Aalto University Department of Applied Physics

PHYS-EV0007 — Machine Learning from and for Quantum Science

Project: Neural quantum states for fermions. Transformer NQS for molecules, and neural-backflow wave functions for the 1D spinless t–V model

Presentation: H. Shang et al., Nat. Commun. **16**, 8464 (2025)
Simulation: NetKet lattice-fermions tutorial

Anouar Moustaj
anouar.moustaj@aalto.fi

Topic nr 2
Project nr 3

Project description

Neural quantum states (NQS) have emerged as a competitive variational framework for interacting fermionic systems, both in *ab initio* quantum chemistry, where molecular Hamiltonians are discretized in a single-particle basis, and in *lattice* settings, where the physics is captured by simple tight-binding models with local interactions. In both cases, the main difficulty is the antisymmetry of the wave function, which must be built into the ansatz itself. The two main routes are (i) *antisymmetric ansätze*, in which a Slater determinant (or a sum of determinants) is augmented by a neural correlator and (ii) *autoregressive neural ansätze*, in which a generative neural network (often a Transformer) parametrizes the conditional amplitudes of the occupation string and enforces particle-number symmetry by construction.

This project is structured along two **independent** axes:

- **Presentation axis (scientific paper).** The student reads and presents the recent paper H. Shang et al., Nat. Commun. **16**, 8464 (2025): *Solving the many-electron Schrödinger equation with a transformer-based framework*, which introduces the QiankunNet architecture, a decoder-only Transformer with autoregressive sampling, and applies it to molecular Hamiltonians, including cases where coupled-cluster methods break down.
- **Simulation axis (hands-on with NetKet).** The student reproduces the NetKet tutorial *Lattice Fermions: from Slater Determinants to Neural Backflow Transformations*, which introduces three variational wave functions for spinless lattice fermions, a Slater determinant, a neural Jastrow × Slater state, and a neural-backflow Slater state, and then *extends* the tutorial by scanning the nearest-neighbour interaction V/t across the full phase diagram of the 1D t–V model, from the non-interacting limit through the Luttinger liquid and across the Berezinskii–Kosterlitz–Thouless (BKT) transition into the charge-density-wave (CDW) phase.

Learning outcomes

- Gain fluency with *fermionic* neural quantum states: why antisymmetry must be built into the ansatz and how the Slater / Jastrow–Slater / neural-backflow hierarchy does so.

- Understand the second-quantized electronic-structure problem and how a Transformer can autoregressively parametrize its wave function with built-in particle-number symmetry.
- Learn how to use NetKet's fermionic primitives (`SpinOrbitalFermions`, the `create` / `destroy` / `number` operators, a number-conserving `MetropolisExchange` sampler, and VMC with stochastic reconfiguration) to solve a lattice fermion model.
- Understand the Jordan–Wigner mapping between the 1D spinless t–V model and the spin-$\frac{1}{2}$ XXZ chain, and use the Bethe-ansatz solution as an exact reference against which to benchmark the NQS data across the Luttinger-liquid to CDW transition.

Presentation

- Explain the formalism used in the paper: the second-quantized electronic-structure problem, the QiankunNet decoder-only Transformer ansatz, and the autoregressive conditional amplitudes $\log \psi(\boldsymbol{n}) = \sum_i \log p(n_i \mid n_{<i}) + i\phi(\boldsymbol{n})$.
- Understand the role of the layer-wise Monte-Carlo-tree- search sampler and of physics-informed (CI-based) initialization in enforcing particle-number symmetry and stabilizing VMC optimization.
- Explain the benchmarks reported in the paper (CCSD, CCSD(T), DMRG, FCI on molecules of increasing difficulty).
- Explain the advantages and limitations of the method (no sample rejection, exact normalization, enforced particle-number symmetry; dependence on the orbital basis, cost of deep Transformers, absence of built-in antisymmetry in the occupation-number representation).

Useful complementary readings include G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial neural networks*, Science **355**, 602 (2017), for background on NQS in general, and K. Choo, A. Mezzacapo, and G. Carleo, *Fermionic neural-network states for ab-initio electronic structure*, Nature Communications **11**, 2368 (2020), for an earlier, non-Transformer fermionic NQS approach to chemistry.

Simulations

The simulation axis is a hands-on exercise whose purpose is to expose the student to fermionic NQS in the simplest possible lattice setting. The 1D spinless t–V model is chosen because it is non-trivial (interacting, with a Luttinger-liquid-to-CDW quantum phase transition) and yet has an *exact* reference solution via a Jordan–Wigner mapping to the XXZ spin chain.

1. Model: spinless t–V fermions on a chain

The Hamiltonian studied in the NetKet *lattice-fermions* tutorial is, in its original 2D version,

$$H = -t \sum_{\langle i,j \rangle} (c_i^\dagger c_j + c_j^\dagger c_i) + V \sum_{\langle i,j \rangle} n_i n_j, \tag{1}$$

with $c_i^\dagger, c_i$ spinless fermionic creation / annihilation operators on site i, $n_i = c_i^\dagger c_i$ the on-site number, $\langle i, j \rangle$ a sum over nearest neighbours on the chosen graph, t the hopping amplitude and V the nearest-neighbour repulsion.

For the *extension* we specialize Eq. (1) to a 1D periodic chain of length L, and work at half filling $N_f = L/2$. In this setting the model admits an exact solution: a Jordan–Wigner transformation

maps Eq. (1) onto the spin-$\frac{1}{2}$ XXZ chain

$$H_{\mathrm{XXZ}} = -\frac{t}{2}\sum_i (\sigma_i^x\sigma_{i+1}^x + \sigma_i^y\sigma_{i+1}^y) + \frac{V}{4}\sum_i \sigma_i^z\sigma_{i+1}^z + \text{const.}, \tag{2}$$

with anisotropy $\Delta = V/(2t)$. The XXZ chain is integrable (Bethe ansatz), and at half filling in the fermion language it exhibits:

- a **gapless Luttinger-liquid** phase for $-1 < \Delta < 1$, i.e. $-2 < V/t < 2$, with continuously varying Luttinger parameter K;
- a **Berezinskii–Kosterlitz–Thouless (BKT)** transition at $\Delta = 1$, i.e. $V/t = 2$, and
- a **charge-density-wave (CDW)** gapped phase for $\Delta > 1$, i.e. $V/t > 2$, characterized by long-range order at wave vector $q = \pi$.

For the repulsive-fermion problem we scan $V/t \in [0, 4]$, so that both the Luttinger-liquid phase and the BKT transition into the CDW are covered.

2. Reproducing the NetKet lattice-fermions tutorial

As a first step, the student reproduces the NetKet tutorial *Lattice Fermions* on its default geometry: a 4×4 square lattice with $N_f = 5$ spinless fermions at $t = 1$, $V = 4$. Then, the student implements all three wave functions discussed there: the bare Slater determinant (mean-field reference), the neural Jastrow×Slater state, and the neural-backflow state, in which the orbital matrix becomes configuration-dependent, $M^{\mathrm{bf}}_{\alpha,i}(\boldsymbol{n}) = M_{\alpha,i} + F_{\alpha,i}(\boldsymbol{n})$. Unlike the Jastrow factor, the backflow modifies the *nodal structure* of the determinant and is therefore strictly more expressive. Each wave function is optimised with VMC and stochastic reconfiguration and benchmarked against the exact ground-state energy ($E_{\mathrm{gs}} \simeq -6.859$) obtained by sparse diagonalisation of the Hamiltonian. The student is expected to report the relative error of each ansatz and reproduce the hierarchy observed in the tutorial: $\sim 25\%$ error for the mean-field Slater determinant at $V/t = 4$, reducing to $\sim 1\%$ for the Jastrow–Slater and backflow states.

3. Extension: phase diagram of the 1D t–V model

Once the tutorial runs correctly, the student *specializes* the workflow to the one-dimensional chain and *extends* it by scanning the interaction strength V/t to reconstruct the well-known phase diagram of the 1D t–V model at half filling.

Scan. Fix $t = 1$ and work on a 1D chain with periodic boundary conditions at half filling, $N_f = L/2$. The recommended chain sizes are $L \in \{12, 16, 20\}$ (reasonable sizes for a regular laptop; you may want to modify `n_discard_per_chain` and `n_samples`). You are also free to explore how large a system your machine can handle. Scan

$$V/t \in \{0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 4.0\}$$

with three additional points at $V/t \in \{1.8, 2.1, 2.2\}$ inside the critical window around the expected BKT transition at $V/t = 2$.

Compute plan. Run the full V/t scan with the *Slater* and *Jastrow–Slater* ansätze (few hundred to two thousand parameters, a few minutes per point on a laptop). Run the *neural-backflow* ansatz only at five representative couplings, $V/t \in \{0.4, 1.2, 2.0, 2.4, 3.2\}$, which is enough to demonstrate the ansatz hierarchy without paying for the full sweep three times.

Observables. At each V/t, measure the following:

- **Ground-state energy per site** $E_0/L = \langle H \rangle / L$, compared to the Bethe-ansatz reference.
- **Energy variance** $\sigma_E^2 = \langle H^2 \rangle - \langle H \rangle^2$, as a quality-control diagnostic (vanishing for the true eigenstate).
- **CDW structure factor**

$$S_{\mathrm{CDW}}(q = \pi) = \frac{1}{L} \sum_{i,j} (-1)^{i-j} \left[\langle n_i n_j \rangle - \langle n_i \rangle \langle n_j \rangle \right].$$

  Since the ordered CDW state doubles the unit cell, $S_{\mathrm{CDW}}(\pi)$ grows like L deep in the CDW phase and remains $O(1)$ in the Luttinger-liquid phase.
- **Single-particle correlator** $G(r) = \left\langle c_{i+r}^\dagger c_i \right\rangle$ on distances $r = 1, \ldots, L/2$. In the Luttinger-liquid phase this decays as a power law $r^{-(K+K^{-1})/2}$ with a coupling-dependent Luttinger parameter K; in the CDW phase the correlator is exponentially suppressed.
- **Density–density correlator** $\langle n_i n_j \rangle - \langle n_i \rangle \langle n_j \rangle$ in real space, and its long-distance envelope.

Plots. The student should produce:

1. E_0/L vs V/t compared to the Bethe-ansatz reference, one curve per ansatz (Slater, Jastrow–Slater, Backflow), clearly showing the hierarchy of accuracies.
2. $S_{\mathrm{CDW}}(\pi)/L$ vs V/t for $L = 12$, 16, 20.
3. $\log G(r)$ vs $\log r$ at three representative values of V/t (weak interaction, $V/t = 2$, strong interaction).
4. σ_E^2 vs V/t, one curve per ansatz, to show that the neural-backflow ansatz retains low variance.

Discuss the following

- The location of the BKT transition in the finite-size NQS data
- The relative performance of the three ansätze across the phase diagram: where does the Slater determinant already suffice? Where is the Jastrow important? Where does the neural backflow become essential?

Deliverables

- A concise report on the article in the form of a presentation.
- A repo with organized, well-documented code, and a notebook with working examples for the simulations.

Working practices and tools

You are strongly encouraged to recycle existing material. You are expected to **read, reuse, and adapt** existing reference implementations pipelines rather than re-derive or re-implement every algorithmic detail from scratch; the pedagogical goal of this project is to *understand and apply* the method, not to reproduce boilerplate. Any code you borrow must be clearly attributed in your repository (e.g. in a `README.md` or as in-source comments) and integrated cleanly with your

own contributions. *Use of LLMs (ChatGPT, Claude, Gemini, ...) is permitted and encouraged for onboarding.* In particular, LLMs are very effective at:

- explaining unfamiliar programming syntax and idioms;
- summarising sections of the paper or cross-referencing related literature;
- debugging installation issues, bookkeeping, and programming recommendations;
- generating boilerplate (plotting scripts, parameter sweeps, unit tests, diagnostic helpers).

You remain, however, fully responsible for the correctness, clarity, and scientific content of what you submit: LLM output must be checked, understood, and, where appropriate, cited. Treat an LLM as a capable but occasionally wrong collaborator, not as an oracle.

Programming language: Python

Packages required: `NetKet` (for NQS and VMC), `JAX` and `Flax` for automatic differentiation and neural-network layers (dependencies of NetKet). `NumPy`, `SciPy`, `matplotlib`

References

- H. Shang et al., Nat. Commun. **16**, 8464 (2025).
- G. Carleo and M. Troyer, Science **355**, 602 (2017).
- K. Choo et al., Nat. Commun. **11**, 2368 (2020).
- D. Luo and B. K. Clark, Phys. Rev. Lett. **122**, 226401 (2019).
- F. Vicentini et al., SciPost Phys. Codebases **7** (2022).
- NetKet tutorial, Lattice Fermions: from Slater Determinants to Neural Backflow Transformations.
- C. N. Yang and C. P. Yang, Phys. Rev. **150**, 321 (1966).
- T. Giamarchi, *Quantum Physics in One Dimension*, Oxford University Press (2004).