One More Time: Revisiting Neural Quantum States from a Reinforcement Learning Perspective

Juan Agustin Duque[*1,2] **Sergio García-Heredia**[*2] **Vinicius Hernandes**[3] **Eliška Greplová**[3] **Thomas Spriggs**[†3] **Aaron Courville**[†1] **Anna Dawid**[†2]

Abstract

Neural quantum states (NQS) provides a flexible and scalable framework for approximating quantum many-body wavefunctions. Among NQS parameterizations, autoregressive models are especially attractive because they enable exact, independent sampling from the Born distribution, avoiding the autocorrelation and mixing issues of Markov chain methods. Yet their optimization remains comparatively underexplored: Adam is a scalable method but ignores function space geometry, while stochastic reconfiguration is principled but costly and numerically fragile in large models. To address this gap, we show that variational energy minimization can be viewed as an advantage policy-gradient problem over the Born distribution, motivating trust-region optimization for NQS training. We introduce *Proximal Wavefunction Optimization* (PWO), a trust-region algorithm that clips probability-ratio changes in the amplitude channel and wrapped phase increments in the phase channel. PWO avoids explicit matrix inversion, reuses samples across inner updates, and preserves the scalability of first-order optimization. Across Ising, Heisenberg, and frustrated J_1–J_2 spin chains, PWO improves stability and wall-clock convergence over Adam, minSR, and SPRING. Finally, we fine-tune a 1.5B-parameter RWKV-7 model as a neural quantum state, demonstrating NQS optimization at a scale over three orders of magnitude beyond prior work.

[*]Equal contribution [†] Equal senior supervision [1]Mila – Quebec AI Institute, Université de Montréal, Montréal, Canada [2]$\langle$aQa$^L\rangle$ Applied Quantum Algorithms – Leiden Institute of Advanced Computer Science & Leiden Institute of Physics, Leiden University, Leiden, The Netherlands [3]QuTech and Kavli Institute of Nanoscience, Delft University of Technology, Delft, The Netherlands. Correspondence to: Juan Agustin Duque <juanduquevan@gmail.com>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

1. Introduction

Quantum physics seeks to predict and understand the behavior of many interacting quantum particles. However, determining the exact ground (lowest-energy) state of a system of N quantum bits generally requires diagonalizing a $2^N \times 2^N$ Hamiltonian matrix, quickly rendering exact methods intractable at scale. Variational approaches address this exponential complexity by optimizing a parameterized wavefunction ansatz to approximate the ground state. Neural quantum states (NQS) (Carleo & Troyer, 2017; Lange et al., 2024) build upon this by using neural networks as expressive wavefunction representations, capable of capturing complex correlations (Nomura & Imada, 2021) and high entanglement (Gauvin-Ndiaye et al., 2025), as well as scaling to high-dimensional systems (Pescia et al., 2024), where traditional methods struggle.

The practical success of NQS depends not only on expressivity but also on the ability to sample from and optimize the variational distribution. State-of-the-art NQS methods (Viteritti et al., 2023) primarily rely on Markov Chain Monte Carlo (MCMC) to estimate energies and gradients, which can introduce autocorrelation, slow mixing, and unreliable exploration in difficult regimes (Wolff, 1990; Del Debbio et al., 2004). Autoregressive NQS remove this sampling bottleneck by factorizing the Born distribution and enabling exact, independent sampling (Sharir et al., 2020; Hibat-Allah et al., 2020; Moss et al., 2025). As a result, they provide a clean setting for variational optimization: samples can be drawn directly from the current wavefunction, and training dynamics are no longer confounded by Markov chain limitations. Despite these advantages, their adoption remains limited, as stable optimization is a central bottleneck for training accurate autoregressive models.

Existing optimization methods expose a sharp trade-off. First-order optimizers such as Adam (Kingma & Ba, 2014) are computationally efficient and scale naturally to large neural networks, but they ignore the geometry of the variational wavefunction and can converge unstably or inaccurately in NQS applications (Pfau et al., 2020; Liu et al., 2025). Stochastic reconfiguration (SR), and scalable variants such as minSR (Chen & Heyl, 2024), are more geometrically

principled because they approximate natural-gradient descent in wavefunction space. However, they require solving large and often ill-conditioned linear systems, with costs that become prohibitive for large networks or large sample regimes. Thus, autoregressive NQS resolve an important sampling problem, but stable first-order optimization of large autoregressive wavefunction models remains a central obstacle.

Contributions. This gap points to a missing optimization principle in NQS. In this work, we observe that the variational objective in NQS is mathematically equivalent to a policy-gradient objective in reinforcement learning (RL) under certain assumptions. While this connection has been implicitly present since early work (Carleo & Troyer, 2017), it has not been formalized or leveraged to design modern optimizers. In this work, we show that the *RL-style trust-region optimization improves NQS training scalability and stability.* Our contributions are as follows:

- We prove that, for stoquastic Hamiltonians, the variational energy gradient is exactly an advantage policy-gradient estimator over the Born distribution, formalizing a correspondence between NQS optimization and policy-gradient RL.
- We introduce *Proximal Wavefunction Optimization (PWO)*, an algorithm for autoregressive NQS inspired by Proximal Policy Optimization (PPO) (Schulman et al., 2017).
- We show that PWO improves stability and convergence speed of autoregressive NQS compared to Adam and minSR on standard benchmarks (1D Ising and Heisenberg models).
- Finally, we demonstrate the scalability of PWO by fine-tuning a 1.5 billion parameter RWKV-7 LLM (Peng et al., 2025) on the 1-D Ising model.

2. Background

2.1. Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm in which an agent learns to make decisions by interacting with an environment. At each time step t, the agent observes a state s_t, selects an action a_t, and then receives a reward r_{t+1} and a new state s_{t+1} from the transition function $P(\cdot|s_t, a_t)$. The goal of the agent is to learn a *policy* that maximizes the expected return over time. Let $\pi_{\boldsymbol{\theta}}$ denote a policy parameterized by $\boldsymbol{\theta}$. Formally, the return is defined over a trajectory $\tau := (s_0, a_0, r_1, s_1, a_1, r_2, \dots)$, which is a sequence of states, actions, and rewards generated by the agent-environment interaction. Following the notation of (Agarwal et al., 2021), the probability of a trajectory τ under $\pi_{\boldsymbol{\theta}}$ is given by

$$\mathcal{P}_{\boldsymbol{\theta}}(\tau) = \mu(s_0) \prod_{t=0}^{\infty} \pi_{\boldsymbol{\theta}}(a_t|s_t) P(s_{t+1}|s_t, a_t), \tag{1}$$

where $\mu(s_0)$ denotes the initial distribution over states. The objective of RL is to find a policy that maximizes the expected *discounted return*, which is captured by the state-value and action-value functions of a policy $\pi_{\boldsymbol{\theta}}$:

$$V^{\pi_{\boldsymbol{\theta}}}(s) := \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right], \tag{2}$$

$$Q^{\pi_{\boldsymbol{\theta}}}(s, a) := \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s,\ a_0 = a\right], \tag{3}$$

respectively. In policy optimization, an agent maximizes its expected return by performing gradient ascent with a *policy gradient* estimator (Williams, 1992) of the form:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(\mu) = \mathbb{E}_{\tau \sim \mathcal{P}_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_{\boldsymbol{\theta}}}(s_t, a_t) \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)\right]. \tag{4}$$

Here $A^{\pi_{\boldsymbol{\theta}}}(s, a) := Q^{\pi_{\boldsymbol{\theta}}}(s, a) - V^{\pi_{\boldsymbol{\theta}}}(s)$ denotes the advantage of the agent taking action a while in state s.

2.2. Neural Quantum States

A central task in quantum many-body physics is to approximate the ground-state wavefunction of a system with many interacting particles. For a system of N spin-$1/2$ particles, a configuration is a binary vector $\mathbf{s} \in \{\pm 1\}^N$, and the wavefunction can be viewed as a complex vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$ (see Appendix A) indexed by all 2^N configurations. Equivalently, it defines a function $\mathbf{s} \mapsto \psi(\mathbf{s})$, where $\psi(\mathbf{s}) \in \mathbb{C}$ is the amplitude assigned to configuration $\mathbf{s}$:

$$|\psi\rangle = \sum_{\mathbf{s} \in \{\pm 1\}^N} \psi(\mathbf{s}) \, |\mathbf{s}\rangle, \tag{5}$$

where we use the Dirac, bra-ket, notation (see Appendix A). The exponential size of this vector makes explicit representations intractable for large N. Neural quantum states (NQS) address this by parameterizing the amplitude function with a neural network, $f_{\boldsymbol{\theta}}$, so that $f_{\boldsymbol{\theta}}(\mathbf{s})$ provides a compact approximation to $\psi(\mathbf{s})$ (Carleo & Troyer, 2017; Dawid et al., 2025). In practice, NQS models output the logarithm of the complex amplitude,

$$f_{\boldsymbol{\theta}}(\mathbf{s}) = \log \psi_{\boldsymbol{\theta}}(\mathbf{s}) = \log |\psi_{\boldsymbol{\theta}}(\mathbf{s})| + i\, \phi(\psi_{\boldsymbol{\theta}}(\mathbf{s})), \tag{6}$$

which separates the wavefunction into a log-modulus and a phase, usually handled by separate networks (channels). This representation is numerically convenient because amplitudes can vary over many orders of magnitude. By the

Born rule, normalized wavefunctions induce a probability distribution over configurations, $P_{\boldsymbol{\theta}}(\mathbf{s}) \propto |\psi_{\boldsymbol{\theta}}(\mathbf{s})|^2$. In this work, we focus on autoregressive NQS, which explicitly factorize $P_{\boldsymbol{\theta}}$ and enable exact independent sampling from the Born distribution.

2.3. Finding the Ground State

The Hamiltonian $\hat{H}$ is the energy operator of a quantum system: it encodes the interactions, external fields, and kinetic terms that determine which wavefunctions have low or high energy. A popular application of the NQS framework is to find the ground state of quantum systems. Given a description of a physics system within a Hamiltonian, $\hat{H}$, the ground state is the eigenstate $|\psi_0\rangle$ corresponding to the lowest eigenvalue E_0. This can be cast as a minimization problem:

$$E_0 \leq E[\psi] = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle}, \quad \forall\, |\psi\rangle \in \mathcal{H}, \tag{7}$$

where the equality holds if and only if $|\psi\rangle = |\psi_0\rangle$ – this is known as the *variational principle*. Taking this variational formulation of the ground state as the starting point, so-called variational methods define a parametric family of states $|\psi_{\boldsymbol{\theta}}\rangle$ and approximate the ground state by solving

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{\langle\psi_{\boldsymbol{\theta}}|\hat{H}|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle}. \tag{8}$$

In the case of NQS, this family of functions is given by a neural network, as already discussed.

2.4. Variational Monte Carlo

To efficiently compute a minimizer of Eq. (8), NQS uses Monte Carlo estimates. This approach is known as variational Monte Carlo (VMC). In particular, the energy can be written as

$$E[\psi_{\boldsymbol{\theta}}] = \frac{\langle\psi_{\boldsymbol{\theta}}|\hat{H}|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle} = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\right], \tag{9}$$

where we define

$$\mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) = \frac{|\psi_{\boldsymbol{\theta}}(\mathbf{s})|^2}{\sum_{\mathbf{s}'} |\psi_{\boldsymbol{\theta}}(\mathbf{s}')|^2}, \tag{10}$$

$$E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) := \sum_{\mathbf{s}'} \frac{\psi_{\boldsymbol{\theta}}(\mathbf{s}')}{\psi_{\boldsymbol{\theta}}(\mathbf{s})} \langle\mathbf{s}|\hat{H}|\mathbf{s}'\rangle\,. \tag{11}$$

With this formulation, NQS estimates the loss function by sampling a set of spin configurations $\{\mathbf{s}_i\}_{i=1}^M$ from the parameterized distribution $\mathcal{P}_{\boldsymbol{\theta}}$ and computing the sample mean of $E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s}_i)$:

$$L(\boldsymbol{\theta}) \approx \bar{E}^{\text{loc}}_{\boldsymbol{\theta}} \equiv \frac{1}{M}\sum_{i=1}^{M} E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s}_i). \tag{12}$$

By differentiating Eq. (9) (see Appendix B.2) it is possible to show that its gradient with respect to $\boldsymbol{\theta}$ can be expressed as an expected value over $\mathcal{P}_{\boldsymbol{\theta}}$ and estimated using either MCMC sampling or direct sampling of autoregressive models:

$$\partial_{\boldsymbol{\theta}_i} L(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[2\,\text{Re}\left\{\left(E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\right)O_i(\mathbf{s})^*\right\}\right], \tag{13}$$

for $i = 1, \ldots, P$, where $O_i(\mathbf{s}) = \partial_{\boldsymbol{\theta}_i} \log \psi_{\boldsymbol{\theta}}$ are the so-called score functions.

2.5. Stochastic Reconfiguration

Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951) is a simple and effective optimization method in many scenarios. However, in the context of VMC, it can suffer from slow convergence and instability due to the complex geometry of the parameter space. The key issue is that small changes in the parameters, $\boldsymbol{\theta}$, do not necessarily correspond to small changes in the quantum state $|\psi_{\boldsymbol{\theta}}\rangle$.

Stochastic reconfiguration (SR) (Sorella, 1998), which generalizes natural gradient descent (Amari, 1998) to VMC, preconditions the gradient $\boldsymbol{\nabla}_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ to ensure that the distance between $|\psi_{\boldsymbol{\theta}}\rangle$ at one training iteration and the next is small. This distance is given by the infidelity between the two states and approximated to second order by the Fubini-Study metric,

$$S_{ij} = \text{Re}\left\{\text{Cov}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[O_i^*(\mathbf{s}), O_j(\mathbf{s})\right]\right\}, \quad i,j = 1,\ldots,P, \tag{14}$$

which encodes the local geometry of the Hilbert space at the current parameter configuration. In mathematical terms, SR modifies the update rule in the following way:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \mathbf{S}^{-1}\boldsymbol{\nabla}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \tag{15}$$

where $\eta \in \mathbb{R}$ is the learning rate. We discuss the high computational costs of SR and its more recent improvements in Appendix C.1, motivating the use of first-order methods.

3. Proximal Wavefunction Optimization

Here, we introduce a new optimization procedure for training NQS. We first draw an equivalence between the current NQS optimization paradigm and the policy gradient update in RL, and then use this to motivate applying Proximal Policy Optimization to NQS.

Proposition 3.1 (Policy-gradient form of variational energy minimization)**.** *Assume the Hamiltonian is stoquastic, i.e. its matrix representation in a chosen computational basis has non-positive off-diagonal elements. Then we can assume*

Algorithm 1 Proximal Wavefunction Optimization (PWO)

Input: Hamiltonian $\hat{H}$, neural quantum state ψ_θ, batch size M, inner epochs K, amplitude clip ϵ, phase clip δ
Initialize $\boldsymbol{\theta}$
while *not converged* **do**
  $\boldsymbol{\theta}_{\text{old}} \leftarrow \boldsymbol{\theta}$
  sample $\{\mathbf{s}_i\}_{i=1}^M \sim \mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}$
  Cache reference log-probabilities, $\log \mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}_i)$; phases, $\phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}_i)$; local energies $E^{\text{loc}}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}_i)$; and normalized real and imaginary advantages, A_i^R and A_i^I respectively, of $\{\mathbf{s}_i\}_{i=1}^M$.
  for $k = 1, \ldots, K$ **do**
    Compute current log-probabilities $\log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}_i)$ and phases $\phi_{\boldsymbol{\theta}}(\mathbf{s}_i)$ for all samples $\{\mathbf{s}_i\}_{i=1}^M$
    $r_i \leftarrow \exp(\log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}_i) - \log \mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}_i))$
    $\Delta\phi_i \leftarrow 2\,\text{atan}\big(\sin(\phi_{\boldsymbol{\theta}}(\mathbf{s}_i) - \phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}_i)), \cos(\phi_{\boldsymbol{\theta}}(\mathbf{s}_i) - \phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}_i))\big)$
    $\ell_i^R \leftarrow \max\big(r_i A_i^R,\ \text{clip}(r_i, 1-\epsilon, 1+\epsilon)A_i^R\big)$
    $\ell_i^I \leftarrow \text{stop_gradient}(r_i) \cdot \max\big(\Delta\phi_i A_i^I,\ \text{clip}(\Delta\phi_i, -\delta, \delta)A_i^I\big)$
    Update $\boldsymbol{\theta}$ by descending on the surrogate objective: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \boldsymbol{\nabla}_{\boldsymbol{\theta}} \sum_{i=1}^M \big(\ell_i^R + \ell_i^I\big)/M$.

$f_{\boldsymbol{\theta}} = \log \psi_{\boldsymbol{\theta}} = \log |\psi_{\boldsymbol{\theta}}|$, *and the gradient of the variational energy can be written in policy-gradient form as*

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\big(E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\text{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\big)\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\big]. \quad (16)$$

In particular, variational energy minimization is equivalent to an advantage policy-gradient update over configurations. For a proof, see Appendix B.1.

This proposition establishes a direct link between the NQS and RL frameworks (see also Tab. 1) by relating Eqs. (2.4) and (3.1) which motivates our algorithm, *Proximal Wavefunction Optimization* (PWO). The key observation is that the SR gradient is the result of solving the approximate constraint optimization problem of minimizing the expected energy while keeping the change in infidelity smaller than some quantity, δ. An analogous problem has been studied in the RL literature for a long time, but referred to as *trust region optimization* (Kakade & Langford, 2002; Schulman et al., 2015; 2017). The fundamental idea is similar to that of SR: maximizing the expected reward while keeping the change in the total variation distance smaller than some quantity, δ.

Proximal Policy Optimization (PPO) (Schulman et al., 2017), introduces a simple heuristic that empirically shows monotonic improvements by constraining updates to keep the updated policy close to the previous one. Applied to NQS, PPO would minimize the following surrogate loss:

$$L^{\text{clip}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}\left[\max\left(r(\boldsymbol{\theta})A^R, \text{clip}(r(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon)A^R\right)\right], \quad (17)$$

where $r(\boldsymbol{\theta}) = \mathcal{P}_{\boldsymbol{\theta}}/\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}$ is the importance sampling ratio between the updated and previous distributions, ϵ is the amplitude clip, the max comes from solving a minimization problem, and

$$A^R := \text{Re}\left\{E^{\text{loc}}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})\right\} - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}\left[\text{Re}\left\{E^{\text{loc}}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})\right\}\right], \quad (18)$$

is the equivalent of the advantage estimate. In the case of RL, clipping ensures that policy updates remain conservative, thereby maintaining stability during optimization.

While PPO is a natural fit for the amplitude channel, it does not by itself account for the full VMC gradient. As shown in Appendix B.3, the variational gradient also contains an imaginary component, which governs how the phase of the wavefunction should evolve. We, therefore, introduce a second proximal objective that controls phase updates directly by constraining the wrapped phase increment between the current and reference models. Concretely, letting $\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s})$ denote the wrapped phase difference (see Appendix B.4), we minimize the clipped phase surrogate loss

$$L^{\text{clip}}_{\text{phase}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}\Big[\text{sg}\big(r(\boldsymbol{\theta})\big) \times \max\Big(\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s})A^I(\mathbf{s}),\ \text{clip}\big(\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s}), -\delta, \delta\big)A^I(\mathbf{s})\Big)\Big], \quad (19)$$

where $\text{sg}\big(r(\boldsymbol{\theta})\big)$ denotes the stop-gradient operator (see Appendix B.5), and

$$A^I := \text{Im}\left\{E^{\text{loc}}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})\right\} - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}\left[\text{Im}\left\{E^{\text{loc}}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})\right\}\right]. \quad (20)$$

The imaginary part of the local energy has zero expectation, so the centering is retained only to mirror the RL advantage. Equation (19) is the phase analogue of PPO: instead of clipping a probability ratio, it clips the phase increment itself, preventing abrupt rotations of the wavefunction while still following the imaginary part of the VMC gradient. Using the phase increment instead of the ratio of phases ensures that the gradient of the combined loss (Eqs. (3)+(19))

matches the original one (Eq. (2.4)) when we are on-policy, *i.e.*, when the importance sampling ratio is one.

PWO combines the amplitude and phase losses into an *efficient* (see Appendix C.2) first-order optimization procedure. For the amplitude channel, Proposition 3.1 shows that, when the phase is fixed, variational energy minimization takes the form of an advantage-weighted policy-gradient update over configurations, which naturally motivates a PPO-style clipped surrogate based on importance ratios. For the phase channel, Appendix B.4 shows that the imaginary part of the VMC gradient can be conservatively optimized through a surrogate that constrains the phase increment. In practice, PWO performs K inner updates on a frozen batch of configurations sampled from the reference Born distribution $\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}$, clipping probability-ratio changes for the amplitude and wrapped phase increments for the phase. Because the same batch is reused across inner steps, the phase objective includes detached importance weights to correct for the mismatch between the reference and current sampling distributions (see Appendix B.5). Algorithm 1 summarizes the resulting procedure.

4. Experiments

4.1. Spin-chain benchmarks

We evaluate PWO for 1.5M-parameter autoregressive NQS on three different spin systems. Each systems consists of a one-dimensional spin-1/2 chain of $N = 12$ sites with periodic boundary conditions, so that site indices are understood modulo N. Let $\hat{\sigma}_i^\alpha$, $\alpha \in \{x, y, z\}$, denote the Pauli operator on site i, and let $\hat{\mathbf{S}}_i = 1/2(\hat{\sigma}_i^x, \hat{\sigma}_i^y, \hat{\sigma}_i^z)$. For each Hamiltonian, we minimize the variational energy $E[\psi_{\boldsymbol{\theta}}]$ and report the relative error with respect to the exact ground-state energy E_0, which can be computed for these relatively small systems. We also report the V-score (Wu et al., 2024), a scale-invariant convergence metric based on the energy variance that vanishes for exact eigenstates. The three Hamiltonians chosen offer increasing difficulty, as the underlying physics requires more complicated functions to represent. We use the same NQS architecture and the same number of Monte Carlo samples for all optimizers, so differences in performance reflect optimization rather than model capacity. For details about the hyperparameter search and the used architecture, see Appendices D.1 and D.2.

Ising Model.

$$\hat{H}_{\text{Ising}} = -J \sum_{i=1}^{N} \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - h \sum_{i=1}^{N} \hat{\sigma}_i^x. \tag{21}$$

Here, $J = 1$ is the ferromagnetic coupling strength and $h = 1$ is the transverse-field strength. This Hamiltonian is a standard sign-problem-free benchmark: the diagonal interaction term favors aligned spin configurations, while the transverse field introduces quantum fluctuations by flipping individual spins. Therefore, Proposition 3.1 holds exactly.

Heisenberg Chain.

$$\hat{H}_{\text{H}} = J \sum_{i=1}^{N} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1}, \tag{22}$$

where $J = 0.25$ is the antiferromagnetic exchange coupling. Compared to the Ising model, the Heisenberg chain requires the NQS to represent stronger quantum correlations induced by spin-exchange terms, making it a more stringent test of optimization stability.

Heisenberg J_1–J_2 Chain.

$$\hat{H}_{J_1-J_2} = J_1 \sum_{i=1}^{N} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1} + J_2 \sum_{i=1}^{N} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+2}. \tag{23}$$

We set $J_1 = 1$ and $J_2 = 0.5$, *i.e.*, $J_2/J_1 = 0.5$. At this ratio the next-nearest-neighbor interaction maximally frustrates the nearest-neighbor antiferromagnetic coupling, placing the system at a known quantum critical point where a highly entangled ground state with a complex sign structure makes both variational optimization and Monte Carlo sampling particularly challenging (Sorella, 1998). Unlike $\hat{H}_{\text{Ising}}$, the Hamiltonians $\hat{H}_{\text{H}}$ and $\hat{H}_{J_1-J_2}$ require a nontrivial sign/phase structure, so the wavefunction must generally be represented with complex amplitudes. This setting evaluates whether the clipped phase surrogate (Eq. (19)) remains a useful practical heuristic. All results are computed over 10 random seeds. Since several runs reach high-precision errors, numerical instabilities can produce extreme outliers. Mean-and-standard-deviation summaries are therefore poorly suited to this regime, as a single unstable run can distort both the central estimate and the uncertainty band. Following Agarwal et al. (2022), we instead report interquartile statistics: at each evaluation time, curves show the interquartile mean (IQM), computed by averaging the middle 50% of seeds after discarding the lowest and highest quartiles. Shaded regions denote the interquartile range, from the 25th to the 75th percentile. This provides robust central estimates while still showing run-to-run variability. The figures with all individual seeds plotted are shown in Appendices E.1-E.2. We discuss limitations of our approach in Appendix C.3.

4.2. Convergence and Stability Results

Figure 1 shows that PWO converges the fastest of the four optimizers on the Ising model, reaching relative error 10^{-7} in approximately 5 minutes, while minSR requires approximately 30 minutes to reach the same accuracy. The V-score follows the same trend: PWO rapidly suppresses energy fluctuations, indicating that the proximal objective is not merely improving the energy estimate, but driving the state

Reinforcement Learning		Variational Monte Carlo	
Policy	$\pi_{\boldsymbol{\theta}}(a \mid s)$	Born distribution	$\mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) \propto \lvert\psi_{\boldsymbol{\theta}}(\mathbf{s})\rvert^2$
Advantage	$A^{\pi_{\boldsymbol{\theta}}}(s_t, a_t)$	Centered local energy	$\Delta E(\mathbf{s}) = E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - E[\psi_{\boldsymbol{\theta}}]$
Policy gradient	$A^{\pi_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}$	VMC force	$2\,\mathrm{Re}[\Delta E(\mathbf{s}) \nabla_{\boldsymbol{\theta}} \log \psi^*_{\boldsymbol{\theta}}(\mathbf{s})]$
KL trust region	$D_{\mathrm{KL}}(\pi_{\boldsymbol{\theta}_{\mathrm{old}}} \Vert \pi_{\boldsymbol{\theta}})$	Infidelity	$\mathcal{I}(\psi_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}+\delta\boldsymbol{\theta}})$
Fisher matrix	$\mathbf{F}$	Fubini–Study metric	$\mathbf{S}$

Table 1. Correspondence between RL policy-gradient methods and VMC methods for NQS.

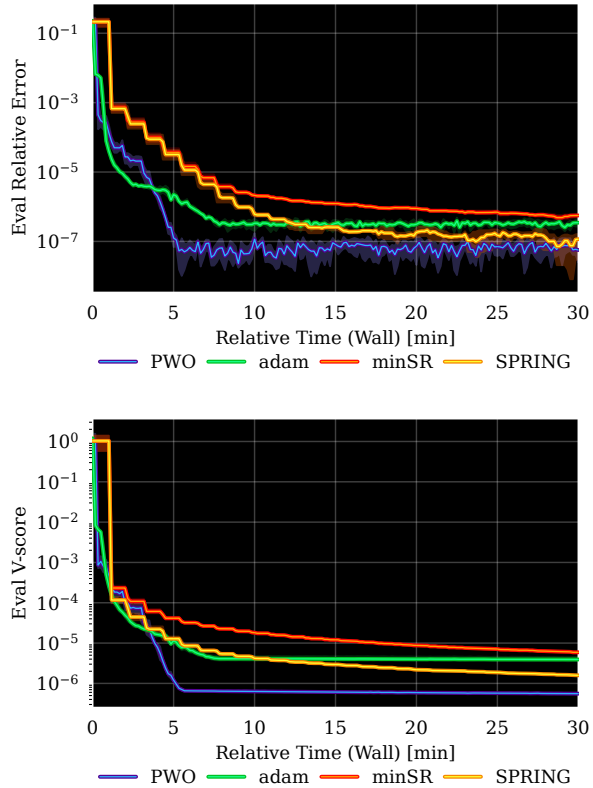


Figure 1. Comparison of PWO, Adam, minSR, and SPRING on the transverse-field Ising model over 10 random seeds. All methods in this comparison were run with 1024 samples on a single NVIDIA L40S GPU.

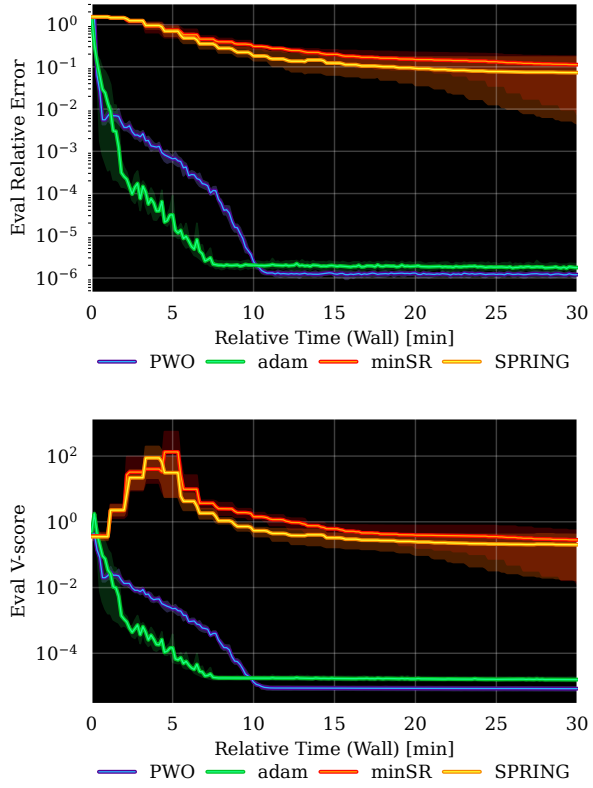


Figure 2. Comparison of PWO, Adam, minSR, and SPRING on the nearest-neighbor Heisenberg chain over 10 random seeds. MinSR is numerically unstable on this Hamiltonian, with 3 out of 10 runs producing NaNs; for visualization, NaN trajectories are continued at their best finite value before collapse. All methods were run with 1024 samples on a single NVIDIA L40S GPU.

toward an eigenstate. Adam and SPRING also make steady progress, but require more time to reach the same accuracy. minSR is stable on this easier Hamiltonian, but its per-step cost makes it substantially slower in wall-clock time.

On the Heisenberg chain, Figure 2 demonstrates that PWO remains stable and reaches relative error 10^{-6} in approximately 10 minutes. The contrast with minSR and SPRING is sharper than in the Ising case: minSR is numerically unstable, with 3 out of 10 runs producing NaNs, and remains near relative error 10^{-1} after 30 minutes. Despite using curvature-inspired updates, these methods struggle to make progress and exhibit numerical instability. Adam is competitive on this benchmark, suggesting that the nearest-neighbor chain is not sufficient by itself to separate first-order methods. However, PWO matches the stability of Adam while preserving the proximal multi-step update structure that becomes decisive in the frustrated case below.

The frustrated J_1–J_2 chain gives the clearest separation between optimizers. In Figure 3 PWO reaches relative error 10^{-7} in approximately 15 minutes while maintaining a steadily decreasing V-score, showing that the learned state is converging in both energy and variance. In contrast, Adam plateaus several orders of magnitude above PWO due to outliers, while minSR and SPRING remain near relative error 10^{-1} after 30 minutes and suffer from severe numerical

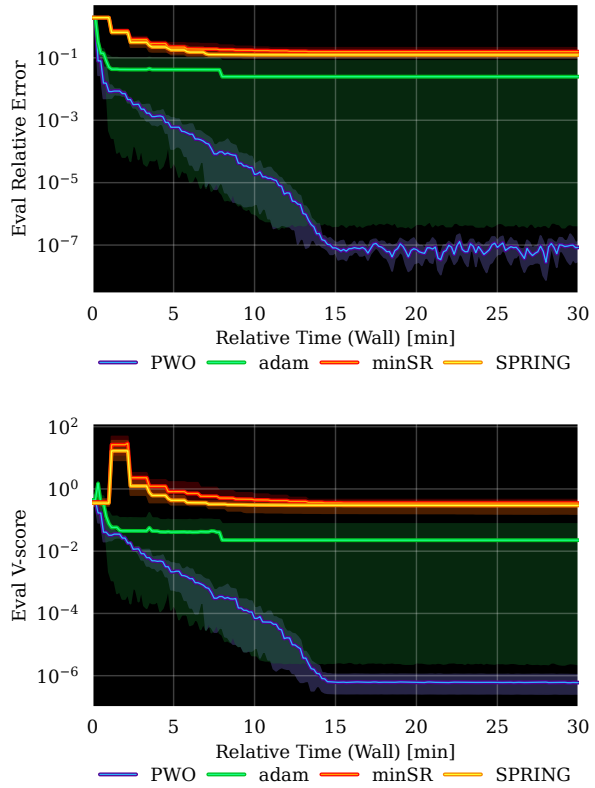


Figure 3. Comparison of PWO, Adam, minSR, and SPRING on the Heisenberg J_1–J_2 chain over 10 random seeds. MinSR is highly unstable on this Hamiltonian, with 6 out of 10 runs producing NaNs. All methods were run with 1024 samples on a single NVIDIA L40S GPU.

instability. This is the strongest empirical evidence for the central claim of the paper: a PPO-style proximal objective can retain the low wall-clock cost of first-order optimization while providing the stability needed to train expressive NQS on difficult quantum Hamiltonians. We also plot all individual seed curves without aggregation or filtering for greater clarity and visibility in Appendix E.1.

4.3. Scalability Experiments

Figure 4 compares the scaling behavior of PWO, Adam, and minSR for increasing NQS sizes (see Appendix D.6 for the scaling procedure). PWO exhibits consistent stability and monotonic energy improvement as NQS size grows, reaching near-final energies within the first 30 minutes. In contrast, minSR is both unstable and never reaches competitive energy values. Adam performs best for the tiny network, but degrades in stability and efficiency for small and medium NQS, where PWO is both faster and more reliable. In particular, achieving comparable energy scales with the medium NQS requires roughly $4\times$ more time with Adam than with PWO. In Appendix C.2, we show that PWO converges faster than other optimizers because it does more optimization steps per unit of time. Although Adam attains lower energies after extended training of 4 hours, the observed scaling trend suggests that this advantage may diminish for yet larger networks.

4.4. NQS Fine-tuning of Large Language Models

As a final scalability test, we instantiate the NQS with a pretrained RWKV-7 large language model (LLM). RWKV is a natural architecture for autoregressive NQS because it combines sequence-model expressivity with recurrent inference: unlike Transformer-based autoregressive models, it does not require a growing key-value cache, and its per-token inference cost and memory footprint remain constant in sequence length (Peng et al., 2025). This is particularly attractive for NQS sampling, where configurations are generated sequentially and model evaluation is repeatedly invoked inside Monte Carlo optimization. In addition, recent JAX implementations for hyperscale RWKV fine-tuning provide a practical foundation for adapting billion-parameter RWKV models to nonstandard optimization objectives (Sarkar et al., 2026). Figure 5 shows fine-tuning results for a 1.5B-parameter RWKV-7 model on the transverse-field Ising chain. PWO achieves a lower final relative error and V-score than Adam, while maintaining stable training across the full wall-clock budget. This improvement is not meant to establish that LLMs or large networks (Moss et al., 2026) are the right inductive bias for one-dimensional Hamiltonians; rather, it shows that the proximal objective is effective when the ansatz is scaled by more than three orders of magnitude beyond existing NQS in the literature (Rende et al., 2025).

5. Related Work

Reinforcement Learning and Ground State Search. Ground state search has been treated as an optimal control problem via Feynman-Kac representations (Barr et al., 2020), and RL frameworks have been developed for lattice models by treating stoquastic Hamiltonians as reward functions (Gispen & Lamacraft, 2022). Recent meta-learning approaches (Jae et al., 2025) leverage RL to optimize the quantum state learning process itself. However, these works apply RL by either reformulating the problem theoretically or optimizing the learning procedure. We instead use RL as the primary optimization framework applying it directly on the NQS ansatz.

Importance Sampling and Trust Regions. Sample reuse and proximal objectives have been previously explored for non-autoregressive NQS optimization. (Yang et al., 2020) introduced importance sampling gradient optimization (ISGO), which reuses Markov chain samples across multiple gradient updates by reweighting local energies with

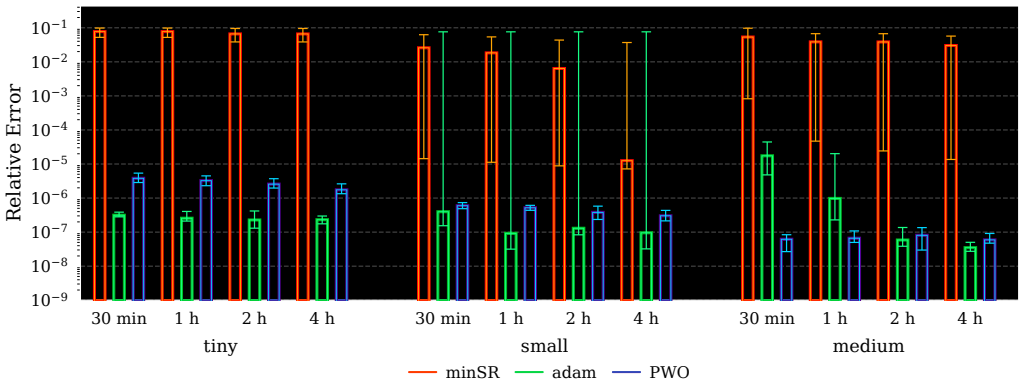


Figure 4. Wall-clock scaling comparison across model sizes and optimization methods. Bars show the interquartile mean relative error over seeds, with error bars indicating the interquartile range. Results are grouped by model size and wall-clock time, and run on a single NVIDIA A100 GPU.

the probability ratio between the current and reference wavefunctions. Building on this, (Chen et al., 2022) introduced a PPO-inspired a cosine penalty on phase variation. Our work formalizes the analogy with RL in Proposition 3.1, uses a different clipping objective directly on the imaginary gradient instead of a penalty, and makes the policy-gradient correspondence exact, allowing stable training of billion-parameter models for the first time.

6. Conclusion

We introduced Proximal Wavefunction Optimization (PWO), a scalable first-order optimizer for neural quantum states (NQS) derived from an explicit connection between variational Monte Carlo and reinforcement learning (RL). In the fixed-phase stoquastic setting, we showed that variational energy minimization can be written as an advantage policy-gradient objective over the Born distribution, where spin configurations play the role of actions and centered local energies act as advantages. This reformulates NQS optimization into a trust-region policy-optimization problem, motivating a PPO-style algorithm that clips probability-ratio changes in the amplitude channel and extends naturally to complex wavefunctions through a clipped phase-increment surrogate. Across spin-chain benchmarks, PWO improves wall-clock convergence and stability over Adam, minSR, and SPRING, with the largest gains in frustrated regimes where optimization is most fragile. Finally, by fine-tuning a 1.5B-parameter RWKV-7 model as an autoregressive NQS, we bring NQS training into the regime of modern large-scale sequence modeling. More broadly, our results bridge the RL and NQS communities, enabling further mutual inspiration. By unlocking large autoregressive NQS, PWO opens the door to studying whether scale alone, in the spirit of the "bitter lesson", can overcome current barriers in quantum many-body simulation.

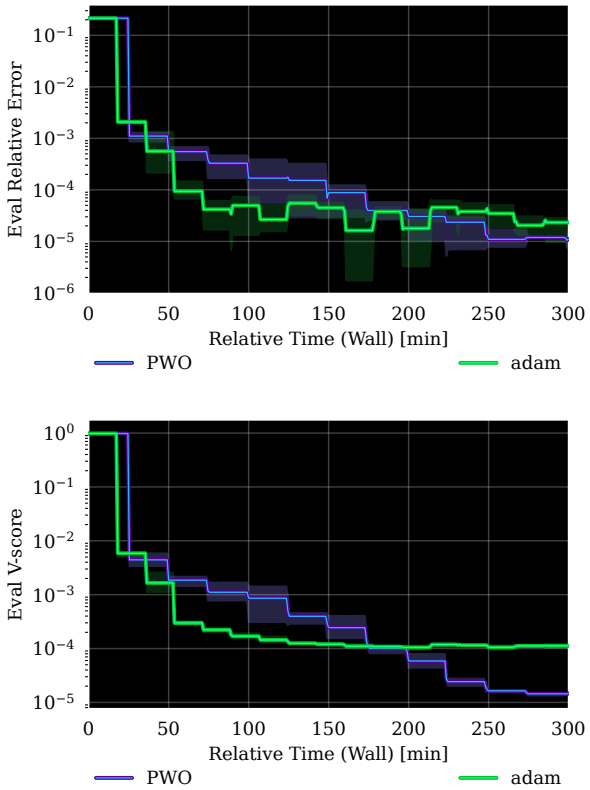


Figure 5. Fine-tuning curves of a 1.5B-parameter RWKV7 Large Language Model on the 1-D Ising Model. The experiment is run on a single NVIDIA L40S GPU, limiting it to 150 samples.

Acknowledgements

This work was supported by the Netherlands Organization for Scientific Research (NWO/OCW), as part of Quantum Limits (project number SUMMIT.1.1016). This publication is also part of the project Optimal Digital-Analog Quantum Circuits with File No. NGF.1582.22.026 of the research programme NGF Quantum Delta NL 2022, which is (partly) financed by the Dutch Research Council (NWO) and the Dutch National Growth Fund initiative Quantum Delta NL. In addition, Juan Agustín Duque is supported by the St-Pierre-Larochelle Scholarship at the University of Montreal and by Aaron Courville's CIFAR AI Chair in Representations that Generalize Systematically.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agarwal, A., Jiang, N., Kakade, S., and Sun, W. *Reinforcement Learning: Theory and Algorithms*. Draft/Preprint, 2021. URL https://rltheorybook.github.io/.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., and Bellemare, M. G. Deep reinforcement learning at the edge of the statistical precipice. *arXiv preprint arXiv:2108.13264*, 2022. doi: 10.48550/arXiv.2108.13264. URL https://arxiv.org/abs/2108.13264.

Amari, S.-i. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746. URL https://doi.org/10.1162/089976698300017746.

Barr, A., Gispen, W., and Lamacraft, A. Quantum ground states from reinforcement learning. In *Proceedings of the First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pp. 635–653, 2020. URL https://proceedings.mlr.press/v107/barr20a.html.

Carleo, G. and Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. doi: 10.1126/science.aag2302. URL https://doi.org/10.1126/science.aag2302.

Chen, A. and Heyl, M. Empowering deep neural quantum states through efficient optimization. *Nature Physics*, 20(9):1476–1481, 2024. doi: 10.1038/s41567-024-02566-1. URL https://doi.org/10.1038/s41567-024-02566-1.

Chen, H., Hendry, D. G., Weinberg, P. E., and Feiguin, A. Systematic improvement of neural network quantum states using lanczos. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=qZUHvvtbzy.

Choo, K., Mezzacapo, A., and Carleo, G. Fermionic neural-network states for ab-initio electronic structure. *Nature Communications*, 11(1):2368, 2020. doi: 10.1038/s41467-020-15724-9. URL https://doi.org/10.1038/s41467-020-15724-9.

Dawid, A., Arnold, J., Requena, B., Gresch, A., Płodzień, M., Donatella, K., Nicoli, K. A., Stornati, P., Koch, R., Büttner, M., Okuła, R., Muñoz Gil, G., Vargas-Hernández, R. A., Cervera-Lierta, A., Carrasquilla, J., Dunjko, V., Gabrié, M., Huembeli, P., van Nieuwenburg, E., Vicentini, F., Wang, L., Wetzel, S. J., Carleo, G., Greplová, E., Krems, R., Marquardt, F., Tomza, M., Lewenstein, M., and Dauphin, A. *Machine Learning in Quantum Sciences*. Cambridge University Press, 2025. ISBN 9781009504935. doi: 10.1017/9781009504942. URL https://doi.org/10.1017/9781009504942.

Del Debbio, L., Manca, G. M., and Vicari, E. Critical slowing down of topological modes. *Physics Letters B*, 594(3–4):315–323, 2004. doi: 10.1016/j.physletb.2004.05.038. URL https://doi.org/10.1016/j.physletb.2004.05.038.

Drissi, M., Keeble, J. W. T., Rozalén Sarmiento, J., and Rios, A. Second-order optimization strategies for neural network quantum states. *Philosophical Transactions of the Royal Society A*, 382(2275):20240057, 2024. doi: 10.1098/rsta.2024.0057. URL https://doi.org/10.1098/rsta.2024.0057.

Gauvin-Ndiaye, C., Tindall, J., Moreno, J. R., and Georges, A. Mott transition and volume law entanglement with neural quantum states. *Physical Review Letters*, 134(7):076502, 2025. doi: 10.1103/PhysRevLett.134.076502. URL https://doi.org/10.1103/PhysRevLett.134.076502.

Gispen, W. and Lamacraft, A. Ground states of quantum many body lattice models via reinforcement learning. In *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145 of *Proceedings of Machine Learning Research*, pp. 369–385, 2022. URL https://proceedings.mlr.press/v145/gispen22a.html.

Goldshlager, G., Abrahamsen, N., and Lin, L. A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions. *Journal of Computational Physics*, 516:113351, 2024. doi: 10.1016/j.jcp.2024.113351. URL https://doi.org/10.1016/j.jcp.2024.113351.

Hermann, J., Schätzle, Z., and Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, 2020. doi: 10.1038/s41557-020-0544-y. URL https://doi.org/10.1038/s41557-020-0544-y.

Hibat-Allah, M., Ganahl, M., Hayward, L. E., Melko, R. G., and Carrasquilla, J. Recurrent neural network wave functions. *Physical Review Research*, 2 (2):023358, 2020. doi: 10.1103/PhysRevResearch.2.023358. URL https://doi.org/10.1103/PhysRevResearch.2.023358.

Horn, R. A. and Johnson, C. R. *Matrix Analysis*. Cambridge University Press, Cambridge, 2 edition, 2012. ISBN 9780521839402. doi: 10.1017/CBO9780511810817. URL https://doi.org/10.1017/CBO9780511810817.

Jae, J., Hong, J., Choo, J., and Kwon, Y.-D. Reinforcement learning to learn quantum states for heisenberg scaling accuracy. *Advanced Quantum Technologies*, 8(10), 2025. doi: 10.1002/qute.202500206. URL https://doi.org/10.1002/qute.202500206.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002. URL https://dl.acm.org/doi/10.5555/645531.656005.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. doi: 10.48550/arXiv.1412.6980. URL https://arxiv.org/abs/1412.6980.

Lange, H., Van de Walle, A., Abedinnia, A., and Bohrdt, A. From architectures to applications: A review of neural quantum states. *arXiv preprint arXiv:2402.09402*, 2024. doi: 10.48550/arXiv.2402.09402. URL https://arxiv.org/abs/2402.09402.

Liu, J., Tang, Y., and Zhang, P. Efficient optimization of variational autoregressive networks with natural gradient. *Physical Review E*, 111(2):025304, 2025. doi: 10.1103/PhysRevE.111.025304. URL https://doi.org/10.1103/PhysRevE.111.025304.

Martens, J. and Grosse, R. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2408–2417, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Moss, M. S., Wiersema, R., Hibat-Allah, M., Carrasquilla, J., and Melko, R. G. Leveraging recurrence in neural network wavefunctions for large-scale simulations of heisenberg antiferromagnets on the square lattice. *Physical Review B*, 112(13):134450, 2025. ISSN 2469-9950. doi: 10.1103/6ccd-wzhz.

Moss, M. S., Orfi, A., Roth, C., Sengupta, A. M., Georges, A., Sels, D., Dawid, A., and Valenti, A. Double descent: When do neural quantum states generalize? *Physical Review E*, 113(4):045303, 2026. ISSN 2470-0045. doi: 10.1103/cwmj-fxr4. URL https://link.aps.org/doi/10.1103/cwmj-fxr4.

Nomura, Y. and Imada, M. Dirac-type nodal spin liquid revealed by refined quantum many-body solver using neural-network wave function, correlation ratio, and level spectroscopy. *Physical Review X*, 11(3):031034, 2021. doi: 10.1103/PhysRevX.11.031034. URL https://doi.org/10.1103/PhysRevX.11.031034.

Nys, J., Pescia, G., Sinibaldi, A., and Carleo, G. Ab-initio variational wave functions for the time-dependent many-electron Schrödinger equation. *Nature Communications*, 15(1):9404, 2024. doi: 10.1038/s41467-024-53672-w. URL https://doi.org/10.1038/s41467-024-53672-w.

Peng, B., Zhang, R., Goldstein, D., Alcaide, E., Du, X., Hou, H., Lin, J., Liu, J., Lu, J., Merrill, W., Song, G., Tan, K., Utpala, S., Wilce, N., Wind, J. S., Wu, T., Wuttke, D., and Zhou-Zheng, C. RWKV-7 "goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*, 2025. doi: 10.48550/arXiv.2503.14456. URL https://arxiv.org/abs/2503.14456.

Pescia, G., Nys, J., Kim, J., Lovato, A., and Carleo, G. Message-passing neural quantum states for the homogeneous electron gas. *Physical Review B*, 110(3):035108, 2024. doi: 10.1103/PhysRevB.110.035108. URL https://doi.org/10.1103/PhysRevB.110.035108.

Pfau, D., Spencer, J. S., Matthews, A. G. D. G., and Foulkes, W. M. C. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020. doi: 10.1103/PhysRevResearch.2.033429. URL https://doi.org/10.1103/PhysRevResearch.2.033429.

Ren, Y. and Goldfarb, D. Efficient subsampled Gauss-Newton and natural gradient methods for training neural networks. *arXiv preprint arXiv:1906.02353*, 2019. doi: 10.48550/arXiv.1906.02353. URL https://arxiv.org/abs/1906.02353.

Rende, R., Viteritti, L. L., Bardone, L., Becca, F., and Goldt, S. A simple linear algebra identity to optimize large-scale neural network quantum states. *Communications Physics*, 7(1):260, 2024. doi: 10.1038/s42005-024-01732-4. URL https://doi.org/10.1038/s42005-024-01732-4.

Rende, R., Viteritti, L. L., Becca, F., Scardicchio, A., Laio, A., and Carleo, G. Foundation neural-networks quantum states as a unified ansatz for multiple hamiltonians. *Nature Communications*, 16(1):7213, 2025. doi: 10.1038/s41467-025-62098-x.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

Sarkar, B., Fellows, M., Duque, J. A., Letcher, A., León Villares, A., Sims, A., Wibault, C., Samsonov, D., Cope, D., Liesen, J., Li, K., Seier, L., Wolf, T., Berdica, U., Mohl, V., Goldie, A. D., Courville, A., Sevegnani, K., Whiteson, S., and Foerster, J. N. Evolution strategies at the hyperscale. *arXiv preprint arXiv:2511.16652*, 2026. doi: 10.48550/arXiv.2511.16652. URL https://arxiv.org/abs/2511.16652.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*, 2015. doi: 10.48550/arXiv.1502.05477. URL https://arxiv.org/abs/1502.05477.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347. URL https://arxiv.org/abs/1707.06347.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet—a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018. doi: 10.1063/1.5019779. URL https://doi.org/10.1063/1.5019779.

Sharir, O., Levine, Y., Wies, N., Carleo, G., and Shashua, A. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Physical Review Letters*, 124(2):020503, 2020. doi: 10.1103/PhysRevLett.124.020503. URL https://doi.org/10.1103/PhysRevLett.124.020503.

Shokry, A., Santini, A., and Vicentini, F. When less is more: Approximating the quantum geometric tensor with block structures. *arXiv preprint arXiv:2510.08430*, 2025. doi: 10.48550/arXiv.2510.08430. URL https://arxiv.org/abs/2510.08430.

Smith, L. N. and Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2018. doi: 10.48550/arXiv.1708.07120. URL https://arxiv.org/abs/1708.07120.

Sorella, S. Green function Monte Carlo with stochastic reconfiguration. *Physical Review Letters*, 80(20):4558–4561, 1998. doi: 10.1103/PhysRevLett.80.4558. URL https://doi.org/10.1103/PhysRevLett.80.4558.

Viteritti, L. L., Rende, R., and Becca, F. Transformer variational wave functions for frustrated quantum spin systems. *Physical Review Letters*, 130(23):236401, 2023. ISSN 0031-9007. doi: 10.1103/physrevlett.130.236401. URL https://link.aps.org/doi/10.1103/PhysRevLett.130.236401.

von Glehn, I., Spencer, J. S., and Pfau, D. A self-attention ansatz for ab-initio quantum chemistry. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=xveTeHVlF7j.

Wang, J.-Q., He, R.-Q., and Lu, Z.-Y. Generalized Lanczos method for systematic optimization of neural-network quantum states. *Physical Review B*, 113(8):085120, 2026. doi: 10.1103/PhysRevB.113.085120. URL https://doi.org/10.1103/PhysRevB.113.085120.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Wolff, U. Critical slowing down. *Nuclear Physics B - Proceedings Supplements*, 17:93–102, 1990. doi: 10.1016/0920-5632(90)90224-I. URL https://doi.org/10.1016/0920-5632(90)90224-I.

Wu, D., Rossi, R., Vicentini, F., Astrakhantsev, N., Becca, F., Cao, X., Carrasquilla, J., Ferrari, F., Georges, A., Hibat-Allah, M., Imada, M., Läuchli, A. M., Mazzola, G., Mezzacapo, A., Millis, A., Moreno, J. R., Neupert, T., Nomura, Y., Nys, J., Parcollet, O., Pohle, R., Romero, I., Schmid, M., Silvester, J. M., Sorella, S., Tocchio, L. F.,

Wang, L., White, S. R., Wietek, A., Yang, Q., Yang, Y., Zhang, S., and Carleo, G. Variational benchmarks for quantum many-body problems. *Science*, 386(6719):296–301, 2024. doi: 10.1126/science.adg9774. URL https://doi.org/10.1126/science.adg9774.

Yang, L., Leng, Z., Yu, G., Patel, A., Hu, W.-J., and Pu, H. Deep learning-enhanced variational Monte Carlo method for quantum many-body physics. *Physical Review Research*, 2(1):012039, 2020. doi: 10.1103/PhysRevResearch.2.012039. URL https://doi.org/10.1103/PhysRevResearch.2.012039.

A. Dirac Notation

This appendix provides a self-contained introduction to the Dirac (bra-ket) notation used throughout the paper.

Hilbert space. A quantum state lives in a complex vector space $\mathcal{H}$ called a *Hilbert space*, equipped with an inner product. For a system of N spin-1/2 particles, the Hilbert space is $\mathcal{H} = (\mathbb{C}^2)^{\otimes N}$, which has dimension 2^N. A natural orthonormal basis is the *computational* (spin) basis, whose elements $|\mathbf{s}\rangle$ are indexed by binary spin configurations $\mathbf{s} \in \{\pm 1\}^N$.

Ket vectors. A quantum state is written as a *ket* $|\psi\rangle$, which is simply a column vector in $\mathcal{H}$. Any state can be expanded in the computational basis,

$$|\psi\rangle = \sum_{\mathbf{s}\in\{\pm 1\}^N} \psi(\mathbf{s})\ |\mathbf{s}\rangle\,, \tag{24}$$

where each coefficient $\psi(\mathbf{s}) \in \mathbb{C}$ is the *amplitude* of configuration $\mathbf{s}$. The vector $|\psi\rangle$ is therefore completely specified by the mapping $\mathbf{s} \mapsto \psi(\mathbf{s})$, which NQS parameterize with a neural network.

Bra vectors and inner product. The *bra* $\langle\psi|$ is the conjugate transpose (Hermitian adjoint) of $|\psi\rangle$,

$$\langle\psi| = |\psi\rangle^\dagger\,. \tag{25}$$

The *inner product* of two states is written $\langle\phi|\psi\rangle$, which evaluates to the complex number $\sum_{\mathbf{s}} \phi(\mathbf{s})^* \psi(\mathbf{s})$. When $\phi = \psi$ this gives the squared norm $\langle\psi|\psi\rangle = \sum_{\mathbf{s}} |\psi(\mathbf{s})|^2 \geq 0$, with equality only for the zero vector. The basis states are orthonormal:

$$\langle\mathbf{s}|\mathbf{s}'\rangle = \delta_{\mathbf{s},\mathbf{s}'}\,, \tag{26}$$

where $\delta_{\mathbf{s},\mathbf{s}'}$ is the Kronecker delta.

Completeness relation. The basis states form a complete set, meaning the *identity operator* $\mathbb{1}$ can be resolved as

$$\sum_{\mathbf{s}} |\mathbf{s}\rangle\,\langle\mathbf{s}| = \mathbb{1}. \tag{27}$$

This is used repeatedly in derivations (e.g. Appendix B.2) to insert a basis expansion and convert abstract operator equations into sums over spin configurations.

Operators and matrix elements. A quantum *operator* $\hat{A}$ is a linear map $\mathcal{H} \to \mathcal{H}$, represented as a $2^N \times 2^N$ matrix in the computational basis. The entry at row $\mathbf{s}$ and column $\mathbf{s}'$ is the *matrix element*

$$\langle\mathbf{s}|\hat{A}|\mathbf{s}'\rangle = \langle\mathbf{s}|\Big(\hat{A}\,|\mathbf{s}'\rangle\Big) = (\mathbf{A})_{\mathbf{s},\mathbf{s}'}. \tag{28}$$

An operator is *Hermitian* (or self-adjoint) if $\hat{A} = \hat{A}^\dagger$, which implies real eigenvalues. The Hamiltonian $\hat{H}$ is always Hermitian because energy is a real-valued observable.

Expectation value. The expected value of an observable $\hat{A}$ in state $|\psi\rangle$ is

$$\langle\hat{A}\rangle_\psi := \frac{\langle\psi|\hat{A}|\psi\rangle}{\langle\psi|\psi\rangle}. \tag{29}$$

For the Hamiltonian, this gives the variational energy $E[\psi] = \langle\psi|\hat{H}|\psi\rangle\,/\,\langle\psi|\psi\rangle$ that NQS minimizes. When $|\psi\rangle$ is normalized ($\langle\psi|\psi\rangle = 1$), the denominator can be dropped.

Summary table.

Dirac symbol	Linear-algebra equivalent	Meaning in this paper
$\lvert\psi\rangle$	column vector in $\mathbb{C}^{2^N}$	quantum state / wavefunction
$\langle\psi\rvert$	conjugate row vector	dual / adjoint state
$\langle\phi\vert\psi\rangle$	$\boldsymbol{\phi}^\dagger\boldsymbol{\psi}$ (dot product)	inner product; overlap of two states
$\langle\psi\vert\psi\rangle$	$\lVert\boldsymbol{\psi}\rVert^2$	squared norm; $= 1$ for normalized states
$\lvert\mathbf{s}\rangle$	standard basis vector $e_{\mathbf{s}}$	spin-configuration basis state
$\langle\mathbf{s}\vert\hat{H}\vert\mathbf{s}'\rangle$	$(\mathbf{H})_{\mathbf{s},\mathbf{s}'}$ matrix entry	Hamiltonian matrix element
$\langle\psi\vert\hat{H}\vert\psi\rangle$	$\boldsymbol{\psi}^\dagger\mathbf{H}\boldsymbol{\psi}$ (quadratic form)	(unnormalized) variational energy
$\sum_{\mathbf{s}} \lvert\mathbf{s}\rangle\,\langle\mathbf{s}\rvert$	$\mathbb{1}_{2^N}$ (identity matrix)	completeness / resolution of identity

B. Mathematical Statements

B.1. Proof of Proposition 3.1

Proposition B.1 (Policy-gradient form of variational energy minimization)**.** *Assume the Hamiltonian is stoquastic, i.e. its matrix representation in a chosen computational basis has non-positive off-diagonal elements. Then we can assume $f_{\boldsymbol{\theta}} = \log \psi_{\boldsymbol{\theta}} = \log |\psi_{\boldsymbol{\theta}}|$, and the gradient of the variational energy can be written in policy-gradient form as*

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\big(E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\big)\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\big]. \tag{16}$$

In particular, variational energy minimization is equivalent to an advantage policy-gradient update over configurations. For a proof, see Appendix B.1.

Proof. We follow the proof strategy of the policy gradient theorem (Agarwal et al., 2021) to connect the standard NQS/VMC gradient estimator with REINFORCE-style updates when the probability distribution is parameterized autoregressively.

Differentiating gives

$$\begin{aligned}
\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] &= \boldsymbol{\nabla}_{\boldsymbol{\theta}} \sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\, E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) && (30)\\
&= \sum_{\mathbf{s}} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\, E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) \;+\; \sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) && (31)\\
&= \sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\, E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) \;+\; \sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) && (32)\\
&= \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\big] \;+\; \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\boldsymbol{\nabla}_{\boldsymbol{\theta}} E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\big]. && (33)
\end{aligned}$$

We now analyze the second term. Define the log-derivative observable

$$\mathbf{O}(\mathbf{s}) := \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \psi_{\boldsymbol{\theta}}(\mathbf{s}),$$

then

$$\begin{aligned}
\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\boldsymbol{\nabla}_{\boldsymbol{\theta}} E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\big] &= \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \sum_{\mathbf{s}'} \frac{\psi_{\boldsymbol{\theta}}(\mathbf{s}')}{\psi_{\boldsymbol{\theta}}(\mathbf{s})} \langle \mathbf{s}|\hat{H}|\mathbf{s}'\rangle\right] && (34)\\
&= \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[\sum_{\mathbf{s}'} \langle \mathbf{s}|\hat{H}|\mathbf{s}'\rangle\, \frac{(\psi_{\boldsymbol{\theta}}(\mathbf{s})\boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_{\boldsymbol{\theta}}(\mathbf{s}') - \boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_{\boldsymbol{\theta}}(\mathbf{s})\psi_{\boldsymbol{\theta}}(\mathbf{s}'))}{\psi_{\boldsymbol{\theta}}(\mathbf{s})^2}\right] && (35)\\
&= \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[\sum_{\mathbf{s}'} \langle \mathbf{s}|\hat{H}|\mathbf{s}'\rangle\, \frac{\psi_{\boldsymbol{\theta}}(\mathbf{s}')}{\psi_{\boldsymbol{\theta}}(\mathbf{s})}(\mathbf{O}(\mathbf{s}') - \mathbf{O}(\mathbf{s}))\right] && (36)\\
&= \sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) \sum_{\mathbf{s}'} \hat{H}_{\mathbf{s},\mathbf{s}'} \frac{\psi_{\boldsymbol{\theta}}(\mathbf{s}')}{\psi_{\boldsymbol{\theta}}(\mathbf{s})}(\mathbf{O}(\mathbf{s}') - \mathbf{O}(\mathbf{s})), && (37)
\end{aligned}$$

where $\hat{H}_{\mathbf{s},\mathbf{s}'} = \langle \mathbf{s}|\hat{H}|\mathbf{s}'\rangle$, which gives the double-sum form

$$\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\boldsymbol{\nabla}_{\boldsymbol{\theta}} E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\big] = \frac{1}{Z_{\boldsymbol{\theta}}} \sum_{\mathbf{s},\mathbf{s}'} \psi_{\boldsymbol{\theta}}(\mathbf{s})\psi_{\boldsymbol{\theta}}(\mathbf{s}')\, \hat{H}_{\mathbf{s},\mathbf{s}'}(\mathbf{O}(\mathbf{s}') - \mathbf{O}(\mathbf{s})). \tag{38}$$

At this point, the only remaining question is whether the second term in (38) vanishes. This happens when the Hamiltonian is stoquastic in the chosen basis. In that case, its matrix elements are real and Hermiticity implies $\hat{H}_{\mathbf{s},\mathbf{s}'} = \hat{H}_{\mathbf{s}',\mathbf{s}}$. Moreover, according to Perron-Frobenius (Horn & Johnson, 2012), all the components of the ground state wavefunction of a stoquastic Hamiltonian are real and strictly positive, so $\psi_{\boldsymbol{\theta}}(\mathbf{s}) = |\psi_{\boldsymbol{\theta}}(\mathbf{s})|$ and the log-derivative observable $\mathbf{O}(\mathbf{s})$ is real-valued.

Under this assumption, the factor $\psi_{\boldsymbol{\theta}}(\mathbf{s})\psi_{\boldsymbol{\theta}}(\mathbf{s}')\hat{H}_{\mathbf{s},\mathbf{s}'}$ in (38) is symmetric under exchanging $\mathbf{s}$ and $\mathbf{s}'$, whereas the difference $\mathbf{O}(\mathbf{s}') - \mathbf{O}(\mathbf{s})$ is antisymmetric. Therefore, each term in the double sum cancels with the term obtained by swapping $\mathbf{s}$ and $\mathbf{s}'$, and the whole sum is zero. Hence,

$$\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\boldsymbol{\nabla}_{\boldsymbol{\theta}} E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\big] = 0. \tag{39}$$

Substituting back into (33) yields the pure score-function (policy-gradient) estimator:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\big]. \tag{40}$$

Moreover, for any constant c that doesn't depend on $\mathbf{s}$,

$$\begin{aligned}
\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[c\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})] &= c\sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) && (41)\\
&= c\sum_{\mathbf{s}} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) && (42)\\
&= c\boldsymbol{\nabla}_{\boldsymbol{\theta}} \sum_{\mathbf{s}} \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) && (43)\\
&= c\boldsymbol{\nabla}_{\boldsymbol{\theta}} 1 = 0. && (44)
\end{aligned}$$

So we may subtract any constant baseline without bias (Agarwal et al., 2021). Choosing the baseline as the global energy $E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]$ gives the form

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\big[\big(E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\big)\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})\big], \tag{45}$$

which is analogous to the Advantage form of the REINFORCE estimator (Agarwal et al., 2021). □

B.2. Variational Gradient Derivation

In this section, we show how to derive the well known VMC-gradient in Eq. (2.4) (Carleo & Troyer, 2017) from Eq. (9). We start from the general expression for the energy, valid for any (not necessarily normalized) state $|\psi_{\boldsymbol{\theta}}\rangle$:

$$E[\psi_{\boldsymbol{\theta}}] = \frac{\langle\psi_{\boldsymbol{\theta}}|\hat{H}|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle}. \tag{46}$$

Differentiating with respect to θ_k using the quotient rule and the Hermiticity of $\hat{H}$ gives:

$$\partial_{\theta_k} E[\psi_{\boldsymbol{\theta}}] = \frac{2\,\mathrm{Re}\left\{(\partial_{\theta_k}\langle\psi_{\boldsymbol{\theta}}|)\hat{H}|\psi_{\boldsymbol{\theta}}\rangle\right\}}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle} - E[\psi_{\boldsymbol{\theta}}]\frac{2\,\mathrm{Re}\left\{(\partial_{\theta_k}\langle\psi_{\boldsymbol{\theta}}|)|\psi_{\boldsymbol{\theta}}\rangle\right\}}{(\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle)^2}. \tag{47}$$

Since the final result is norm-invariant, we now choose the convenient normalization $\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle = 1$, which simplifies the expression to:

$$\partial_{\theta_k} E[\psi_{\boldsymbol{\theta}}] = 2\,\mathrm{Re}\left\{(\partial_{\theta_k}\langle\psi_{\boldsymbol{\theta}}|)\big(\hat{H} - E[\psi_{\boldsymbol{\theta}}]\big)|\psi_{\boldsymbol{\theta}}\rangle\right\}. \tag{48}$$

We insert the completeness relation $\sum_{\mathbf{s}} |\mathbf{s}\rangle\langle\mathbf{s}| = \mathbb{1}$ twice and use $\mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) = |\psi_{\boldsymbol{\theta}}(\mathbf{s})|^2$:

$$\begin{aligned}
(\partial_{\theta_k}\langle\psi_{\boldsymbol{\theta}}|)\big(\hat{H} - E[\psi_{\boldsymbol{\theta}}]\big)|\psi_{\boldsymbol{\theta}}\rangle &= \sum_{\mathbf{s}} (\partial_{\theta_k}\langle\psi_{\boldsymbol{\theta}}|)|\mathbf{s}\rangle\, \langle\mathbf{s}|\big(\hat{H} - E[\psi_{\boldsymbol{\theta}}]\big)|\psi_{\boldsymbol{\theta}}\rangle && (49)\\
&= \sum_{\mathbf{s}} \partial_{\theta_k}\psi^*_{\boldsymbol{\theta}}(\mathbf{s}) \left(\sum_{\mathbf{s}'} \langle\mathbf{s}|\hat{H}|\mathbf{s}'\rangle\langle\mathbf{s}'|\psi_{\boldsymbol{\theta}}\rangle - E[\psi_{\boldsymbol{\theta}}]\,\psi_{\boldsymbol{\theta}}(\mathbf{s})\right) && (50)\\
&= \sum_{\mathbf{s}} \psi^*_{\boldsymbol{\theta}}(\mathbf{s}) O^*_k(\mathbf{s}) \left(\sum_{\mathbf{s}'} \hat{H}_{\mathbf{s},\mathbf{s}'}\psi_{\boldsymbol{\theta}}(\mathbf{s}') - E[\psi_{\boldsymbol{\theta}}]\,\psi_{\boldsymbol{\theta}}(\mathbf{s})\right) && (51)\\
&= \sum_{\mathbf{s}} |\psi_{\boldsymbol{\theta}}(\mathbf{s})|^2\, O^*_k(\mathbf{s})\, \big(E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - E[\psi_{\boldsymbol{\theta}}]\big) && (52)\\
&= \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[O^*_k(\mathbf{s})\Big(E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\Big)\right], && (53)\\
& && (54)
\end{aligned}$$

where in the last step we used that $E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]$. Therefore,

$$\partial_{\theta_k} E[\psi_{\boldsymbol{\theta}}] = 2\,\mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[O^*_k(\mathbf{s})\Big(E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\Big)\right]\right\}. \tag{55}$$

B.3. Variational Gradient Decomposition

For the development of a practical algorithm we can decompose the VMC gradient into its probability and phase components as follows. Starting from the standard VMC/NQS gradient estimator we can write

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[\left(E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]\right)\mathbf{O}(\mathbf{s})^*\right]\right\}. \tag{56}$$

Recall that $*$ is the complex complement, and

$$E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathrm{Re}\left\{E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\right\} + i\cdot\mathrm{Im}\left\{E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\right\}, \quad \mathbf{O}(\mathbf{s}) = \mathrm{Re}\left\{\mathbf{O}(\mathbf{s})\right\} + i\cdot\mathrm{Im}\left\{\mathbf{O}(\mathbf{s})\right\}. \tag{57}$$

For convenience, we denote $E_R := \mathrm{Re}\left\{E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\right\}$ and $E_I := \mathrm{Im}\left\{E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})\right\}$ (analogously for $\mathbf{O}(\mathbf{s})$). We can now write

$$\begin{aligned}
\boldsymbol{\nabla}_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] &= 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[\left((E_R - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})]) + iE_I\right)(O_R - iO_I)\right]\right\} && (58)\\
&= 2\ \mathrm{Re}\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[(E_R - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})])O_R \\
&\quad - i(E_R - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})])O_I + iE_I O_R + E_I O_I]\} && (59)\\
&= 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}\left[(E_R - \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E^{\mathrm{loc}}_{\boldsymbol{\theta}}(\mathbf{s})])O_R\right]\right\} + 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I O_I]\right\}, && (60)
\end{aligned}$$

which shows exactly how to update our model, when we parameterize the real and imaginary parts of the amplitude separately. Moreover, we can subtract a constant baseline, c, from the second term (corresponding to the imaginary parts of our parameterization) to reduce the variance:

$$\begin{aligned}
2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I(O_I - c)]\right\} &= 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I O_I]\right\} - 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I c]\right\} && (61)\\
&= 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I O_I]\right\} - 2c\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I]\right\} && (62)\\
&= 2\ \mathrm{Re}\left\{\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I O_I]\right\}, && (63)
\end{aligned}$$

where line 63 comes from the reality of expected values under Hermitian operators. So like before, we may subtract a baseline to reduce variance.

To find the variance-minimizing constant, consider the one-sample estimator for the imaginary contribution to the gradient

$$g_I(\mathbf{s}; c) := 2\, E_I(\mathbf{s})(O_I(\mathbf{s}) - c). \tag{64}$$

Since $\mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[E_I(\mathbf{s})] = 0$, subtracting any constant c leaves the estimator unbiased:

$$\mathbb{E}[g_I(\mathbf{s}; c)] = 2\,\mathbb{E}[E_I(\mathbf{s})\, O_I(\mathbf{s})]\,. \tag{65}$$

Therefore, the optimal constant is the one that minimizes $\mathbb{V}(g_I(\mathbf{s}; c))$, or equivalently its second moment:

$$\underset{c}{\mathrm{argmin}}\ \mathbb{E}\Big[E_I(\mathbf{s})^2(O_I(\mathbf{s}) - c)^2\Big]\,. \tag{66}$$

Differentiating with respect to c and setting the derivative to zero gives

$$\begin{aligned}
0 &= \frac{\partial}{\partial c}\mathbb{E}\left[E_I(\mathbf{s})^2(O_I(\mathbf{s}) - c)^2\right] && (67)\\
&= -2\,\mathbb{E}\left[E_I(\mathbf{s})^2(O_I(\mathbf{s}) - c)\right], && (68)
\end{aligned}$$

hence the variance-minimizing baseline is

$$c^\star = \frac{\mathbb{E}\big[E_I(\mathbf{s})^2\, O_I(\mathbf{s})\big]}{\mathbb{E}\left[E_I(\mathbf{s})^2\right]}. \tag{69}$$

If $O_I(\mathbf{s})$ is vector-valued, this formula is applied coordinate-wise. We try using this minimizer in practice, but its implementation requires adding an extra forward and backwards pass that make it significantly slower compared to vanilla gradient expression.

B.4. Clipping the Phase Increment

The phase of the wavefunction is only defined modulo 2π, so the natural notion of distance between two phases is not the raw difference $\phi_{\boldsymbol{\theta}}(\mathbf{s}) - \phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})$, but the wrapped increment

$$\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s}) := \operatorname{atan2}\big(\sin(\phi_{\boldsymbol{\theta}}(\mathbf{s}) - \phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})), \cos(\phi_{\boldsymbol{\theta}}(\mathbf{s}) - \phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s}))\big), \tag{70}$$

which always lies in $[-\pi, \pi]$. This is the shortest angular displacement between the old and new phases. A natural way to measure how much the phase changes is to compare the corresponding unit-modulus complex numbers $e^{i\phi_{\boldsymbol{\theta}}(\mathbf{s})}$ and $e^{i\phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})}$. Their squared Euclidean distance is

$$\left|e^{i\phi_{\boldsymbol{\theta}}(\mathbf{s})} - e^{i\phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})}\right|^2 = 2\big(1 - \cos(\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s}))\big). \tag{71}$$

Therefore, bounding the wrapped phase increment by $|\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s})| \leq \delta$ directly bounds the change in phase on the unit circle:

$$\left|e^{i\phi_{\boldsymbol{\theta}}(\mathbf{s})} - e^{i\phi_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})}\right|^2 \leq 2(1 - \cos\delta) \approx \delta^2 \qquad \text{for small } \delta. \tag{72}$$

In other words, clipping $\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s})$ prevents abrupt rotations of the wavefunction phase and enforces a local trust region on the phase channel. This is the phase analogue of PPO clipping: instead of limiting how much the probability ratio can change, we limit how far the phase can move along the circle in a single update.

B.5. Importance Sampling Ratio for the Phase Update

During the inner updates of PWO, we sample configurations once from the reference distribution $\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}$ and then reuse the same batch for multiple optimization steps. As a result, after the first inner step the cached samples are no longer distributed according to the current Born distribution $\mathcal{P}_{\boldsymbol{\theta}}$. This creates an off-policy mismatch in the phase update. Let

$$h_{\boldsymbol{\theta}}(\mathbf{s}) := \max\big(\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s})A^I(\mathbf{s}), \text{clip}\Delta\phi_{\boldsymbol{\theta}}(\mathbf{s}), -\delta, \delta)A^I(\mathbf{s})\big)\,, \tag{73}$$

denote the clipped phase surrogate for a single configuration, where $A^I(\mathbf{s})$ is the imaginary advantage computed from the reference batch. The phase objective we would like to optimize is the expectation of this quantity under the *current* distribution:

$$J_{\text{phase}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[h_{\boldsymbol{\theta}}(\mathbf{s})]\,. \tag{74}$$

However, our samples come from $\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}$ rather than $\mathcal{P}_{\boldsymbol{\theta}}$. Using importance sampling, we can rewrite this objective as

$$J_{\text{phase}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}[r_{\boldsymbol{\theta}}(\mathbf{s})\, h_{\boldsymbol{\theta}}(\mathbf{s})]\,, \qquad r_{\boldsymbol{\theta}}(\mathbf{s}) := \frac{\mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s})}{\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})} = \exp(\log\mathcal{P}_{\boldsymbol{\theta}}(\mathbf{s}) - \log\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s})). \tag{75}$$

This shows why the importance sampling ratio is necessary: without it, the batch average would estimate an expectation under the wrong distribution. At the same time, we do *not* want the phase update to differentiate through the ratio itself. Differentiating the reweighted objective naively gives us the following

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} J_{\text{phase}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}[r_{\boldsymbol{\theta}}(\mathbf{s})\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}}(\mathbf{s})] + \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}[h_{\boldsymbol{\theta}}(\mathbf{s})\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s})]\,. \tag{76}$$

The first term is the desired phase gradient under the current distribution. The second term appears only because the samples are reused off-policy, and it differentiates through the sampling correction itself. This introduces an additional score-function term that changes the probability mass, whereas the role of the phase objective is only to control the evolution of the phase. For this reason, PWO treats the importance ratio as a pure change-of-measure weight and stops gradients through it:

$$\widetilde{J}_{\text{phase}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}[\text{sg}(r_{\boldsymbol{\theta}}(\mathbf{s}))\, h_{\boldsymbol{\theta}}(\mathbf{s})]\,. \tag{77}$$

Differentiating now yields

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \widetilde{J}_{\text{phase}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}}[\text{sg}(r_{\boldsymbol{\theta}}(\mathbf{s}))\, \boldsymbol{\nabla}_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}}(\mathbf{s})] = \mathbb{E}_{\mathbf{s}\sim\mathcal{P}_{\boldsymbol{\theta}}}[\boldsymbol{\nabla}_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}}(\mathbf{s})]\,. \tag{78}$$

Therefore, the detached importance ratio is exactly what allows us to reuse samples from the reference model while still taking a phase gradient corresponding to the current distribution. Note that if the amplitude were kept fixed during the inner loop, then $\mathcal{P}_{\boldsymbol{\theta}} = \mathcal{P}_{\boldsymbol{\theta}_{\text{old}}}$ and $r_{\boldsymbol{\theta}}(\mathbf{s}) \equiv 1$, so no correction would be needed. In PWO, however, amplitude and phase updates are interleaved while the same batch is reused, making this correction necessary.

C. Additional Sections

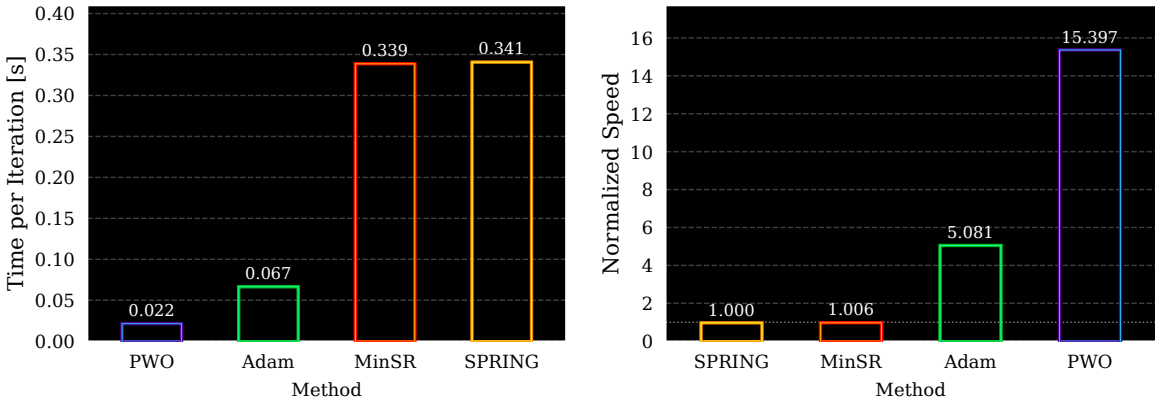
C.1. Numerical Cost of Stochastic Reconfiguration and Improvements

A significant bottleneck arises when using the original SR formulation for optimization. The vanilla implementation involves inverting a $P \times P$ matrix, where P denotes the number of parameters. Alternatively, (Chen & Heyl, 2024) showed that this inversion problem can be recast into one that inverts a matrix of size $M \times M$, where M is the number of samples used per iteration in the Monte Carlo estimation – this goes by the name of minSR and builds upon ideas from the machine learning optimization literature (Ren & Goldfarb, 2019). The required operations for minSR are $O(M^2P) + O(M^3)$ instead of $O(P^3)$, and the memory usage is only $O(MP)$ instead of $O(P^2)$. It was used to train a Vision Transformer with around 3×10^5 parameters, giving the lowest energy so far for the difficult benchmark of the critical point of the 10×10 J1-J2 Heisenberg system (Rende et al., 2024).

Despite the improvements brought by minSR, several implementations of NQS optimization, especially for electronic structure problems, employ KFAC (Martens & Grosse, 2015) instead of SR. The $O(M^3)$ cost can still be prohibitive when training networks for problems that require a large number of samples; KFAC provides a practical compromise: it avoids explicit matrix inversion through block-diagonal approximation, offering scalability at the potential cost of missing information about the geometry of the Hilbert space. KFAC is commonly used for applications to fermionic systems (Choo et al., 2020), where NQS achieved remarkable results, including architectures like FermiNet (Pfau et al., 2020), SchNet (Schütt et al., 2018), PauliNet (Hermann et al., 2020), and more recent transformer-based approaches (von Glehn et al., 2023).

Recent advances have aimed to improve upon both approaches. These include SPRING, which combines minSR and KFAC to avoid explicit inversion (Goldshlager et al., 2024), methods that exploit block-diagonal structure in the quantum geometric tensor to improve conditioning and scalability (Nys et al., 2024; Shokry et al., 2025), and variational Lanczos techniques that accelerate convergence by extracting information across multiple eigenstates (Wang et al., 2026). For a more detailed analysis of improvements in second-order optimization of NQS, the reader can refer to (Drissi et al., 2024).

C.2. Relative Speed of PWO per Iteration



(a) Wall-clock time per optimizer iteration.

(b) Normalized iteration speed relative to SPRING.

Figure 6. Per-iteration computational cost and normalized iteration speed for PWO and the baseline optimizers on the Heisenberg J_1–J_2 chain with the default architecture D.2. Both measurements are obtained on a single NVIDIA L40S GPU. Normalized speed is reported as the number of optimizer iterations per unit time relative to SPRING, so higher is faster.

Beyond convergence in wall-clock time, it is useful to isolate the computational cost of each optimization update. Figure 6b compares the normalized iteration speed (optimization steps per unit of time) of PWO against the baselines. The speed advantage over minSR is expected: PWO is a first-order method and does not require constructing or inverting the stochastic-reconfiguration matrix. In contrast, minSR replaces the original $P \times P$ SR solve by an $M \times M$ solve, where M is the number of samples, but it still requires expensive Jacobian contractions and a matrix inversion or linear solve. PWO avoids this curvature computation entirely.

The comparison with Adam is more subtle. A single PWO update has additional bookkeeping relative to Adam, including probability ratios, clipping terms, and wrapped phase increments. However, in VMC the dominant cost is often not the optimizer algebra itself, but sampling configurations and evaluating the corresponding local energies. PWO computes local energies and advantages once for a sampled batch and then reuses that batch across multiple proximal inner epochs. Adam, in contrast, performs a single update per sampled batch, so the local-energy overhead is paid again for each parameter update. Thus, PWO can amortize the expensive energy computation across several optimization steps, making its effective per-update cost competitive with Adam despite the more structured surrogate objective.

C.3. Limitations of PWO

Our experiments are a controlled optimizer comparison rather than a comprehensive NQS benchmark: all main results use one-dimensional spin-$1/2$ chains with $N = 12$, periodic boundary conditions, and exact ground-state energies. Thus, the results do not yet establish performance on larger systems, higher-dimensional lattices, fermionic systems, or continuum electronic-structure problems. The policy-gradient equivalence in Proposition 3.1 also holds under restrictive assumptions, namely stoquastic Hamiltonians; the full complex-valued setting is handled by the clipped phase surrogate, which is a practical approximation rather than an exact trust-region method. Finally, although PWO avoids explicit curvature solves and reuses samples across inner updates, its performance may still depend on local-energy evaluation cost, sampling efficiency, estimator variance, numerical precision, and hyperparameter tuning.

D. Experimental Details

D.1. Hyperparameter search

For each method, we perform a small hyperparameter search over the learning rate using a grid sweep and select the configuration with the best validation performance. For PWO, we additionally tune the clipping parameters that control the amplitude and phase surrogate objectives. Specifically, after selecting an initial learning rate, we independently grid search each clipping parameter while holding the best values found so far fixed. This sequential procedure provides a simple and computationally tractable way to tune the additional PWO-specific hyperparameters without requiring an exhaustive joint sweep over all combinations.

For Adam-based methods, including PWO, we use a cosine one-cycle learning-rate schedule (Smith & Topin, 2018). The learning rate is first increased to a peak value during a short warmup phase and is then annealed smoothly to a small final value by cosine decay. The peak learning rate is selected by the grid search above, while the other schedule parameters are fixed across runs. For PWO, which performs multiple optimization epochs per sampled batch, we scale the transition horizon by the number of PPO epochs so that the schedule evolves on a comparable outer-iteration timescale. For minSR and SPRING, a constant learning rate performed better than scheduled variants.

Because the Heisenberg and J1–J2 chains are more challenging optimization problems than the transverse-field Ising model, we use longer learning-rate schedules for these settings (compare Appendices D.3-D.5). In particular, we allocate more transition steps to the Heisenberg chain than to Ising, and more transition steps to J1–J2 than to Heisenberg. This gives the optimizer a longer annealing horizon on the harder Hamiltonians, while keeping the schedule structure fixed across problems. To improve readability, we report the method-specific hyper-parameters for each Hamiltonian separately. The Ising configuration is shown first, followed by the Heisenberg and J1–J2 settings in the subsequent subsections.

D.2. Neural Quantum State Architecture

Across experiments, we use the same autoregressive recurrent neural network for PWO and minSR, ensuring that performance differences arise from the optimizer rather than the parametrization. Given a spin configuration $\mathbf{s} \in \{\pm 1\}^N$, we map spins to binary tokens, prepend a beginning-of-sequence token, and embed the resulting sequence with learned token embeddings of dimension 32 and learned positional embeddings. The embedded inputs are projected to dimension 256, followed by a $\tanh$ nonlinearity and layer normalization. The backbone consists of GRU layers with residual connections between recurrent blocks. From the shared recurrent representation, we use two separate two-layer MLP heads and GELU activations: an amplitude head, which outputs conditional log-probabilities via a log-softmax, and a phase head, which outputs bounded phases through a $\tanh$ nonlinearity scaled by π. The model has 1.4M parameters. Unless otherwise stated, all methods and Hamiltonians use the same architecture. The shared hyperparameters are summarized in Table 2.

Architecture hyperparameter	Value
Token embedding dimension	32
Site embeddings	Learned
Backbone input dimension	256
Backbone nonlinearity	$\tanh$
Backbone normalization	Layer normalization
Recurrent backbone	GRU
Number of GRU layers	3
GRU hidden dimension	256
Amplitude head	2-layer MLP
Phase head	2-layer MLP
Head hidden dimension	256
Head activation	GELU
Phase output	$\pi \tanh(\cdot)$
Phase scale	π

Table 2. Shared NQS architecture used across all Hamiltonians and optimization methods. The same autoregressive GRU-based parameterization is used for Adam, PWO, minSR, and SPRING.

D.3. Ising Model

Hyperparameter	Adam	PWO	minSR	SPRING
Optimizer / method	Adam	Adam + PWO	minSR	SPRING
Learning rate	10^{-5}	10^{-5}	10^{-2}	10^{-2}
Peak learning rate	10^{-4}	10^{-4}	–	–
Transition steps	5,000	20,000	–	–
PPO epochs	–	4	–	–
PPO clip ϵ	–	10^{-3}	–	–
Advantage normalization	–	Yes	–	–
Phase loss	–	$\Delta\phi$ clip	–	–
Phase coefficient	–	1.0	–	–
Phase clip	–	0.3	–	–
Center imaginary advantage	–	Yes	–	–
Normalize imaginary advantage	–	Yes	–	–
Phase Jacobian baseline	–	Yes	–	–
SR diagonal shift	–	–	10^{-2}	10^{-2}
NTK / minSR mode	–	–	Yes	Yes
On-the-fly SR	–	–	Yes	Yes
SPRING momentum	–	–	–	0.8

Table 3. Hyperparameters used for the Ising experiments. All methods use $N = 12$, periodic boundary conditions, $J = 1$, $h = 1$, complex-valued neural quantum states, 1024 training samples, exact diagonalization for evaluation, and evaluation every 200 iterations. Adam and PWO use the cosine one-cycle schedule.

D.4. Heisenberg Chain

Hyperparameter	Adam	PWO	minSR	SPRING
Optimizer / method	Adam	Adam + PWO	minSR	SPRING
LR parameter / constant LR	10^{-5}	10^{-5}	10^{-3}	10^{-3}
Peak learning rate	3×10^{-4}	10^{-4}	–	–
Transition steps	10,000	40,000	–	–
PPO epochs	–	4	–	–
PPO clip ϵ	–	10^{-3}	–	–
Advantage normalization	–	Yes	–	–
Phase loss	–	$\Delta\phi$ clip	–	–
Phase coefficient	–	1.0	–	–
Phase clip	–	0.3	–	–
Center imaginary advantage	–	Yes	–	–
Normalize imaginary advantage	–	Yes	–	–
Phase Jacobian baseline	–	Yes	–	–
SR diagonal shift	–	–	10^{-2}	10^{-2}
NTK / minSR mode	–	–	Yes	Yes
On-the-fly SR	–	–	Yes	Yes
SPRING momentum	–	–	–	0.8

Table 4. Hyperparameters for the Heisenberg-chain experiments. All methods use $N = 12$, periodic boundary conditions, coupling $J = 0.25$, no sign rule, complex-valued neural quantum states, 1024 training samples, exact evaluation, and evaluation every 200 iterations. Adam and PWO use a cosine one-cycle learning-rate schedule; minSR and SPRING use a constant learning rate.

D.5. Heisenberg J_1–J_2 Chain

Hyperparameter	Adam	PWO	minSR	SPRING
Optimizer / method	Adam	Adam + PWO	minSR	SPRING
LR parameter / constant LR	10^{-5}	10^{-5}	10^{-3}	10^{-3}
Peak learning rate	3×10^{-4}	10^{-4}	–	–
Transition steps	10,000	40,000	–	–
PPO epochs	–	4	–	–
PPO clip ϵ	–	10^{-3}	–	–
Advantage normalization	–	Yes	–	–
Phase loss	–	$\Delta\phi$ clip	–	–
Phase coefficient	–	1.0	–	–
Phase clip	–	0.3	–	–
Center imaginary advantage	–	Yes	–	–
Normalize imaginary advantage	–	Yes	–	–
Phase Jacobian baseline	–	Yes	–	–
SR diagonal shift	–	–	10^{-2}	10^{-2}
NTK / minSR mode	–	–	Yes	Yes
On-the-fly SR	–	–	Yes	Yes
SPRING momentum	–	–	–	0.8

Table 5. Hyperparameters for the frustrated Heisenberg J_1–J_2 experiments. All methods use $N = 12$, periodic boundary conditions, couplings $J_1 = 1$ and $J_2 = 0.5$, no sign rule, complex-valued neural quantum states, 1024 training samples, exact evaluation, and evaluation every 200 iterations. Adam and PWO use a cosine one-cycle learning-rate schedule; minSR and SPRING use a constant learning rate.

D.6. Scaling Experiment

To study how performance scales with model capacity, we train three model sizes on the frustrated Heisenberg J_1–J_2 chain ($N = 12$, $J_1 = 1$, $J_2 = 0.5$, periodic boundary conditions) using PWO, Adam, and minSR. All three sizes share the same autoregressive GRU architecture described in Appendix D.2; they differ only in the number of recurrent layers and hidden dimensions. Table 6 summarises the sizes and the corresponding total parameter counts.

Size	GRU layers	RNN hidden	Head hidden	Parameters
Tiny	1	64	64	44,452
Small	2	128	128	269,156
Medium	3	256	256	1,456,356

Table 6. Model sizes used in the scaling experiment. All models use an embedding dimension of 32 and are evaluated on the frustrated Heisenberg J_1–J_2 chain with $N = 12$ sites. Parameter counts include all weights and biases of the full autoregressive network (embedding, recurrent backbone, amplitude head, and phase head).

The hyperparameters used for each optimization method in the scaling experiment are listed in Table 7. The schedule and clipping parameters are identical to those of the main J_1–J_2 experiments (see Tables 5 and 2); only the number of transition steps differs to account for the change in model size.

Hyperparameter	Adam	PWO	minSR
Optimizer / method	Adam	Adam + PWO	minSR
Learning rate	10^{-5}	10^{-5}	10^{-3}
Peak learning rate	3×10^{-4}	10^{-4}	–
Transition steps	40,000	40,000	–
PPO epochs	–	4	–
PPO clip ϵ	–	10^{-3}	–
Advantage normalization	–	Yes	–
Phase loss	–	$\Delta\phi$ clip	–
Phase coefficient	–	1.0	–
Phase clip	–	0.3	–
Center imaginary advantage	–	Yes	–
Normalize imaginary advantage	–	Yes	–
Phase Jacobian baseline	–	Yes	–
SR diagonal shift	–	–	10^{-2}
NTK / minSR mode	–	–	Yes
On-the-fly SR	–	–	Yes
Training samples		2,048	
Seeds		10	

Table 7. Hyperparameters for the scaling experiment on the frustrated Heisenberg J_1–J_2 chain ($N = 12$, $J_1 = 1$, $J_2 = 0.5$, periodic boundary conditions). All methods and sizes use the same optimization hyperparameters; only the model architecture varies across runs (see Table 6). Adam and PWO use a cosine one-cycle learning-rate schedule; minSR uses a constant learning rate.

D.7. RWKV-7 on Ising Model

Hyperparameter	Adam	PWO
Optimizer / method	Adam	Adam + PWO
Model	RWKV-7	RWKV-7
Model size	1.5B	1.5B
Learning rate	10^{-5}	10^{-5}
Transition steps	1,200	4,800
Decay rate	0.5	0.5
PPO epochs	–	4
PPO clip ϵ	–	10^{-3}
Advantage normalization	–	Yes
Batch size	150	150
Machine power	2	2
Evaluation samples	4,096	4,096
Evaluation batch size	128	128
Exact diagonalization	Yes	Yes

Table 8. Hyperparameters for the RWKV-7 fine-tuning experiments on the transverse-field Ising model. Both methods use an autoregressive RWKV-7 model with 1.5B parameters, $N = 12$, periodic boundary conditions, $J = 1$, $h = 1$, exact diagonalization for evaluation, and samples drawn exactly from the autoregressive Born distribution. Adam corresponds to the single-epoch first-order baseline, while PWO performs four proximal inner epochs per sampled batch.

E. Additional Figures

E.1. Individual Seed Plots for All Hamiltonias

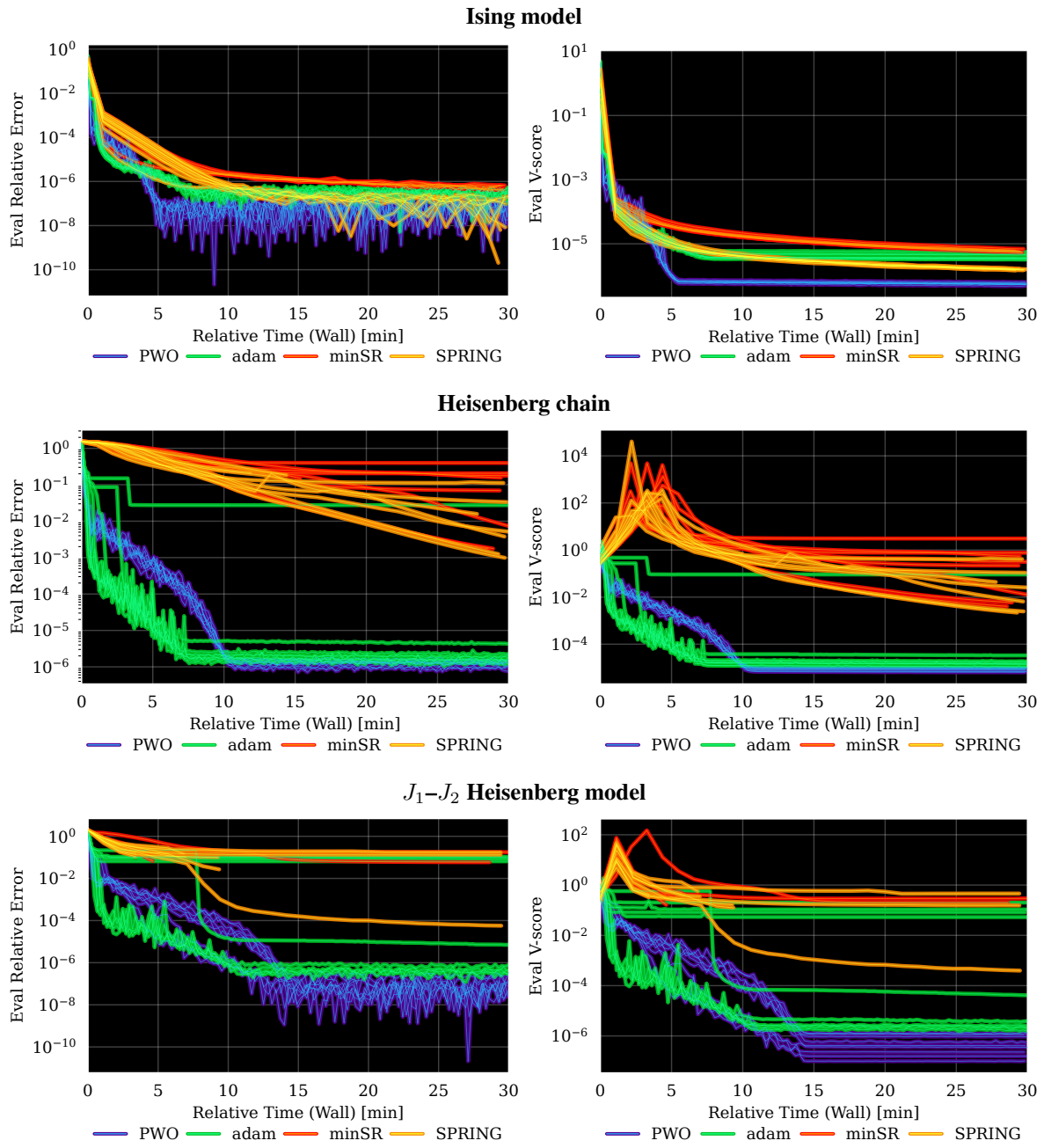


Figure 7. Individual-seed learning curves for all Hamiltonians. Each row corresponds to one Hamiltonian, and columns show relative error and V-score, respectively. Each curve corresponds to one random seed.

E.2. Individual Seed Plots for RWKV7 Fine-tuning

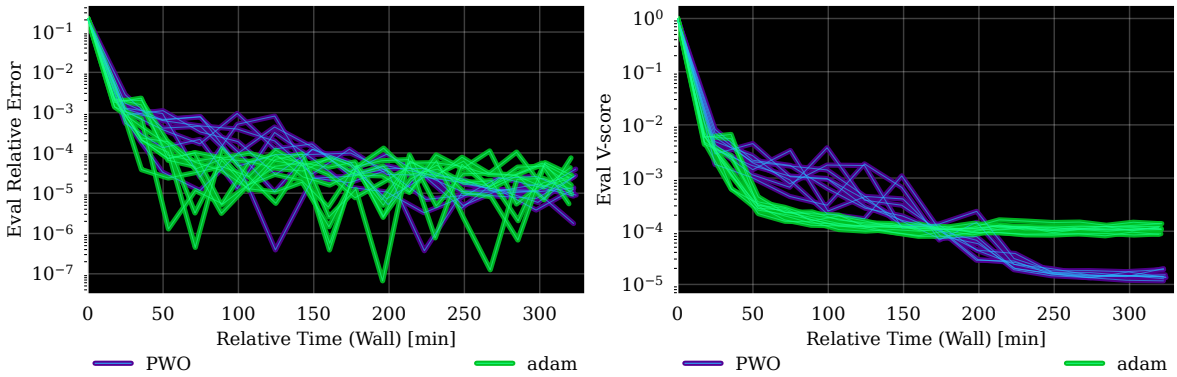


Figure 8. Individual-seed fine-tuning curves for the 1.5B-parameter RWKV-7 neural quantum state on the transverse-field Ising model. Each curve corresponds to one random seed, with relative error shown on the left and V-score on the right. PWO consistently remains stable across seeds and reaches lower final error and variance than the Adam baseline, indicating that the proximal objective improves robustness even in the billion-parameter regime.