




mdopt: A code-agnostic tensor-network decoder for quantum error-correcting codes

Aleksandr Berezutskii ¹

¹ Institut Quantique & Département de Physique, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Daniel S. Katz](#) 

Reviewers:

- [@HectorMozo3110](#)
- [@burgholzer](#)
- [@WhiteSymmetry](#)

Submitted: 23 September 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

mdopt is an open-source Python library for code-agnostic decoding of quantum error-correcting codes using tensor networks (in particular, MPS and MPO). Given only the parity-check matrices and a noise model, mdopt builds a tensor-network representation of the decoding problem in the form of MPS-MPO evolution and contracts it to approximate (or, for small instances, exact) maximum-likelihood posteriors over logical operators. Depending on the number of logical qubits, the code either reads out the posterior maximum directly or uses a DMRG-like variational algorithm to find the most likely logical correction. The library targets researchers who wish to benchmark codes and noise models beyond simple settings, while retaining a clear and reproducible workflow in pure Python.

Statement of need

Decoding is central to assessing the performance of quantum codes and hardware. Many widely used decoders are tailored to specific code families, e.g., Minimum Weight Perfect Matching (MWPM) for matching-like detection graphs. While these tools are powerful, researchers frequently need a generic decoder to compare heterogeneous codes and noise models under a common, transparent methodology. Tensor-network methods provide such perspective: decoding can be cast as contracting a TN derived from the code to compute likelihoods of logical events given a syndrome (Ferris & Poulin, 2014). TN decoders have achieved state-of-the-art accuracy for 2D codes (Chubb, 2021) and have recently been extended beyond 2D and to circuit-level noise (Piveteau et al., 2024). However, prior TN decoders primarily target codes with just a few logical qubits; open-source tools for multi-logical quantum LDPC (qLDPC) codes have been lacking. mdopt addresses this gap with a DMRG-like variational sweep that directly reads out the most likely logical configuration across multiple logical qubits, providing a novel, code-agnostic TN decoder for qLDPC codes, while remaining lightweight and reproducible.

mdopt fills the gap by providing an open-source, Python-first implementation of a code-agnostic TN decoder:

- **Inputs:** parity-check matrices, a description of logical observables, and a parametric noise model.
- **Outputs:** logical-error posteriors and sample-level decisions (success/failure), with utilities to aggregate failure-rate curves.
- **Scope:** any CSS/LDPC stabilizer code; independent or biased Pauli noise; exact or approximate contractions.
- **Integration:** minimal API, command-line scripts for large parameter sweeps (e.g., via SLURM), and Jupyter-friendly helpers.

41 This combination complements specialized open-source decoders such as PyMatching (MWPM)
42 (Higgott, 2021), general QEC simulators with decoder plug-ins like qecsim (Tuckett, 2020),
43 and stabilizer-circuit tooling such as Stim (Gidney, 2021).

44 Functionality

45 Given the code parity checks, logical operators and the noise model, mdopt constructs a tensor
46 network whose contraction yields posteriors for logical observables conditioned on a measured
47 syndrome. Users choose a contraction strategy (exact, or approximate via low-rank MPS/MPO
48 truncations) and accuracy controls.

49 Minimal example

```
import numpy as np
import qecstruct as qec
from examples.decoding.decoding import decode_css

# Define a small instance of the surface code
LATTICE_SIZE = 3
surface_code = qec.hypergraph_product(
    qec.repetition_code(LATTICE_SIZE),
    qec.repetition_code(LATTICE_SIZE),
)

# Input an error and choose decoder controls
logicals, success = decode_css(
    code=surface_code,
    error="IIXIIIIIIIII",
    bias_prob=0.1,
    bias_type="Bitflip",
    chi_max=64,
    renormalise=True,
    contraction_strategy="Optimised",
    tolerance=0,
    silent=False,
)
```

50 Acknowledgements

51 This work was supported by the Ministère de l'Économie, de l'Innovation et de l'Énergie du
52 Québec through its Research Chair in Quantum Computing, an NSERC Discovery grant, and
53 the Canada First Research Excellence Fund. This work made use of the compute infrastructure
54 of Calcul Québec and the Digital Research Alliance of Canada.

55 References

- 56 Chubb, C. T. (2021). General tensor network decoding of 2D pauli codes. *arXiv Preprint*
57 *arXiv:2101.04125*. <https://doi.org/10.48550/arXiv.2101.04125>
- 58 Ferris, A. J., & Poulin, D. (2014). Tensor networks and quantum error correction. *Physical*
59 *Review Letters*, 113(3), 030501. <https://doi.org/10.1103/PhysRevLett.113.030501>
- 60 Gidney, C. (2021). Stim: A fast stabilizer circuit simulator. *Quantum*, 5, 497. <https://doi.org/10.22331/q-2021-07-06-497>
- 61

- 62 Higgott, O. (2021). PyMatching: A python package for decoding quantum codes with
63 minimum-weight perfect matching, arXiv e-prints. *arXiv Preprint arXiv:2105.13082*. <https://doi.org/10.48550/arXiv.2105.13082>
64
- 65 Piveteau, C., Chubb, C. T., & Renes, J. M. (2024). Tensor-network decoding beyond 2d. *PRX*
66 *Quantum*, 5(4), 040303. <https://doi.org/10.1103/PRXQuantum.5.040303>
- 67 Tuckett, D. K. (2020). *Tailoring surface codes: Improvements in quantum error correction*
68 *with biased noise* [PhD thesis, University of Sydney]. <https://doi.org/10.25910/x8xw-9077>

DRAFT