PHYS-EV0007 — Machine Learning from and for Quantum Science

Project: Neural quantum states architectures and entanglement.
From the Heisenberg chain to the XXZ phase diagram

Presentation: Y. Levine et al., Phys. Rev. Lett. **122**, 065301 (2019)
Simulation: NetKet Ground-State Heisenberg tutorial

Anouar Moustaj
anouar.moustaj@aalto.fi

Topic nr 2
Project nr 2

Project description

Many different architectures for neural-quantum-state (NQS) have been proposed for various different situations. A central question is *which architectures of neural network can efficiently represent the ground states of physically relevant Hamiltonians?* This is simultaneously a question about expressive power in the machine-learning sense and about entanglement in the many-body-physics sense: a variational ansatz is "expressive enough" for a given target state precisely when its parameters can be tuned to reproduce the entanglement structure of that state without an exponential blow-up in the number of parameters. This project is structured along two **independent** axes:

- **Presentation axis (scientific paper).** The student reads and presents the paper Y. Levine et al., Phys. Rev. Lett. **122**, 065301 (2019): *Quantum Entanglement in Deep Learning Architectures*. The paper constructs tensor-network equivalents of deep convolutional and recurrent networks, identifies *information reuse* as the mechanism that enhances their entanglement capacity, and proves that such architectures can support volume-law entanglement polynomially more efficiently than Restricted Boltzmann machines (RBMs).
- **Simulation axis (hands-on with NetKet).** The student reproduces the NetKet tutorial *Ground-State: Heisenberg model*, which walks through a sequence of NQS architectures of increasing expressive power on the spin-$\frac{1}{2}$ antiferromagnetic Heisenberg chain (a Jastrow ansatz, a plain RBM, a translation-symmetric RBM, and feed-forward neural networks of one and two hidden layers), and then *extends* the tutorial by scanning the z-anisotropy of the XXZ chain to reconstruct its phase diagram: from the ferromagnetic phase, through the gapless XY phase, across the Berezinskii–Kosterlitz–Thouless (BKT) transition at $\Delta = 1$, and into the Ising-like antiferromagnet.

Learning outcomes

- Gain fluency with NQS for spin systems: how architectural expressivity translates into entanglement capacity and why certain phases of matter are harder to represent than others.
- Understand the tensor-network / neural-network dictionary of Levine et al., and the role of *information reuse* (shared convolutional channels, recurrent hidden states) in supporting volume-law entanglement with polynomial resources.

- Learn how to use NetKet's spin primitives to train various NQS architectures.
- Construct the XXZ phase diagram as a function of the anisotropy Δ using NQS.

Presentation

- Explain the expressivity–entanglement dictionary: area law vs. volume law and the MPS bond-dimension bound linking parameter count to representable entanglement.
- Understand the tensor-network equivalents of deep convolutional and recurrent networks and the role of *information reuse* in supporting volume-law entanglement with polynomial resources.
- Explain the main quantitative claim: deep ConvNets / RNNs can represent volume-law states with polynomially many parameters, whereas shallow RBMs cannot.
- Explain the advantages and limitations of the result (rigorous representability guarantee; silence on trainability, sign structure, and VMC optimisation in practice).

Useful complementary readings include earlier work by some of the same authors, *Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design* (arXiv:1704.01552 (2018), which develops the tensor-network / deep-network dictionary in more detail, and 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.

Simulations

The simulation axis is a hands-on exercise on the spin-$\frac{1}{2}$ Heisenberg / XXZ chain, following the NetKet tutorial *Ground-State: Heisenberg model* and extending it to a full phase diagram. The 1D XXZ chain is chosen because it is non-trivial (gapless Luttinger-liquid phase, BKT transition, Ising-like AFM phase) and yet admits an *exact* Bethe-ansatz solution against which every NQS data point can be benchmarked.

1. Model: the XXZ chain

The tutorial targets the isotropic Heisenberg antiferromagnet on a 1D chain with periodic boundary conditions,

$$H_{\text{Heis}} \;=\; \sum_{i=1}^{L} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_{i+1} \;=\; \sum_{i=1}^{L} \big(\sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z\big), \tag{1}$$

with $\sigma_{L+1}^{\alpha} \equiv \sigma_1^{\alpha}$ and chain length $L = 22$ by default.

For the *extension* we generalise Eq. (1) to the anisotropic (XXZ) chain,

$$H_{\text{XXZ}}(\Delta) \;=\; \sum_{i=1}^{L} \Big(\sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \Delta\, \sigma_i^z \sigma_{i+1}^z\Big), \tag{2}$$

with Δ the easy-axis anisotropy. The XXZ chain is Bethe-ansatz integrable (Yang & Yang 1966) and, at total $S^z_{\text{tot}} = 0$, exhibits a **gapless XY (Luttinger-liquid) phase** for $-1 < \Delta \le 1$, with power-law correlations $\langle \sigma_0^z \sigma_r^z \rangle \sim r^{-\eta}$ and a continuously varying Luttinger parameter K; a **Berezinskii–Kosterlitz–Thouless (BKT) transition** at $\Delta = 1$ (exactly the Heisenberg point of the tutorial) and a **gapped Ising-like antiferromagnet** for $\Delta > 1$, with long-range Néel order along the z-axis. We scan the repulsive portion $\Delta \in [0, 3]$, covering the XY phase, the BKT transition, and the Ising-AFM phase.

2. Reproducing the NetKet GS-Heisenberg tutorial

As a first step, the student reproduces the NetKet tutorial *Ground-State: Heisenberg model* on its default geometry: a 1D chain of length $L = 22$ with periodic boundary conditions at the isotropic point $\Delta = 1$. The student then implements the four wave functions discussed there: a Jastrow ansatz, a plain RBM, a translation-symmetric RBM, and a shallow feed-forward neural network (with an optional deeper variant). Each model is optimised with VMC and stochastic reconfiguration and benchmarked against the exact ground-state energy $E_0 \simeq -39.1475$ obtained by Lanczos diagonalisation. The student is expected to report the relative error of each ansatz and reproduce the hierarchy observed in the tutorial: all four ansätze reach the exact energy within a few parts per thousand, with the symmetric RBM getting there at roughly 5% of the plain-RBM parameter count and the deeper FFNN delivering the best energy per unit of compute.

3. Extension: phase diagram of the XXZ chain

Once the tutorial runs correctly at the Heisenberg point ($\Delta = 1$), the student replaces the `Heisenberg` operator by the explicit XXZ Hamiltonian of Eq. (2) (built via Pauli operators) and scans Δ across the phase diagram.

Scan. Work on a 1D chain with periodic boundary conditions at total $S^z_{\text{tot}} = 0$. The recommended chain sizes are $L \in \{16, 22, 28\}$ (reasonable sizes for a regular laptop; you may want to increase `n_samples` and `n_discard_per_chain` as L grows). You are also free to explore how large a system your machine can handle. Scan

$$\Delta \in \{0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.5, 3.0\}$$

with three additional points at $\Delta \in \{0.9, 1.0, 1.1\}$ inside the critical window around the expected BKT transition at $\Delta = 1$.

Compute plan. Run the full Δ scan with the *Jastrow*, *plain RBM* and *Symmetric RBM* ansätze (a few hundred to a thousand parameters, a few minutes per point on a laptop). Run the *deeper FFNN* ansatz only at four representative couplings, $\Delta \in \{0.4, 1.0, 1.6, 2.5\}$.

Observables. At each Δ, measure the following:

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

$$S_{\text{AFM}}(q = \pi) = \frac{1}{L} \sum_{i,j} (-1)^{i-j} [\left\langle \sigma^z_i \sigma^z_j \right\rangle - \langle \sigma^z_i \rangle \left\langle \sigma^z_j \right\rangle].$$

- **Transverse correlator** $C^x(r) = \langle \sigma^x_0 \sigma^x_r \rangle$ on distances $r = 1, \ldots, L/2$.
- **Longitudinal connected correlator** $\left\langle \sigma^z_i \sigma^z_j \right\rangle - \langle \sigma^z_i \rangle \left\langle \sigma^z_j \right\rangle$ in real space, and its long-distance envelope.

Plots. The student should produce:

1. E_0/L vs Δ compared to the Bethe-ansatz reference, one curve per ansatz (Jastrow, RBM, RBMSymm, deeper FFNN where available), clearly showing the hierarchy of accuracies.
2. $S_{\mathrm{AFM}}(\pi)/L$ vs Δ for $L \in \{16, 22, 28\}$.
3. $\log C^x(r)$ vs $\log r$ at three representative anisotropies (deep in the XY phase, at the Heisenberg point, and in the Ising AFM phase).
4. σ_E^2 vs Δ, one curve per ansatz, to show which architecture retains low variance deepest into the gapped phases.

Discuss the following

- The location of the BKT transition in the finite-size NQS data
- The hallmark physical observations in each phase
- The relative performance of the four ansätze across the phase diagram: where does Jastrow already suffice? Where does imposing translation symmetry (`RBMSymm`) give a qualitative gain? Where does the deeper FFNN become essential? Tie these empirical observations back to the expressivity / entanglement hierarchy proposed by Levine et al.

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

- Y. Levine et al., Phys. Rev. Lett. **122**, 065301 (2019).
- Y. Levine et al., arXiv:1704.01552 (2018).
- G. Carleo and M. Troyer, Science **355**, 602 (2017).
- F. Vicentini et al., SciPost Phys. Codebases **7** (2022).
- NetKet tutorial, Ground-State: Heisenberg model.
- C. N. Yang and C. P. Yang, Phys. Rev. **150**, 321 (1966).
- T. Giamarchi, *Quantum Physics in One Dimension*, Oxford University Press (2004).