Getting Started with BPDecoderPlus

BP+OSD Decoder for Quantum Error Correction

Si-Yuan Chen*, Meng-Kun Liu*, Shen Yang* and Jin-Guo Liu*

2026-01-26

Outline

香港科技大學(廣州) THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Outline

What Are We Decoding?

Quantum error correction protects logical qubits by encoding them in many physical qubits.

Physical errors → Syndrome measurements → **Decoder** → Logical error prediction

The decoder's job: Given syndrome measurements, infer what errors occurred and whether they flipped the logical qubit.

Key Challenge

- Degeneracy: identical syndromes.
- Must predict the **logical effect**, not the exact physical errors

Goal

Minimize the **logical error rate** - the probability of incorrect logical predictions

Circuit-Level vs Code-Capacity Noise

Model	Description	Realism
Code-capacity	Errors only on data qubits, perfect measurements	Simplified
Circuit-level	Errors on all operations, noisy measurements	**Realistic**

BPDecoderPlus uses **circuit-level noise** - the realistic model where measurement operations themselves can fail.

Code-Capacity

- Single round of perfect measurements
- Error model: p on each data qubit
- Useful for theoretical analysis

Circuit-Level

- Multiple rounds of noisy measurements
- Errors on gates, idles, measurements
- Required for real hardware

Detection Events

We don't directly use raw syndrome bits. Instead, we use **detection events**:

$$\text{Detection event} = \text{syndrome}[\text{round } t] \oplus \text{syndrome}[\text{round } t-1]$$

A detection event fires (value = 1) when the syndrome **changes** between rounds.

Why Detection Events?

- Raw syndromes are noisy (measurement errors)
- Detection events **cancel** measurement errors that persist across rounds
- Stim's detector error model (DEM) is defined in terms of detection events

Example Timeline

Round	$t-1$	t	$t+1$
Syndrome s	0	1	1
Detection d	—	1	0

Detection $d_t = s_t \oplus s_{t-1}$

Outline

Pipeline Steps

Step	Input	Output	Purpose
1. Generate Circuit	Parameters (d, r, p)	`.stim` file	Define noisy quantum operations
2. Extract DEM	`.stim` circuit	`.dem` file	Map errors → detections
3. Build H Matrix	`.dem` file	H, priors, obs_flip	Decoder input format
4. Sample Syndromes	`.stim` circuit	`.npz` file	Training/test data
5. Decode	H + syndromes	Predictions	Infer logical errors

The DEM: Detector Error Model

The DEM is the crucial link between physical errors and observable detection events.

```
error(0.01) D0 D5 L0
```

This entry means: *"There's a 1% probability of an error that triggers detectors 0 and 5, and flips the logical observable."*

DEM Specifies

- **What errors** can occur (each line is one error mechanism)
- **Which detectors** fire (D0, D1, etc.)
- **Logical effect**
- **Probability** (the number in parentheses)

Generated By

- Stim's circuit analysis (Gidney, 2021)
- Automatic error propagation
- Decomposition of correlated errors

Outline

Factor Graph from H Matrix

The parity check matrix H defines a **factor graph** (Tanner graph) (Kschischang et al., 2001):

- **Variable nodes** (circles): Error mechanisms (columns of H)
- **Check nodes** (squares): Detectors (rows of H)
- **Edges**: $H[i, j] = 1$ connects detector i to error j

	e_1	e_2	e_3	e_4
D_0	1	1	0	1
D_1	0	1	1	1

D_0 checks: $e_1 \oplus e_2 \oplus e_4$
D_1 checks: $e_2 \oplus e_3 \oplus e_4$

Example: $H = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$

- 4 error variables, 2 detectors
- e_2 and e_4 connected to both checks

Message Passing Intuition

BP iteratively passes "beliefs" between nodes (Pearl, 1988):

$$e \xrightarrow{\mu_{e\to D}} D \qquad D \xrightarrow{\mu_{D\to e}} e$$

Step 1
Variable → Check

"Here's my current probability of being an error"

Step 2
Check → Variable

"Given what others told me, here's what you should be to satisfy the parity"

Step 3
Repeat

Until convergence or max iterations reached

After convergence, each variable has a **marginal probability** of being an error.

The Degeneracy Problem

BP works perfectly on trees, but quantum codes have loops!

On loopy graphs, BP can:

- Fail to converge
- Converge to wrong probabilities
- Output invalid solutions ($H \cdot e \neq$ syndrome)

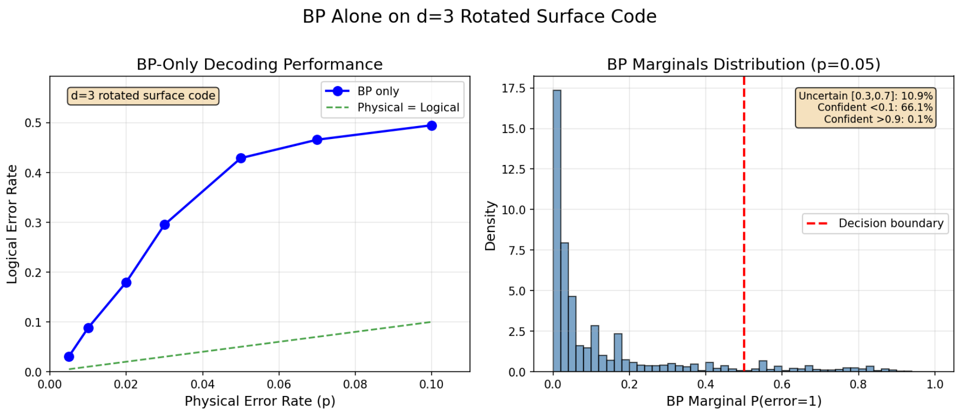
The Degeneracy Problem

Most Critically

BP outputs **probabilities**, but rounding them doesn't guarantee a valid solution.

Why Quantum Codes Are Hard

- Classical LDPC: Each error has unique syndrome signature
- **Quantum surface codes**: Multiple error patterns produce the **same syndrome** (degeneracy)
- BP gets "confused" by equivalent solutions



BP's marginals produce invalid error pattern

Outline

The Key Insight

OSD (Ordered Statistics Decoding) (Fossorier & Lin, 1995) forces a **unique, valid solution** by treating decoding as a system of linear equations:

$$\boxed{H \cdot e = s \pmod 2}$$

Given syndrome s, find error vector e that satisfies this constraint.

BP Provides

- Soft information (probabilities)
- **Unreliable** hard decisions
- No validity guarantee

OSD Guarantees

- **Valid** solutions ($H \cdot e = s$)
- Uses BP's confidence to guide search
- Polynomial time complexity

OSD-0 Algorithm in 3 Steps

Step	**Operation**	**Result**
1. Sort	Order columns by $\lvert \text{LLR} \rvert$ descending	High confidence columns first
2. Row Reduce	Gaussian elimination on H	$[I \mid P]$ form with pivots
3. Solve	Back-substitution with s	Valid error vector e

Result: A valid codeword that respects BP's confident decisions.

OSD-W: Search for Better Solutions

OSD-0 fixes non-pivot bits to BP's decision. **OSD-W** searches over 2^W combinations of the W least confident non-pivot bits.

BP Marginals $\rightarrow$ Sort by Confidence $\rightarrow$ Gaussian Elimination $\rightarrow$ $\boxed{\text{OSD-0 / OSD-W}}$ $\rightarrow$ Best Solution

Method	**Candidates**	**Trade-off**
OSD-0	1	Fast, single solution
OSD-W	2^W	Search W least confident bits
OSD-CS	$1 + k + \binom{W}{2}$	Efficient: weight-0,1,2 patterns

OSD-W: Search for Better Solutions

OSD Order Trade-off

- **OSD-0**: Fast, single solution
- **OSD-10**: $2^{\{10\}}$ candidates, better accuracy
- **OSD-20**: Near-optimal, slower

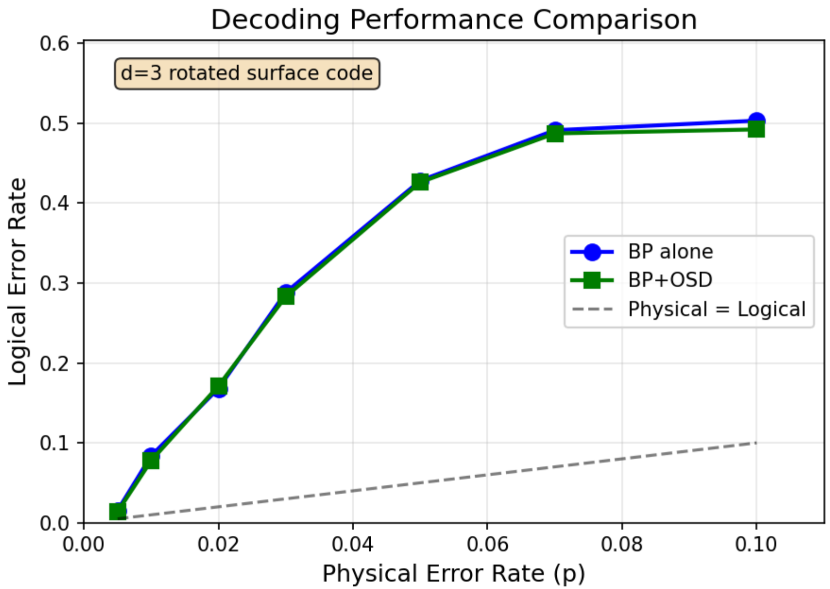
Selection Criterion

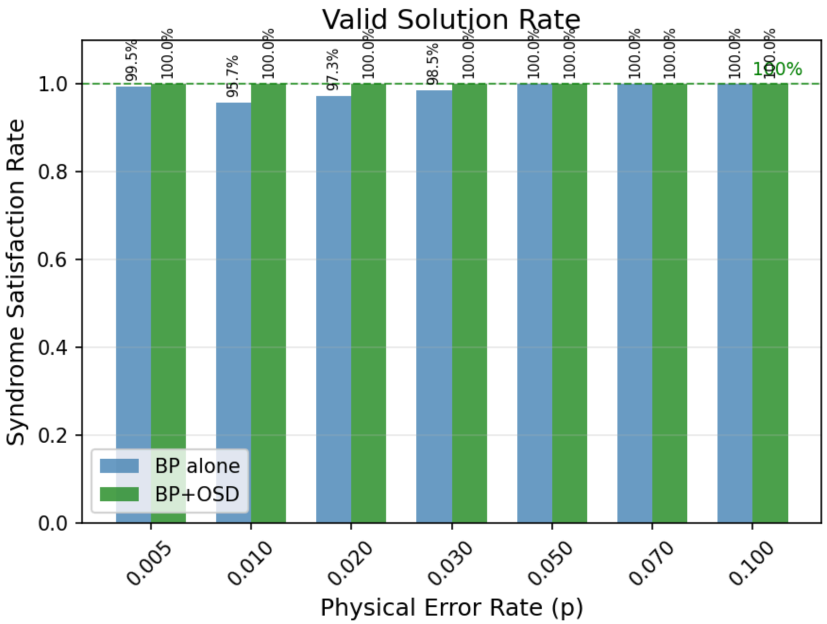
Pick the candidate with lowest **soft-weighted cost**:

$$\mathrm{cost}(e) = \sum_i |\mathrm{LLR}_i| \cdot e_i$$

The Fix in Action

BP+OSD Solves the Hard-Decision Failure





The same syndrome, now decoded correctly with OSD post-processing.

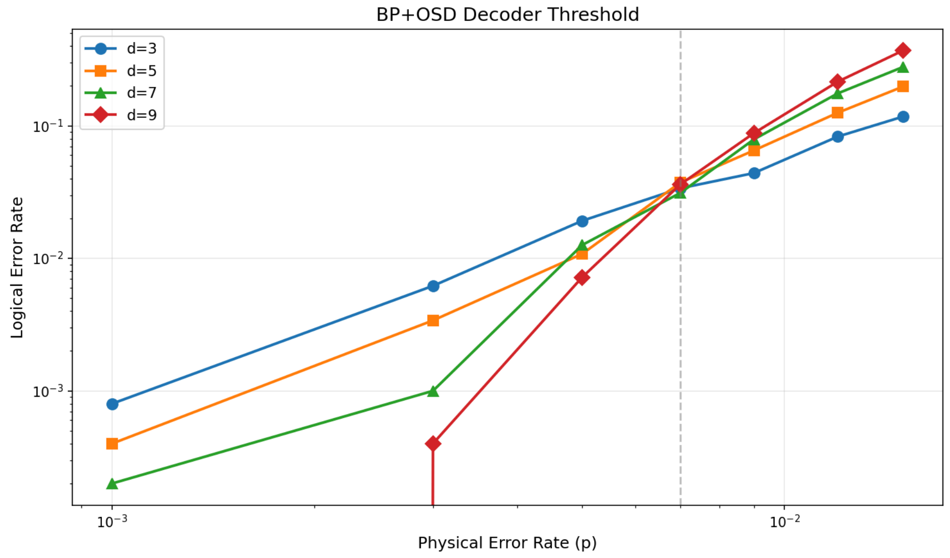
OSD guarantees $H \cdot e = s$

香港科技大學(廣州) THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Outline

Threshold Results

The **threshold** is the physical error rate below which larger codes perform better.

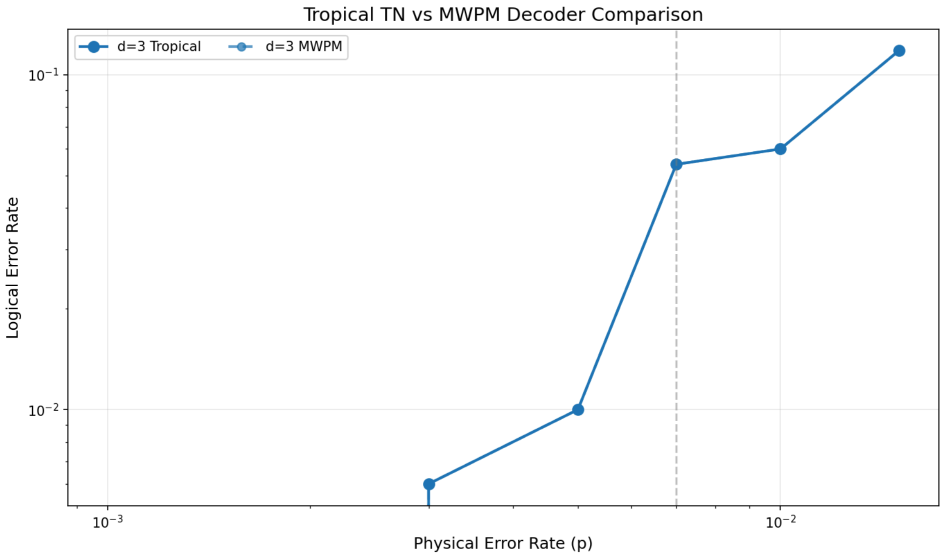


Decoder	Threshold	Notes
BP (damped)	N/A	Fast, limited by loops
BP+OSD	$\sim 0.7\%$	Near-optimal
MWPM	$\sim 0.7\%$	Gold standard (Higgott & Gidney, 2025)

BP+OSD achieves **near-optimal threshold** with good computational efficiency.

Tropical Results

The **threshold** for tropical tensor network decoder require more memory. Further approximate contraction method?



Decoder	Threshold	Notes
BP (damped)	N/A	Fast, limited by loops
BP+OSD	$\sim 0.7\%$	Near-optimal
MWPM	$\sim 0.7\%$	Gold standard (Higgott & Gidney, 2025)

Tropical is too resource consuming!

Summary

BPDecoderPlus: Fast and accurate decoding for quantum error correction

Key Takeaways

1. **Decoding** = syndrome $\longrightarrow$ error prediction
2. **DEM** maps physical errors to detections
3. **BP** provides soft information but may fail
4. **OSD** guarantees valid solutions
5. **BP+OSD** achieves near-optimal threshold

Next Steps

- Try: `uv run python examples/minimal_example.py`
- CLI: `uv run bpdecoder --help`
- Docs: `docs/usage_guide.md`
- Math: `docs/mathematical_description.md`

香港科技大學(廣州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Summary

GitHub: `GiggleLiu/BPDecoderPlus`

Bibliography

Bibliography

[1] C. Gidney, "Stim: a fast stabilizer circuit simulator," *Quantum*, vol. 5, p. 497, 2021.

[2] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[3] J. Pearl, "Probabilistic reasoning in intelligent systems: networks of plausible inference," *Morgan Kaufmann*, 1988.

[4] M. P. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[5] O. Higgott and C. Gidney, "Sparse blossom: correcting a million errors per core second with minimum-weight matching," *Quantum*, vol. 9, p. 1600, 2025.