Generative flow-based warm start of the variational quantum eigensolver

npj | quantum information

Article

Published in partnership with The University of New South Wales

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

Generative flow-based warm start of the variational quantum eigensolver

Hang Zou[1], Martin Rahm[2], Anton Frisk Kockum[3] & Simon Olsson[1] ✉

Hybrid quantum-classical algorithms like the variational quantum eigensolver (VQE) show promise for quantum simulations on near-term quantum devices, but are often limited by complex objective functions and expensive optimization procedures. Here, we propose Flow-VQE, a generative framework leveraging conditional normalizing flows with parameterized quantum circuits to efficiently generate high-quality variational parameters. By embedding a generative model into the VQE optimization loop through preference-based training, Flow-VQE enables quantum gradient-free optimization and offers a systematic approach for parameter transfer, accelerating convergence across related problems through warm-started optimization. We compare Flow-VQE to a number of standard benchmarks through numerical simulations on molecular systems, including hydrogen chains, water, ammonia, and benzene. We find that Flow-VQE generally outperforms baseline optimization algorithms, achieving computational accuracy with fewer circuit evaluations (improvements range from modest to more than two orders of magnitude) and, when used to warm-start the optimization of new systems, accelerates subsequent fine-tuning by up to 50-fold compared with Hartree–Fock initialization. Therefore, we believe Flow-VQE can become a pragmatic and versatile paradigm for leveraging generative modeling to reduce the costs of variational quantum algorithms.

The advent of noisy intermediate-scale quantum (NISQ)[1] computers[2–4] has spurred significant interest in hybrid quantum-classical algorithms[5–8] that can leverage current quantum hardware capabilities while mitigating their limitations using conventional computation. Among them, the variational quantum eigensolver (VQE)[6,9] is a widely adopted approach that uses parameterized quantum circuits, together with classical optimization of the parameters, to approximate the ground state of a many-body Hamiltonian. Although VQE does not overcome the worst-case computational complexity of quantum simulation—unstructured variational optimization is NP-hard[10]—it remains a promising heuristic framework, combining physically motivated ansätze, initialization schemes, and optimization strategies, whose practical value warrants continued investigation[6,11]. Here, we propose a machine learning approach for learning the distribution of good variational parameters and producing promising initial parameter guesses, taking into account physical information about the system.

A major challenge of variational quantum algorithms is that the optimization landscape is typically highly nonconvex and noisy. Phenomena such as barren plateaus[12,13] and the proliferation of local minima[14] substantially hinder convergence, often necessitating a large number of quantum circuit evaluations to achieve acceptable accuracy. Gradient-based optimization methods incur considerable overhead from quantum gradient estimation[15–17], whereas gradient-free methods typically require even more queries and scale poorly with increasing parameter dimensionality[18,19]. Furthermore, conventional optimization procedures tend to focus on isolated optimal parameter configurations, and thus may not fully exploit the underlying structure of the parameter landscape, which can limit their capacity for transferring knowledge between related tasks.

One promising strategy to address these challenges is to warm-start the variational search—beginning not from an uninformed point but from states or parameters already shaped by physical principles or prior computation. Based on the object the prior acts on, warm starts can be broadly grouped into state-based, parameter-based, and cross-instance transfer approaches. In practice, state-based routes bias initial states toward physically meaningful or otherwise simple approximations, such as chemistry-inspired references[20,21], classically guided initializations[22–24], and tensor-network pretraining compilations[25–27]. Parameter-based routes precondition the variational degrees of freedom using empirical regularities of the

[1]Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden. [2]Department of Chemistry and Chemical Engineering, Chalmers University of Technology, Gothenburg, Sweden. [3]Department of Microtechnology and Nanoscience, Chalmers University of Technology, Gothenburg, Sweden. ✉e-mail: simonols@chalmers.se

landscape. For example, effective-depth extrapolation progressively activates blocks and reuses shallow-depth optima to seed deeper circuits[28–31]. In addition, cross-instance transfer commonly materializes as parameter-based initialization, carrying forward learned parameters as priors to position new but related problems close to trainable basins, where "related" is grounded in domain structure—e.g., molecular geometry[32], shared symmetry[33], smooth continuation in Hamiltonian parameters[34], or graph structure[35]. These categories are non-mutually exclusive and can be combined—for example, pretraining compilation can dovetail seamlessly with effective-depth extrapolation.

Building on the success of heuristic warm-start methods, recent efforts have increasingly explored machine learning-based approaches that aim to enhance flexibility and applicability across diverse problem settings. Models trained on precomputed quantum data have been employed to produce effective initial parameters through supervised learning[36,37] and generative modeling[38,39], enabling generalization to unseen problem instances while supporting extremely fast sampling during inference. In parallel, emerging data-free paradigms integrate machine learning models directly into the quantum optimization loop to provide dynamic guidance and modify the cost landscape[40–43]. Extending this idea, meta-learning frameworks train across a distribution of tasks to acquire generalizable optimization behaviors. Rather than fitting solutions, meta-learners directly interact with optimization trajectories during training, progressively distilling shared initialization strategies from multiple tasks to enable rapid adaptation to new structures[44–48].

In this article, we introduce Flow-VQE, a probabilistic framework that learns the distribution of "high-quality" variational parameters that yield low-energy quantum states. By modeling this distribution, Flow-VQE equips VQE algorithms with an adaptable, learnable prior, which allows for the one-shot generation of effective initial solutions and significantly diminishes the need for costly iterative optimization from scratch. Flow-VQE leverages flow-based generative models[49–52] to explicitly model conditional probability distributions over variational parameters, conditioned on relevant contextual information of quantum systems. We develop a preference-based optimization approach for Flow-VQE, enabling efficient training while circumventing quantum gradient calculations. In doing so, the generative model replaces the classical optimizer, turning direct gradient-query interactions into a sampling-based dialog with the quantum circuit. Additionally, training the model on a suite of multi-objective tasks enables it to acquire meta-initialized heuristics for new systems.

We empirically validate Flow-VQE through state-vector simulation experiments on various quantum chemical systems. For the systems we tested, Flow-VQE reaches computational accuracy in single-molecule tasks with up to two orders of magnitude fewer circuit evaluations than gradient-descent algorithms. When used as a warm start for subsequent fine-tuning, it delivers as much as 50-fold acceleration at small learning rates, while the total training overhead remains no greater than that required to optimize five molecules by conventional methods. Our results comprehensively demonstrate that Flow-VQE significantly reduces quantum circuit evaluations and reliably generates high-quality variational parameters for initialization, paving the way for its establishment as a foundational technique for future warm-starts in variational quantum algorithms.

Results

Overview of flow-VQE

Generative modeling aims to maximize the likelihood of target data, a goal seemingly ill-suited to VQE optimization, since it presupposes access to optimal parameter samples—resources generally unavailable in this context. Nonetheless, as we will show, this challenge can be overcome by leveraging the flexibility of normalizing flows (NFs): they can be trained based on self-sampled data, with the training process guided by external quality evaluations. A compelling precedent for this paradigm is Boltzmann generators[53], which utilize NFs to learn complex equilibrium distributions by using a physical energy function to guide training in the absence of explicit training data.

Inspired by this approach, we treat the VQE energy expectation as an implicit surrogate for distributional fitness, thereby guiding the flow model to concentrate probability density in low-energy regions of parameter space. Here, the term "implicit" indicates that the model is trained not through labels or target parameters, but rather through indirect feedback provided by measured energy expectations. The main idea of Flow-VQE is as follows (see Fig. 1):

Latent sampling: Draw samples $\mathbf{z} \sim p(\mathbf{z})$ from a simple, analytically tractable prior $p(\mathbf{z})$.

Conditional transformation: Use an invertible neural network $f_{\boldsymbol{\tau}}$ to map $\mathbf{z}$ into variational parameters $\boldsymbol{\theta} = f_{\boldsymbol{\tau}}(\mathbf{z}; \boldsymbol{\gamma})$, conditioned on the problem-specific context $\boldsymbol{\gamma}$, yielding samples $\boldsymbol{\theta} \sim p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})$.

Likelihood evaluation: Explicitly determine the log-likelihood of the generated samples $\log p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})$.

Energy measurement: Run parameterized quantum circuits with $\boldsymbol{\theta}$ to measure the energy expectation value $E(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\hat{H}(\boldsymbol{\gamma})|\psi(\boldsymbol{\theta})\rangle$, providing the implicit training signal for the generative model.

Preference-based training: Identify low-energy parameter sets, map them back to the latent space via $f_{\boldsymbol{\tau}}^{-1}(\boldsymbol{\theta}; \boldsymbol{\gamma})$, and update the model parameters $\boldsymbol{\tau}$ by maximizing the log-likelihood of these samples.

The general objective for Flow-VQE is to minimize the expected energy across problem instances:

$$\begin{aligned}\boldsymbol{\tau}^* &= \arg\min_{\boldsymbol{\tau}} \mathcal{L}(\boldsymbol{\tau}) \\ &= \arg\min_{\boldsymbol{\tau}} \mathbb{E}_{\boldsymbol{\gamma}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})}[E(\boldsymbol{\theta})]\right].\end{aligned} \tag{1}$$

This objective defines a generic paradigm for generative initialization in VQE, where the model distribution $p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})$ specifies the sampling of parameters. In doing so, the model is encouraged to assign higher probability to parameter regions that yield low energy across diverse problem instances.

A central challenge in Flow-VQE is optimizing the variational energy functional $\mathcal{L}(\boldsymbol{\tau})$. The gradient with respect to the classical parameters, $\nabla_{\boldsymbol{\tau}}\mathcal{L}$, cannot be evaluated directly using standard gradient-computation techniques like automatic differentiation[54]. This intractability arises because the energy expectation, $E(\boldsymbol{\theta})$, is not a differentiable programmatic function but an observable measured from a quantum processor. To address this challenge, we introduce a preference-based optimization that constructs a surrogate objective from energy-based preferences, thereby circumventing the prohibitive costs of quantum gradient estimators (e.g., the parameter-shift rule[55]) and yielding a sample-efficient training paradigm for Flow-VQE, as detailed next.

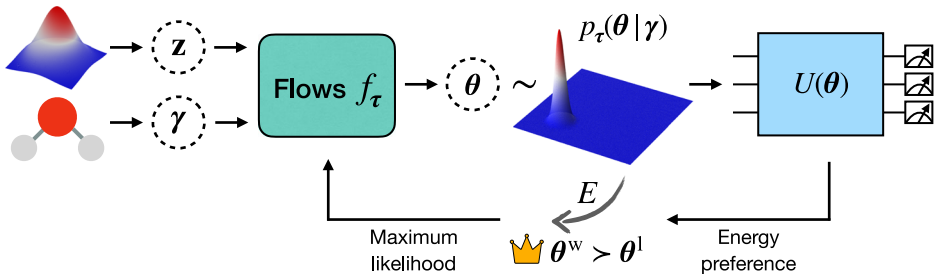

Fig. 1 | Schematic overview of the Flow-VQE framework. Latent samples $\mathbf{z}$ conditioned on the molecular context $\boldsymbol{\gamma}$ are transformed into variational parameters $\boldsymbol{\theta}$ via normalizing flows and evaluated by quantum circuits. Preference comparisons identify low-energy winners, which are retained in a buffer and used to update the model through maximum likelihood training.

Preference-based optimization

Drawing inspiration from recent advances in preference fine-tuning for large language models[56,57], we propose a preference-based optimization approach for Flow-VQE, which leverages direct performance preferences among sampled parameters to construct an informative supervision signal.

Specifically, we define a preference relation: $\boldsymbol{\theta}_i \succ \boldsymbol{\theta}_j$ if $E(\boldsymbol{\theta}_i) < E(\boldsymbol{\theta}_j)$, where $E(\boldsymbol{\theta})$ is the energy expectation value measured for a given sample $\boldsymbol{\theta}$. We maintain and dynamically update an 'elite memory buffer' $\mathcal{B} = \{(\boldsymbol{\theta}_k^w, \boldsymbol{\gamma}_k)\}_{w=1}^{M}$, which stores the top M samples under the energy-based preference ordering for each molecular structure k. To optimize the Flow-VQE distribution $p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})$, we perform maximum likelihood training over the samples from the buffer $\mathcal{B}$:

$$\mathcal{L}(\boldsymbol{\tau}) = -\mathbb{E}_{(\boldsymbol{\theta}_k^w, \boldsymbol{\gamma}_k)\sim\mathcal{B}}[\log p_{\boldsymbol{\tau}}(\boldsymbol{\theta}_k^w|\boldsymbol{\gamma}_k)], \tag{2}$$

where the log-likelihood can be explicitly computed by leveraging normalizing flows [see (7) in the *Methods* section].

In contrast to standard generative modeling, which focuses on samples from a fixed target dataset, our method performs maximum likelihood estimation on a self-sampled dataset whose quality is progressively improved through dynamic selection based on the preference criterion. Notably, the objective in (2) provides a specific guiding mechanism for the general objective in (1), as both are functionally aligned to encourage the generation of higher-quality samples.

An alternative strategy is standard policy-gradient methods such as REINFORCE[58], which depend on noisy online rewards and therefore exhibit high variance and poor sample efficiency, particularly in quantum chemistry applications (see Section I of the *Supplementary Information* for further discussion). In contrast, our preference-based approach decouples model updates from immediate online sampling and focusing exclusively on high-quality samples, thereby reducing distributional drift, lowering gradient variance, and amplifying learning signals even in low-energy-differential regimes. This mitigates the instability often observed in policy-gradient methods when low-quality samples dominate under limited sampling budgets. As a result, the model can improve reliably with small exploratory batches, substantially enhancing sample efficiency in quantum-constrained settings.

The pseudocode for Flow-VQE training via preference optimization is presented in Algorithm 1.

Algorithm 1. Training Flow-VQE based on preference optimization.

Require Molecular conditions $\{\boldsymbol{\gamma}_k\}_{k=1}^{K}$, Hamiltonians $\{\widehat{H}(\boldsymbol{\gamma}_k)\}_{k=1}^{K}$, flow model $p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})$, training epochs T, batch size B, buffer size M

Initialize flow parameters $\boldsymbol{\tau}$, memory buffers $\mathcal{B} = \{D_k = \varnothing\}_{k=1}^{K}$
for epoch = 1 to T **do**
  for each $\boldsymbol{\gamma}_k$ in $\{\boldsymbol{\gamma}_k\}_{k=1}^{K}$ **do**
    Sample parameters from the flow model: $\{(\boldsymbol{\theta}_i^{\text{new}}, \log p_{\boldsymbol{\tau}}(\boldsymbol{\theta}_i^{\text{new}}|\boldsymbol{\gamma}_k))\}_{i=1}^{B} \sim p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma}_k)$
    for $i = 1$ to B **do**
      Evaluate energy: $E_i^{\text{new}} \leftarrow \langle\psi(\boldsymbol{\theta}_i^{\text{new}})|\widehat{H}(\boldsymbol{\gamma}_k)|\psi(\boldsymbol{\theta}_i^{\text{new}})\rangle$
      Add to memory buffer: $D_k \leftarrow D_k \cup \{(\boldsymbol{\theta}_i^{\text{new}}, E_i^{\text{new}})\}$ **end for**
    Sort D_k by ascending energy
    if $|D_k| > M$ **then**
      $D_k \leftarrow \{(\boldsymbol{\theta}_k^w, E_k^w) \in D_k | w \in \{1, \ldots, M\}\}$ {Keep top M}
    end if
  end for
  Preference optimization:
  $\boldsymbol{\theta}^W \leftarrow \varnothing$
  for each D_k, corresponding to $\boldsymbol{\gamma}_k$ **do**
    $\boldsymbol{\theta}^W \leftarrow \boldsymbol{\theta}^W \cup \{(\boldsymbol{\theta}_k^w, \boldsymbol{\gamma}_k) | (\boldsymbol{\theta}_k^w, E_k^w) \in D_k\}$
  end for
  Compute loss: $\mathcal{L} \leftarrow -\mathbb{E}_{(\boldsymbol{\theta}_k^w, \boldsymbol{\gamma}_k)\in\boldsymbol{\theta}^W}[\log p_{\boldsymbol{\tau}}(\boldsymbol{\theta}_k^w|\boldsymbol{\gamma}_k)]$
  Update: $\boldsymbol{\tau} \leftarrow \boldsymbol{\tau} - \eta\nabla_{\boldsymbol{\tau}}\mathcal{L}$
end for
return $p_{\boldsymbol{\tau}}(\boldsymbol{\theta}|\boldsymbol{\gamma})$

Simulated systems

Here, we simulate four molecular systems undergoing specific conformational changes to benchmark the Flow-VQE framework. The systems studied are: a linear hydrogen chain (H_4) with simultaneous stretching of neighboring H–H bonds; water (H_2O) with symmetric stretching of both O–H bonds; ammonia (NH_3) undergoing nitrogen pyramidal inversion while its hydrogen atoms remain fixed in a plane; and benzene (C_6H_6) with the stretching of a single C–H bond. Detailed atomic coordinates are provided in Section II of the *Supplementary Information*, and further details on the basis sets and active spaces are available in the subsection *Implementation details* of the *Methods* section. We use these systems to investigate Flow-VQE in a dual role: as a standalone optimizer for direct energy minimization, and as a warm-start parameter generator for conventional VQE routines. We denote the two training regimes for Flow-VQE as Flow-VQE-S (trained on a single molecular geometry) and Flow-VQE-M (trained on multiple geometries).

Single-molecule optimization

We evaluate five optimization algorithms on H_2O and H_4 molecular systems: gradient descent (GD), quantum natural-gradient simultaneous perturbation stochastic approximation (QNSPSA)[18], Adam, and the two variants of our Flow-VQE method (Flow-VQE-S and Flow-VQE-M). While Flow-VQE-M is primarily designed as a parameter generator for warm-starting downstream tasks, it inherently performs multi-objective joint optimization. Accordingly, we include its convergence behavior in our comparative assessment.

We adopt the number of quantum circuit evaluations, independent of measurement shot counts, as the primary performance metric for comparing optimization algorithms. The total number of circuit evaluations required by each method is as follows: $2dN_{\text{epoch}}$ for GD and Adam, where d is the variational parameter dimension and N_{epoch} is the total number of optimization iterations; $6N_{\text{epoch}}$ for QNSPSA, which is a gradient-free algorithm with constant complexity; and BN_{epoch} for Flow-VQE-S and Flow-VQE-M, with a batch size of $B = 2$.

Figure 2 illustrates the number of circuit evaluations required by each algorithm to achieve convergence within computational accuracy, defined as an error not exceeding 1.6×10^{-3} Hartree relative to exact diagonalization at the same level of theory[59]. Here, the exact diagonalization results are assumed to be known for benchmarking purposes.

For the H_2O molecule (Fig. 2a), Flow-VQE-S consistently outperforms all baseline optimizers across the test structures, e.g., achieving improvements of one to two orders of magnitude over traditional GD. Compared to the competitive Adam baseline, Flow-VQE-S achieves approximately a two- to five-fold reduction in the number of circuit evaluations.

Flow-VQE-M also performs comparably to Adam, despite being trained in a more challenging multi-objective setting. While Flow-VQE-M does not match the convergence efficiency of Flow-VQE-S, this is expected: Flow-VQE-M is designed to generalize across multiple structures, rather than optimize a single fixed instance. Thus, it trades off some task-specific performance in favor of broader applicability and transferable warm-start capability.

For the H_4 system (Fig. 2b), which spans a broader range of bond lengths (0.6Å to 2.8Å), Flow-VQE-S generally outperforms the baseline optimizers across most bond lengths. The only exception arises at 0.6Å, where the optimization target is close to the initial state, allowing Adam to converge quickly. While Flow-VQE-S still retains an overall advantage, its margin over Adam is reduced—an effect that may be attributed to the more rugged energy landscapes introduced by the hardware-efficient ansatz, whose trainability remains a significant challenge. Furthermore, the performance gap between Flow-VQE-M and Flow-VQE-S narrows in this setting, implying that broader distributional exploration may contribute to avoiding certain optimization traps.

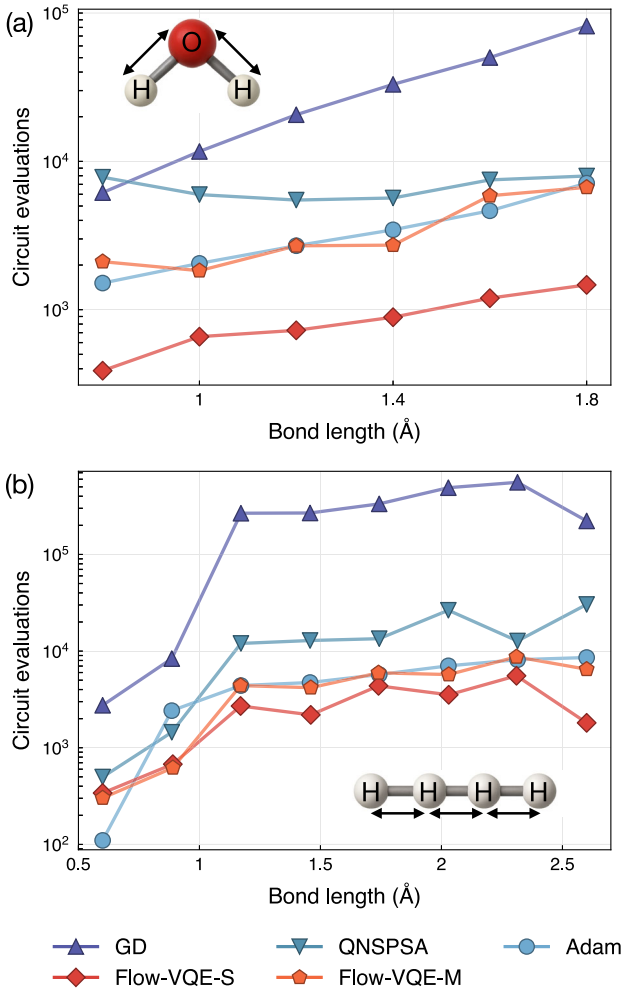

Fig. 2 | Number of quantum circuit evaluations required to achieve computational accuracy with different optimizers. a H_2O: optimization over six uniformly spaced bond lengths in [0.8, 1.8]Å. **b** H_4: optimization over eight bond lengths in [0.6, 2.6]Å. The arrows in each molecular diagram represent changing bond lengths. All baseline optimizers use a learning rate of $\eta = 0.02$.

Generating potential-energy surfaces

We illustrate the generative capability of Flow-VQE-M by analyzing its performance in producing approximate variational parameters for the ansätze along the potential energy surfaces (PESs) of H_2O and H_4 molecular systems. For experimental evaluation, we train Flow-VQE-M on six molecular geometries of H_2O and eight of H_4. After 5000 training epochs (equivalent to 10,000 circuit evaluations per molecule), we evaluate performance by uniformly selecting 50 molecular structures across the bond-length domain. For each structure, we sample 16 parameter vectors from the flow model and assess both the minimum and mean energies as primary metrics.

The results presented in Fig. 3, which include the PESs of the representative systems H_2O and H_4, substantiate the high quality of parameters generated by Flow-VQE-M. When evaluated in the corresponding ansätze, these generated parameters yield energy expectation values closely align with the exact solutions, significantly outperforming Hartree–Fock (HF) baselines. Overall, energy errors increase in stretched bond-length regions due to strong correlation effects[60] and the limited expressivity of the employed ansätze. Nevertheless, Flow-VQE-M maintains high-quality results, demonstrating robust performance even under these challenging conditions. Notably, most generated points already achieve computational accuracy, leading to high-quality PESs. These points can therefore serve as strong initializations for VQE, facilitating rapid convergence to the desired precision.

Warm-start post-training

We select high-error samples from the previous example of H_2O and H_4 for post-training with the Adam optimizer, comparing convergence behavior under different learning rates. Concurrently, we demonstrate the performance gap between Flow-VQE-M warm-start and HF initialization.

The results, presented in Table 1, show that across all settings, Flow-VQE-M consistently demonstrates faster convergence and lower energy errors. In practice, learning rates η are typically chosen between 0.01 and 0.1; for our benchmarks we set $\eta = 0.02$. At this rate, Flow-VQE-M reduces the number of circuit evaluations required to reach computational accuracy by over 27-fold for H_2O and more than 11-fold for H_4. Moreover, this advantage grows even more pronounced at smaller η. For instance, at $\eta = 0.001$, Flow-VQE-M cuts the required evaluations by more than 50-fold for H_2O and over 36-fold for H_4, while achieving significantly lower minimum energy errors. Complete optimization trajectories corresponding to the results in Table 1 are further provided in Figs. S1 and S2, in Section III of the *Supplementary Information*. These results underscore the effectiveness of Flow-VQE-M as a robust warm-start strategy that substantially enhances sample efficiency and convergence precision in the post-training stage, even under conservative optimization settings.

Estimate of cost advantage

Given the probabilistic nature of Flow-VQE's inference mechanism, deriving a general analytical expression for cost evaluation is infeasible and inherently dependent on implementation details. To provide a representative estimate, we conduct empirical case studies on the NH_3 and C_6H_6 molecules under lightweight training, where both the number of iterations and training configurations are deliberately limited to avoid generating overly high-quality parameters too early and to retain room for warm-started optimization.

Specifically, we train Flow-VQE-M on four configurations for each molecule. For NH_3, the training points correspond to nitrogen displacements $\{-0.5, -0.17, 0.17, 0.5\}$Å along the inversion path, where 0Å denotes the geometric center of the hydrogen plane; each point involves 3000 circuit evaluations (12,000 in total). For C_6H_6, the configurations involve stretching a single C–H bond relative to its equilibrium length (1.084Å) by $\{-0.3, 0.0, 0.3, 0.6\}$Å, with 6000 evaluations per point (24,000 in total). The generated PESs for both molecules are available in Fig. S3 in Section IV of the *Supplementary Information*.

Additionally, we benchmark Flow-VQE-M against the conventional parameter-transfer method in VQE (see the *Methods* section). In our PT protocol, we choose the 0Å configuration as the reuse point for each molecule and optimize it via Flow-VQE-S, using 2000 circuit evaluations for NH_3 and 6000 for C_6H_6 (sufficient to reach computational accuracy), thus defining the pre-training cost for each system.

We use the number of circuit evaluations required to reach computational accuracy as the cost metric. As shown earlier in Fig. 2, Flow-VQE-M incurs a number of evaluations during training comparable to that of standard VQE with Adam. This suggests that, in the systems tested here, Flow-VQE-M offers a clear quantum resource advantage when training points are also required for a task, due to its additional generative capability. In what follows, we focus exclusively on the cost evaluation for unseen test configurations.

Let C_{pre} denote the total pre-training cost, and $\bar{C}_{\text{post}}$ the average post-training cost, per test point. Although individual VQE procedures require significantly different optimization efforts, we approximate the cost growth using average values for simplicity. Accordingly, the total cost of using a typical warm-start method (such as Flow-VQE or PT) for n_{test} test points is $C_{\text{total}} = C_{\text{pre}} + \bar{C}_{\text{post}} \cdot n_{\text{test}}$. In contrast, standard VQE incurs a cost of approximately $\bar{C}_{\text{VQE}} \cdot n_{\text{test}}$. Given that $\bar{C}_{\text{post}} < \bar{C}_{\text{VQE}}$, the gentler cost scaling of warm-start methods highlights their growing advantage as n_{test} increases.

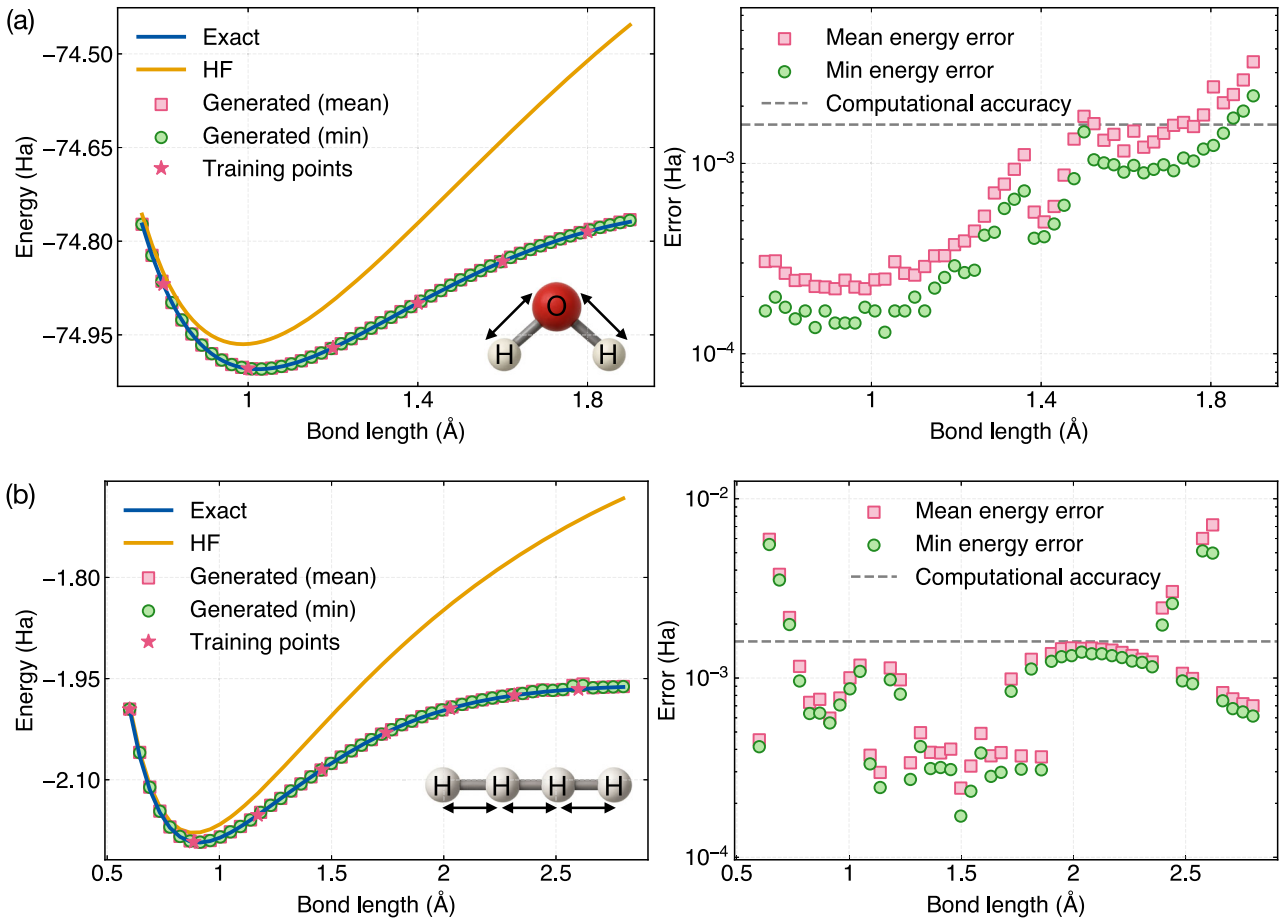

Fig. 3 | Potential-energy surfaces (left) and corresponding errors (right) evaluated using parameters generated by Flow-VQE-M. Generation is performed on 50 uniformly spaced test points in [0.75, 1.9]Å for (**a**) H_2O and [0.6, 2.8]Å for (**b**) H_4. The training points are consistent with those used in Fig. 2.

Table 1 | Post-training comparison of Flow-VQE-M (FVM) and HF initializations for H_2O at a bond length of 1.90Å and H_4 at 2.58Å under different learning rates (η)

Molecule	η	$N_{c.a.}$		ΔE_{min}	
		FVM	HF	FVM	HF
H_2O	0.02	**324**	8748	$\mathbf{5.932 \times 10^{-4}}$	5.932×10^{-4}
	0.005	**432**	30672	$\mathbf{5.935 \times 10^{-4}}$	6.000×10^{-4}
	0.001	**1944**	102276	$\mathbf{5.941 \times 10^{-4}}$	1.033×10^{-3}
H_4	0.02	**770**	8910	$\mathbf{5.205 \times 10^{-8}}$	2.489×10^{-7}
	0.005	**440**	16720	$\mathbf{1.319 \times 10^{-9}}$	3.202×10^{-9}
	0.001	**1980**	72270	$\mathbf{1.053 \times 10^{-10}}$	2.579×10^{-4}

After initialization, we perform standard VQE by running 1000 iterations of the Adam optimizer. Reported metrics include the number of circuit evaluations required to reach computational accuracy ($N_{c.a.}$) and the minimum energy error achieved over iterations (ΔE_{min}). Bold numbers highlight the better results.

Figure 4 compares the training costs using standard VQE, PT, and Flow-VQE-M warm-start strategies for NH_3 and C_6H_6, along with corresponding scaling estimates. For NH_3, standard VQE requires an average of $\bar{C}_{VQE} = 5,265$ circuit evaluations per test point, reduced to $\bar{C}_{post} = 2,527$ by Flow-VQE-M and $\bar{C}_{post} = 4,165$ by PT. With pre-training costs of $C_{pre} = 12,000$ for Flow-VQE-M and $C_{pre} = 2,000$ for PT, Flow-VQE-M achieves a net cost advantage over standard VQE beyond five test points, and over PT beyond six points. For C_6H_6, the respective values are $\bar{C}_{VQE} = 10,787$, $\bar{C}_{post} = 2,153$ (Flow-VQE-M), and $\bar{C}_{post} = 5,873$ (PT), with corresponding pre-training costs of 24, 000 and 6, 000. Here, Flow-VQE-M outperforms standard VQE after three test points and PT after five points. These estimates are instance-dependent and not intended as universal benchmarks, but they illustrate the practical advantages of Flow-VQE-M in scenarios requiring repeated evaluations across chemical space.

We attribute these savings to Flow-VQE-M's ability to embed problems into a latent space where similarity captures the closeness of the optimal variational parameters, a shift from conventional PT. Unlike PT, which relies on heuristic reuse based on geometrical proximity, Flow-VQE-M conditions its parameter distribution on rich, task-specific embeddings spanning diverse molecular configurations. While these embeddings are not explicitly engineered to encode structural or electronic features, the model benefits from exposure to a broad training distribution, allowing it to capture latent task similarities. Notably, advances in molecular descriptors[61,62] provide a fertile ground for designing more informative embeddings.

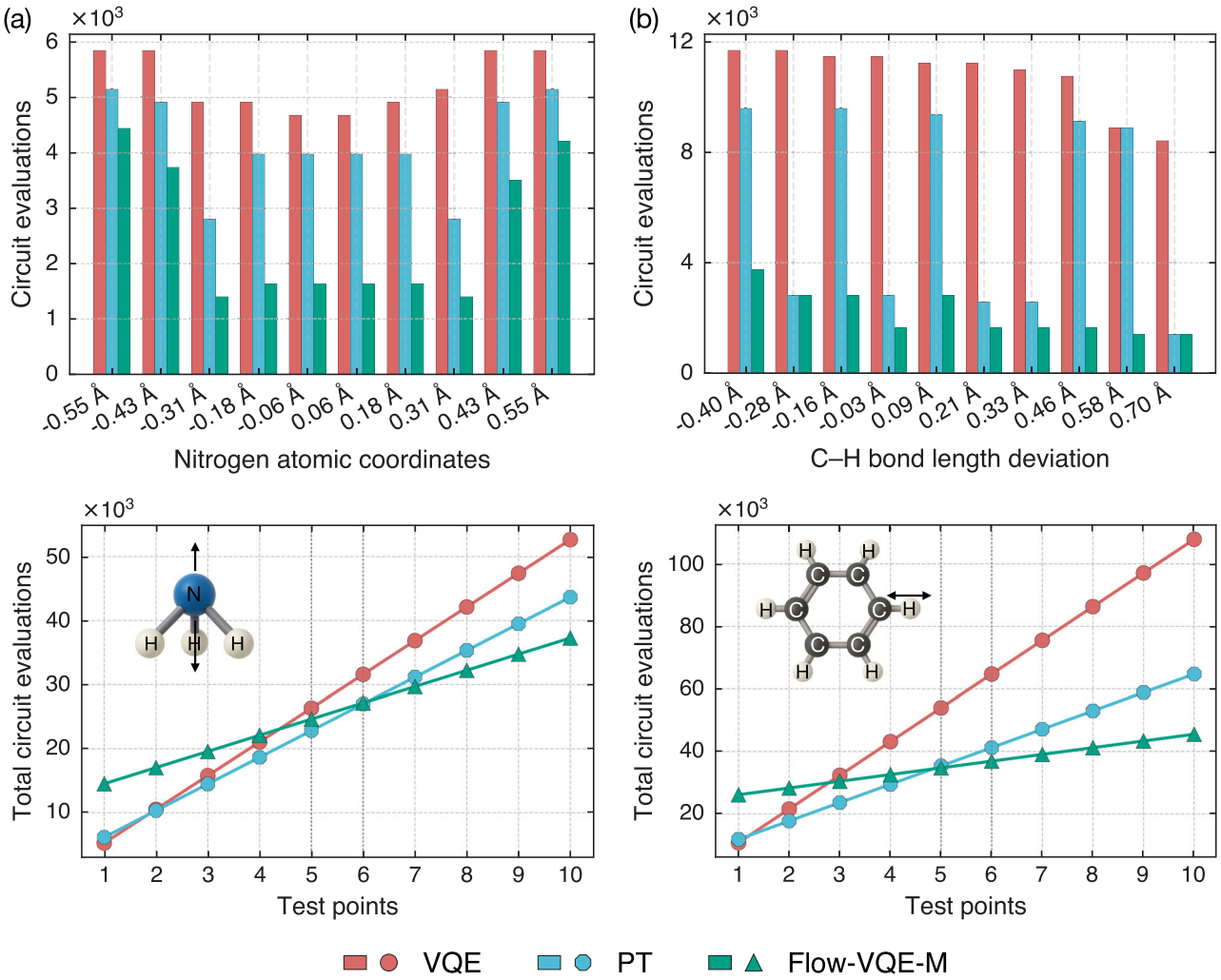

Fig. 4 | Circuit evaluation costs for achieving computational accuracy under HF, PT, and Flow-VQE-M initializations. For (**a**) NH_3 and (**b**) C_6H_6, the upper panels show circuit evaluation counts across different molecular geometries; the lower panels present approximate average scaling as the number of test points increases, where the intercept indicates the one-time pre-training cost for each strategy, and the slope reflects the average post-training cost per additional test point. The arrows in each molecular diagram represent modes of atomic displacement. Optimization is performed using the Adam optimizer with a learning rate of $\eta = 0.02$. Note that evaluation counts are recorded as integer multiples of the number of variational parameters; hence, circuit evaluations with similar initialization contributions differ by less than a full iteration, resulting in bars of identical height.

Nevertheless, even in its current form, Flow-VQE-M is capable of generalizing beyond geometric similarity. This is exemplified in the case of C_6H_6, where reused parameters from the 0 Å configuration yields unexpectedly fast convergence for distant configurations, suggesting the presence of deeper task correlations not captured by geometry alone. In contrast to the manual identification of transferable patterns required by PT, Flow-VQE-M learns how contextual embeddings modulate the parameter distribution, enabling more consistent and scalable transfer across chemically diverse settings.

Discussion

We propose Flow-VQE, a probabilistic approach that recasts variational quantum optimization as a generative modeling task. By leveraging preference-based training, Flow-VQE eliminates the need for quantum gradient estimation and progressively refines its sampling distribution to generate high-quality variational parameters. Our results on representative molecular systems show that Flow-VQE enables lower quantum resource costs during optimization and supports transferable warm starts by extracting generalizable features from diverse molecular structures. When generating hundreds of variational parameters, the classical training component remains easily tractable, positioning Flow-VQE as a practical and resource-efficient approach for variational quantum simulation.

Flow-VQE, as a black-box optimizer for VQE, may be less precise than gradient-based methods in smooth, differentiable energy landscapes, where local curvature can be finely exploited. However, its gradient-free nature may offer improved robustness for noisy VQE—a hypothesis that remains to be systematically validated. In addition, the preference-based optimization in Flow-VQE tends to concentrate probability density over time, which can reduce sample diversity and limit exploratory behavior. Mitigating this limitation requires further investigation into entropy-regularized objectives or diversity-promoting preference selection strategies[63,64].

While our numerical experiments extend only to 12-qubit systems and 117 variational parameters, several features of Flow-VQE promise broad quantum-resource savings, even when scaling to larger molecules. On the classical side, the computational overhead in modern normalizing-flow architectures scales linearly in data dimensions, keeping training and inference practical even as the number of variational parameters (d) reaches tens of thousands[65]. Expanding on this, our intuition is that the primary bottleneck for scaling Flow-VQE will not be the classical generative model, but rather the intrinsic properties of the VQE energy landscape itself.

Recent research shows that overparametrization induces a 'computational phase transition' in the optimization landscape: beyond a critical parameter threshold, spurious local minima vanish and trainability improves markedly[66]. Flow-VQE is poised to leverage this regime, as its per-epoch quantum cost scales with the batch size B rather than the parameter dimension d. Consequently, it handles highly parameterized circuits efficiently, reaping a landscape with fewer traps without the concern of increased gradient-estimation cost.

Barren plateaus, however, pose a more fundamental challenge for variational methods, as the exponentially flat landscape causes the feedback signal to vanish. While not a complete remedy, Flow-VQE mitigates this pathology. Beyond its inherent warm-start advantage, Flow-VQE offers a practical mitigation strategy by adaptively scheduling computational resources, chiefly the batch size B. Although we use $B = 2$ throughout this study, the batch size (and the shot budget[67]) can be dynamically adjusted to trade cost for signal quality. In an exploratory phase, one uses a small B to sweep the landscape cheaply and globally. As the parameters enter the ground-state basin, B are increased to reduce estimator variance and sharpen energy differences. Because this policy is independent of the variational dimension d, it bypasses the quartic parameter growth typical of chemically motivated ansätze[68] and the even steeper scaling of many hardware-efficient ansätze[69].

Recent work links barren-plateau avoidance to classically simulable polynomial subspaces and calls for rethinking the role of variational computation[70]. In this spirit, we can view Flow-VQE as a natural front end to quantum subspace methods[71–73]. A trained flow model can, in essentially constant time per sample, generate many diverse parameter sets whose associated trial states have high ground-state overlap. These non-orthogonal yet linearly independent states span a compact subspace, thereby reducing the effective dimensionality of the ensuing (generalized) eigenvalue problem. In this reframing, Flow-VQE acts as an auxiliary sampler to shift the objective from "finding one optimum" to "sampling a high-quality basis", echoing calls for quantum-enhanced classical simulation.

In our present experiments, Flow-VQE warm-starts optimization only across different geometries of the same molecule; transfer across molecular species remains outside our current scope. We anticipate that cross-species generalization will be feasible when paired with explicit molecular representation learning to distill transferable and chemistry-aware features[61,62] to distill the most relevant features. Furthermore, when approximate reference data are available — such as historically calculated inaccurate data — the flow model can be pre-trained entirely with classical computation on this corpus, and then fine-tuned through its standard interaction with the quantum device. This two-stage regimen shifts the bulk of optimization off-hardware and making larger systems economically accessible. Capitalizing on these gains, our immediate priority is to scale Flow-VQE to systems exceeding 20 qubits, thereby furnishing a decisive, empirical assessment of its ability to treat larger, more chemically realistic molecules.

We want to emphasize that the most stubborn training obstacles in VQE stem largely from the ansatz structure itself[13,74]. Genuine progress requires Flow-VQE and ansatz designs to advance together. To this end, Flow-VQE can be extended beyond fixed templates to generate circuits directly as symbolic gate sequences. By treating gate selection and ordering as a unified 'operator space', this approach enables end-to-end optimization and provides a principled foundation for quantum architecture search[75]. Prior studies have shown that operator orderings crucially affect ansatz's expressivity and accuracy[76–79]. Leveraging discrete normalizing flows[80,81] allows Flow-VQE to efficiently search such combinatorial space. This facilitates the automated discovery of expressive, Hamiltonian-aware circuits and sets the stage for multimodal quantum circuit generation[82].

Although the present work focuses on VQE, the framework extends naturally to other variational quantum algorithms—such as the quantum approximate optimization algorithm[83]—whenever a well-defined loss function exists. Overall, this work highlights the promise of integrating generative modeling with quantum optimization to expand the algorithmic design space and advance hybrid quantum-classical computing.

Methods

Variational quantum eigensolver

The VQE algorithm is a hybrid quantum-classical optimization framework designed to approximate the ground-state energies of complex quantum systems[6,9,84]. Given a Hamiltonian operator $\widehat{H}$ defined on an n-qudit (D-level quantum system) Hilbert space $\mathcal{H} \in \mathbb{C}^{D^n \times D^n}$, the VQE seeks to determine its minimal eigenvalue E_0, defined as

$$E_0 = \min_{\psi \in \mathcal{H}, |\psi|=1} \langle \psi | \widehat{H} | \psi \rangle. \tag{3}$$

Although the VQE can in principle be applied to arbitrary multilevel quantum systems, in this work, we focus on the case $D = 2$, i.e., standard qubit circuits. To do so, the VQE involves preparing a trial quantum state using a variational ansatz $U(\boldsymbol{\theta})$:

$$\psi(\boldsymbol{\theta}) = U(\boldsymbol{\theta})|0\rangle^{\otimes n}, \tag{4}$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d) \in \mathbb{R}^d$ is a vector of d trainable parameters, and $U(\boldsymbol{\theta}) : \mathbb{R}^d \to \mathcal{U}(2^n)$ is a smooth mapping from the parameter space to the unitary group acting on $\mathcal{H}$. This formulation rephrases the eigenvalue problem as a variational optimization problem in $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{\theta}), \tag{5}$$

where $\mathcal{L}(\boldsymbol{\theta}) := \langle \psi(\boldsymbol{\theta}) | \widehat{H} | \psi(\boldsymbol{\theta}) \rangle$ is the variational energy functional (expected energy of the trial state).

The optimization proceeds as a quantum-classical loop, where a classical optimizer iteratively updates parameters based on an objective function evaluated on a quantum device. The parameter update strategy generally falls into two categories. Gradient-based methods such as gradient descent, while common, are computationally expensive; estimating quantum gradients via techniques like the parameter-shift rule[55] incurs a cost of $\mathcal{O}(d)$ per iteration, making it substantially more expensive than classical backpropagation, which computes all parameter derivatives in a single pass. Alternatively, gradient-free methods can be developed under different design principles and have received significant attention[18,19,85–87]. By avoiding explicit gradient computations, these approaches can potentially be more practical in scenarios where evaluating quantum gradients is costly or infeasible, such as under hardware noise or in the presence of non-differentiable objectives.

Normalizing flows

Normalizing flows (NFs)[49,50,52] are a class of generative models predicated on the principle of invertible maps between probability densities. To see how NFs work, consider a latent vector $\mathbf{z} \in \mathbb{R}^d$ with an associated simple base distribution $p_Z(\mathbf{z})$, typically a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The fundamental objective is to construct a bijective and differentiable (*diffeomorphic*) map $f_{\boldsymbol{\tau}} : \mathbb{R}^d \to \mathbb{R}^d$ with model parameters $\boldsymbol{\tau} \in \mathbb{R}^p$, which transforms $p_Z(\mathbf{z})$ into a modeling distribution $p_X(\mathbf{x})$ that approximates a complex, unknown data distribution.

The transformation is governed by the change-of-variables formula:

$$p_X(\mathbf{x}) = p_Z(\mathbf{z}) \left| \det \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right| = p_Z(f_{\boldsymbol{\tau}}^{-1}[\mathbf{x}]) |\det \mathbf{J}(f_{\boldsymbol{\tau}}^{-1}[\mathbf{x}])|, \tag{6}$$

where $\mathbf{J}(f_{\boldsymbol{\tau}}^{-1}[\mathbf{x}])$ denotes the Jacobian matrix of the inverse mapping $f_{\boldsymbol{\tau}}^{-1}$ evaluated at point $\mathbf{x}$. To model complex distributions tractably, NFs employ compositional transformations: $f_{\boldsymbol{\tau}} = f_{\boldsymbol{\tau}_K} \circ f_{\boldsymbol{\tau}_{K-1}} \circ \cdots \circ f_{\boldsymbol{\tau}_1}$, where K is the number of transformations.

In the training phase, NFs apply the inverse transformation to map each data point from the empirical data set $\mathcal{D} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ into the latent space via $\mathbf{z} = f_{\boldsymbol{\tau}}^{-1}(\mathbf{x})$. The corresponding log-likelihood can be

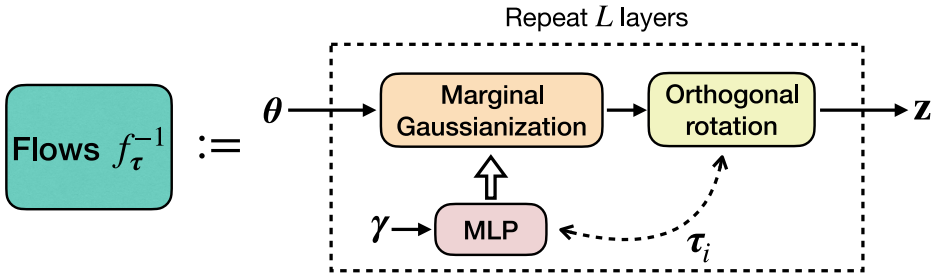

Fig. 5 | Conditional normalizing flow architecture used in this work. During training, the inverse mapping ($\boldsymbol{\theta} \to \boldsymbol{z}$) is realized by L repeated Gaussianization flow layers, each comprising a marginal Gaussianization block parameterized by an MLP conditioned on the molecular context $\boldsymbol{\gamma}$, followed by a learnable orthogonal rotation. All trainable parameters $\boldsymbol{\tau}$ reside in the MLPs and the rotations.

evaluated by

$$\log p_X(\mathbf{x};\boldsymbol{\tau}) = \log p_Z(\mathbf{z}) + \sum_{j=1}^{K} \log\left|\det \mathbf{J}(f_{\tau_j}^{-1}[\mathbf{x}])\right|. \tag{7}$$

The model parameters $\boldsymbol{\tau}$ are trained via maximum likelihood estimation over the target data:

$$\mathcal{L}(\boldsymbol{\tau};\mathcal{D}) = -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\log p_X(\mathbf{x};\boldsymbol{\tau})]. \tag{8}$$

At inference (generation) time, we sample $\mathbf{z} \sim p_Z(\mathbf{z})$ from the base distribution and execute the forward transformation $\mathbf{x} = f_{\boldsymbol{\tau}}(\mathbf{z})$. For compositional flows, this entails a sequential application of transformations:

$$\mathbf{z} \sim p_Z(\mathbf{z}) \xrightarrow{f_{\tau_1}} \mathbf{h}^{(1)} \xrightarrow{f_{\tau_2}} \mathbf{h}^{(2)} \xrightarrow{\cdots} \xrightarrow{f_{\tau_K}} \mathbf{x}, \tag{9}$$

where $\mathbf{h}^{(j)}$ denotes the intermediate output after the j-th transformation.

The compositional design of NFs makes the model highly expressive while keeping each Jacobian tractable. Nonetheless, log-likelihood evaluation remains computationally intensive, since computing Jacobian determinants at each layer scales as $\mathcal{O}(Kd^3)$ for general diffeomorphic maps. To alleviate this cost, modern flow architectures often employ sparse Jacobians that permit more efficient determinant algorithms, such as affine coupling flows[88,89], autoregressive flows[90,91], and spline-based flows[92,93]. In contrast, the generation process exhibits remarkable efficiency, with $\mathcal{O}(Kd)$ complexity, as it involves mere function evaluations without determinant calculations, enabling rapid sample synthesis.

Gaussianization flows

We use Gaussianization flows[94] as the backbone NF architecture in this work. Gaussianization flows support efficient likelihood evaluation for fast training, enable rapid sampling, exhibit greater robustness to data transformations, and generalize more effectively on small datasets than other mainstream flow models[94].

Gaussianization flows construct a structured diffeomorphism on $\mathbb{R}^d$, built upon the rotation-based iterative Gaussianization[95]. Each layer alternates (i) marginal Gaussianization, where dimension-wise monotone maps push forward empirical marginals to a univariate Gaussian distribution, and (ii) orthogonal rotations, which redistribute dependencies across coordinates while preserving Gaussian marginals. Formally, the bijective mapping is constructed through L alternating layers:

$$f_{\boldsymbol{\tau}} = \Psi_{\boldsymbol{\tau}_L} \circ R_L \circ \cdots \circ \Psi_{\boldsymbol{\tau}_1} \circ R_1. \tag{10}$$

Each marginal Gaussianization layer $\Psi_{\boldsymbol{\tau}_i}$ acts independently on the coordinates of $\mathbf{x}$,

$$\Psi_{\boldsymbol{\tau}_i}(\mathbf{x}) = \left(\Psi_{\boldsymbol{\tau}_i}^{(1)}(x^{(1)}), \ldots, \Psi_{\boldsymbol{\tau}_i}^{(d)}(x^{(d)})\right)^{\top}, \tag{11}$$

where each dimension-wise component is defined as

$$\Psi_{\boldsymbol{\tau}_i}^{(k)}(x^{(k)}) = \Phi^{-1}\left[F_{\boldsymbol{\tau}_i}^{(k)}\left(x^{(k)}\right)\right],\ k = 1, \ldots, d. \tag{12}$$

Here, Φ denotes the cumulative distribution function (CDF) of the standard Gaussian, and Φ^{-1} is its inverse. The intermediate mapping is parameterized as

$$F_{\boldsymbol{\tau}_i}^{(k)}(x^{(k)}) = \frac{1}{P}\sum_{j=1}^{P} \Phi\left[\exp\left(a_{ij}^{(k)}\right)x^{(k)} + b_{ij}^{(k)}\right], \tag{13}$$

a combination of P rescaled and shifted Gaussian CDFs with learnable parameters $\left\{a_j^{(k)}, b_j^{(k)}\right\}_{j=1}^{P}$. To remove dependencies across coordinates, each marginal Gaussianization layer is followed by an orthogonal rotation

$$R_i = \exp\left(A_i - A_i^{\top}\right), \tag{14}$$

where $A_i \in \mathbb{R}^{d\times d}$ is a learnable parameter matrix. The skew-symmetric form $A_i - A_i^{\top}$ guarantees that the exponential map lies in the orthogonal group with unit determinant, ensuring invertibility and stability throughout training. These parameterizations of $F_{\boldsymbol{\tau}_i}^{(k)}$ and R_i are not unique, but adopted here for simplicity and stability.

Conditional architecture

Here, we employ conditional Gaussianization flows, making the invertible blocks explicitly aware of the molecular context, as illustrated in Fig. 5. To achieve this, we introduce conditioner networks implemented as multilayer perceptrons (MLPs). The overall set of trainable parameters $\boldsymbol{\tau}$ comprises two components: (i) the parameters of the MLPs themselves, and (ii) the learnable matrices that specify the orthogonal rotations. During the forward pass, the MLP takes $\boldsymbol{\gamma}$ as input and generates the layer-specific parameters that configure the marginal Gaussianization maps. This division of labor allows MLPs to specifically adapt the marginal distributions based on the molecular input, while the learned rotations provide a stable, problem-agnostic mechanism to mix coordinates. Once this conditional mapping is defined, the training process takes the candidate parameters as input, mapping them to the latent space via the inverse transformation $f_{\boldsymbol{\tau}}^{-1}(\boldsymbol{\theta};\boldsymbol{\gamma})$.

Parameter transfer in the variational quantum eigensolver

Traditional parameter transfer (PT) accelerates the VQE algorithm by reusing optimized parameters from structurally adjacent problem instances[32,35]. Let $\boldsymbol{\theta}_k^*$ denote the optimized variational parameters for a VQE task $\mathcal{T}_k$. The PT strategy initializes the subsequent task $\mathcal{T}_{k+1}$ using $\boldsymbol{\theta}_k^*$ as a warm start: $\boldsymbol{\theta}_{k+1}^{(0)} := \boldsymbol{\theta}_k^*$. The selection of transferable task pairs typically relies on certain heuristics that aim to preserve underlying similarity between instances. One example is using small geometric perturbations[32], such as those along a potential energy surface, where configurations differ by a small Euclidean distance, e.g., $\|\boldsymbol{R}_{k+1} - \boldsymbol{R}_k\| \sim 0.1$Å. In the absence of further task-specific guidance, such perturbations are commonly assumed

—with high empirical confidence—to keep $\boldsymbol{\theta}_k^*$ within the attraction basin of $\mathcal{T}_{k+1}$, thereby enabling efficient parameter reuse.

Implementation details

Molecular Hamiltonians are constructed with `OpenFermion`[96], based on one- and two-electron integrals obtained in the chosen basis sets (STO-3G for H_2O, NH_3, and C_6H_6; cc-pVDZ for H_4). Active spaces are chosen as (4e, 4o) for H_4, (6e, 5o) for H_2O, and (6e, 6o) for NH_3 and C_6H_6. The fermionic Hamiltonians are mapped to qubit Hamiltonians using the Jordan–Wigner transformation, resulting in the following qubit requirements: 5 qubits for H_4 (after applying $\mathbb{Z}_2$ symmetry tapering[97]), 10 qubits for H_2O, and 12 qubits for both NH_3 and C_6H_6.

Quantum-circuit simulations are carried out in `PennyLane`[98], using a 10-layer hardware-efficient R_Y-linear ansatz[99] for H_4 (55 parameters) and a Givens-based singles and doubles (GSD) ansatz[100,101] for the other molecules, resulting in 54 parameters for H_2O and 117 parameters for both NH_3 and C_6H_6. Further details of the ansatz structures are provided in Section V and Figure S5 of the *Supplementary Information*.

Gaussianization flows are implemented in `PyTorch`[102] with `Zuko`[103]. Each flow layer employs a mixture of 32 Gaussian CDFs. Each Gaussianization transformation is parameterized by a multi-layer perceptron comprising three hidden layers with 256 units each and exponential linear unit (ELU) activation functions[104]. The base distribution uses a multivariate normal $\mathcal{N}(\boldsymbol{\mu} = \boldsymbol{0}, \boldsymbol{\Sigma} = 0.01 \cdot \boldsymbol{I})$, ensuring that initial samples remain close to the HF reference point. We use seven flow layers for H_4 and ten layers for all other systems in Flow-VQE-S. For Flow-VQE-M, we employ 20 flow layers for each molecule. To guide conditional generation, we construct the context vector $\boldsymbol{\gamma}$ for each molecular geometry by passing its coefficients of Pauli Hamiltonian.

We train the flow models using the Adam optimizer[105] on a single Nvidia A40 GPU (48 GB) or on an Apple M4 Pro with 24 GB unified memory, using a learning rate and weight decay of 0.0001, empirically selected from small-scale simulations. To enhance exploration and reduce overfitting under limited sample regimes, we add zero-mean Gaussian noise with variance 0.001 to the winning parameters before evaluating sample likelihoods during Flow-VQE training. In preference-based optimization, all experiments use a batch size of $B = 2$ per sampling and a buffer size of $M = 2$, retaining at most two winning samples per molecular configuration. This deliberately small-scale setup demonstrates the sample efficiency of our method in low-budget scenarios.

Data availability

All data and trained models supporting the findings of this study are available at https://doi.org/10.5281/zenodo.17278313.

Code availability

The code used to perform the numerical simulations is available at https://github.com/olsson-group/Flow-VQE.

Received: 15 July 2025; Accepted: 4 December 2025;
Published online: 23 December 2025

References

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
2. Kim, Y. et al. Evidence for the utility of quantum computing before fault tolerance. *Nature* **618**, 500 (2023).
3. Bluvstein, D. et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **626**, 58–65 (2024).
4. Acharya, R. et al. Quantum error correction below the surface code threshold. *Nature* **638**, 920–926 (2025).
5. Montanaro, A. Quantum algorithms: an overview. *npj Quantum Inf.* **2**, 15023 (2016).
6. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
7. Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L. & Coles, P. J. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci.* **2**, 567 (2022).
8. Dalzell, A. M. et al. Quantum Algorithms: A Survey of Applications and End-to-End Complexities (Cambridge University Press, 2025).
9. Peruzzo, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).
10. Bittel, L. & Kliesch, M. Training variational quantum algorithms is NP-hard. *Phys. Rev. Lett.* **127**, 120502 (2021).
11. Zimborás, Z. et al. Myths around quantum computation before full fault tolerance: What no-go theorems rule out and what they don't. Preprint at https://doi.org/10.48550/arXiv.2501.05694 (2025).
12. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812 (2018).
13. Larocca, M. et al. Barren plateaus in variational quantum computing. *Nat. Rev. Phys.* **7**, 174–189 (2025).
14. Anschuetz, E. R. & Kiani, B. T. Quantum variational algorithms are swamped with traps. *Nat. Commun.* **13**, 7760 (2022).
15. Sweke, R. et al. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum* **4**, 314 (2020).
16. Stokes, J., Izaac, J., Killoran, N. & Carleo, G. Quantum natural gradient. *Quantum* **4**, 269 (2020).
17. Fitzek, D., Jonsson, R. S., Dobrautz, W. & Schäfer, C. Optimizing variational quantum algorithms with qBang: efficiently interweaving metric and momentum to navigate flat energy landscapes. *Quantum* **8**, 1313 (2024).
18. Gacon, J., Zoufal, C., Carleo, G. & Woerner, S. Simultaneous perturbation stochastic approximation of the quantum fisher information. *Quantum* **5**, 567 (2021).
19. Wanner, M., Jonasson, J., Carlsson, E. & Dubhashi, D. Variational quantum optimization with continuous bandits. Preprint at https://doi.org/10.48550/arXiv.2502.04021 (2025).
20. Tubman, N. M. et al. Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices. Preprint at https://doi.org/10.48550/arXiv.1809.05523 (2018).
21. Fomichev, S. et al. Initial state preparation for quantum chemistry on quantum computers. *PRX Quantum* **5**, 040339 (2024).
22. Truger, F., Barzen, J., Leymann, F. & Obst, J. Warm-Starting the VQE with Approximate Complex Amplitude Encoding. In *Proc. of the 1st International Conference on Quantum Software*, 15–26 (2025).
23. Mitarai, K., Suzuki, Y., Mizukami, W., Nakagawa, Y. O. & Fujii, K. Quadratic Clifford expansion for efficient benchmarking and initialization of variational quantum algorithms. *Phys. Rev. Res.* **4**, 033012 (2022).
24. Egger, D. J., Mareček, J. & Woerner, S. Warm-starting quantum optimization. *Quantum* **5**, 479 (2021).
25. Huggins, W., Patil, P., Mitchell, B., Whaley, K. B. & Stoudenmire, E. M. Towards quantum machine learning with tensor networks. *Quantum Sci. Technol.* **4**, 024001 (2019).
26. Dborin, J., Barratt, F., Wimalaweera, V., Wright, L. & Green, A. G. Matrix product state pre-training for quantum machine learning. *Quantum Sci. Technol.* **7**, 035014 (2022).
27. Rudolph, M. S. et al. Synergistic pretraining of parametrized quantum circuits via tensor networks. *Nat. Commun.* **14**, 8367 (2023).
28. Lee, X., Saito, Y., Cai, D. & Asai, N. Parameters fixing strategy for quantum approximate optimization algorithm. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 10–16 (2021).
29. Puig, R., Drudis, M., Thanasilp, S. & Holmes, Z. Variational quantum simulation: a case study for understanding warm starts. *PRX Quantum* **6**, 010317 (2025).

30. Skolik, A., McClean, J. R., Mohseni, M., Van Der Smagt, P. & Leib, M. Layerwise learning for quantum neural networks. *Quantum Mach. Intell.* **3**, 5 (2021).
31. Grant, E., Wossnig, L., Ostaszewski, M. & Benedetti, M. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum* **3**, 214 (2019).
32. Skogh, M., Leinonen, O., Lolur, P. & Rahm, M. Accelerating variational quantum eigensolver convergence using parameter transfer. *Electron. Struct.* **5**, 035002 (2023).
33. Lyngfelt, I. & García-Álvarez, L. Symmetry-informed transferability of optimal parameters in the Quantum Approximate Optimization Algorithm. *Phys. Rev. A* **111**, 022418 (2025).
34. Mele, A. A., Mbeng, G. B., Santoro, G. E., Collura, M. & Torta, P. Avoiding barren plateaus via transferability of smooth solutions in a Hamiltonian variational ansatz. *Phys. Rev. A* **106**, L060401 (2022).
35. Rohe, T. et al. Accelerated VQE: Parameter recycling for similar recurring problem instances. In *International Conference on Innovations for Community Services*, 63–78 (2025).
36. Tao, Y. et al. Exploring accurate potential energy surfaces via integrating Variational Quantum Eigensolver with machine learning. *J. Phys. Chem. Lett.* **13**, 6420–6426 (2022).
37. Yao, Q. et al. Machine learning accelerates precise excited-state potential energy surface calculations on a quantum computer. *J. Phys. Chem. Lett.* **15**, 7061–7068 (2024).
38. Ceroni, J. et al. Generating approximate ground states of molecules using quantum machine learning. Preprint at https://doi.org/10.48550/arXiv.2210.05489 (2023).
39. Zhang, S. et al. Diffusion-enhanced optimization of variational quantum eigensolver for general hamiltonians. Preprint at https://doi.org/10.48550/arXiv.2501.05666 (2025).
40. Miao, J., Hsieh, C.-Y. & Zhang, S.-X. Neural-network-encoded variational quantum algorithms. *Phys. Rev. Appl.* **21**, 014053 (2024).
41. Mesman, K., Tang, Y., Moller, M., Chen, B. & Feld, S. NN-AE-VQE: Neural network parameter prediction on autoencoded variational quantum eigensolvers. Preprint at https://doi.org/10.48550/arXiv.2411.15667 (2025).
42. Rivera-Dean, J., Huembeli, P., Acín, A. & Bowles, J. Avoiding local minima in variational quantum algorithms with neural networks. Preprint at https://doi.org/10.48550/arXiv.2104.02955 (2021).
43. Zhang, S.-X. et al. Variational quantum-neural hybrid eigensolver. *Phys. Rev. Lett.* **128**, 120502 (2022).
44. Verdon, G. et al. Learning to learn with quantum neural networks via classical neural networks. Preprint at https://doi.org/10.48550/arXiv.1907.05415 (2019).
45. Cervera-Lierta, A., Kottmann, J. S. & Aspuru-Guzik, A. Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation. *PRX Quantum* **2**, 020329 (2021).
46. Sauvage, F. et al. FLIP: A flexible initializer for arbitrarily-sized parametrized quantum circuits. Preprint at https://doi.org/10.48550/arXiv.2103.08572 (2021).
47. Kamata, Y., Tran, Q. H., Endo, Y. & Oshima, H. Molecular quantum transformer. Preprint at https://doi.org/10.48550/arXiv.2503.21686 (2025).
48. Chang, R.-Y., Lin, Y.-C., Hsu, P.-C., Huang, T.-W. & Kuo, E.-J. Accelerating parameter initialization in quantum chemical simulations via LSTM-FC-VQE. *IEEE Access* **13**, 189725–189734 (2025).
49. Tabak, E. G. & Vanden-Eijnden, E. Density estimation by dual ascent of the log-likelihood. *Commun. Math. Sci.* **8**, 217–233 (2010).
50. Rezende, D. J. & Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning* **37**, 1530–1538 (ICML, 2015).
51. Kobyzev, I., Prince, S. J. & Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 3964–3979 (2020).
52. Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S. & Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.* **22**, 1–64 (JMLR, 2021).
53. Noé, F., Olsson, S., Köhler, J. & Wu, H. Boltzmann generators: sampling equilibrium states of many-body systems with deep learning. *Science* **365**, eaaw1147 (2019).
54. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 1–43 (JMLR, 2018).
55. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
56. Rafailov, R. et al. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems* Vol. 36, 53728–53741 https://arxiv.org/abs/2305.18290 (2023).
57. Xu, H. et al. Contrastive preference optimization: pushing the boundaries of LLM performance in machine translation. In *International Conference on Machine Learning*, 55204–55224 (ICML, 2024).
58. Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992).
59. Lolur, P. et al. Reference-state error mitigation: a strategy for high accuracy quantum computation of chemistry. *J. Chem. Theory Comput.* **19**, 783–789 (2023).
60. Sheka, E. *Stretching and Breaking of Chemical Bonds, Correlation of Electrons, and Radical Properties of Covalent Species*, 111–161 (Elsevier, 2015). https://doi.org/10.1016/bs.aiq.2014.07.005.
61. Wang, L. et al. Quantum chemical descriptors in quantitative structure–activity relationship models and their applications. *Chemometrics Intell. Lab. Syst.* **217**, 104384 (2021).
62. von Lilienfeld, O. A., Müller, K.-R. & Tkatchenko, A. Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem.* **4**, 347–358 (2020).
63. Lanchantin, J. et al. Diverse preference optimization. Preprint at https://doi.org/10.48550/arXiv.2501.18101 (2025).
64. Maus, N., Wu, K., Eriksson, D. & Gardner, J. Discovering many diverse solutions with bayesian optimization. Preprint at https://doi.org/10.48550/arXiv.2210.10953 (2023).
65. Zhai, S. et al. Normalizing flows are capable generative models. In *International Conference on Machine Learning* (ICML, 2025).
66. Larocca, M., Ju, N., García-Martín, D., Coles, P. J. & Cerezo, M. Theory of overparametrization in quantum neural networks. *Nat. Comput. Sci.* **3**, 542–551 (2023).
67. Liang, S., Zhu, L., Liu, X., Yang, C. & Li, X. Artificial-intelligence-driven shot reduction in quantum measurement. *Chem. Phys. Rev* **5**, 041403 (2024).
68. Anand, A. et al. A quantum computing view on unitary coupled cluster theory. *Chem. Soc. Rev.* **51**, 1659–1684 (2022).
69. Sim, S., Johnson, P. D. & Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv. Quantum Technol.* **2**, 1900070 (2019).
70. Cerezo, M. et al. Does provable absence of barren plateaus imply classical simulability? *Nat. Commun.* **16**, 7907 (2025).
71. Motta, M. et al. Subspace methods for electronic structure simulations on quantum computers. *Electron. Struct.* **6**, 013001 (2024).
72. Robledo-Moreno, J. et al. Chemistry beyond the scale of exact diagonalization on a quantum-centric supercomputer. *Sci. Adv.* **11**, eadu9991 (2025).

73. Kanno, K. et al. Quantum-Selected Configuration Interaction: classical diagonalization of Hamiltonians in subspaces selected by quantum computers. Preprint at https://doi.org/10.48550/arXiv.2302.11320 (2023).
74. Holmes, Z., Sharma, K., Cerezo, M. & Coles, P. J. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum* **3**, 010313 (2022).
75. Martyniuk, D., Jung, J. & Paschke, A. Quantum architecture search: a survey. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 1695-1706 (IEEE, 2024). https://doi.org/10.1109/QCE60285.2024.00198.
76. Grimsley, H. R., Claudino, D., Economou, S. E., Barnes, E. & Mayhall, N. J. Is the Trotterized UCCSD ansatz chemically well-defined? *J. Chem. Theory Comput.* **16**, 1–6 (2019).
77. Grimsley, H. R., Economou, S. E., Barnes, E. & Mayhall, N. J. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nat. Commun.* **10**, 3007 (2019).
78. Campbell, E. Random compiler for fast Hamiltonian simulation. *Phys. Rev. Lett.* **123**, 070503 (2019).
79. Nakaji, K. et al. The generative quantum eigensolver (GQE) and its application for ground state search. Preprint at https://doi.org/10.48550/arXiv.2401.09253 (2024).
80. Tran, D., Vafa, K., Agrawal, K., Dinh, L. & Poole, B. Discrete flows: Invertible generative models of discrete data. In *Advances in Neural Information Processing Systems* **32** (NIPS, 2019).
81. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P. & Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems* **34** (NeurIPS, 2021).
82. Fürrutter, F., Chandani, Z., Hamamura, I., Briegel, H. J. & Muñoz-Gil, G. Synthesis of discrete-continuous quantum circuits with multimodal diffusion models. Preprint at https://doi.org/10.48550/arXiv.2506.01666 (2025).
83. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at https://doi.org/10.48550/arXiv.1411.4028 (2014).
84. McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *N. J. Phys.* **18**, 023023 (2016).
85. Lavrijsen, W., Tudor, A., Müller, J., Iancu, C. & De Jong, W. Classical optimizers for noisy intermediate-scale quantum devices. In *2020 IEEE International Conference on Quantum Computing and Engineering (QEC)*, 267–277 (IEEE, 2020).
86. Chivilikhin, D. et al. MoG-VQE: Multiobjective genetic variational quantum eigensolver. Preprint at https://doi.org/10.48550/arXiv.2007.04424 (2020).
87. Jäger, J., Kaldenbach, T. N., Haas, M. & Schultheis, E. Fast gradient-free optimization of excitations in variational quantum eigensolvers. *Commun. Phys.* **8**, 418 (2025).
88. Dinh, L., Sohl-Dickstein, J. & Bengio, S. Density estimation using Real NVP. In *International Conference on Learning Representations* (ICLR, 2017).
89. Kingma, D. P. & Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems* **31** (NeurIPS, 2018).
90. Kingma, D. P. et al. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems* **29** (NeurIPS, 2016).
91. Papamakarios, G., Pavlakou, T. & Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems* **30** (NeurIPS, 2017).
92. Müller, T., McWilliams, B., Rousselle, F., Gross, M. & Novák, J. Neural importance sampling. *ACM Trans. Graph.* **38**, 1–19 (2019).
93. Durkan, C., Bekasov, A., Murray, I. & Papamakarios, G. Cubic-Spline Flows. In *International Conference on Machine Learning* (ICML, 2019).
94. Meng, C., Song, Y., Song, J. & Ermon, S. Gaussianization flows. In *International Conference on Artificial Intelligence and Statistics*, 4336–4345 (PMLR, 2020).
95. Laparra, V., Camps-Valls, G. & Malo, J. Iterative gaussianization: from ICA to random rotations. *IEEE Trans. Neural Netw.* **22**, 537–549 (2011).
96. McClean, J. R. et al. OpenFermion: the electronic structure package for quantum computers. *Quantum Sci. Technol.* **5**, 034014 (2020).
97. Bravyi, S., Gambetta, J. M., Mezzacapo, A. & Temme, K. Tapering off qubits to simulate fermionic Hamiltonians. Preprint at https://doi.org/10.48550/arXiv.1701.08213 (2017).
98. Bergholm, V. et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. Preprint at https://doi.org/10.48550/arXiv.1811.04968 (2022).
99. Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242–246 (2017).
100. Arrazola, J. M. et al. Universal quantum circuits for quantum chemistry. *Quantum* **6**, 742 (2022).
101. Xia, R. & Kais, S. Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations. *Quantum Sci. Technol.* **6**, 015001 (2020).
102. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. In *Advances in neural information processing systems* **32** (NeurIPS, 2019).
103. Rozet, F. et al. Zuko: Normalizing flows in pytorch. https://pypi.org/project/zuko (2022).
104. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). In *International Conference on Learning Representations* (ICLR, 2016).
105. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. Preprint at https://doi.org/10.48550/arXiv.1412.6980 (2017).

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The work was further supported by a joint project between WASP and the Wallenberg Center for Quantum Technology (WACQT), funded by the Knut and Alice Wallenberg Foundation. Preliminary results were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Alvis (project: NAISS 2025/22-480), partially funded by the Swedish Research Council through grant agreement no.~2022-06725. A.F.K. acknowledges support from the Swedish Foundation for Strategic Research (grant numbers FFL21-0279 and FUS21-0063). A.F.K. and M.R. also acknowledges support from the Horizon Europe program HORIZON-CL4-2022-QUANTUM-01-SGA via the project 101113946 OpenSuperQPlus100. S.O. thanks Morten Kjaergaard (UCPH) for discussions during the early conceptualization of the presented work and Werner Dobrautz (ScaDS.AI) for discussions and help during the early implementation phase.

Author contributions

S.O. conceptualized the work. H.Z. and S.O. developed methods and designed experiments with input from A.F.K. and M.R.; H.Z. implemented methods and experiments under the supervision of S.O.; S.O. and A.F.K. procured funding for the study. H.Z. and S.O. wrote the main manuscript text. All authors reviewed the manuscript.

Funding

Open access funding provided by Chalmers University of Technology.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-025-01159-x.

Correspondence and requests for materials should be addressed to Simon Olsson.