_i Z_i + \sum_{i<j} J_{ij}\, Z_i Z_j, \quad J_{ij} = \frac{C_6}{|\mathbf{r}_i - \mathbf{r}_j|^6}, \tag{2.14}$$

which describes arrays of neutral atoms excited to Rydberg states. Each atom at position $\mathbf{r}_i$ encodes a qubit in its ground and Rydberg states, Ω and Δ are set by the Rabi frequency and the detuning of the driving laser, and J_{ij} is the van der Waals interaction with coefficient C_6. Because the couplings J_{ij} depend on the atomic positions, deviations of the atoms from their intended positions translate directly into errors in the realized Hamiltonian. In **Paper II**, we infer the in-plane displacements of atoms in simulated Rydberg arrays from local observables measured after short-time evolution under H_{2D}.

2.1.3 Quantum circuits

Quantum circuit model

In the circuit model of quantum computation, an n-qubit register is initialized in the reference state $|0\rangle^{\otimes n}$, transformed by a sequence of quantum gates, and finally measured [68]. Each gate is a unitary operator acting nontrivially on a small number of qubits, typically one or two. Gates acting on disjoint sets of qubits can be applied simultaneously and are grouped into layers. A circuit of depth L thus implements the unitary

$$U = U_L U_{L-1} \cdots U_1, \tag{2.15}$$

where each layer U_l is a tensor product of gates acting on disjoint qubits. A frequently used layout on hardware with nearest-neighbor connectivity is the *brickwall* layer, in which two-qubit gates act first on the qubit pairs $(1,2),(3,4),\ldots$ and then on the pairs $(2,3),(4,5),\ldots$, so that two consecutive sublayers couple every pair of neighboring qubits.

Quantum gates

The elementary single-qubit gates include the Pauli rotations

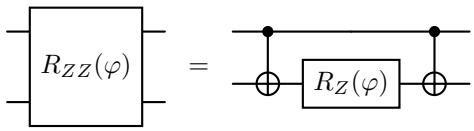
$$R_\sigma(\varphi) = e^{-i\varphi\sigma/2} = \cos\frac{\varphi}{2}\, I - i \sin\frac{\varphi}{2}\, \sigma, \quad \sigma \in \{X, Y, Z\}, \tag{2.16}$$

together with the Hadamard gate $\mathrm{H} = (X+Z)/\sqrt{2}$, the phase gate $S = \mathrm{diag}(1, i)$, and the gate $T = \mathrm{diag}(1, e^{i\pi/4})$. The standard two-qubit entangling gate is the controlled-NOT gate,

$$\mathrm{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X, \tag{2.17}$$

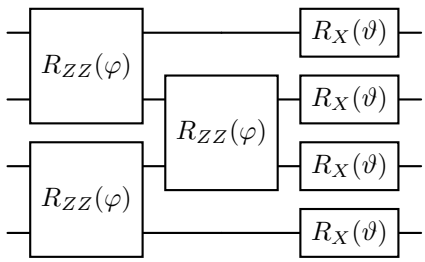
which flips the target qubit conditioned on the state of the control qubit. More generally, for any Pauli string $\sigma \in \mathcal{P}_n$, the multi-qubit Pauli rotation

$R_\sigma(\varphi) = e^{-i\varphi\sigma/2}$ can be compiled into single-qubit gates and CNOT gates [68]. Such rotations are the building blocks of the Trotter circuits of Eq. (2.10) and of many variational circuits. For example, the two-qubit Ising rotation $R_{ZZ}(\varphi) = e^{-i\varphi Z\otimes Z/2}$ is compiled into two CNOT gates and a single-qubit Z rotation,



(2.18)

For the one-dimensional TFIM of Eq. (2.13), each first-order Trotter step with step size δt then consists of a brickwall layer of Ising rotations followed by a layer of single-qubit rotations, with $\varphi = -2j\,\delta t$ and $\vartheta = 2h\,\delta t$, as shown for $N = 4$,



(2.19)

Clifford gates and universality

The Clifford group consists of the unitaries that map Pauli strings to Pauli strings under conjugation, $CPC^\dagger \in \pm\mathcal{P}_n$ for all $P \in \mathcal{P}_n$. It is generated by H, S, and CNOT, and contains the Pauli rotations in Eq. (2.16) with angles that are integer multiples of $\pi/2$. By the Gottesman–Knill theorem, circuits composed of Clifford gates acting on $|0\rangle^{\otimes n}$ and followed by Pauli measurements can be simulated efficiently on a classical computer [69, 70]. Adding a non-Clifford gate, such as T or a Pauli rotation with a generic angle, yields a universal gate set, with which any unitary can be approximated to arbitrary accuracy [68]. Non-Clifford operations are therefore the source of computational power beyond classical simulation, and their number and arrangement largely determine the cost of many of the simulation methods discussed in Section 2.2.

Non-Clifford operations are also the costly part of fault-tolerant quantum computation. No quantum error-correcting code admits a universal set of transversal gates, which act independently on the physical qubits of a code block and thus do not spread errors [71]. Non-Clifford gates are instead typically implemented by gate teleportation, which consumes ancillary resource states known as magic states [72, 73]. Magic state distillation [73, 74] and, more recently, magic state cultivation [75] are leading contributors to the resulting space-time overhead.

Parameterized quantum circuits

A parameterized quantum circuit (PQC) is a circuit whose gates depend on a vector of continuous parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d) \in \mathbb{R}^d$. A common form is

$$U(\boldsymbol{\theta}) = \prod_{j=1}^{d} W_j \, e^{-i\theta_j G_j/2}, \tag{2.20}$$

where the G_j are Hermitian generators, often Pauli strings, and the W_j are fixed gates. The circuit prepares the parameterized state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle^{\otimes n}$, and the choice of the circuit structure $U(\boldsymbol{\theta})$ is referred to as the *ansatz*. Ansätze range from hardware-efficient circuits, built from layers of native gates, to problem-inspired circuits whose structure reflects the physics of the target problem [20, 21]. PQCs provide a compact parameterization of a family of quantum states, which can serve, for example, as the candidate states in quantum state tomography [76], as in **Paper II**.

The ansatz determines the set of reachable states $\{|\psi(\boldsymbol{\theta})\rangle : \boldsymbol{\theta} \in \mathbb{R}^d\}$, and its *expressibility* characterizes how well these states cover the Hilbert space [77]. A common measure compares the distribution $P_{\mathrm{PQC}}(F)$ of fidelities $F = |\langle\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta}')\rangle|^2$ between states generated by independently sampled parameters with the corresponding distribution for Haar-random states,

$$\mathrm{Expr} = D_{\mathrm{KL}}(P_{\mathrm{PQC}}(F) \,\|\, P_{\mathrm{Haar}}(F)), \quad P_{\mathrm{Haar}}(F) = (N-1)(1-F)^{N-2}, \tag{2.21}$$

where $N = 2^n$, and $P_{\mathrm{Haar}}(F)$ is the exact fidelity distribution between two independent Haar-random pure states in an N-dimensional Hilbert space [78]. Smaller values of Expr indicate higher expressibility. Highly expressive ansätze can represent a broad range of states but tend to exhibit *barren plateaus*, in which the variance of the gradients of a loss function $\ell(\boldsymbol{\theta})$ vanishes exponentially with the number of qubits, $\mathrm{Var}_{\boldsymbol{\theta}}[\partial_k \ell(\boldsymbol{\theta})] \in \mathcal{O}(b^{-n})$ with $b > 1$ [38, 39].

The VQE uses a PQC to approximate the ground state of a Hamiltonian H [58]. By the variational principle, the energy $\ell(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\, H\, |\psi(\boldsymbol{\theta})\rangle$, which serves as the loss function, is bounded from below by the ground-state energy E_0. The energy is estimated on the quantum processor by measuring the Pauli terms of H, and a classical optimizer updates $\boldsymbol{\theta}$ to minimize it. Each evaluation of $\ell(\boldsymbol{\theta})$ therefore requires many circuit executions. Together with barren plateaus and a nonconvex landscape whose optimization is NP-hard in general [79], this measurement cost makes the choice of initial parameters critical. In **Paper I**, we propose a generative model that learns a conditional distribution over favorable circuit parameters and uses it to warm-start the VQE across related Hamiltonians, which may help alleviate these difficulties in practice.

2.1.4 Quantum channels

A general physical evolution of a quantum state is described by a quantum channel, a linear map $\mathcal{E}$ that is completely positive and trace preserving (CPTP).

Every channel admits a Kraus representation [68]

$$\mathcal{E}(\rho) = \sum_k K_k \rho K_k^\dagger, \quad \sum_k K_k^\dagger K_k = I. \tag{2.22}$$

Unitary evolution corresponds to the special case $\mathcal{U}(\rho) = U\rho U^\dagger$ with a single Kraus operator. The adjoint channel $\mathcal{E}^\dagger$ is defined by $\mathrm{Tr}[\mathcal{E}(\rho)O] = \mathrm{Tr}[\rho\,\mathcal{E}^\dagger(O)]$ for all ρ and O, and acts on observables as $\mathcal{E}^\dagger(O) = \sum_k K_k^\dagger O K_k$. This duality generalizes Eq. (2.9) to open-system dynamics and underlies the Heisenberg-picture simulation methods of Section 2.2.

Since Pauli strings form an operator basis, a channel can equivalently be represented by its Pauli transfer matrix (PTM) [80], the real $4^n \times 4^n$ matrix with entries

$$[R_\mathcal{E}]_{PQ} = \frac{1}{2^n}\mathrm{Tr}[P\,\mathcal{E}(Q)], \quad P, Q \in \mathcal{P}_n. \tag{2.23}$$

Composition of channels corresponds to multiplication of their PTMs, and the PTM of the adjoint channel is the transpose, $R_{\mathcal{E}^\dagger} = R_\mathcal{E}^\mathsf{T}$.

Pauli channels

A Pauli channel is a channel of the form

$$\Lambda(\rho) = \sum_{P\in\mathcal{P}_n} p_P\, P\rho P, \quad p_P \geq 0, \quad \sum_{P\in\mathcal{P}_n} p_P = 1, \tag{2.24}$$

which applies the Pauli error P with probability p_P. Pauli channels are self-adjoint and diagonal in the Pauli basis. Each Pauli string is an eigenoperator,

$$\Lambda^\dagger(Q) = \Lambda(Q) = f_Q\, Q, \quad f_Q = \sum_{P\in\mathcal{P}_n} (-1)^{\langle P,Q\rangle} p_P, \tag{2.25}$$

where the eigenvalues f_Q are called Pauli fidelities. The relation between error probabilities and fidelities is a Walsh–Hadamard transform and can be inverted as $p_P = 4^{-n}\sum_Q (-1)^{\langle P,Q\rangle} f_Q$ [81]. A Pauli channel therefore acts on an observable by rescaling each of its Pauli components by the corresponding fidelity.

Pauli channels are not merely a mathematically convenient special case. Pauli twirling conjugates a noise channel by uniformly random Pauli strings,

$$\Lambda^\mathrm{T}(\rho) = \frac{1}{4^n}\sum_{P\in\mathcal{P}_n} P\,\Lambda(P\rho P)\,P, \tag{2.26}$$

which maps an arbitrary channel Λ to a Pauli channel whose fidelities equal the diagonal PTM elements $[R_\Lambda]_{QQ}$ of Eq. (2.23). For Clifford gates such as the CNOT, the random Pauli frames can be propagated through the gate and absorbed into neighboring single-qubit layers. Randomized compiling [82] uses this construction to tailor the noise of a circuit into Pauli form while leaving its ideal action unchanged on average, converting coherent errors into stochastic Pauli errors.

2.1.5 Noise models

Noisy quantum circuits

We write $\mathcal{U}_l(\rho) = U_l \rho U_l^\dagger$ for the unitary channel of the l-th layer in Eq. (2.15). On noisy hardware, we model the implementation of each ideal layer $\mathcal{U}_l$ as $\mathcal{U}_l$ followed by a noise channel Λ_l, so that a circuit of depth L acting on the initial state ρ_0 realizes

$$\mathcal{E} = \Lambda_L \circ \mathcal{U}_L \circ \cdots \circ \Lambda_1 \circ \mathcal{U}_1, \quad \langle O \rangle_{\text{noisy}} = \mathrm{Tr}[O\, \mathcal{E}(\rho_0)]\,. \tag{2.27}$$

This description assumes Markovian noise, in which each Λ_l depends only on the layer it accompanies and not on the history of the computation. It is often further assumed that the noise is stationary, so that identical layers carry identical noise channels throughout the circuit and across repetitions of the experiment. Drift and non-Markovian effects violate these assumptions and require extended models.

Sparse noise models

A general n-qubit Pauli channel has $4^n - 1$ free parameters, so scalable characterization requires structural assumptions on the noise. A common assumption is locality, in which the noise accompanying a gate acts only on, or near, the qubits involved in that gate. A simple gate-local model assumes that each two-qubit gate acting on qubits k and $k+1$ is followed by a two-local Pauli channel

$$\Lambda_k(\rho) = \Big(1 - \sum_{P \in \mathcal{P}_2^*} p_{k,P}\Big)\rho + \sum_{P \in \mathcal{P}_2^*} p_{k,P}\, P\rho P, \tag{2.28}$$

where $\mathcal{P}_2^* = \{I, X, Y, Z\}^{\otimes 2} \setminus \{II\}$ contains the 15 nontrivial two-qubit Pauli strings, embedded on qubits k and $k+1$. For a brickwall layer of $n-1$ nearest-neighbor gates, the number of parameters grows linearly as $15(n-1)$. A closely related description is the sparse Pauli–Lindblad model [83], in which the noise channel is generated as $\Lambda = \exp(\mathcal{L})$ by the Lindbladian

$$\mathcal{L}(\rho) = \sum_{k \in \mathcal{K}} \lambda_k \left(P_k \rho P_k - \rho\right), \tag{2.29}$$

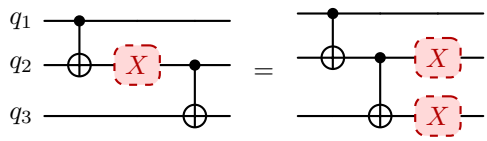
where $\mathcal{K}$ indexes a set of low-weight Pauli strings compatible with the device connectivity and $\lambda_k \geq 0$ are the model rates.

Gauge freedom and identifiability

Not every parameter of a noise model can be determined from measurement data. Distinct parameter configurations may produce identical measurement statistics for a given set of experiments, in which case the corresponding parameters are not identifiable from those data. In the local model of Eq. (2.28), each noisy gate $\Lambda_k \circ \mathcal{U}_k$ forms a single element of the gate set, and measurement outcomes depend only on how these elements compose along each circuit. Errors can

therefore be redistributed between different gate-set elements without changing any outcome.

As an example, consider CNOT_{12} followed by CNOT_{23}, where qubit 2 is the target of the first gate and the control of the second. An X error on qubit 2, produced by the noise channel Λ_1 of the first gate with probability $p_{1,IX}$, propagates through the second gate as $\mathrm{CNOT}_{23}\, X_2 = X_2 X_3\, \mathrm{CNOT}_{23}$, where errors are shown as dashed boxes,



(2.30)

It is thus indistinguishable from an X_2X_3 error produced by the noise channel Λ_2 of the second gate with probability $p_{2,XX}$, and the two probabilities affect the observations only through their combination. More generally, a *gauge* transformation inserts an invertible map and its inverse between consecutive gate-set elements. For the noisy gates $\mathcal{G}_k = \Lambda_k \circ \mathcal{U}_k$, the transformations that preserve the gate-local Pauli model are generated by invertible Pauli channels $\mathcal{D}$ on the shared qubits,

$$\tilde{\mathcal{G}}_1 = \mathcal{D} \circ \mathcal{G}_1, \qquad \tilde{\mathcal{G}}_2 = \mathcal{G}_2 \circ \mathcal{D}^{-1}, \tag{2.31}$$

so that $\tilde{\mathcal{G}}_2 \circ \tilde{\mathcal{G}}_1 = \mathcal{G}_2 \circ \mathcal{G}_1$. The noise parameters along these gauge directions are therefore not identifiable [84, 85]. In **Paper II**, the gauge parameters show broad posterior distributions, whereas the gauge-free parameters are inferred accurately.

Quantum error mitigation

The noise described above biases the expectation values away from their ideal values. QEM aims to reduce this bias by estimating the ideal expectation value from noisy executions, without the qubit overhead of QEC [25]. Most QEM methods combine noisy expectation values of a set of modified circuits through classical post-processing,

$$\langle O \rangle_{\mathrm{mit}} = g\big(\langle O \rangle^{(1)}_{\mathrm{noisy}}, \ldots, \langle O \rangle^{(m)}_{\mathrm{noisy}}\big), \tag{2.32}$$

and differ in how the circuits and the function g are constructed [25]. For instance, zero-noise extrapolation runs the circuit at amplified noise levels and extrapolates to zero noise [86, 87]. Probabilistic error cancellation samples circuits from a quasi-probability decomposition of the inverse noise channel, which requires a learned noise model [83, 86]. Learning-based methods instead obtain g on related circuits, typically (near-)Clifford variants of the target circuit whose ideal expectation values can be computed classically [88–90].

2.2 Classical simulation of quantum circuits

Exact simulation of quantum circuits stores 2^n amplitudes for a pure state or 4^n entries for a density matrix, so its cost grows exponentially with the number of

qubits and is limited to a few tens of qubits in practice. Approximate methods instead exploit structure in the circuit, the noise, or the entanglement of the simulated states or operators to reduce this cost [91, 92]. Which method is suitable also depends on the information required from the simulation, ranging from individual outcome probabilities to samples or expectation values.

We first introduce the hierarchy of simulation tasks in Section 2.2.1. We then review stabilizer simulation, tensor networks, and Pauli propagation in Sections 2.2.2–2.2.4. Finally, we discuss how such simulators define statistical models in Section 2.2.5, which connects them to the inference problems of Section 2.4.

2.2.1 Simulation hierarchy

Consider a quantum model specified by parameters $\boldsymbol{\theta}$, which may describe the circuit, its noise, or the underlying Hamiltonian. Simulation tasks for such a model can be ordered by the information they provide about its measurement outcomes b [93]. Strong simulation computes the probability $p(b|\boldsymbol{\theta})$ of any given outcome to high precision, including its marginals over subsets of qubits, and thus gives direct access to the likelihood of observed measurement outcomes. Weak simulation instead generates any number of independent samples $b \sim p(\cdot|\boldsymbol{\theta})$, or from a distribution close to it, so that the distribution is accessible only implicitly through its samples [94]. Weaker still is the estimation of expectation values $\langle O \rangle$ of bounded observables to additive precision ϵ. Such estimates generally do not suffice for sampling. The distribution of computational-basis outcomes is determined by the expectation values of all 2^n Pauli strings composed of I and Z, whereas estimation methods such as Pauli propagation, discussed below, typically access only a few of them.

A quantum computer naturally performs the two weaker tasks by repeatedly running and measuring the circuit. Estimating an outcome probability from N samples, however, only achieves an additive error of $\mathcal{O}(1/\sqrt{N})$, whereas for generic circuits the outcome probabilities are exponentially small in n. Computing them exactly or to multiplicative precision is #P-hard in general [95]. An efficient quantum algorithm for this task on arbitrary circuits would therefore allow quantum computers to solve every counting problem in #P, implying $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BQP}$. Since $\mathsf{P}^{\#\mathsf{P}}$, the class of problems solvable in polynomial time with access to an oracle for counting problems, contains the entire polynomial hierarchy [96], this inclusion would in particular imply $\mathsf{NP} \subseteq \mathsf{BQP}$. This inclusion is therefore widely believed *not* to hold, and there exist oracles relative to which $\mathsf{NP} \not\subseteq \mathsf{BQP}$ [97]. In contrast, the classical complexity of each task depends on the structure of the circuit. Some circuit families admit efficient classical weak simulation while their strong simulation is #P-hard [93], whereas for others, such as noiseless random circuits, even weak simulation is believed to be classically hard [98, 99].

2.2.2 Stabilizer simulation

Clifford circuits admit an efficient classical description through the *stabilizer* formalism [69, 70], which describes a state by the Pauli operators that leave

it invariant rather than by its 2^n amplitudes. A signed Pauli string $g \in \pm\mathcal{P}_n$ stabilizes $|\psi\rangle$ if $g\,|\psi\rangle = |\psi\rangle$. A stabilizer state is uniquely determined, up to a global phase, by n independent and mutually commuting stabilizer generators $g_1, \ldots, g_n$, which generate the group of all its stabilizers under multiplication. Each generator imposes a constraint $g_j\,|\psi\rangle = |\psi\rangle$ and halves the dimension of the subspace of states satisfying the constraints, so that n of them reduce the 2^n-dimensional Hilbert space to a single state. For example, $|0\rangle^{\otimes n}$ is stabilized by $Z_1, \ldots, Z_n$. Under a unitary U, the relation $(UgU^\dagger)\,U\,|\psi\rangle = U\,|\psi\rangle$ shows that the evolved state is stabilized by the conjugated operators $Ug_jU^\dagger$. For a Clifford gate these are again signed Pauli strings, so it suffices to update the n generators. For instance, a Hadamard gate on qubit 1 followed by CNOT_{12} maps the generators (Z_1, Z_2) of $|00\rangle$ to (X_1X_2, Z_1Z_2), which stabilize the Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$.

Storing each generator as $2n+1$ bits, the state requires only $\mathcal{O}(n^2)$ bits, and Clifford gates and Pauli measurements are simulated in polynomial time, with exact outcome probabilities [70, 100]. The stabilizer formalism therefore provides strong simulation of Clifford circuits, as stated by the Gottesman–Knill theorem. Since a Pauli error maps a stabilizer state to another stabilizer state, noisy Clifford circuits with Pauli noise remain efficiently simulable. Non-Clifford gates, in contrast, map Pauli strings to linear combinations of Pauli strings, such as $TXT^\dagger = (X+Y)/\sqrt{2}$, and thus leave the stabilizer formalism. Extensions that decompose non-stabilizer states into sums of stabilizer states remain applicable, but their cost grows exponentially with the number of non-Clifford gates [101].

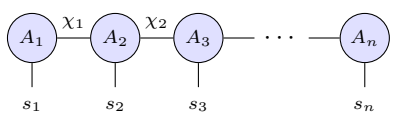
2.2.3 Tensor networks

Tensor-network methods compress the quantum state by exploiting limited entanglement [102, 103]. The most widely used example is the matrix product state (MPS), which represents an n-qubit pure state as

$$|\psi\rangle = \sum_{s_1,\ldots,s_n \in \{0,1\}} A_1^{s_1} A_2^{s_2} \cdots A_n^{s_n}\,|s_1 s_2 \cdots s_n\rangle\,, \tag{2.33}$$

where each $A_i^{s_i}$ is a $\chi_{i-1} \times \chi_i$ matrix with $\chi_0 = \chi_n = 1$. Equivalently, each A_i is a three-index tensor with one *physical* index s_i and two *virtual* indices of dimensions χ_{i-1} and χ_i, which are contracted between neighboring tensors. In graphical notation,

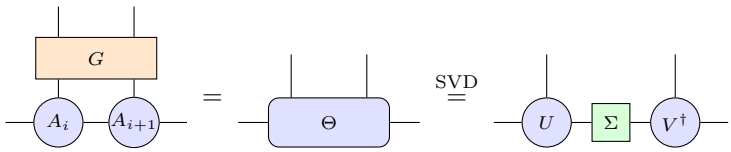
$|\psi\rangle$ =



(2.34)

The Schmidt rank across the cut between sites i and $i+1$ is at most χ_i, so the entanglement entropy across every cut is bounded by $\log \chi$, where $\chi = \max_i \chi_i$ is the maximal *bond dimension*. An MPS requires only $\mathcal{O}(n\chi^2)$ memory, in contrast to the 2^n amplitudes of the full state vector.

Applying a two-qubit gate G to neighboring sites i and $i+1$ proceeds in three steps. The two site tensors and the gate are contracted into a single tensor Θ, which is reshaped into a $2\chi_{i-1} \times 2\chi_{i+1}$ matrix and factorized by a singular value decomposition (SVD),

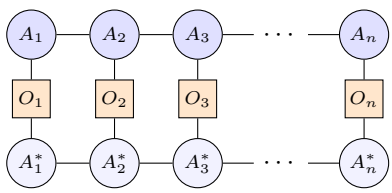


(2.35)

where Σ is the diagonal matrix of singular values. Up to 2χ of them can be nonzero, so the bond dimension may double. Retaining only the χ largest singular values restores it, and the new site tensors are U and $\Sigma V^\dagger$. In the mixed canonical form centered on sites i and $i+1$, the tensors A_k with $k < i$ are left isometries, $\sum_s A_k^{s\dagger} A_k^s = I$, and those with $k > i+1$ are right isometries, $\sum_s A_k^s A_k^{s\dagger} = I$. If the MPS is brought into this form before the gate is applied, the singular values $\sigma_1 \geq \sigma_2 \geq \cdots$ of Θ coincide with the Schmidt coefficients of $|\psi\rangle$ across the cut, and the truncated state $|\tilde{\psi}\rangle$, before renormalization, is the closest state to $|\psi\rangle$ with Schmidt rank at most χ across this cut, with error $\epsilon_{\text{trunc}} = \| \, |\psi\rangle - |\tilde{\psi}\rangle \, \|^2 = \sum_{\alpha > \chi} \sigma_\alpha^2$. The simulation is exact if no nonzero singular value is ever discarded.

Expectation values are obtained by contracting the MPS with the observable and the conjugate MPS. For an observable $O = \bigotimes_i O_i$ that factorizes over sites, such as a Pauli string,

$\langle\psi| O |\psi\rangle \;=$



(2.36)

Contracting this network column by column from left to right, each step absorbs one site and keeps an intermediate $\chi \times \chi$ matrix, so the total cost is $\mathcal{O}(n\chi^3)$. A single amplitude $\langle s_1 \cdots s_n | \psi \rangle$ is obtained in the same way by fixing all physical indices, and samples can be drawn sequentially site by site from the conditional marginal distributions. MPS therefore support both strong and weak simulation, which is efficient whenever the bond dimension required for the desired accuracy grows at most polynomially with n.

Ground states of gapped one-dimensional Hamiltonians are a standard example of states that admit efficient MPS approximations [104]. They obey an *area law*, meaning that the entanglement entropy across a bipartition scales with the size of the boundary between the two subsystems rather than with the volume of the smaller one [105]. Another example is the output of *shallow* one-dimensional circuits with nearest-neighbor two-qubit gates. Each layer contains at most one gate across any cut, which increases the Schmidt rank by at most a factor of four, so that a circuit of depth D yields $\chi \leq 4^D$. This bound is polynomial in n for $D = \mathcal{O}(\log n)$ [106]. Generic circuits whose depth grows

linearly with n, in contrast, generate volume-law entanglement and require a bond dimension exponential in n [107].

2.2.4 Pauli propagation

Pauli propagation evaluates expectation values in the Heisenberg picture [92, 108] and can be viewed as an extension of stabilizer simulation beyond Clifford circuits. Clifford gates map Pauli strings to Pauli strings without *branching*, whereas non-Clifford gates generate linear combinations of Pauli strings. For the noisy circuit of Eq. (2.27), the duality between a channel and its adjoint gives

$$\langle O\rangle_{\text{noisy}} = \mathrm{Tr}\big[\rho_0\,\mathcal{E}^\dagger(O)\big]\,, \quad \mathcal{E}^\dagger = \mathcal{U}_1^\dagger \circ \Lambda_1^\dagger \circ \cdots \circ \mathcal{U}_L^\dagger \circ \Lambda_L^\dagger. \tag{2.37}$$

The observable is expanded as $O = \sum_\alpha c_\alpha P_\alpha$, and the adjoint layers are applied while tracking the coefficients c_α of all Pauli strings in the expansion. By linearity, it suffices to specify how each operation acts on a single Pauli string P.

1. A Clifford gate C maps a Pauli string to a single Pauli string up to a sign, $C^\dagger P C = \pm P'$, so that the number of strings is preserved. Single-qubit Clifford gates preserve the weight $|P|$, whereas a two-qubit Clifford gate can change it by at most one, such as $\mathrm{CNOT}\,(X\otimes I)\,\mathrm{CNOT} = X\otimes X$.

2. A Pauli channel rescales the coefficient of each string by its fidelity, $\Lambda^\dagger(P) = f_P\,P$, according to Eq. (2.25).

3. A Pauli rotation $R_\sigma(\varphi)$ with $\sigma \in \mathcal{P}_n$ and a generic angle leaves P invariant if the two commute, and otherwise splits it into two strings,

$$R_\sigma(\varphi)^\dagger\,P\,R_\sigma(\varphi) = \begin{cases} P, & \langle\sigma, P\rangle = 0, \\ \cos\varphi\,P + \sin\varphi\,(i\sigma P), & \langle\sigma, P\rangle = 1. \end{cases} \tag{2.38}$$

After all layers have been applied, the expectation value follows from the overlaps of the propagated strings with the initial state, so Pauli propagation operates at the level of expectation-value estimation.

Only gates within the backward light cone of the observable contribute [Fig. 2.1(a)], and the cost of the method is governed by the number of Pauli strings generated by the branching in Eq. (2.38). Since this number can grow exponentially with the number of non-Clifford rotations, practical implementations truncate the expansion [Fig. 2.1(b)], discarding strings whose coefficients fall below a threshold, $|c_\alpha| < \epsilon$, or whose weight exceeds a cutoff, $|P_\alpha| > w_{\max}$. Noise justifies the weight truncation physically. Under local noise, the fidelity f_P of a string typically decreases exponentially with its weight, so contributions of high-weight strings to low-weight observables are strongly suppressed. For noisy circuits, this mechanism yields classical simulation algorithms with polynomial cost under suitable conditions [109–111].

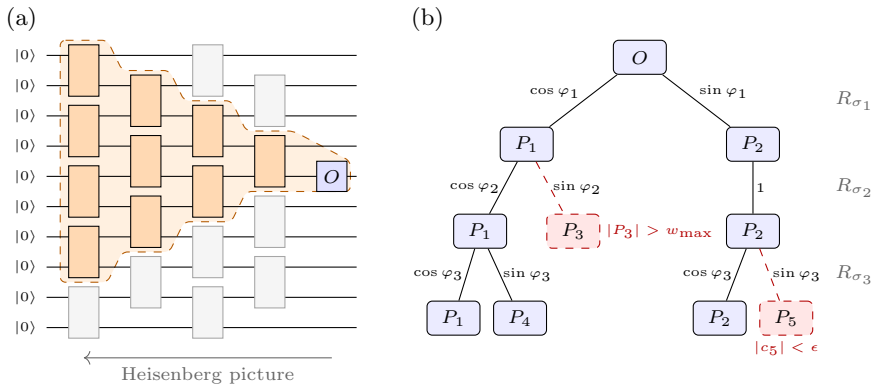


Figure 2.1: Pauli propagation. (a) Backward light cone of a single-qubit observable O in a one-dimensional brickwall circuit of nearest-neighbor two-qubit gates. Only the highlighted gates act nontrivially on O, and the support of the propagated Pauli strings grows by at most one qubit on each side per layer. (b) Branching and truncation. A Pauli rotation $R_{\sigma_l}(\varphi_l)$ splits each anticommuting string into two, following Eq. (2.38), and leaves commuting strings unchanged (edge labeled 1). Strings whose weight exceeds $w_{\max}$ or whose coefficient magnitude falls below ϵ are discarded (dashed red).

2.2.5 Simulators as statistical models

Stabilizer simulation is exact but restricted to circuits dominated by Clifford gates. Tensor networks and Pauli propagation apply to general circuits and trade accuracy for computational cost through truncation parameters, such as the bond dimension χ for MPS and the weight cutoff $w_{\max}$ for Pauli propagation. Despite their differences, such simulators provide statistical descriptions of the underlying physical processes by mapping system parameters to simulated observations. They can, in turn, be used to infer the unknown system parameters underlying observed data. The simulation capabilities required depend on the inference method: methods based on explicit likelihoods require strong simulation, whereas the likelihood-free methods of Section 2.4 require only samples of the observations. This ability to perform inference from simulated samples is central to **Paper II**, which employs all three types of simulators to generate training data for likelihood-free inference of quantum systems.

2.3 Generative modeling with normalizing flows

This section provides background on generative modeling relevant to the appended papers. We first describe generative models and their training by density estimation in Section 2.3.1. We then introduce normalizing flows, together with the architectures used in **Paper I** and **Paper II**, in Section 2.3.2.

2.3.1 Generative models

Let $p(\boldsymbol{x})$ be an unknown distribution on $\mathbb{R}^d$ from which only samples $\{\boldsymbol{x}_i\}_{i=1}^N$ are available. A generative model is a parameterized family of distributions $q_{\boldsymbol{\phi}}(\boldsymbol{x})$ that is fitted to approximate p. Two operations characterize such a model: generating new samples $\boldsymbol{x} \sim q_{\boldsymbol{\phi}}$ and evaluating the probability density $q_{\boldsymbol{\phi}}(\boldsymbol{x})$ at a given point. Generative models differ in how well they support each of these operations. Autoregressive models factorize the density by the chain rule, $q_{\boldsymbol{\phi}}(\boldsymbol{x}) = \prod_{i=1}^{d} q_{\boldsymbol{\phi}}(x_i|x_1, \ldots, x_{i-1})$, which gives exact densities but requires d sequential steps for sampling [112, 113]. Energy-based models specify the density in unnormalized form, $q_{\boldsymbol{\phi}}(\boldsymbol{x}) \propto \exp[-E_{\boldsymbol{\phi}}(\boldsymbol{x})]$, with a generally intractable partition function, and typically rely on Markov chain Monte Carlo (MCMC) for sampling [114]. Variational autoencoders allow straightforward sampling through latent variables but provide only a lower bound on the density [115]. Implicit models such as generative adversarial networks support only sampling [116]. Diffusion and flow-based models transform a simple base distribution into the data distribution, and the form of this transformation governs sampling and density evaluation, as discussed next.

Diffusion models and flows

Score-based diffusion models admit a continuous-time formulation in which a forward stochastic differential equation (SDE) progressively perturbs data with noise. Samples can be generated by simulating the corresponding reverse-time SDE, whose drift depends on the marginal score $\nabla_x \log p_t(x)$ estimated from data [117]. Rewriting the forward Fokker–Planck equation as a continuity equation yields the probability-flow ordinary differential equation (ODE), whose velocity field depends on the same score. When solved backward from the forward terminal distribution p_T using the exact score, the ODE and the reverse-time SDE both recover the forward marginal p_t at every time $t \in [0, T]$ [117]. Under suitable regularity conditions, the ODE defines a deterministic, invertible transport from p_T to the data distribution. The resulting continuous normalizing flow (CNF) has a model density that can be evaluated using the instantaneous change-of-variables formula [117, 118].

Flow matching [119], rectified flow [120], and stochastic interpolants [121] learn the velocity field of a CNF directly. They use conditional velocities associated with prescribed probability paths as regression targets, so the regression itself requires no ODE simulation. A single stochastic interpolant can give rise to both an ODE and a family of SDEs with tunable diffusion coefficients, all sharing the interpolant's one-time marginal distributions [122].

The normalizing flows used in this thesis are discrete-time flows composed of a finite sequence of invertible layers with tractable Jacobian determinants. Sampling proceeds by applying these layers in sequence to a sample from the base distribution, and model densities follow from the change-of-variables formula [123].

Maximum-likelihood training

When the density can be evaluated, the parameters are fitted by maximizing the average log-likelihood of the samples,

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \frac{1}{N} \sum_{i=1}^{N} \log q_{\boldsymbol{\phi}}(\boldsymbol{x}_i). \tag{2.39}$$

In the limit $N \to \infty$, this is equivalent to minimizing the forward Kullback–Leibler (KL) divergence

$$D_{\mathrm{KL}}(p \,\|\, q_{\boldsymbol{\phi}}) = \mathbb{E}_p[\log p(\boldsymbol{x}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{x})] = -\mathbb{E}_p[\log q_{\boldsymbol{\phi}}(\boldsymbol{x})] - H(p), \tag{2.40}$$

since the entropy $H(p)$ does not depend on $\boldsymbol{\phi}$. The forward KL divergence diverges if $q_{\boldsymbol{\phi}}$ vanishes on any part of the support of p, so that the fitted model tends to cover all regions of high probability under p. The reverse divergence $D_{\mathrm{KL}}(q_{\boldsymbol{\phi}} \,\|\, p)$ instead requires evaluating p, up to normalization, rather than sampling from it, and tends to concentrate $q_{\boldsymbol{\phi}}$ on a subset of the modes of p. It is used when the target is given as an unnormalized density, such as a Boltzmann distribution [51]. Both appended papers train normalizing-flow models by maximum likelihood, on variational circuit parameters in **Paper I** and on quantum-system parameters paired with simulated observations in **Paper II**.

2.3.2 Normalizing flows

A normalizing flow represents $q_{\boldsymbol{\phi}}$ as the transformation of a simple base density $q_0(\boldsymbol{z})$, typically a standard normal distribution, under a *diffeomorphism* $f_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^d$, so that $\boldsymbol{x} = f_{\boldsymbol{\phi}}(\boldsymbol{z})$ with $\boldsymbol{z} \sim q_0$ [123, 124]. The *change-of-variables* formula gives the density

$$q_{\boldsymbol{\phi}}(\boldsymbol{x}) = q_0\big(f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x})\big) \left| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|. \tag{2.41}$$

Sampling requires one evaluation of $f_{\boldsymbol{\phi}}$, and density evaluation requires one evaluation of $f_{\boldsymbol{\phi}}^{-1}$ together with its Jacobian determinant. In practice, the flow is composed of simpler layers, $f_{\boldsymbol{\phi}} = f_T \circ \cdots \circ f_1$, whose log-determinants add,

$$\log q_{\boldsymbol{\phi}}(\boldsymbol{x}) = \log q_0(\boldsymbol{z}_0) - \sum_{t=1}^{T} \log \left| \det \frac{\partial f_t(\boldsymbol{z}_{t-1})}{\partial \boldsymbol{z}_{t-1}} \right|, \tag{2.42}$$

where $\boldsymbol{z}_t = f_t(\boldsymbol{z}_{t-1})$ and $\boldsymbol{z}_T = \boldsymbol{x}$. Since a general Jacobian determinant costs $\mathcal{O}(d^3)$ operations, flow architectures are designed to be invertible with triangular or diagonal Jacobians, whose determinants can be computed in $\mathcal{O}(d)$ operations.

Coupling and autoregressive layers

The two most common layer types transform each coordinate by a monotone scalar function $g(\,\cdot\,;\boldsymbol{\psi})$ whose parameters $\boldsymbol{\psi}$ depend only on coordinates that

precede it in a fixed ordering. Written in the direction $\boldsymbol{x} \to \boldsymbol{z}$ used for density evaluation, an autoregressive layer conditions each coordinate on all preceding ones,

$$z_i = g\big(x_i;\, \boldsymbol{\psi}_i(x_1, \ldots, x_{i-1})\big), \quad i = 1, \ldots, d, \tag{2.43}$$

whereas a coupling layer splits the coordinates into two blocks $\boldsymbol{x} = (\boldsymbol{x}_A, \boldsymbol{x}_B)$, leaves the first unchanged, and transforms the second conditioned on the first [125],

$$\boldsymbol{z}_A = \boldsymbol{x}_A, \quad z_{B,i} = g\big(x_{B,i};\, \boldsymbol{\psi}_i(\boldsymbol{x}_A)\big). \tag{2.44}$$

In both cases, the Jacobian $\partial \boldsymbol{z}/\partial \boldsymbol{x}$ is therefore (block) lower triangular, with diagonal entries $\partial g(x_i; \boldsymbol{\psi}_i)/\partial x_i$ for the transformed coordinates and one for the unchanged ones. A coupling layer is a special case of an autoregressive layer with two blocks instead of d [126, 127], and the two differ mainly in their computational trade-offs. In a coupling layer, both density evaluation and sampling require a single network evaluation. In an autoregressive layer, all parameters $\boldsymbol{\psi}_i$ are produced in a single pass of a masked neural network [126, 128], so that density evaluation is parallel over coordinates, whereas sampling inverts Eq. (2.43) sequentially over the d coordinates. In return, an autoregressive layer is more flexible, because every coordinate is conditioned on others. For either layer type, the coordinates are permuted between consecutive layers so that every coordinate can depend on every other.

Monotone scalar transformations

The flexibility of both layer types is determined by the scalar function g. The simplest choice is affine, $g(x) = ax + b$ with $a > 0$, as in Real NVP and the masked autoregressive flow [125, 126]. Neural spline flows replace it by a monotone rational-quadratic spline, built from K segments on an interval $[-B, B]$ and extended by the identity outside it, whose bin widths, heights, and knot derivatives form the parameters $\boldsymbol{\psi}_i$ [127]. Such splines are considerably more flexible than affine maps, remain analytically invertible, and can be used in both coupling and autoregressive layers [127]. **Paper II** uses autoregressive neural spline flows.

Gaussianization flows

Gaussianization flows follow a different construction, in which no coordinate is conditioned on another within a layer [129]. They instead alternate learnable orthogonal rotations R_t, which mix the coordinates, with marginal Gaussianization layers Ψ_t, which act independently on each coordinate,

$$\begin{aligned} f_{\boldsymbol{\phi}}^{-1} &= \Psi_T \circ R_T \circ \cdots \circ \Psi_1 \circ R_1, \\ \Psi_t^{(k)}(x) &= \Phi^{-1}\Big(\frac{1}{J}\sum_{j=1}^{J} \Phi\big(e^{\alpha_{t,j}^{(k)}} x + \beta_{t,j}^{(k)}\big)\Big), \end{aligned} \tag{2.45}$$

where Φ is the standard normal *cumulative distribution function* (CDF) and $\alpha_{t,j}^{(k)}, \beta_{t,j}^{(k)}$ are trainable parameters of coordinate k. Each marginal layer

is a learned version of the probability integral transform. The mixture $F_t^{(k)}(x) = \frac{1}{J}\sum_j \Phi\big(e^{\alpha_{t,j}^{(k)}} x + \beta_{t,j}^{(k)}\big)$ is a monotone map onto $(0,1)$, with positive slopes ensured by the exponentials, and serves as a model of the true marginal CDF of coordinate k. If it matched the true marginal exactly, $F_t^{(k)}(x)$ would be uniformly distributed on $(0,1)$, and Φ^{-1} would map it to a standard normal variable. Each layer thus *Gaussianizes* every coordinate separately. The rotations are orthogonal matrices, $R_t^\mathsf{T} R_t = I$, which are inverted by transposition and preserve volume, $|\det R_t| = 1$. To keep them orthogonal during training, each R_t is parameterized, for example, as a product of Householder reflections $I - 2\boldsymbol{v}\boldsymbol{v}^\mathsf{T}/\|\boldsymbol{v}\|^2$ with trainable vectors $\boldsymbol{v}$ [129]. Each rotation mixes the coordinates so that the subsequent marginal layer acts along new directions, and all dependencies between coordinates are captured in this way. Since the Jacobian of Ψ_t is diagonal, the log-determinant is again a sum of scalar derivatives. **Paper I** uses this parameterization.

Conditional flows

A conditional flow models a family of densities $q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{c})$ indexed by a context vector $\boldsymbol{c}$, by letting the parameters of each layer depend on $\boldsymbol{c}$, for example $\boldsymbol{\psi}_i = \boldsymbol{\psi}_i(x_1, \ldots, x_{i-1}, \boldsymbol{c})$ in Eq. (2.43). For every fixed $\boldsymbol{c}$, Eq. (2.41) holds unchanged. High-dimensional contexts are often compressed by an embedding network trained jointly with the flow. Given pairs $\{(\boldsymbol{x}_i, \boldsymbol{c}_i)\}_{i=1}^N$ drawn from a joint distribution $p(\boldsymbol{x}, \boldsymbol{c})$, maximum-likelihood training minimizes

$$-\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{c})] = \mathbb{E}_{p(\boldsymbol{c})}\big[D_{\mathrm{KL}}\big(p(\cdot|\boldsymbol{c}) \,\|\, q_{\boldsymbol{\phi}}(\cdot|\boldsymbol{c})\big)\big] + \text{const}, \tag{2.46}$$

the forward KL divergence of Eq. (2.40) averaged over contexts. A single trained model thus covers all contexts, and a new context requires only a forward pass rather than retraining. This property is referred to as *amortization*, whose accuracy relies on the model generalizing to contexts not seen during training. In **Paper I**, the flow generates circuit parameters conditioned on Hamiltonian coefficients, and in **Paper II**, it generates system parameters conditioned on observations.

2.4 Simulation-based inference

Consider parameters $\boldsymbol{\theta}$ of a quantum system and an observation $\boldsymbol{x}$. Bayesian inference combines a prior $\pi(\boldsymbol{\theta})$ with the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ into the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{p(\boldsymbol{x})}, \quad p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})\,d\boldsymbol{\theta}, \tag{2.47}$$

from which point estimates, such as the posterior mean, and credible regions are derived. Evaluating Eq. (2.47) requires the likelihood. For many physical models, however, the likelihood is typically intractable to evaluate but is still defined implicitly by a simulator that generates observations from parameters, as discussed in Section 2.2.5.

Simulation-based inference (SBI) addresses this setting by inferring $\boldsymbol{\theta}$ from samples generated by the simulator instead of likelihood evaluations [53]. Neural SBI methods use generative models such as those of Section 2.3 as conditional density estimators, trained on samples from simulators such as those of Section 2.2. We first review traditional likelihood-free methods in Section 2.4.1. We then introduce neural posterior estimation, the approach used in **Paper II**, in Section 2.4.2.

2.4.1 Approximate Bayesian computation and beyond

Traditional SBI methods are based on Monte Carlo sampling. In its simplest form, approximate Bayesian computation (ABC) draws $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$, simulates $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$, and accepts $\boldsymbol{\theta}$ if

$$\rho\big(\boldsymbol{s}(\boldsymbol{x}), \boldsymbol{s}(\boldsymbol{x}_0)\big) \leq \epsilon, \tag{2.48}$$

where $\boldsymbol{x}_0$ is the measured observation, $\boldsymbol{s}$ denotes summary statistics, ρ a distance, and ϵ a tolerance [130]. The accepted parameters are samples from an approximate posterior that approaches $p(\boldsymbol{\theta}|\boldsymbol{x}_0)$ as $\epsilon \to 0$ if the summary statistics are sufficient. The acceptance rate, however, vanishes in this limit and decreases rapidly with the dimension of $\boldsymbol{s}$. ABC with MCMC improves the efficiency by proposing parameters from a Markov chain instead of the prior [131], and sequential Monte Carlo variants propagate a population of parameters through a decreasing sequence of tolerances [132, 133]. Synthetic-likelihood methods instead approximate the distribution of the summary statistics at each $\boldsymbol{\theta}$ by a tractable density, such as a Gaussian or a kernel density estimate, fitted to repeated simulations. This density then replaces the intractable likelihood in standard MCMC [134]. All these methods require new simulations for every observation. Neural SBI methods instead train a neural network on simulated pairs $(\boldsymbol{\theta}_i, \boldsymbol{x}_i)$ to approximate the likelihood [135], the likelihood ratio [136], or the posterior itself [137, 138], referred to as neural likelihood, ratio, and posterior estimation (NLE, NRE, and NPE), respectively [53].

2.4.2 Neural posterior estimation

As summarized in Fig. 2.2, NPE draws $\boldsymbol{\theta}_i \sim \pi(\boldsymbol{\theta})$, simulates $\boldsymbol{x}_i \sim p(\boldsymbol{x}|\boldsymbol{\theta}_i)$, and fits an approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})$ by

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{\pi(\boldsymbol{\theta})\, p(\boldsymbol{x}|\boldsymbol{\theta})}\left[-\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})\right]. \tag{2.49}$$

A sufficiently flexible estimator can thus approximate $p(\boldsymbol{\theta}|\boldsymbol{x})$ for observations in the support of the prior predictive distribution $p(\boldsymbol{x})$. Normalizing flows with efficiently evaluable densities are well suited to this objective. Score matching [139] and flow matching [140] offer alternative objectives that avoid explicit evaluation of the model density during training. Training requires only sampling access to the simulator. Once trained, the model provides *amortized inference* by returning an approximate posterior for a new observation $\boldsymbol{x}_0$ without additional simulations or retraining.

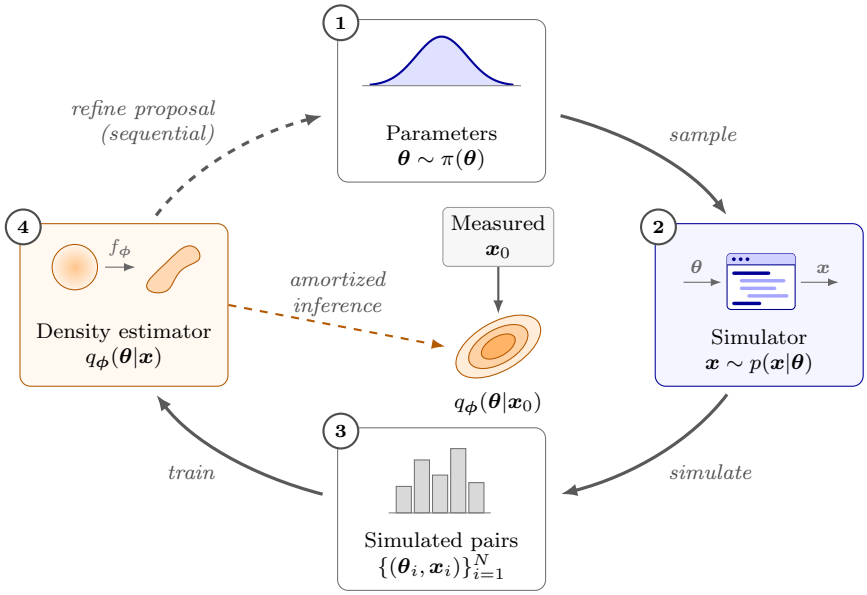


Figure 2.2: Simulation-based inference with neural posterior estimation. Parameters drawn from the prior $\pi(\boldsymbol{\theta})$ are passed through a simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$, and a neural density estimator $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})$ is trained on the resulting pairs $\{(\boldsymbol{\theta}_i, \boldsymbol{x}_i)\}_{i=1}^N$. Once trained, the estimator returns the posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x}_0)$ for any measured observation $\boldsymbol{x}_0$ without further simulation. Sequential variants close the loop by drawing new parameters from a proposal concentrated near the current posterior estimate.

When only a single $\boldsymbol{x}_0$ is of interest, sequential NPE draws parameters in later rounds from a proposal $\tilde{\pi}(\boldsymbol{\theta})$ concentrated near the current posterior estimate, which reduces the number of simulations required [137, 138]. Maximum-likelihood training on $\tilde{\pi}$ instead of π yields the proposal posterior $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{\theta}|\boldsymbol{x})\, \tilde{\pi}(\boldsymbol{\theta})/\pi(\boldsymbol{\theta})$, which is biased relative to $p(\boldsymbol{\theta}|\boldsymbol{x})$. Sequential methods therefore either modify the training loss to account for this difference [138] or draw from truncated versions of the prior [141]. In both cases, the resulting estimator is no longer amortized over observations. **Paper II** applies NPE to noise learning, state tomography, and Hamiltonian learning, and extends its application to quantum error mitigation.

Since the true posterior is unknown, NPE is validated on simulated test pairs with known parameters. Point estimates can be compared with the true parameters, and calibration can be tested by simulation-based calibration, which checks that the rank of the true parameter among posterior samples is uniformly distributed [142]. Posteriors obtained by NPE can be overconfident, with credible regions smaller than they should be, particularly for small simulation budgets [143]. If distinct parameters produce the same distribution of observations, as for the gauge-related noise parameters of Section 2.1.5, the posterior cannot concentrate along the corresponding directions. The meaning

of the posterior width depends on the simulator. If the simulator is stochastic, for example through shot noise, the width partly reflects the stochasticity of the observations. If it is instead deterministic, $\boldsymbol{x} = F(\boldsymbol{\theta})$, the likelihood is a Dirac distribution, $p(\boldsymbol{x}|\boldsymbol{\theta}) = \delta(\boldsymbol{x} - F(\boldsymbol{\theta}))$, and the posterior is supported on the level set $\{\boldsymbol{\theta} \in \operatorname{supp} \pi : F(\boldsymbol{\theta}) = \boldsymbol{x}\}$, so its width reflects only non-identifiability and approximation error. Finally, the posterior is only as reliable as the simulator. Under *model misspecification*, the simulator does not accurately describe the device, and experimental data may lie outside the distribution of simulated observations. NPE can then return biased posteriors even when it is well calibrated on simulated data [144, 145].

Chapter 3

Summary of Included Papers

This chapter summarizes the appended papers. Both papers use conditional normalizing flows, introduced in Section 2.3.2, trained by maximum likelihood. **Paper I** learns a distribution over the circuit parameters of the variational quantum eigensolver, and **Paper II** learns posterior distributions over the parameters of quantum systems from classical simulations.

3.1 Generative flow-based warm start of the variational quantum eigensolver

Paper I addresses the optimization cost of the variational quantum eigensolver (VQE) described in Section 2.1.3.

Problem

The VQE minimizes the energy of a parameterized circuit by classical optimization, and each evaluation of the energy requires repeated circuit executions. The landscape is typically nonconvex and may exhibit barren plateaus and many local minima, so that convergence often requires a large number of circuit evaluations. Gradient-based optimizers incur the additional cost of estimating gradients on the quantum device, whereas gradient-free optimizers typically require even more evaluations. Moreover, conventional optimizers treat each Hamiltonian separately and do not transfer knowledge between related problems, such as molecular geometries along a potential-energy surface.

Contribution

We propose Flow-VQE, which models a conditional distribution $q_{\phi}(\boldsymbol{\theta}|\boldsymbol{\gamma})$ over circuit parameters $\boldsymbol{\theta}$ with a conditional Gaussianization flow, described in Section 2.3.2. The context $\boldsymbol{\gamma}$ is constructed from the Pauli coefficients of the

molecular Hamiltonian. Since no samples of optimal parameters are available in advance, the flow is trained by reinforcement learning on its own samples, rewarding parameters with lower measured energies. This energy-preference training replaces the traditional optimizer and requires no quantum gradients. A model trained on several geometries then generates parameters for unseen geometries by sampling alone, and these parameters serve as warm starts for subsequent optimization.

Results

We benchmark Flow-VQE in state-vector simulations of hydrogen chains, water, ammonia, and benzene. As an optimizer for a single geometry, Flow-VQE reaches computational accuracy with fewer circuit evaluations than the baseline optimizers, with improvements ranging from modest to more than two orders of magnitude. A model trained on multiple geometries generates parameters that closely reproduce the potential-energy surfaces. For the small molecules studied, using these parameters as a warm start accelerates subsequent fine-tuning by up to 50-fold compared with Hartree–Fock initialization.

3.2 Simulation-based quantum system inference with neural posterior estimation

Paper II develops the simulation-based inference framework of Section 2.4 for the characterization of quantum systems, using the classical simulators described in Section 2.2.

Problem

Learning quantum systems, such as Hamiltonians, states, or noise processes, from measurement data is central to the characterization and verification of quantum devices. Standard Bayesian inference for such tasks requires evaluating the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$, which generally involves simulating the full quantum system at a cost that grows exponentially with system size. Classical likelihood-free methods avoid this evaluation but require many simulations and must be rerun for every new observation. Characterization tasks such as noise tracking and cross-device benchmarking are, however, carried out repeatedly on large devices.

Contribution

We introduce simulation-based quantum system inference, a framework that pairs polynomial-cost classical simulators, such as Pauli propagation and tensor networks, with neural posterior estimation. A conditional neural spline flow is trained once on simulated pairs of parameters and observations and then maps any new observation to its posterior without retraining. The framework treats otherwise tailor-made characterization tasks within a single template, and the posterior uncertainty indicates which parameters are not identifiable from the chosen measurements. We apply it to four inverse problems, namely

Pauli noise learning, quantum error mitigation, quantum state tomography, and Hamiltonian learning. For error mitigation, we use the inferred noise in protocols that require an explicit noise model, and we further introduce a digital-twin strategy, in which simulators based on the inferred noise generate training data for learning-based error mitigation.

Results

In Pauli noise learning, the model infers sparse Pauli noise in shallow circuits of up to 50 qubits, and the required training data grow approximately linearly with system size. With the inferred noise, probabilistic error cancellation and zero-noise extrapolation reach an accuracy comparable to that obtained with the true noise parameters. The digital-twin strategy outperforms zero-noise extrapolation for Trotterized Ising dynamics. In quantum state tomography, the states are prepared by a fixed parameterized quantum circuit, and a single estimator infers its parameters, reconstructing randomly sampled 12-qubit states with fidelities typically exceeding 0.99. In Hamiltonian learning, the model infers atomic displacements in simulated 9×9 Rydberg atom arrays from local observables, with site-resolved uncertainties that indicate where further calibration may be needed. Overall, the examples demonstrate efficient characterization of systems with up to 81 qubits and up to 735 inferred parameters.

Chapter 4

Discussion and Future Work

This thesis has studied how conditional generative models can reduce the cost of forward and inverse problems in quantum computing. Both appended papers use conditional normalizing flows that learn distributions over continuous parameters and are trained once for a family of related problems.

In **Paper I**, generative models combined with reinforcement learning provide a framework that amortizes the cost of variational optimization. Related works, in which machine-learning models generate circuits or circuit parameters for families of ground-state problems instead of optimizing each instance separately, point to an emerging trend in variational quantum algorithms [146–149].

In **Paper II**, generative models combined with classical simulators provide a framework that amortizes Bayesian inference of quantum systems. This extends the role of classical simulation techniques beyond competing with current quantum devices in benchmarks [150–152]. Through neural posterior estimation, classical simulators become practical tools for inverse problems and pave the way toward scalable characterization with uncertainty quantification for large quantum devices.

Building on these results, we are currently pursuing several directions, summarized in Fig. 4.1. They extend the generative models to multimodal spaces, which combine discrete and continuous variables, and involve structure and representation learning for quantum states.

Generating gate sequences and continuous angles jointly extends **Paper I** to multimodal circuit design, building on quantum architecture search [153, 154]. Such models may find circuits at a favorable balance of expressibility, trainability, and hardware awareness. Pretrained large language models can additionally provide an informative prior for quantum circuit generation [155, 156]. Similarly, simulation-based model inference extends **Paper II** by learning joint posteriors over discrete model components and their continuous parameters [157]. It may support inverse quantum design, in which quantum models are designed for desired properties [54, 158, 159]. With quantum computers serving as the simulators, such model inference may become a setting for demonstrating

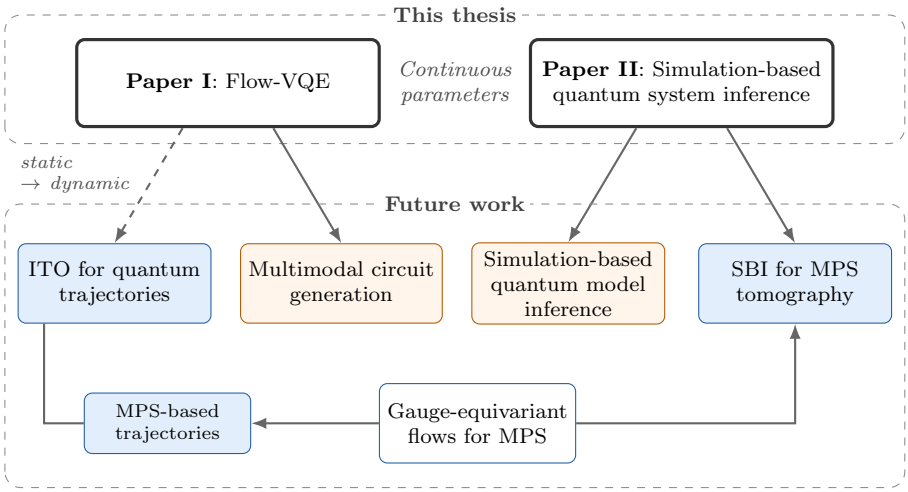


Figure 4.1: Overview of future work. **Paper I** and **Paper II** learn distributions over continuous parameters. Orange boxes extend the generative models to multimodal design spaces that combine discrete and continuous variables. Blue boxes involve structure and representation learning for quantum states. Implicit transfer operators (ITOs) learn the dynamics of quantum trajectories and extend the research problem from static to dynamic (dashed arrow). Gauge-equivariant flows can reduce the sample complexity of learning from matrix product state (MPS) data, such as MPS-based trajectories, and also serve SBI for MPS tomography.

quantum advantage in the future.

Implicit transfer operators learn the transition probabilities of Markov dynamics and generate long trajectories in far fewer steps than direct simulation of the underlying dynamics [160–164]. We are applying them to quantum-trajectory methods for open quantum systems [165]. Scalable simulation of such trajectories is enabled by MPS [166], whose tensor representation is highly redundant because of gauge symmetry [167]. Gauge-equivariant flows build this symmetry in as an inductive bias, so that the redundancy need not be learned from data [168–170]. In addition, SBI can parameterize MPS directly for quantum state tomography, broadening the demonstrations in **Paper II**. Here, gauge-equivariant flows can reduce the required simulation budget. Finally, all of these projects may benefit from models that exploit subsystem structure, such as the locality of physical interactions. Such structure allows what is learned on small systems to be applied to large quantum devices, as demonstrated with methods that characterize or simulate large systems through local or partial descriptions [83, 171, 172] and with graph neural networks [149, 162, 173].

These directions build directly on the work of this thesis and leave broad room for generative models to advance many aspects of quantum simulation and quantum system inference.

Use of Aids

Claude Opus 5.5 was used under the author's direction and supervision to assist with language editing of the introductory chapters, searching for and verifying references, and preparing the Ti*k*Z code of figures. The cover illustration was generated with the assistance of GPT-6.1 Sol. All ideas and viewpoints are the author's own. All content was reviewed and revised by the author, who takes full responsibility for this thesis.

Bibliography

[1] R. P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics **21**, 467–488 (1982) (cit. on p. 3).

[2] D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, Proceedings of the Royal Society of London A **400**, 97–117 (1985) (cit. on p. 3).

[3] J. Preskill, The physics of quantum information, arXiv:2208.08064 (2022) (cit. on p. 3).

[4] A. Streltsov, G. Adesso and M. B. Plenio, Colloquium: Quantum coherence as a resource, Reviews of Modern Physics **89**, 041003 (2017) (cit. on p. 3).

[5] R. Horodecki, P. Horodecki, M. Horodecki and K. Horodecki, Quantum entanglement, Reviews of Modern Physics **81**, 865–942 (2009) (cit. on p. 3).

[6] J. Preskill, Quantum computing and the entanglement frontier, arXiv:1203.5813 (2012) (cit. on p. 3).

[7] A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proceedings of the London Mathematical Society **s2-42**, 230–265 (1936) (cit. on p. 3).

[8] E. Bernstein and U. Vazirani, Quantum complexity theory, SIAM Journal on Computing **26**, 1411–1473 (1997) (cit. on p. 3).

[9] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing **26**, 1484–1509 (1997) (cit. on p. 3).

[10] S. Lloyd, Universal quantum simulators, Science **273**, 1073–1078 (1996) (cit. on pp. 3, 9).

[11] A. W. Harrow, A. Hassidim and S. Lloyd, Quantum algorithm for linear systems of equations, Physical Review Letters **103**, 150502 (2009) (cit. on p. 3).

[12] P. Rebentrost, A. Steffens, I. Marvian and S. Lloyd, Quantum singular-value decomposition of nonsparse low-rank matrices, Physical Review A **97**, 012327 (2018) (cit. on p. 4).

[13] S. Aaronson, How much structure is needed for huge quantum speedups?, arXiv:2209.06930 (2022) (cit. on p. 4).

[14] D. Coppersmith, An approximate Fourier transform useful in quantum factoring, arXiv:quant-ph/0201067 (2002) (cit. on p. 4).

[15] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, arXiv:quant-ph/9511026 (1995) (cit. on pp. 4, 9).

[16] A. Gilyén, Y. Su, G. H. Low and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (2019), pp. 193–204 (cit. on p. 4).

[17] 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 (cit. on p. 4).

[18] G. Brassard, P. Høyer, M. Mosca and A. Tapp, Quantum amplitude amplification and estimation, in Quantum computation and information, Vol. 305 (American Mathematical Society, 2002), pp. 53–74 (cit. on p. 4).

[19] A. Ambainis, Quantum walk algorithm for element distinctness, SIAM Journal on Computing **37**, 210–239 (2007) (cit. on p. 4).

[20] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics **3**, 625–644 (2021) (cit. on pp. 4, 13).

[21] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Reviews of Modern Physics **94**, 015004 (2022) (cit. on pp. 4, 13).

[22] J. Robledo-Moreno, M. Motta, H. Haas, A. Javadi-Abhari, P. Jurcevic, W. Kirby, S. Martiel, K. Sharma, S. Sharma, T. Shirakawa, I. Sitdikov, R.-Y. Sun, K. J. Sung, M. Takita, M. C. Tran, S. Yunoki and A. Mezzacapo, Chemistry beyond the scale of exact diagonalization on a quantum-centric supercomputer, Science Advances **11**, eadu9991 (2025) (cit. on p. 4).

[23] H.-Y. Huang, S. Choi, J. R. McClean and J. Preskill, Vast world of quantum advantage, Physical Review X **16**, 030501 (2026) (cit. on p. 4).

[24] J. Eisert and J. Preskill, Mind the gaps: The fraught road to quantum advantage, arXiv:2510.19928 (2026) (cit. on p. 4).

[25] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean and T. E. O'Brien, Quantum error mitigation, Reviews of Modern Physics **95**, 045005 (2023) (cit. on pp. 4, 16).

[26] D. J. Spencer, S. P. Jain, A. Tanggara, Z. Sun, T. Haug, D. Khu and K. Bharti, Quantum error correction and fault tolerance: A comprehensive tutorial, arXiv:2605.29137 (2026) (cit. on p. 4).

[27] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, SIAM Journal on Computing **38**, 1207–1282 (2008) (cit. on p. 4).

[28] E. Knill, R. Laflamme and W. H. Zurek, Resilient quantum computation, Science **279**, 342–345 (1998) (cit. on p. 4).

[29] Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, Nature **638**, 920–926 (2025) (cit. on p. 4).

[30] V. Sivak, M. Newman and P. Klimov, Optimization of decoder priors for accurate quantum error correction, Physical Review Letters **133**, 150603 (2024) (cit. on p. 4).

[31] J. Bausch, A. W. Senior, F. J. H. Heras et al., Learning high-accuracy error decoding for quantum processors, Nature **635**, 834–840 (2024) (cit. on p. 4).

[32] M. Sarovar, T. Proctor, K. Rudinger, K. Young, E. Nielsen and R. Blume-Kohout, Detecting crosstalk errors in quantum information processors, Quantum **4**, 321 (2020) (cit. on p. 4).

[33] M. McEwen, L. Faoro, K. Arya et al., Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits, Nature Physics **18**, 107–111 (2022) (cit. on p. 4).

[34] T. Proctor, M. Revelle, E. Nielsen, K. Rudinger, D. Lobser, P. Maunz, R. Blume-Kohout and K. Young, Detecting and tracking drift in quantum information processors, Nature Communications **11**, 5396 (2020) (cit. on p. 4).

[35] V. Gebhart, R. Santagati, A. A. Gentile, E. M. Gauger, D. Craig, N. Ares, L. Banchi, F. Marquardt, L. Pezze and C. Bonato, Learning quantum systems, Nature Reviews Physics **5**, 141–156 (2023) (cit. on pp. 4, 5).

[36] Y. Du, Y. Zhu, Y.-H. Zhang, M.-H. Hsieh, P. Rebentrost, W. Gao, Y.-Z. You, J. Eisert, G. Chiribella, D. Tao et al., Artificial intelligence for representing and characterizing quantum systems, Nature Reviews Physics **8**, 579–596 (2026) (cit. on pp. 4, 5).

[37] A. Hashim, L. B. Nguyen, N. Goss, B. Marinelli, R. K. Naik, T. Chistolini, J. Hines, J. Marceaux, Y. Kim, P. Gokhale, T. Tomesh, S. Chen, L. Jiang, S. Ferracin, K. Rudinger, T. Proctor, K. C. Young, I. Siddiqi and R. Blume-Kohout, Practical introduction to benchmarking and characterization of quantum computers, PRX Quantum **6**, 030202 (2025) (cit. on p. 4).

[38] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush and H. Neven, Barren plateaus in quantum neural network training landscapes, Nature Communications **9**, 4812 (2018) (cit. on pp. 4, 13).

[39] Z. Holmes, K. Sharma, M. Cerezo and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, PRX Quantum **3**, 010313 (2022) (cit. on pp. 4, 13).

[40] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp et al., Does provable absence of barren plateaus imply classical simulability?, Nature Communications **16**, 7907 (2025) (cit. on p. 4).

[41] R. Puig, M. Drudis, S. Thanasilp and Z. Holmes, Variational quantum simulation: A case study for understanding warm starts, PRX Quantum **6**, 010317 (2025) (cit. on p. 4).

[42] H. Zou, M. Rahm, A. F. Kockum and S. Olsson, Generative flow-based warm start of the variational quantum eigensolver, npj Quantum Information **12**, 5 (2025) (cit. on pp. 4, 5).

[43] S. Noy and W. Zhang, Experimental evidence on the productivity effects of generative artificial intelligence, Science **381**, 187–192 (2023) (cit. on p. 4).

[44] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, High-resolution image synthesis with latent diffusion models, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022), pp. 10684–10695 (cit. on p. 4).

[45] A. Polyak, A. Zohar, A. Brown et al., Movie Gen: A cast of media foundation models, arXiv:2410.13720 (2024) (cit. on p. 4).

[46] B. Zitkovich, T. Yu, S. Xu et al., RT-2: Vision-language-action models transfer web knowledge to robotic control, in Proceedings of the 7th Conference on Robot Learning, Vol. 229, PMLR (2023), pp. 2165–2183 (cit. on p. 4).

[47] A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász, M. Lackenby, G. Williamson, D. Hassabis and P. Kohli, Advancing mathematics by guiding human intuition with AI, Nature **600**, 70–74 (2021) (cit. on p. 4).

[48] J. Jumper, R. Evans, A. Pritzel et al., Highly accurate protein structure prediction with AlphaFold, Nature **596**, 583–589 (2021) (cit. on pp. 4, 5).

[49] C. Zeni, R. Pinsler, D. Zügner et al., A generative model for inorganic materials design, Nature **639**, 624–632 (2025) (cit. on pp. 4, 5).

[50] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno and B. Schölkopf, Real-time gravitational wave science with neural posterior estimation, Physical Review Letters **127**, 241103 (2021) (cit. on pp. 4, 5).

[51] F. Noé, S. Olsson, J. Köhler and H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science **365**, eaaw1147 (2019) (cit. on pp. 5, 23).

[52] I. Price, A. Sanchez-Gonzalez, F. Alet, T. R. Andersson, A. El-Kadi, D. Masters, T. Ewalds, J. Stott, S. Mohamed, P. Battaglia, R. Lam and M. Willson, Probabilistic weather forecasting with machine learning, Nature **637**, 84–90 (2025) (cit. on p. 5).

[53] K. Cranmer, J. Brehmer and G. Louppe, The frontier of simulation-based inference, Proceedings of the National Academy of Sciences **117**, 30055–30062 (2020) (cit. on pp. 5, 26).

[54] M. Krenn, J. S. Kottmann, N. Tischler and A. Aspuru-Guzik, Conceptual understanding through efficient automated design of quantum optical experiments, Physical Review X **11**, 031044 (2021) (cit. on pp. 5, 33).

[55] Y. Alexeev, M. H. Farag, T. L. Patti et al., Artificial intelligence for quantum computing, Nature Communications **16**, 10829 (2025) (cit. on p. 5).

[56] G. Acampora, A. Ambainis, N. Ares et al., Quantum computing and artificial intelligence: status and perspectives, arXiv:2505.23860 (2025) (cit. on p. 5).

[57] M. Chen, Y. Gan, X. Jin et al., When AI meets quantum information: A comprehensive review, arXiv:2607.00365 (2026) (cit. on p. 5).

[58] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nature Communications **5**, 4213 (2014) (cit. on pp. 5, 13).

[59] H. Zou, A. F. Kockum, M. Rahm and S. Olsson, Simulation-based quantum system inference with neural posterior estimation, arXiv:2609.34995 (2026) (cit. on p. 5).

[60] J. Kempe, A. Kitaev and O. Regev, The complexity of the local Hamiltonian problem, SIAM Journal on Computing **35**, 1070–1097 (2006) (cit. on p. 9).

[61] S. Lee, J. Lee, H. Zhai, Y. Tong, A. M. Dalzell, A. Kumar, P. Helms, J. Gray, Z.-H. Cui, W. Liu, M. Kastoryano, R. Babbush, J. Preskill, D. R. Reichman, E. T. Campbell, E. F. Valeev, L. Lin and G. K.-L. Chan, Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry, Nature Communications **14**, 1952 (2023) (cit. on p. 9).

[62] S. Fomichev, K. Hejazi, M. S. Zini, M. Kiser, J. Fraxanet, P. A. M. Casares, A. Delgado, J. Huh, A.-C. Voigt, J. E. Mueller and J. M. Arrazola, Initial state preparation for quantum chemistry on quantum computers, PRX Quantum **5**, 040339 (2024) (cit. on p. 9).

[63] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe and S. Zhu, Theory of Trotter error with commutator scaling, Physical Review X **11**, 011020 (2021) (cit. on p. 9).

[64] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin and X. Yuan, Quantum computational chemistry, Reviews of Modern Physics **92**, 015003 (2020) (cit. on p. 10).

[65] P. Jordan and E. Wigner, Über das Paulische Äquivalenzverbot, Zeitschrift für Physik **47**, 631–651 (1928) (cit. on p. 10).

[66] S. Bravyi, J. M. Gambetta, A. Mezzacapo and K. Temme, Tapering off qubits to simulate fermionic Hamiltonians, arXiv:1701.08213 (2017) (cit. on p. 10).

[67] O. Simard, A. Dawid, J. Tindall, M. Ferrero, A. M. Sengupta and A. Georges, Learning interactions between Rydberg atoms, PRX Quantum **6**, 030324 (2025) (cit. on p. 11).

[68] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information (Cambridge University Press, 2000) (cit. on pp. 11, 12, 14).

[69] D. Gottesman, The Heisenberg representation of quantum computers, arXiv:quant-ph/9807006 (1998) (cit. on pp. 12, 17).

[70] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Physical Review A **70**, 052328 (2004) (cit. on pp. 12, 17, 18).

[71] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, Physical Review Letters **102**, 110502 (2009) (cit. on p. 12).

[72] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, Nature **402**, 390–393 (1999) (cit. on p. 12).

[73] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Physical Review A **71**, 022316 (2005) (cit. on p. 12).

[74] D. Litinski, Magic state distillation: Not as costly as you think, Quantum **3**, 205 (2019) (cit. on p. 12).

[75] C. Gidney, N. Shutty and C. Jones, Magic state cultivation: growing T states as cheap as CNOT gates, arXiv:2409.17595 (2024) (cit. on p. 12).

[76] Y. Liu, D. Wang, S. Xue, A. Huang, X. Fu, X. Qiang, P. Xu, H.-L. Huang, M. Deng, C. Guo, X. Yang and J. Wu, Variational quantum circuits for quantum state tomography, Physical Review A **101**, 052316 (2020) (cit. on p. 13).

[77] S. Sim, P. D. Johnson and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, Advanced Quantum Technologies **2**, 1900070 (2019) (cit. on p. 13).

[78] K. Życzkowski and H.-J. Sommers, Average fidelity between random quantum states, Physical Review A **71**, 032313 (2005) (cit. on p. 13).

[79] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-hard, Physical Review Letters **127**, 120502 (2021) (cit. on p. 13).

[80] D. Greenbaum, Introduction to quantum gate set tomography, arXiv:1509.02921 (2015) (cit. on p. 14).

[81] R. Harper, S. T. Flammia and J. J. Wallman, Efficient learning of quantum noise, Nature Physics **16**, 1184–1188 (2020) (cit. on p. 14).

[82] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, Physical Review A **94**, 052325 (2016) (cit. on p. 14).

[83] E. van den Berg, Z. K. Minev, A. Kandala and K. Temme, Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors, Nature Physics **19**, 1116–1121 (2023) (cit. on pp. 15, 16, 34).

[84] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young and R. Blume-Kohout, Gate set tomography, Quantum **5**, 557 (2021) (cit. on p. 16).

[85] S. Chen, Y. Liu, M. Otten, A. Seif, B. Fefferman and L. Jiang, The learnability of Pauli noise, Nature Communications **14**, 52 (2023) (cit. on p. 16).

[86] K. Temme, S. Bravyi and J. M. Gambetta, Error mitigation for short-depth quantum circuits, Physical Review Letters **119**, 180509 (2017) (cit. on p. 16).

[87] Y. Li and S. C. Benjamin, Efficient variational quantum simulator incorporating active error minimization, Physical Review X **7**, 021050 (2017) (cit. on p. 16).

[88] P. Czarnik, A. Arrasmith, P. J. Coles and L. Cincio, Error mitigation with Clifford quantum-circuit data, Quantum **5**, 592 (2021) (cit. on p. 16).

[89] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin and Y. Li, Learning-based quantum error mitigation, PRX Quantum **2**, 040330 (2021) (cit. on p. 16).

[90] H. Zou, E. Magnusson, H. Brunander, W. Dobrautz and M. Rahm, Multireference error mitigation for quantum computation of chemistry, Digital Discovery **4**, 2521–2533 (2025) (cit. on p. 16).

[91] Y. Zhou, E. M. Stoudenmire and X. Waintal, What limits the simulation of quantum computers?, Physical Review X **10**, 041038 (2020) (cit. on p. 17).

[92] M. S. Rudolph, T. Jones, Y. Teng, A. Angrisani and Z. Holmes, Pauli propagation: A computational framework for simulating quantum systems, PRX Quantum **7**, 032001 (2026) (cit. on pp. 17, 20).

[93] M. Van den Nest, Classical simulation of quantum computation, the Gottesman–Knill theorem, and slightly beyond, Quantum Information and Computation **10**, 258–271 (2010) (cit. on p. 17).

[94] C. Ferrie and C. E. Granade, Likelihood-free methods for quantum parameter estimation, Physical Review Letters **112**, 130402 (2014) (cit. on p. 17).

[95] S. Aaronson, Quantum computing, postselection, and probabilistic polynomial-time, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **461**, 3473–3482 (2005) (cit. on p. 17).

[96] S. Toda, PP is as hard as the polynomial-time hierarchy, SIAM Journal on Computing **20**, 865–877 (1991) (cit. on p. 17).

[97] C. H. Bennett, E. Bernstein, G. Brassard and U. Vazirani, Strengths and weaknesses of quantum computing, SIAM Journal on Computing **26**, 1510–1523 (1997) (cit. on p. 17).

[98] M. J. Bremner, R. Jozsa and D. J. Shepherd, Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **467**, 459–472 (2011) (cit. on p. 17).

[99] A. W. Harrow and A. Montanaro, Quantum computational supremacy, Nature **549**, 203–209 (2017) (cit. on p. 17).

[100] C. Gidney, Stim: a fast stabilizer circuit simulator, Quantum **5**, 497 (2021) (cit. on p. 18).

[101] S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, Physical Review Letters **116**, 250501 (2016) (cit. on p. 18).

[102] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, Physical Review Letters **91**, 147902 (2003) (cit. on p. 18).

[103] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Annals of Physics **326**, 96–192 (2011) (cit. on p. 18).

[104] M. B. Hastings, An area law for one-dimensional quantum systems, Journal of Statistical Mechanics: Theory and Experiment **2007**, P08024 (2007) (cit. on p. 19).

[105] J. Eisert, M. Cramer and M. B. Plenio, Colloquium: Area laws for the entanglement entropy, Reviews of Modern Physics **82**, 277–306 (2010) (cit. on p. 19).

[106] R. Jozsa, On the simulation of quantum circuits, arXiv:`quant-ph/0603163` (2006) (cit. on p. 19).

[107] A. Nahum, J. Ruhman, S. Vijay and J. Haah, Quantum entanglement growth under random unitary dynamics, Physical Review X **7**, 031016 (2017) (cit. on p. 20).

[108] P. Rall, D. Liang, J. Cook and W. Kretschmer, Simulation of qubit quantum circuits via Pauli propagation, Physical Review A **99**, 062337 (2019) (cit. on p. 20).

[109] D. Aharonov, X. Gao, Z. Landau, Y. Liu and U. Vazirani, A polynomial-time classical algorithm for noisy random circuit sampling, in Proceedings of the 55th Annual ACM Symposium on Theory of Computing (2023), pp. 945–957 (cit. on p. 20).

[110] T. Schuster, C. Yin, X. Gao and N. Y. Yao, A polynomial-time classical algorithm for noisy quantum circuits, Physical Review X **15**, 041018 (2025) (cit. on p. 20).

[111] Y. Shao, F. Wei, S. Cheng and Z. Liu, Simulating noisy variational quantum algorithms: A polynomial approach, Physical Review Letters **133**, 120603 (2024) (cit. on p. 20).

[112] H. Larochelle and I. Murray, The neural autoregressive distribution estimator, in Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Vol. 15, PMLR (2011), pp. 29–37 (cit. on p. 22).

[113] A. van den Oord, N. Kalchbrenner and K. Kavukcuoglu, Pixel recurrent neural networks, in Proceedings of the 33rd International Conference on Machine Learning, Vol. 48, PMLR (2016), pp. 1747–1756 (cit. on p. 22).

[114] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato and F. J. Huang, A tutorial on energy-based learning, in Predicting structured data (MIT Press, Cambridge, MA, 2006) (cit. on p. 22).

[115] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, in International Conference on Learning Representations (2014) (cit. on p. 22).

[116] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, in Advances in Neural Information Processing Systems, Vol. 27 (2014), pp. 2672–2680 (cit. on p. 22).

[117] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon and B. Poole, Score-based generative modeling through stochastic differential equations, in International Conference on Learning Representations (2021) (cit. on p. 22).

[118] R. T. Q. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud, Neural ordinary differential equations, in Advances in Neural Information Processing Systems, Vol. 31 (2018) (cit. on p. 22).

[119] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel and M. Le, Flow matching for generative modeling, in International Conference on Learning Representations (2023) (cit. on p. 22).

[120] X. Liu, C. Gong and Q. Liu, Flow straight and fast: Learning to generate and transfer data with rectified flow, in International Conference on Learning Representations (2023) (cit. on p. 22).

[121] M. S. Albergo and E. Vanden-Eijnden, Building normalizing flows with stochastic interpolants, in International Conference on Learning Representations (2023) (cit. on p. 22).

[122] M. S. Albergo, N. M. Boffi and E. Vanden-Eijnden, Stochastic interpolants: A unifying framework for flows and diffusions, Journal of Machine Learning Research **26**, 1–80 (2025) (cit. on p. 22).

[123] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, Journal of Machine Learning Research **22**, 1–64 (2021) (cit. on pp. 22, 23).

[124] D. J. Rezende and S. Mohamed, Variational inference with normalizing flows, in Proceedings of the 32nd International Conference on Machine Learning, Vol. 37, PMLR (2015), pp. 1530–1538 (cit. on p. 23).

[125] L. Dinh, J. Sohl-Dickstein and S. Bengio, Density estimation using Real NVP, in International Conference on Learning Representations (2017) (cit. on p. 24).

[126] G. Papamakarios, T. Pavlakou and I. Murray, Masked autoregressive flow for density estimation, in Advances in Neural Information Processing Systems, Vol. 30 (2017), pp. 2335–2344 (cit. on p. 24).

[127] C. Durkan, A. Bekasov, I. Murray and G. Papamakarios, Neural spline flows, in Advances in Neural Information Processing Systems, Vol. 32 (2019), pp. 7511–7522 (cit. on p. 24).

[128] M. Germain, K. Gregor, I. Murray and H. Larochelle, MADE: masked autoencoder for distribution estimation, in Proceedings of the 32nd International Conference on Machine Learning, Vol. 37, PMLR (2015), pp. 881–889 (cit. on p. 24).

[129] C. Meng, Y. Song, J. Song and S. Ermon, Gaussianization flows, in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, Vol. 108, PMLR (2020), pp. 4336–4345 (cit. on pp. 24, 25).

[130] M. A. Beaumont, W. Zhang and D. J. Balding, Approximate Bayesian computation in population genetics, Genetics **162**, 2025–2035 (2002) (cit. on p. 26).

[131] P. Marjoram, J. Molitor, V. Plagnol and S. Tavaré, Markov chain Monte Carlo without likelihoods, Proceedings of the National Academy of Sciences **100**, 15324–15328 (2003) (cit. on p. 26).

[132] S. A. Sisson, Y. Fan and M. M. Tanaka, Sequential Monte Carlo without likelihoods, Proceedings of the National Academy of Sciences **104**, 1760–1765 (2007) (cit. on p. 26).

[133] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin and C. P. Robert, Adaptive approximate Bayesian computation, Biometrika **96**, 983–990 (2009) (cit. on p. 26).

[134] C. Grazian and Y. Fan, A review of approximate Bayesian computation methods via density estimation: Inference for simulator-models, Wiley Interdisciplinary Reviews: Computational Statistics **12**, e1486 (2020) (cit. on p. 26).

[135] G. Papamakarios, D. Sterratt and I. Murray, Sequential neural likelihood: fast likelihood-free inference with autoregressive flows, in Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Vol. 89, PMLR (2019), pp. 837–848 (cit. on p. 26).

[136] J. Hermans, V. Begy and G. Louppe, Likelihood-free MCMC with amortized approximate ratio estimators, in Proceedings of the 37th International Conference on Machine Learning, Vol. 119, PMLR (2020), pp. 4239–4248 (cit. on p. 26).

[137] G. Papamakarios and I. Murray, Fast ϵ-free inference of simulation models with Bayesian conditional density estimation, in Advances in Neural Information Processing Systems, Vol. 29 (2016), pp. 1028–1036 (cit. on pp. 26, 27).

[138] D. Greenberg, M. Nonnenmacher and J. Macke, Automatic posterior transformation for likelihood-free inference, in Proceedings of the 36th International Conference on Machine Learning, Vol. 97, PMLR (2019), pp. 2404–2414 (cit. on pp. 26, 27).

[139] T. Geffner, G. Papamakarios and A. Mnih, Compositional score modeling for simulation-based inference, in Proceedings of the 40th International Conference on Machine Learning, Vol. 202, PMLR (2023), pp. 11098–11116 (cit. on p. 26).

[140] J. Wildberger, M. Dax, S. Buchholz, S. Green, J. H. Macke and B. Schölkopf, Flow matching for scalable simulation-based inference, in Advances in Neural Information Processing Systems, Vol. 36 (2023) (cit. on p. 26).

[141] M. Deistler, P. J. Gonçalves and J. H. Macke, Truncated proposals for scalable and hassle-free simulation-based inference, in Advances in Neural Information Processing Systems, Vol. 35 (2022), pp. 23135–23149 (cit. on p. 27).

[142] S. Talts, M. Betancourt, D. Simpson, A. Vehtari and A. Gelman, Validating Bayesian inference algorithms with simulation-based calibration, arXiv:1804.06788 (2018) (cit. on p. 27).

[143] J. Hermans, A. Delaunoy, F. Rozet, A. Wehenkel, V. Begy and G. Louppe, A trust crisis in simulation-based inference? Your posterior approximations can be unfaithful, Transactions on Machine Learning Research (2022) (cit. on p. 27).

[144] D. Ward, P. Cannon, M. Beaumont, M. Fasiolo and S. M. Schmon, Robust neural posterior estimation and statistical model criticism, in Advances in Neural Information Processing Systems, Vol. 35 (2022) (cit. on p. 28).

[145] M. Schmitt, P.-C. Bürkner, U. Köthe and S. T. Radev, Detecting model misspecification in amortized Bayesian inference with neural networks, in Pattern Recognition, 45th DAGM German Conference, DAGM GCPR 2023, Vol. 14264, Lecture Notes in Computer Science (2023), pp. 541–557 (cit. on p. 28).

[146] K. Nakaji, L. B. Kristensen, R. Kemmoku, J. A. Campos-Gonzalez-Angulo, M. G. Vakili, H. Huang, M. Bagherimehrab, C. Gorgulla, F. Wong, A. McCaskey, J.-S. Kim, T. Nguyen, P. Rao, Q. Gao, M. Sugawara, N. Yamamoto and A. Aspuru-Guzik, The generative quantum eigensolver (GQE) and its application for ground state search, arXiv:2401.09253 (2024) (cit. on p. 33).

[147] S. Minami, K. Nakaji, Y. Suzuki, A. Aspuru-Guzik and T. Kadowaki, Generative quantum combinatorial optimization by means of a novel conditional generative quantum eigensolver, Digital Discovery **4**, 2229–2243 (2025) (cit. on p. 33).

[148] M. Krumtünger, A. Wilms, P. K. Faehrmann, J. Eisert, J. Kottmann, P. A. Erdman and S. Khatri, Reinforcement learning of quantum circuit architectures for molecular potential energy curves, arXiv:2511.16559 (2025) (cit. on p. 33).

[149] D. Bincoletto, K. Stein, J. Motyl and J. S. Kottmann, A transferable machine learning approach to predict quantum circuit parameters for electronic structure problems, Machine Learning: Science and Technology **7**, 035055 (2026) (cit. on pp. 33, 34).

[150] F. Pan, K. Chen and P. Zhang, Solving the sampling problem of the Sycamore quantum circuits, Physical Review Letters **129**, 090502 (2022) (cit. on p. 33).

[151] J. Tindall, M. Fishman, E. M. Stoudenmire and D. Sels, Efficient tensor network simulation of IBM's Eagle kicked Ising experiment, PRX Quantum **5**, 010308 (2024) (cit. on p. 33).

[152] T. Begušić, J. Gray and G. K.-L. Chan, Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance, Science Advances **10**, eadk4321 (2024) (cit. on p. 33).

[153] Y. Du, T. Huang, S. You, M.-H. Hsieh and D. Tao, Quantum circuit architecture search for variational quantum algorithms, npj Quantum Information **8**, 62 (2022) (cit. on p. 33).

[154] S.-X. Zhang, C.-Y. Hsieh, S. Zhang and H. Yao, Differentiable quantum architecture search, Quantum Science and Technology **7**, 045023 (2022) (cit. on p. 33).

[155] A. Gu, A. H. Kemeny, S. F. Yelin and M. D. Lukin, Discovering state-of-the-art quantum circuits with large language models, arXiv:2610.22367 (2026) (cit. on p. 33).

[156] T. de Souza Farias, C. M. Porto, M. da Silva Fonseca, N. A. C. Carpi, A. de Araujo Oliveira and C. J. Villas-Boas, Programming quantum circuit parameters with large language models, arXiv:2611.17246 (2026) (cit. on p. 33).

[157] C. Schröder and J. H. Macke, Simultaneous identification of models and parameters of scientific simulators, in Proceedings of the 41st International Conference on Machine Learning, Vol. 235, PMLR (2024), pp. 43895–43927 (cit. on p. 33).

[158] E. Chertkov and B. K. Clark, Computational inverse method for constructing spaces of quantum models from wave functions, Physical Review X **8**, 031029 (2018) (cit. on p. 33).

[159] C. Kokail, P. E. Dolgirev, R. van Bijnen, D. Gonzalez-Cuadra, M. D. Lukin and P. Zoller, Inverse quantum simulation for quantum material design, arXiv:2601.12239 (2026) (cit. on p. 33).

[160] M. Schreiner, O. Winther and S. Olsson, Implicit transfer operator learning: multiple time-resolution models for molecular dynamics, in Advances in Neural Information Processing Systems, Vol. 36 (2023) (cit. on p. 34).

[161] J. V. Diez, M. Schreiner, O. Engkvist and S. Olsson, Boltzmann priors for implicit transfer operators, in International Conference on Learning Representations (2025) (cit. on p. 34).

[162] J. V. Diez, M. Schreiner and S. Olsson, Transferable generative models bridge femtosecond to nanosecond time-step molecular dynamics, Science Advances **12**, eaed2333 (2026) (cit. on p. 34).

[163] P. Antoniadis, B. Pavesi, S. Olsson and O. Winther, Protein language model embeddings improve generalization of implicit transfer operators, in Proceedings of the 43rd International Conference on Machine Learning (2026) (cit. on p. 34).

[164] W. Chen, N. Costa and J. Zavadlav, SupraTITO: Transferable generative molecular dynamics for supramolecular systems, arXiv:2610.01381 (2026) (cit. on p. 34).

[165] H.-P. Breuer and F. Petruccione, The theory of open quantum systems (Oxford University Press, Oxford, 2007) (cit. on p. 34).

[166] A. Sander, M. Fröhlich, M. Eigel, J. Eisert, P. Gelß, M. Hintermüller, R. M. Milbradt, R. Wille and C. B. Mendl, Large-scale stochastic simulation of open quantum systems, Nature Communications **16**, 11074 (2025) (cit. on p. 34).

[167] J. Haegeman, M. Mariën, T. J. Osborne and F. Verstraete, Geometry of matrix product states: Metric, parallel transport, and curvature, Journal of Mathematical Physics **55**, 021902 (2014) (cit. on p. 34).

[168] J. Köhler, L. Klein and F. Noé, Equivariant flows: Exact likelihood generative learning for symmetric densities, in Proceedings of the 37th International Conference on Machine Learning, Vol. 119, PMLR (2020), pp. 5361–5370 (cit. on p. 34).

[169] T. Cohen, M. Weiler, B. Kicanaoglu and M. Welling, Gauge equivariant convolutional networks and the icosahedral CNN, in Proceedings of the 36th International Conference on Machine Learning, Vol. 97, PMLR (2019), pp. 1321–1330 (cit. on p. 34).

[170] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende and P. E. Shanahan, Equivariant flow-based sampling for lattice gauge theory, Physical Review Letters **125**, 121601 (2020) (cit. on p. 34).

[171] T. Peng, A. W. Harrow, M. Ozols and X. Wu, Simulating large quantum circuits on a small quantum computer, Physical Review Letters **125**, 150504 (2020) (cit. on p. 34).

[172] A. Patel, A. Gaikwad, T. Huang, A. F. Kockum and T. Abad, Selective and efficient quantum state tomography for multiqubit systems, Physical Review Research **8**, 013339 (2026) (cit. on p. 34).

[173] H. Ai and Y. Liu, Scalable parameter design for superconducting quantum circuits with graph neural networks, Physical Review Letters **135**, 040601 (2025) (cit. on p. 34).

Part II

Appended Papers

Paper I

Generative flow-based warm start of the variational quantum eigensolver

H. Zou, M. Rahm, A. F. Kockum, and S. Olsson

npj Quantum Information **12**, 5 (2025)

https://doi.org/10.1038/s41534-025-01159-x

Paper II

Simulation-based quantum system inference with neural posterior estimation

H. Zou, A. F. Kockum, M. Rahm, and S. Olsson

Submitted, under review

https://arxiv.org/abs/2609.34995