A partition function framework for estimating logical error curves in stabilizer codes

Leon Wichette[1], Hans Hohenfeld[2], Elie Mounzer[1], and Linnea Grans-Samuelsson[3]

[1]Robotics Innovation Center, German Research Center for Artificial Intelligence, 28359 Bremen, Germany
[2]Robotics Research Group, University of Bremen, 28359 Bremen, Germany
[3]Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Oxford OX1 3PU, United Kingdom

arXiv:2505.15758v3 [quant-ph] 19 Jul 2026

Based on the mapping between stabilizer quantum error correcting codes and disordered statistical mechanics models, we define a ratio of partition functions that measures the success probability for maximum partition function decoding, which at the Nishimori temperature corresponds to maximum likelihood (ML) decoding. We show that this ratio differs from the similarly defined order probability and describe the decoding strategy whose success rate is described by the order probability. We refer to the latter as a probabilistic partition function decoding and show that it is the strategy that at zero temperature corresponds to maximum probability (MP) decoding. Based on the difference between the two decoders, we discuss the possibility of a maximum partition function decodability boundary outside the order-disorder phase boundary. At zero temperature, the difference between the two ratios measures to what degree MP decoding can be improved by accounting for degeneracy among maximum probability errors, through methods such as ensembling. We consider in detail the example of the toric code under bitflip noise, which maps to the Random Bond Ising Model. We demonstrate that estimation of logical performance through decoding probability and order probability is more sample efficient than estimation by counting failures of the corresponding decoders, especially in the regime of low noise. We consider both uniform noise and noise where qubits are given individual error rates. The latter noise model lifts the degeneracy among maximum probability errors, but we show that ensembling remains useful as long as it also samples less probable errors. We also consider, in less detail, the color code under bitflip and depolarizing noise.

1 Introduction

For large scale quantum computing, high-fidelity qubits and operations are needed. On hardware with limited fidelities, this can be achieved through quantum error correction and fault-tolerant logical operations on the encoded logical qubits. The logical performance can be improved by hardware improvements that reduce the physical error rates, or by the usage of a better quantum error correcting code or a more accurate decoder. In the choice of quantum error correcting code, logical performance must be balanced against the spacetime overhead, while the decoder accuracy must be balanced against the decoder runtime.

Just as there is currently a multitude of approaches to hardware design, the same holds for quantum error correcting codes and decoding approaches, with the list likely to keep growing rapidly. Within this abundance of options, estimates of the logical performance under hardware-realistic noise models provide useful guidance. One important metric of code performance, the optimal threshold, has for scalable families of stabilizer and subsystem codes early on been related to order-disorder phase transitions in Random Bond Ising-type statistical mechanics models along

Leon Wichette: leon.wichette@dfki.de
Hans Hohenfeld: hans.hohenfeld@uni-bremen.de
Elie Mounzer: elie.mounzer@dfki.de
Linnea Grans-Samuelsson: linnea.grans-samuelsson@physics.ox.ac.uk

a certain line in the phase diagram: the *Nishimori line* [1]. This statistical mechanics mapping has been extended from independent qubit Pauli noise to more realistic noise models, including circuit noise and coherent noise [2–8], and used to extract finite size corrections to the threshold for surface codes tailored to biased noise [9]. Similarly, the phase transition along the zero temperature line has been shown to yield the threshold under maximum probability (MP) decoding [1, 2]. More specifically, maximum likelihood (ML) decoding relates to the comparison of certain partition functions at the Nishimori temperature (as detailed in Section 2), while maximum probability decoding relates to the comparison of partition functions at zero temperature. In the toric code under independent identically distributed bitflip noise, which maps to the $\pm J$ Random Bond Ising Model (RBIM), the phase boundary has been shown to be reentrant [10], illustrating the non-optimality of MP decoding. More recently, the RBIM phase boundary has been mapped out through numerical estimates of the order probability [11].

While the order probability allows for the determination of the phase boundary, and hence the threshold of an optimal decoder, we show that it does not measure the success rate of the optimal decoder. Instead, it measures the success rate of what we shall refer to as a probabilistic partition function decoder. The full optimal logical error curves are thus inaccessible through this measure. Analogous to order probability, we introduce a ratio that we name the *decoding probability*, which on the Nishimori line measures the optimal logical error curves. We expect that estimating optimal performance by measuring the decoding probability directly generally requires less sampling overhead than estimation by counting the number of failures of the corresponding ML decoder. Previous works concerning performance estimates via the statistical mechanics mapping have mainly focused on establishing the optimal thresholds of error correcting codes under different noise models (e.g., [2, 3, 7, 12–16]). In [17], full optimal error curves were computed for the surface code under independent identically distributed bitflip noise by counting the number of failures of an ML decoder. Tensor network methods have been used to approximate optimal decoding in a controllable way [2, 4, 17]. Most recently, [4] generalized tensor network decoding to the case of 3D codes, such as the 3D unrotated surface code with depolarizing noise, as well as 2D codes with circuit level noise. There has also been previous work on estimating optimal performance in the surface code through the construction of a lookup table, in order to test the accuracy of different suboptimal surface code decoders [18], although the authors estimated that this method could at most be applied to a distance 9 surface code. On the Nishimori line, measurement of optimal performance by decoding probability corresponds to the measurement by syndrome-averaged minimum infidelity used in [6] (see also the related work [5] by the same authors) as well as the success rate shown in [17]. In [11], the order probability in the Random Bond Ising Model was linked to logical performance rates of the toric code, but the distinction between probabilistic partition function decoding and optimal decoding was not made, and the resulting curves do not show the success rate of an optimal decoder. In more recent work, Ref. [19] focuses on the toric code under bitflip noise with full post-selection, which maps to a clean Ising model where analytic expressions can be derived.

Generalizing beyond ML and MP decoders, one can define decoders that return a correction based on either the maximization of the partition function at a given temperature, or that probabilistically return a correction with the probabilities set by the partition functions. We refer to these as maximum partition function decoders and probabilistic partition function decoders, respectively. (For the former, previous works similarly define minimum free energy decoders [2]. However, in the zero temperature limit, we find that the partition function is the better quantity to consider, since degeneracy can play an important role.) We show that the decoding probability measures the success rates of the former, and the order probability measures the success rates of the latter. This gives access to the logical error curves for any member of these two families of decoders. We relate probabilistic partition function decoding at zero temperature to MP decoding, and maximum partition function decoding at zero temperature to "degeneracy enhanced" MP (dMP) decoding – meaning that, whenever there are multiple maximum probability errors, the decoder returns a correction from the equivalence class containing the largest number of such errors. In matching based decoders, degeneracy enhancement can be approximated through the method of *ensembling* described in [20], where the edge weights of the decoding graph are perturbed in order to sample over multiple matchings. Again, we expect that estimating the success rates through order probability and decoding probability requires fewer samples than by counting the number of

failures of the corresponding decoders. Together, order probability and decoding probability provide a flexible framework for estimating the performance under different decoding schemes. Given a hardware realistic noise model and a set of potential codes that could run on the hardware, such an estimation provides a way to filter out codes with low optimal performance before spending resources on developing a fast decoder. In the development of a fast decoder tailored to a specific code, it also provides a measure for how far from optimality the decoder is, and whether or not there is enough room for improvement to motivate further refinements.

Other than being interesting as tools for performance estimation in stabilizer codes, we believe that the distinction between decoding probability and order probability is of interest for further characterizing the phase diagrams of the statistical mechanics models that the codes map to. The shape of the RBIM phase boundary has historically attracted interest of its own, having originally been expected to be vertical [21] but having later been found to be reentrant [10, 11]. The distinction between order probability and decoding probability opens the possibility that there could exist stabilizer codes whose associated statistical mechanics models have maximum partition function decodability boundaries (defined from the thresholds of maximum partition function decoders) that differ from their phase boundaries. We define in Section 3 conditions under which a model would possess a vertical decodability boundary even with a reentrant phase boundary.

After describing the general framework for performance estimation via decoding probability and order probability, we consider in detail the example of the toric code under both uniform (independent identically distributed) and non-uniform bitflip noise, where in the latter noise model we assign different error probabilities to each qubit. While still a simplified "toy model", non-uniform qubit fidelities represent a scenario that better agrees with hardware observations [22] and introduces a degeneracy lifting into the model where the distinction between MP and dMP decoding vanishes, so that the zero temperature locations of the phase boundary and the decodability boundary must agree. Meanwhile, we find that an improvement from ML over MP remains, although it decreases for larger standard deviations. This implies that there is still room for potential gains from decoder improvements through methods such as ensembling, although in the case of ensembling, the perturbations must now be taken large enough to ensure sampling of not only the most probable errors.

The structure of the paper is as follows.

In Section 2 we summarize the relevant background. We describe ML, MP and dMP decoding, and introduce the mapping between stabilizer codes and disordered statistical mechanics models.

In Section 3 we define maximum partition function decoders and probabilistic partition function decoders, and show that their success rates are measured by the decoding probability and the order probability, respectively. We discuss under which conditions a maximum partition function decoder could reach optimality even away from the Nishimori line.

In Section 4 we present numerical results for the toric code under bitflip noise, using the FKT algorithm to compute the relevant partition functions. We demonstrate that the decoding probability and order probability give estimates of the success rates of ML, MP and dMP decoders, and that significantly fewer samples are required to reach a given confidence interval compared to estimation by counting the number of decoder failures. The relative required sample size depends on p and d, and decreases in the low noise regime, making estimation through partition function ratios especially useful where regular sampling struggles. Comparing the dMP and ML thresholds, we see that the maximum partition function decodability boundary is reentrant in the Random Bond Ising Model.

In Section 5 we focus on the zero temperature limit, and compare to matching based decoders. We discuss the role of boundary conditions and the parity of the code distance in determining the effect of degeneracy. We show how these factors influence the amount of improvement observed from ensembling compared to regular MWPM in the toric code, unrotated surface code and rotated surface code, using the `PyMatching` [23] implementation of MWPM. We also discuss bias, both in MWPM and in ensembling. In addition to regular bitflip noise, we briefly discuss the effects of degeneracy and bias for a noise model focusing on hook errors in the "bare ancilla" syndrome extraction circuit, as well as for combined bitflip and readout noise.

In Section 6, we turn to non-uniform bitflip noise in the toric code, drawing qubit error rates from a uniform distribution with varying standard deviations. We estimate MP, dMP and ML

performance from the relevant partition functions, as well as the improvement from ensembling over regular MWPM using `PyMatching`. We see that all decoding strategies perform better under non-uniform noise than under uniform noise with the same mean error rate, and that MP and dMP decoders now perform identically. Additionally, we see that the performance gains from ensembling decrease as the standard deviation is increased.

Finally, in Section 7, we briefly consider the color code under bitflip and depolarizing noise, focusing on the Nishimori line. In this setting, we use tensor network methods to estimate the partition functions. The results again indicate that the efficiency gain from estimating logical error rates via partition function ratios, instead of counting decoder failures, increases in the low noise regime, although limitations in reaching lower p and larger d make the observed gain less dramatic in the depolarizing case.

2 Background

In this section we will present the preliminary material as well as the notation we use. We also give a brief description of the role the statistical mechanics mapping plays in the estimation of optimal logical performance. For a more detailed discussion, we refer the reader to [1].

2.1 Preliminaries

Consider the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ on n qubits and the Pauli group $\mathcal{P}$ with elements $g \in \mathcal{P}$ given by $g = \lambda g_1 \otimes g_2 \otimes \cdots \otimes g_n$ for $\lambda \in \{\pm 1, \pm i\}$ and where $g_i \in \mathcal{P}_i = \{\mathbb{1}, X, Y, Z\}$ are single Pauli operators. A stabilizer code is a quantum error correcting code defined by a stabilizer group $\mathcal{S} \subset \mathcal{P}$, with $-\mathbb{1} \notin \mathcal{S}$, under which the code space $\mathcal{H}_{\vec{0}}$ is invariant, i.e.

$$\mathcal{H}_{\vec{0}} = \{\psi \in \mathcal{H} : S\psi = \psi, \ \forall S \in \mathcal{S}\}. \tag{1}$$

For a stabilizer group of rank r, the code space encodes the quantum information of $n - r$ logical qubits. The quantum information is subject to noise, described by a noise model. We here consider noise models that contain only errors $e \in \mathcal{P}$. Each such error is assigned a probability p_e. Coherent errors can be written as a decomposition in terms of elements in $\{\mathbb{1}, X, Y, Z\}$, and quantum error correction performed through repeated projective Pauli measurements leads to a digitization of errors, so that the ability to correct a finite set of errors is enough to correct any error [24].

We consider a setting where the generators of the stabilizer group are repeatedly measured. If an error is such that the system is in an eigenspace other than the code space after the following round of stabilizer measurements, we refer to it as a *detectable* error. From the stabilizer measurements we obtain a string of ± 1 eigenvalues $(-1)^{s_1}, (-1)^{s_2}, \ldots, (-1)^{s_r}$, where $\vec{s} \in \mathbb{Z}_2^r$ is referred to as the *syndrome*. The syndrome $\vec{s} = \vec{0}$ is the trivial syndrome characterizing the code space.

In terms of the syndromes, the Hilbert space decomposes as:

$$\mathcal{H} = \bigoplus_{\vec{s}} \mathcal{H}_{\vec{s}}. \tag{2}$$

We say that a Pauli operator f has syndrome $\vec{s}$ if and only if $fS_k = (-1)^{s_k} S_k f$ for all stabilizer generators $S_k \in \mathcal{S}$. The previous statement is also equivalent to saying that f has syndrome $\vec{s}$ if and only if $f\mathcal{H}_0 = \mathcal{H}_{\vec{s}}$. Based on this identification, we can speak not only of the syndrome of an eigenspace, but also of the syndrome of an error.

The role of a *decoder* is to, for any detectable error e, return a correction operator f based on its measured syndrome $\vec{s}$, in such a way that fe act trivially on the encoded information. Let $\mathcal{C}(\mathcal{S})$ be the centralizer of the stabilizer group. For a given syndrome $\vec{s}$, the set of Pauli operators with this syndrome is $g(\vec{s})\mathcal{C}(\mathcal{S})$ for some fixed representative $g(\vec{s})$. The centralizer contains $\bar{L} \in \mathcal{C}(\mathcal{S})\backslash\mathcal{S}$, the logical Pauli operators, as well as the stabilizers themselves. Since the stabilizer group is a subgroup of the centralizer, the set of all Pauli operators with syndrome $\vec{s}$ can be partitioned into a disjoint union of equivalence classes under stabilizer multiplication. Relative to a representative

$g(\vec{s})$, we denote these classes as $\mathcal{C}_{\vec{s},\bar{L}} = g(\vec{s})\bar{L}\mathcal{S}$, referring to them as logical equivalence classes. The decomposition is then given by

$$g(\vec{s})\mathcal{C}(\mathcal{S}) = \bigcup_{\bar{L}} \mathcal{C}_{\vec{s},\bar{L}}. \tag{3}$$

Taking the example of a stabilizer code with a single logical qubit, $\bar{L} \in \{\mathbb{1}, \bar{X}, \bar{Y}, \bar{Z}\}$ and the set of Pauli operators with syndrome $\vec{s}$ is partitioned as follows:

$$g(\vec{s})\mathcal{C}(\mathcal{S}) = \mathcal{C}_{\vec{s},\mathbb{1}} \cup \mathcal{C}_{\vec{s},\bar{X}} \cup \mathcal{C}_{\vec{s},\bar{Y}} \cup \mathcal{C}_{\vec{s},\bar{Z}}, \tag{4}$$

with

$$\mathcal{C}_{\vec{s}}^{(\mathbb{1})} = g(\vec{s})\mathcal{S}, \qquad \mathcal{C}_{\vec{s},\bar{Y}} = g(\vec{s})\bar{Y}\mathcal{S}, \tag{5}$$

$$\mathcal{C}_{\vec{s},\bar{X}} = g(\vec{s})\bar{X}\mathcal{S}, \qquad \mathcal{C}_{\vec{s},\bar{Z}} = g(\vec{s})\bar{Z}\mathcal{S}. \tag{6}$$

Based on these equivalence classes two Pauli operators f and g are considered to be equivalent if they belong to the same logical equivalence class and we write $f \sim_{\mathcal{S}} g$. After syndrome extraction, the task of a decoder boils down to finding a Pauli operator $g \sim_{\mathcal{S}} e$ where e is the error that occurred. The decoder succeeds if g and e belong to the same logical equivalence class $\mathcal{C}_{\vec{s},\bar{L}}$, and fails otherwise.

The optimal decoding strategy is for the decoder to always return a recovery operation from the most likely logical error class $\mathcal{C}_{\vec{s}}^{ML}$ for the given syndrome $\vec{s}$. After the action of the noise channel, stabilizer measurement and extraction of syndrome $\vec{s}$, the (unormalized) state can be written as the following linear combination:

$$\rho(\vec{s}) = \sum_{\bar{L}} P(\mathcal{C}_{\vec{s},\bar{L}}|\vec{s})g(\vec{s})\bar{L}\rho\bar{L}g(\vec{s}), \tag{7}$$

for some fixed, hermitian Pauli $g(\vec{s})$, and logical operators $\bar{L}$. More precisely, a *maximum likelihood decoder* returns a Pauli operator $g \in \mathcal{C}_{\vec{s}}^{ML}$ where

$$\mathcal{C}_{\vec{s}}^{ML} = \arg\max_{\mathcal{C}_{\vec{s},\bar{L}}} P(\mathcal{C}_{\vec{s},\bar{L}}|\vec{s}). \tag{8}$$

Maximum likelihood decoding succeeds with probability

$$P_{success}^{ML} = \sum_{\vec{s}} P(\vec{s})P(\mathcal{C}_{\vec{s}}^{ML}|\vec{s}). \tag{9}$$

It is generally hard to compute the probabilities of the error classes. Meanwhile, it is for some codes and noise models easy to find a maximum probability error. This motivates the alternative strategy of *maximum probability decoding*. In this case, given a syndrome $\vec{s}$, the decoder outputs a recovery operation $e(\vec{s})$ such that $P(e(\vec{s})) \geq P(e'(\vec{s})) \quad \forall e'(\vec{s})$.

In this case, the logical class $\mathcal{C}_{\vec{s}}^{MP}$ that the recovery operator belongs to is given by

$$\mathcal{C}_{\vec{s}}^{MP} = \arg\max_{\mathcal{C}_{\vec{s},\bar{L}}} \max_{e \in \mathcal{C}_{\vec{s},\bar{L}}} P(e) \tag{10}$$

(chosen at random among all such classes in the case of degeneracy) and the decoder succeeds with probability

$$P_{success}^{MP} = \sum_{\vec{s}} P(\vec{s})P(\mathcal{C}_{\vec{s}}^{MP}|\vec{s}). \tag{11}$$

$P_{success}^{MP} \leq P_{success}^{ML}$, with strict inequality in general.

A third decoding strategy of interest in what follows is to return a Pauli operator belonging to the class that contains the largest number $n_{\max}$ of maximum probability errors $e_{\max}$ (errors such that $p(e_{\max}) \geq p(f)\ \forall f$). We refer to this as *degeneracy enhanced MP decoding* (dMP), as it also accounts for the degeneracy among maximum probability errors. If all classes containing

such errors have an equal number of maximum probability errors, dMP decoding reduces to MP decoding. For instance, this holds whenever there is a unique maximum probability error.

The logical class $\mathcal{C}_{\vec{s}}^{dMP}$ that the dMP recovery operator belongs to is given by

$$\mathcal{C}_{\vec{s}}^{dMP} = \arg\max_{\mathcal{C}_{\vec{s},\bar{L}}} n_{\max}(\mathcal{C}_{\vec{s},\bar{L}}|\vec{s}) \tag{12}$$

and the decoder succeeds with probability

$$P_{dMP}^{success} = \sum_{\vec{s}} P(\vec{s}) P(\mathcal{C}_{\vec{s}}^{dMP}|\vec{s}). \tag{13}$$

In summary:

Correction returned by ML decoder:	$e(\vec{s}) \in \mathcal{C}_{\vec{s}}$	s.t.	$P(\mathcal{C}_{\vec{s}}) \geq P(\mathcal{C}'_{\vec{s}}) \quad \forall \mathcal{C}'_{\vec{s}}$.
Correction returned by MP decoder:	$e(\vec{s})$	s.t.	$P(e(\vec{s})) \geq P(e'(\vec{s})) \quad \forall e'(\vec{s})$.
Correction returned by dMP decoder:	$e(\vec{s}) \in \mathcal{C}_{\vec{s}}$	s.t.	$n_{\max}(\mathcal{C}_{\vec{s}}) \geq n_{\max}(\mathcal{C}'_{\vec{s}}) \quad \forall \mathcal{C}'_{\vec{s}}$.

2.2 Error Class Probabilities in the Statistical Mechanical Mapping

Maximum likelihood decoding relies on computing error class probabilities. For stabilizer and subsystem codes, this problem can be mapped to computing partition functions in disordered statistical mechanics models[1], after which existing tools for computation or approximation of partitions functions can be used [4, 25–27]. In particular, this mapping relates the optimal threshold of quantum error correcting codes to a critical point in an order-disorder phase transition in the statistical mechanics model. The statistical mechanics mapping was initially done for the toric code under bitflip noise, but has also been extended to other noise models such as depolarizing noise [12], and correlated noise including circuit level noise [2–4]. Additionally, [28] considers non-uniform noise models with long-range spatio-temporal correlations for the repetition code and the toric code. In this section we give an overview of the mapping, focusing on independent qubit noise in stabilizer codes and referring the reader to [2, 29] for details on the treatment of correlated noise and subsystem codes. We present the example of the toric code under bitflip noise in more detail.

To each stabilizer generator S_k we associate a classical spin degree of freedom $\sigma_k \in \{-1, 1\}$. The goal of the statistical mechanics mapping is to provide, for each error e, a Hamiltonian $H_e(\{\sigma_k\})$, such that the Boltzmann weight of the all spin up configuration $\{\Uparrow\}$ gives the error probability, and the quenched disorder partition function $Z(e)$ (with e seen as the disorder realization) gives error class probability:

$$e^{-\beta H_e(\{\Uparrow\})} = P(e) \tag{14}$$

$$Z(e) = \sum_{\{\sigma_k\}} e^{-\beta H_e(\{\sigma_k\})} = P(\mathcal{C}(e)) \tag{15}$$

for a suitable inverse temperature β. In what follows, it is often convenient to write the partition function in terms of the density of states $g(E)$ as $Z(\mathcal{C}) = \sum_E g_{\mathcal{C}}(E) e^{-\beta E}$, with E the energy.

Below, we show the expression for such a Hamiltonian in a general stabilizer code, for a noise model where each qubit i fails independently, and where the qubit errors are single Pauli operators, $g_i \in \mathcal{P}_i = \{\mathbb{1}, X, Y, Z\}$ with probabilities $p_i(\mathbb{1}), p_i(X), p_i(Y), p_i(Z)$.

Following [2] we write the Hamiltonian in terms of the scalar commutator $[\![.,.]\!] : \mathcal{P} \times \mathcal{P} \to \mathbb{C}$, which is defined by the following normalized trace of the group commutator

$$[\![A, B]\!] \coloneqq \frac{1}{2} \operatorname{Tr}[A, B], \tag{16}$$

with the group commutator being given by $[A, B] \coloneqq ABA^{-1}B^{-1}$.

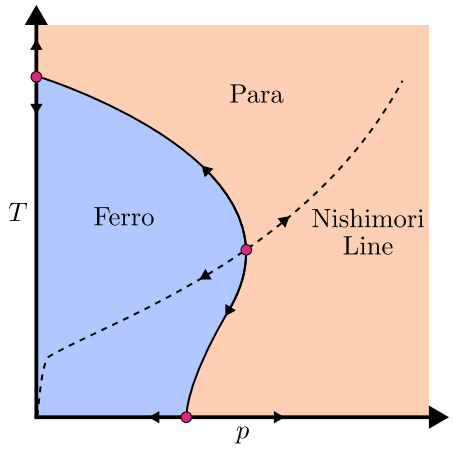


Figure 1: A sketch of the Random Bond Ising Model phase diagram, adapted from [31], with the reentrance of the phase boundary exaggerated for the purpose of illustration. The arrows indicate the renormalization group flow.

The Hamiltonian takes the form

$$H_e(\{\sigma_i\}) = -\sum_i \sum_{g_i \in \mathcal{P}_i} J_i(g_i)\, [\![g_i, e]\!] \prod_{k:[\![g_i,S_k]\!]=-1} \sigma_k \tag{17}$$

with coupling strength $J_i(g_i)$ defined by the Nishimori condition

$$\beta J_i(g_i) = \frac{1}{|\mathcal{P}|} \sum_{f_i \in \mathcal{P}_i} \log p_i(f_i) [\![g_i, f_i^{-1}]\!], \quad \forall i \in \{1, \ldots, n\}, g_i \in \mathcal{P}_i. \tag{18}$$

The Hamiltonian in Eq. (17) is symmetric under stabilizer multiplication of the error configuration as described in [2]. This means that the Hamiltonian of an error e', which differs from another error e by multiplication of a stabilizer generator S_k, gives the same energy for a given spin configuration as the Hamiltonian of the error e gives for a related spin configuration where the corresponding stabilizer spin degrees of freedom is flipped.

For the toric code under bitflip noise, this Hamiltonian reduces to the Hamiltonian of the Random Bond Ising Model (RBIM). Here, the partition function under quenched disorder can be efficiently computed by Pfaffian methods [30]. The toric code and surface code under bitflip noise will be the focus of the numerical examples presented in this paper. We show a sketch of the RBIM phase diagram in Fig. 1.

For the toric code under independent identically distributed bitflip noise, the Hamiltonian of the corresponding statistical mechanics model is defined, for each error configuration e, as

$$H_e = -\sum_{\langle kl \rangle} J_{\langle kl \rangle} \sigma_k \sigma_l - K \tag{19}$$

with the sum taken over all edges $\langle kl \rangle$. The spin degrees of freedom on the vertices $k, l, \ldots$ correspond to the toric code stabilizers of Z-type, and the qubits in the toric code sit on the edges $\langle kl \rangle$.

For an error e the couplings are set to

$$J_{\langle kl \rangle} = \begin{cases} -J, & \text{if } \langle kl \rangle \in e \\ J, & \text{otherwise} \end{cases} \tag{20}$$

with the relation $e^{-2\beta J} = \frac{p}{1-p}$ and $e^{-2\beta K} = p(1-p)$ at the Nishimori line. Here $\langle kl \rangle \in e$ means that the error e has support on the qubit at edge $\langle kl \rangle$.

In what follows, we keep the couplings J fixed and vary the temperature. We denote by T_{Nish} the temperature such that the above relation between $\beta_{\text{Nish}} = 1/T_{\text{Nish}}$ and the couplings J, K is

fulfilled. For identically distributed (uniform) bitflip noise, we fix $J = \pm 1$, $T_{\text{Nish}} = \frac{2}{\ln((1-p)/p)}$. The Hamiltonian in Eq. (19) then fulfills Eqs. (14), so that a decoder that returns a correction based on the largest partition function at $T = T_{\text{Nish}}$ is a maximum likelihood decoder. The interaction strength can not uniformly be normalized in the case of non-uniform bitflip noise with individually sampled qubit error probabilities p_i. Hence, for non-uniform bitflip noise, we set the temperature fixed while varying the couplings: $J_i = \frac{1}{2}\ln\left(\frac{1-p_i}{p_i}\right)$ and $T_{\text{Nish}} = 1$.

3 Success rates for partition function decoders

The goal of this section is to provide a unifying description of the success rates of ML, MP and dMP decoders in terms of ratios of partition functions. Sampling these ratios generally allows for more efficient estimation of the logical performance than directly sampling the number of decoding failures.

3.1 The maximum partition function decoder and the probabilistic partition function decoder

The statistical mechanics mapping relates the probability of an error class $\mathcal{C}_{\vec{s}}$ to the partition function of any error $e \in \mathcal{C}_{\vec{s}}$ evaluated on the Nishimori line. It is also of interest to consider the partition function at other temperatures. In particular, we show how the zero temperature values measure the success rates of MP and dMP decoders. We note that in the zero temperature analysis in [1, 2] the focus has been on energies rather than partition functions, which does not account for the contribution from degeneracy.

In what follows, we denote by $Z^T(\mathcal{C}_{\vec{s}})$ the partition function of any $e \in \mathcal{C}_{\vec{s}}$ evaluated at temperature T. For a given syndrome $\vec{s}$ and temperature T, we denote by $\{\mathcal{C}_{\vec{s}}^{\max}(T)\}$ the set of all classes $\mathcal{C}_{\vec{s}}^{\max}(T)$ such that $Z^T(\mathcal{C}_{\vec{s}}^{\max}(T)) \geq Z^T(\mathcal{C}_{\vec{s}})$ for all $\mathcal{C}_{\vec{s}}$. Below the optimal threshold, in the infinite distance limit and at $T = T_{\text{Nish}}$, this set contains only one element $\mathcal{C}_{\vec{s}}^{\max}(T_{\text{Nish}})$, and its partition function increasingly dominates over the others: $Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}}^{\max}(T_{\text{Nish}})) \to 1$ while $Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}}) \to 0$ for all $\mathcal{C}_{\vec{s}} \neq \mathcal{C}_{\vec{s}}^{\max}(T_{\text{Nish}})$.[1] At temperatures $T \neq T_{\text{Nish}}$, the set $\{\mathcal{C}_{\vec{s}}^{\max}(T)\}$ may contain more than one element, even at noise levels below the optimal threshold. In the infinite temperature limit, it will contain all classes consistent with a given syndrome, while in the zero temperature limit it will contain all classes that contain the largest number $n_{\max}$ of maximum probability errors.

For a given temperature T, we define the following two partition function based decoding strategies:

Definition 1. A *maximum partition function decoder at temperature* T is a decoder that, for each syndrome $\vec{s}$, returns a recovery operation belonging to one of the classes in $\{\mathcal{C}_{\vec{s}}^{\max}(T)\}$, chosen at random each time the decoder is called, with equal probability for each such class.

Definition 2. A *probabilistic partition function decoder at temperature* T is a decoder that, for each syndrome $\vec{s}$, returns a recovery operation belonging to a class $\tilde{\mathcal{C}}_{\vec{s}}$ with probability

$$P_{\text{decoder}}(\tilde{\mathcal{C}}_{\vec{s}}) = \frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})}. \tag{21}$$

In the following proposition, the first statement (which we include for completeness) has been well established, going back to [1], while the distinction between the second and the third statements has to the best of our knowledge not been made.

Proposition 1. The following three statements hold:

1. A maximum partition function decoder at $T = T_{\text{Nish}}$ is an ML decoder.
2. A maximum partition function decoder at $T = 0$ is a dMP decoder.
3. A probabilistic partition function decoder at $T = 0$ is an MP decoder

[1] In the toric code, where the logical operators correspond to nontrivial loops on the torus, an equivalent statement is that the free energy cost of inserting a system-spanning domain wall diverges on the Nishimori line below the optimal threshold.

Proof. We prove each of the three statements in turn.

1. The proof of this statement immediately follows from $P(\mathcal{C}_{\vec{s}}) = Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}})$.

2. Writing the partition function as $Z^T(\mathcal{C}_{\vec{s}}) = \sum_E g_{\mathcal{C}_{\vec{s}}}(E)e^{-E/T}$, with $g_{\mathcal{C}_{\vec{s}}}(E)$ the density of states for the class $\mathcal{C}_{\vec{s}}$, we denote by $E_{\min}(\mathcal{C}_{\vec{s}})$ the lowest energy E such that $g_{\mathcal{C}_{\vec{s}}}(E) \neq 0$, and denote by $E_{\min}(\vec{s}) = \min_{\mathcal{C}_{\vec{s}}} E_{\min}(\mathcal{C}_{\vec{s}})$ the lowest energy among all error classes consistent with the syndrome $\vec{s}$. Normalizing by the lowest energy Boltzmann weight,

$$\lim_{T\to 0} Z^T(\mathcal{C}_{\vec{s}})e^{E_{\min}(\vec{s})/T} = \begin{cases} g(E_{\min}(\mathcal{C}_{\vec{s}})) & \text{if } E_{\min}(\mathcal{C}_{\vec{s}}) = E_{\min}(\vec{s}) \\ 0 & \text{else} \end{cases} \tag{22}$$

with $g(E_{\min}(\mathcal{C}_{\vec{s}}))$ the number of lowest energy disorder realizations present in $\mathcal{C}_{\vec{s}}$. Since the probability of an error is given by its Boltzmann weight at T_{Nish}, and the Boltzmann weight monotonically decreases with E, the lowest energy errors are the most probable and $Z^T(\mathcal{C}_{\vec{s}})e^{E_{\min}(\vec{s})/T} = n_{\max}(\mathcal{C}_{\vec{s}})$, from which the second statement follows.

3. Finally, a probabilistic partition function decoder at zero temperature returns a correction belonging to $\tilde{\mathcal{C}}_{\vec{s}}$ with a probability

$$P_{\text{decoder}}(\tilde{\mathcal{C}}_{\vec{s}}) = \frac{n_{\max}(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} n_{\max}(\mathcal{C}_{\vec{s}})}, \tag{23}$$

which is equivalent to returning a maximum probability error e, with equal probability for each such error. □

3.2 The decoding probability and the order probability

We next introduce decoding probability, alongside the definition of order probability used in [11], and show that the former measures the success rate of maximum partition function decoders, while the latter measures the success rate of probabilistic partition function decoders.

Definition 3. For a temperature T and syndrome $\vec{s}$, let $\mathcal{C}^*_{\vec{s}}(T)$ be an error class chosen at random from $\{\mathcal{C}^{\max}_{\vec{s}}(T)\}$ with uniform probability. The *decoding probability at temperature* T is defined as the following average over disorder realizations e and class choices $\mathcal{C}^*$:

$$P_{\text{d}}(T) = \left\langle \frac{Z^{T_{\text{Nish}}}(\mathcal{C}^*_{\vec{s}}(T))}{\sum_{\mathcal{C}_{\vec{s}}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}})} \right\rangle_{e,\mathcal{C}^*} \equiv \sum_e P(e) \sum_{\mathcal{C}^*_{\vec{s}(e)}(T)} \frac{1}{|\{\mathcal{C}^{\max}_{\vec{s}(e)}(T)\}|} \frac{Z^{T_{\text{Nish}}}(\mathcal{C}^*_{\vec{s}(e)}(T))}{\sum_{\mathcal{C}_{\vec{s}(e)}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}(e)})}. \tag{24}$$

Definition 4. For each disorder realization e, let $\tilde{\mathcal{C}}_{\vec{s}}$ be its error class, $e \in \tilde{\mathcal{C}}_{\vec{s}}$. The *order probability at temperature* T is defined as the following average over disorder realizations e:

$$P_{\text{o}}(T) = \left\langle \frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})} \right\rangle_e \equiv \sum_e P(e) \frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})}. \tag{25}$$

We note that the sum over partition functions in the denominator of Eq. (24) is equal to one, as the partition functions at T_{Nish} are equal to class probabilities. Including the sum explicitly has the advantage of making the expression valid also when the partition functions are computed only up to an overall normalization.

While both the decoding probability and the order probability go to $1/N$ in the limit $T \to \infty$, with N being the number of equivalence classes for each syndrome, their values at finite temperature are generally different. At $T = 0$ their values are equal if, for each syndrome $\vec{s}$, $n_{\max}$ is identical for all classes in $\{\mathcal{C}^{\max}_{\vec{s}}\}$.

Proposition 2. The success rate of a maximum partition function decoder at temperature T is equal to the decoding probability at temperature T.

Proof. We separate the averages over disorder realizations e into averages for each syndrome $\vec{s}$,

$$P_{\text{success}} = \sum_{\vec{s}} P(\vec{s}) P_{\text{success}}(\vec{s}) = \sum_{\vec{s}} P(\vec{s}) \sum_{\mathcal{C}_{\vec{s}}} P(\mathcal{C}_{\vec{s}}|\vec{s}) P_{\text{success}}(\mathcal{C}_{\vec{s}}). \tag{26}$$

For disorder realizations $e \in C_{\vec{s}}$, the decoder succeeds if the recovery operator belongs to $\mathcal{C}_{\vec{s}}$. By definition 1, the recovery operator for a syndrome $\vec{s}$ is chosen at random among $\{\mathcal{C}_{\vec{s}}^{\max}(T)\}$ with probability $P_{\text{decoder}}(\mathcal{C}_{\vec{s}}^{*}(T)) = \frac{1}{|\{\mathcal{C}_{\vec{s}}^{\max}(T)\}|}$. Hence,

$$P_{\text{success}}(\mathcal{C}_{\vec{s}}) = \begin{cases} \frac{1}{|\{\mathcal{C}_{\vec{s}}^{\max}(T)\}|} & \text{if } \mathcal{C}_{\vec{s}} \in \{\mathcal{C}_{\vec{s}}^{\max}(T)\} \\ 0 & \text{else,} \end{cases} \tag{27}$$

so that

$$P_{\text{success}}(\vec{s}) = \sum_{C^* \in \{\mathcal{C}_{\vec{s}}^{\max}(T)\}} P(C^*|\vec{s}) \frac{1}{|\{\mathcal{C}_{\vec{s}}^{\max}(T)\}|}. \tag{28}$$

Finally we substitute

$$P(\mathcal{C}^*|\vec{s}) = Z^{T_{\text{Nish}}}(C^*) = \frac{Z^{T_{\text{Nish}}}(C^*)}{\sum_{C_{\vec{s}}} Z^{T_{\text{Nish}}}(C_{\vec{s}})}, \tag{29}$$

$$P(\vec{s}) = \sum_{\substack{e \\ \vec{s}(e)=\vec{s}}} P(e) \tag{30}$$

to get

$$\begin{aligned} P_{\text{success}} &= \sum_{\vec{s}} P(\vec{s}) \sum_{\mathcal{C}^* \in \{\mathcal{C}_{\vec{s}}^{\max}(T)\}} \frac{1}{|\{\mathcal{C}_{\vec{s}}^{\max}(T)\}|} \frac{Z^{T_{\text{Nish}}}(\mathcal{C}^*)}{\sum_{\mathcal{C}_{\vec{s}}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}})} \\ &= \sum_{e} P(e) \sum_{\mathcal{C}^* \in \{\mathcal{C}_{\vec{s}(e)}^{\max}(T)\}} \frac{1}{|\{\mathcal{C}_{\vec{s}(e)}^{\max}(T)\}|} \frac{Z^{T_{\text{Nish}}}(\mathcal{C}^*)}{\sum_{\mathcal{C}_{\vec{s}(e)}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}(e)})}. \end{aligned} \tag{31}$$

□

We note that at the Nishimori temperature, in a setting with two classes per syndrome, the above per-syndrome measure of logical performance reduces to the following measure given in [6]:

$$P_{\text{failure}} = \sum_{\vec{s}} \min_{q} P_{q,\vec{s}}, \tag{32}$$

where the authors denote by $P_{0,\vec{s}}, P_{1,\vec{s}}$ the class probabilities for the two classes with syndrome $\vec{s}$.

Proposition 3. The success rate of a probabilistic partition function decoder at temperature T is equal to the order probability at temperature T.

Proof. The proof follows the same structure as the proof of Proposition 2. The success rate can again be considered on a per-syndrome basis, with the decoder succeeding whenever the class encountered is the same as the class chosen by the decoder in accordance to Definition 2:

$$\begin{aligned} P_{\text{success}} &= \sum_{\vec{s}} P(\vec{s}) \sum_{\tilde{\mathcal{C}}_{\vec{s}}} P(\tilde{\mathcal{C}}_{\vec{s}}|\vec{s}) P_{\text{decoder}}(\tilde{\mathcal{C}}_{\vec{s}}) = \sum_{\vec{s}} P(\vec{s}) \sum_{\tilde{\mathcal{C}}_{\vec{s}}} P(\tilde{\mathcal{C}}_{\vec{s}}|\vec{s}) \frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})} \\ &= \sum_{e} P(e) \frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}}(e))}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})}. \end{aligned} \tag{33}$$

□

For $T = T_{\text{Nish}}$ the ratio averaged over to obtain the order probability, $\frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})}$, also appears within the expression for coherent information of CSS codes under depolarizing noise found in Ref. [32]. Coherent information provides an alternative method for finding the optimal threshold that can provide accurate estimates at low distances [33], but it generally only provides bounds on the optimal logical success rate itself. An example of a lower bound from coherent information is shown for CSS codes under decoherence in Ref. [34]. At $T = T_{\text{Nish}}$, the probabilistic partition function decoder is also referred to as the *sampling decoder* and, in qubit CSS codes under Pauli noise, is shown in Ref. [35] to correspond to the Petz recovery channel and to share threshold with the optimal decoder.

Not only do the ratios of Definitions 3 and 4 measure the success rates of the decoders defined in Definitions 1 and 2 when averaged over all disorder realizations and class choices, but we expect that for any given finite sample size, an estimate of the success rate computed from these ratios will generally have a narrower confidence interval than an estimate based on counting the number of successes by the corresponding decoder. For each disorder realization, the partition function ratios that are sampled over to estimate decoding probability and order probability,

$$\mathcal{O}_{\text{d}} = \frac{Z^{T_{\text{Nish}}}(\mathcal{C}^*_{\vec{s}}(T))}{\sum_{\mathcal{C}_{\vec{s}}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}})} \tag{34}$$

$$\mathcal{O}_{\text{o}} = \frac{Z^T(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^T(\mathcal{C}_{\vec{s}})}, \tag{35}$$

will yield values in the interval $[0, 1]$, which narrows to $[1/N, 1]$ on the Nishimori line for $\mathcal{O}_{\text{d}}$[2]. Meanwhile, counting the number of successes means sampling over

$$\mathcal{O}' = \begin{cases} 1 & \text{if success} \\ 0 & \text{else} \end{cases} \tag{36}$$

which only takes values in $\{0, 1\}$, leading to a larger sample variance.

3.3 Performance comparisons

There are two natural comparisons for the two types of decoder strategies defined above: performance differences for the same strategy as T is varied, and performance difference between the two strategies for a fixed T.

It has been shown that when T is varied from T_{Nish} to zero in the toric code under independent identically distributed bitflip noise, the phase boundary of the corresponding statistical mechanics model is reentrant. This shows that the probabilistic partition function decoder has a lower threshold at zero temperature than at T_{Nish} in this setting. In models with a reentrant phase boundary, it is interesting to consider whether or not the decodability boundary – defined by the threshold of the maximum partition function decoder – is also reentrant. The distinction between decoding probability and order probability opens the possibility that, as long as the noise rate is below the optimal threshold, maximum partition function decoders can succeed outside the phase boundary. (We expect that a maximum partition function decoder generally has better performance than a probabilistic partition function decoder. At $T = T_{\text{Nish}}$ this is clearly the case: the maximum partition function decoder is optimal, while the probabilistic partition function decoder is suboptimal unless all error classes are equally probable.)

When comparing the present discussion to the phase boundary discussion in [2], it is important to distinguish between a maximum partition function decoder and a minimum free energy decoder. The authors of [2] consider the latter. They note that the negative logarithm defining the free energy F from the partition function,

$$F(T) = -T \ln Z(T), \tag{37}$$

[2]For an error class $\mathcal{C} \in \{\mathcal{C}^{\max}_{\vec{s}}(T_{\text{Nish}})\}$, $Z^{T_{\text{Nish}}}(\mathcal{C}) = P(\mathcal{C}) \geq \frac{1}{N}$.

is monotonically decreasing, and relate the two decoding strategies to each other. However, at zero temperature the free energy decoder will lose the information about degeneracy,

$$\begin{aligned}\lim_{T\to 0} F(T) &= \lim_{T\to 0} -T\ln\left(\sum_E g(E)e^{-E/T}\right)\\ &= \lim_{T\to 0} -T\ln(g(E_{\min})) - T\ln e^{-E_{\min}/T} - T\ln\left(\sum_{E'>E_{\min}} g(E')e^{-E'/T}\right)\\ &= E_{\min},\end{aligned} \tag{38}$$

making it an MP decoder at $T = 0$ even though it is a maximum partition function decoder for all $T > 0$.

Making this distinction, it is clear that while neither the decodability boundary nor phase boundary can extend further to the right than the optimal threshold, lemma 3 of [2] should at $T > 0$ be seen as a statement about the decodability boundary rather than the phase boundary. To phrase the possible distinction between the boundaries differently: even if the free energy difference between error classes does not diverge at $T \neq T_{\text{Nish}}$, the minimum free energy decoder may still succeed at this temperature, provided that the difference remains nonzero and the free energy does diverge at T_{Nish}. Indeed, in such a case a minimum free energy decoder can even retain optimality in spite of a reentrant phase boundary, apart from at $T = 0$. To make a statement that also holds at $T = 0$, we consider a maximum partition function decoder instead: A maximum partition function decoder can retain optimality at $T \neq T_{\text{Nish}}$, as long as at least one of the most likely classes $\mathcal{C}^* \in \{\mathcal{C}_{\vec{s}}^{\max}(T_{\text{Nish}})\}$ has a partition function that remains larger than that of any less likely class.[3] In such cases the decodability boundary is vertical.

Corollary 1. The success rate of a maximum partition function decoder at temperature T is equal to the success rate of an optimal decoder if there exists at least one class $\mathcal{C}_{\vec{s}}^* \in \{\mathcal{C}_{\vec{s}}^{\max}(T_{\text{Nish}})\}$ such that $Z^T(\mathcal{C}_{\vec{s}}^*) > Z^T(\mathcal{C}'_{\vec{s}})$ for all $\mathcal{C}'_{\vec{s}} \notin \{\mathcal{C}_{\vec{s}}^{\max}(T_{\text{Nish}})\}$.

Proof. By Definition 1, if the criterion in Corollary 1 is fulfilled the maximum partition function decoder at temperature T will return a maximally likely error class. By Definition 3 it will then have the same success rate as a maximum partition function decoder at T_{Nish}, which by Proposition 1 is optimal. □

The above shows, in particular, that the thresholds of an MP decoder and a dMP decoder can differ for codes and noise models where the phase boundary is reentrant. This brings us to the second type of comparison: the performance difference between the two decoding strategies at a fixed temperature. In the ordered phase at increasing distance, the two decoding strategies – and hence the two success rates – will increasingly agree, as all partition functions but one go to zero. However, the success rates may differ at finite distance even in the ordered phase, including in cases where their thresholds are the same. In the next section, we study numerically how the distinction between the two decoding strategies plays out in the toric code under bitflip noise, focusing on $T = T_{\text{Nish}}$ and $T = 0$.

We sum up the present section with the three ratios to be sampled over in order to estimate the performance of the three decoders summarized at the end of Section 2.1:

For each disorder realization e, let $\tilde{\mathcal{C}}_{\vec{s}(e)}$ be its error class and $\mathcal{C}_{\vec{s}(e)}^*(T) \in \{\mathcal{C}_{\vec{s}}^{\max}(T)\}$ be chosen at random.

Estimator for the performance of an ML decoder: $\dfrac{Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}}^*(T))}{\sum_{\mathcal{C}_{\vec{s}}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}})}$ at $T = T_{\text{Nish}}$.

Estimator for the performance of an MP decoder: $\dfrac{Z^{T}(\tilde{\mathcal{C}}_{\vec{s}})}{\sum_{\mathcal{C}_{\vec{s}}} Z^{T}(\mathcal{C}_{\vec{s}})}$ at $T = 0$.

Estimator for the performance of a dMP decoder: $\dfrac{Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}}^*(T))}{\sum_{\mathcal{C}_{\vec{s}}} Z^{T_{\text{Nish}}}(\mathcal{C}_{\vec{s}})}$ at $T = 0$.

[3] For noise rates below the optimal threshold there is only one such class at large enough distance, but for finite distances there may be more than one.

4 Partition function based estimates of decoder performance in the toric code under bitflip noise

In this section we focus on the toric code under bitflip noise, mapped to the RBIM as detailed in Section 2.2. We numerically demonstrate Proposition 2 and Proposition 3, and also demonstrate that less samples are needed to reach a certain precision (as measured by the width of the confidence interval) when estimating failure rates using order probability and decoding probability than when counting decoder failures.

We mainly use Pfaffian methods to compute RBIM partition functions, adapting an implementation of the Fisher–Kasteleyn–Temperley (FKT) algorithm by Thomas and Middleton [30] found here. As a complement to this approach, we also adapt a Wang-Landau simulation of partition functions [36]. The Wang-Landau approach allows us to estimate partition values at zero temperature, whereas the FKT algorithm can only approach this limit. Another reason for providing a Wang-Landau implementation is that it can be easily adapted to other codes and noise models, in contrast to the FKT algorithm. However, we expect that tensor network methods are likely to be a more robust choice for settings such as circuit level noise [4]. Further discussion comparing different methods for estimating partition functions can be found in Section 8. We provide a brief description of the Fisher-Kasteleyn-Temperley (FKT) and Wang-Landau (WL) algorithm in Appendix B and C. For an in-depth analysis of these methods, we refer the reader to [25, 26, 30, 36–39]. The source code of our implementation is publicly available at [40–42].

In Fig. 2 we show a comparison between code performance estimates at zero temperature generated by the WL algorithm and performance estimates at $T = 0.1T_{\text{Nish}}$ generated by the FKT algorithm. We see agreement of results within error bars. Further decrease of temperature within the FKT algorithm comes at the cost of significant increase of required bits of precision. A comparison between FKT results at $T = 0.1T_{\text{Nish}}$ and FKT results at $T = 0.01T_{\text{Nish}}$ is shown in Appendix A, Fig. 19, and produces close matching between the logical failure curves. Thus, estimating partition functions with the FKT algorithm at $T = 0.1T_{\text{Nish}}$ indicates to be sufficient to approximate the zero temperature limit with reasonable bits of precision. Additionally, close matching between WL performance estimates at zero temperature and at $T = 0.1T_{\text{Nish}}$, shown in Appendix A, Fig. 20, indicates that $T = 0.1T_{\text{Nish}}$ is a close enough approximation of $T = 0$ for the quantities under consideration. Having established this, all subsequent partition function computations are performed using the FKT algorithm, with $T = 0.1T_{\text{Nish}}$ as a stand-in for zero temperature.

In what follows, we consider 10^4 disorder realizations (samples) for $P \in [0.06, 0.12]$. For each sample, we check whether or not the decoders in Definitions 1 and 2 fail ("estimation by counting"), and compute the ratios in Definitions 3 and 4 ("estimation by ratio"). We compute error bars using bootstrapping [43], in order to have a method that works for both estimation methods. We are using 1000 resamples and 95% confidence level for error estimates by bootstrapping. For the estimation by counting we also compute error bars using a posterior beta distribution with the Jeffreys prior at 95% confidence level, and find that these agree with the error bars computed via bootstrapping.

In Fig. 3 we demonstrate Propositions 2 and 3 at $T = T_{\text{Nish}}$, and in Fig. 4 at $T = 0.1T_{\text{Nish}}$. In both cases we see agreement within error bars between estimation by counting and by ratio.

Comparing the crossing points seen at $T = T_{\text{Nish}}$ to those seen at $T = 0.1T_{\text{Nish}}$, which in the large distance limit measure the thresholds of the different decoders, indicates that not only the phase boundary but also the decodability boundary is reentrant. This shows that in the toric code under bitflip noise, the criterion for Corollary 1 is not fulfilled.

Comparing the two decoding strategies at $T = 0.1T_{\text{Nish}}$ shows the performance difference between MP and dMP decoding. We find that the logical failure rates are almost identical at odd distances, while dMP decoding performs better than MP decoding at small, even distances. The dependence on parity at zero temperature will be discussed in more detail in the next section. At $T = T_{\text{Nish}}$, we find that maximum partition function decoding generally outperforms probabilistic partition function decoding for both even and odd distance, although the threshold is not visibly affected.

Moreover, by comparing decoding probability estimates at $T = T_{\text{Nish}}$ to those at $T = 0.1T_{\text{Nish}}$, which is shown in Fig. 5, we find that dMP decoding closely matches the ML decoding performance

for low code distances or low error rates. ML decoding increasingly outperforms dMP decoding with increasing code distances and error rates.

The size of the error bars are not visible in Fig. 3 and Fig. 4, but are as expected smaller for the estimation by ratio. To illustrate the improved sample efficiency in estimation by ratio compared to estimation by counting, we estimate the fraction of the sample size that is sufficient to obtain the same confidence interval. An example of the confidence interval sizes as a function of sample fraction is shown in Fig. 6 (left) for $p = 0.01, d = 5$. In this example, less than 3% of the sample size is required to reach as precise of an estimate with the ratio method as when using the full sample size and the counting method. Here, the full sample size is just below two million.[4] More generally, the reduction in number of shots needed depends on the desired confidence interval width, as well as on p and d. Assuming a fixed target confidence interval width, for instance determined by requiring that the logical error curves of two different decoders can be distinguished unless they agree up to a certain precision, we compare the required number of shots when estimating the logical error rates by counting and by ratio, respectively. In the counting method, the width of the confidence interval as a function of the number of shots can be found by fitting an ansatz based on a posterior beta distribution with the Jeffreys prior, taking the failure fraction at a given p, d as the fitting parameter. Meanwhile, for the ratio method, we find that the confidence interval width differs from that of the counting method by a scale factor $1/r(p, d)$, independent of the number of samples. An example is shown in Appendix D. The relative required sample size R, which we define through the comparison of shots N needed to reach a target confidence interval width Δ,

$$R = \frac{N_{\text{ratio}}(\Delta)}{N_{\text{counting}}(\Delta)}, \tag{39}$$

saturates as $N \to \infty, \Delta \to 0$, with the saturation value depending on p and d. The saturation is shown for the example of $p = 0.01, d = 5$ in Fig. 6 (right). The saturation value can be straightforwardly determined from $r(p, d)$ by observing that in the limit $N \to \infty$, the width of the confidence interval scales as $1/\sqrt{N}$. The saturation value of the relative required sample size is therefore given by

$$R \xrightarrow[N\to\infty]{} 1/r(p, d)^2. \tag{40}$$

Interestingly, we observe that in the toric code under bitflip noise, the value of $r(p, d)$ increases in the regimes of both large p and low p, as compared to values closer to the threshold. This is shown in Fig. 7. As a consequence, the ratio method becomes increasingly favorable in the low noise regime, where regular sampling struggles.

5 Bias, degeneracy and ensembling in matching based decoding

In this section, we consider how MP and dMP decoders relate to decoding strategies based on minimum weight perfect matching (MWPM). While a matching based decoder will return *some* maximum probability error, it does not necessarily return such errors in an unbiased fashion [44]. Comparing MP to MWPM shows how bias in a given matching algorithm affects the performance.

To approximate dMP decoding, a matching based decoder moreover needs to return not only a single matching, but an unbiased sampling of several such matchings that can be used to estimate the relative fraction of them belonging to each class. Methods such as *ensembling* [20] can return several matchings by calling the matching algorithm multiple times, each time with a random perturbation of the edge weights. We distinguish between weak ensembling, where the perturbation is small enough that we only sample among minimum weight matchings, and strong ensembling, where the perturbation is strong enough that we also sample among higher weight matchings. To approximate dMP decoding, weak ensembling would be used. However, bias may again affect performance, as ensembling does not return matchings in an unbiased fashion. An intuitive example for why this is the case is shown in Fig. 8. The comparison of dMP decoding to weakly ensembled MWPM shows how this bias affects the performance. We also discuss under what conditions weak

[4] A few jobs were canceled due to competing jobs from other users, giving a total sample size of 1999131. Overall, for the confidence interval estimates within this section, the sample sizes vary for different p and d. In all cases, saturation of the relative required sample size was observed, providing a sample size independent measure.

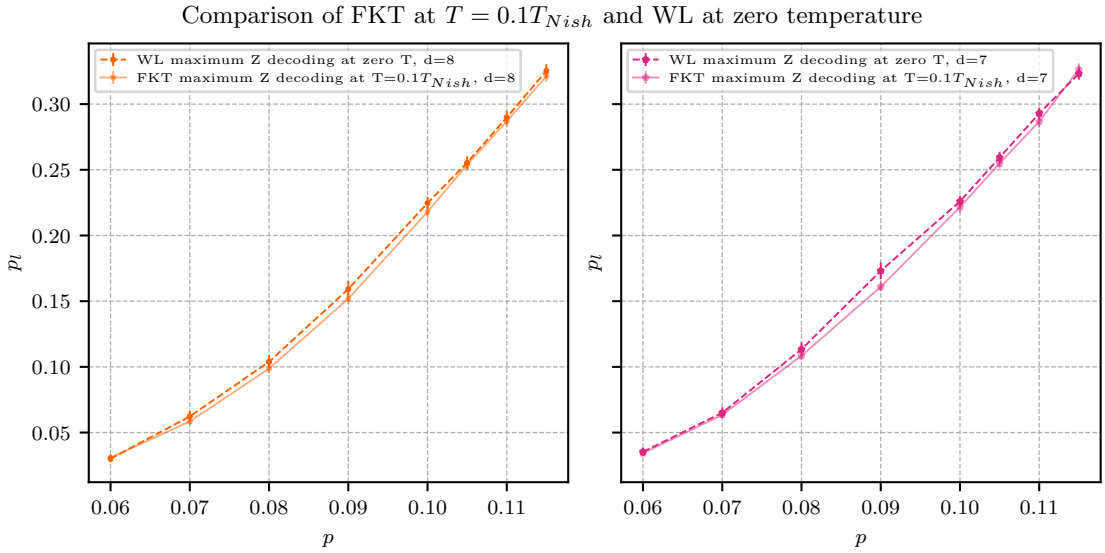


Figure 2: Matching within error bars of maximum partition function decoder performance estimates generated by WL at zero temperature and FKT at $T = 0.1T_{\text{Nish}}$.

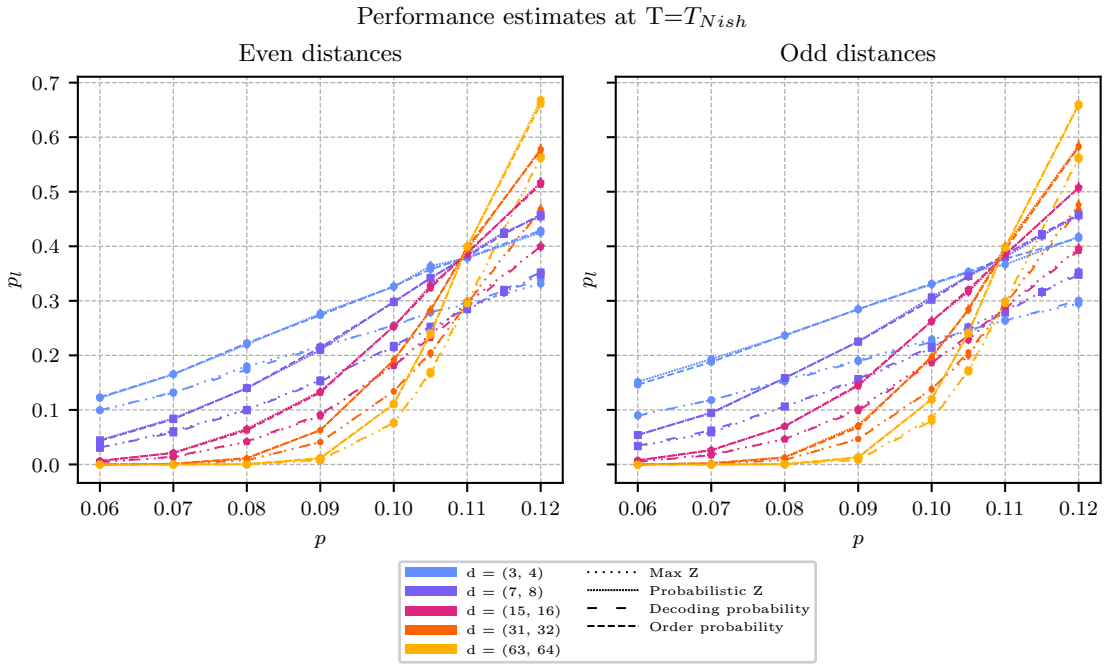


Figure 3: Performance estimates for both counting and ratio methods for probabilistic and maximum partition function decoding at $T = T_{\text{Nish}}$. Here, the performance estimates by counting are labeled "Max Z" and "Probabilistic Z". The success rate of ML decoding is measured by "Max Z" and the "Decoding probability" at $T = T_{\text{Nish}}$. We find matching within error bars between counting and ratio methods. Furthermore, we find improved performance of maximum partition function decoding over probabilistic partition function decoding.

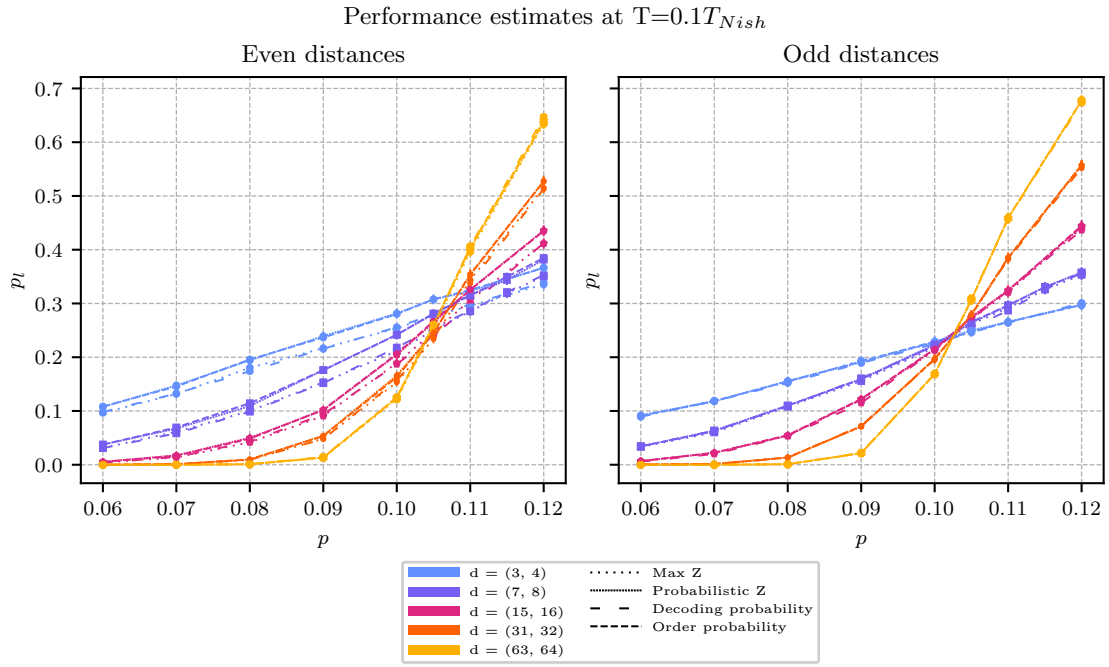


Figure 4: Performance estimates for both counting and ratio methods for probabilistic and maximum partition function decoding at $T = 0.1T_{\text{Nish}}$. Here, the performance estimates by counting are labeled "Max Z" and "Probabilistic Z". The success rate of MP decoding is estimated by "Probabilistic Z" decoding and the "Order probability" while the success rate of dMP decoding is measured by "Max Z" and the "Decoding probability" in the low temperature limit. We find matching within error bars between counting and ratio methods. Furthermore, we find improved performance of maximum partition function decoding over probabilistic partition function decoding for low, even distances, while the performance results match closely for odd distances.

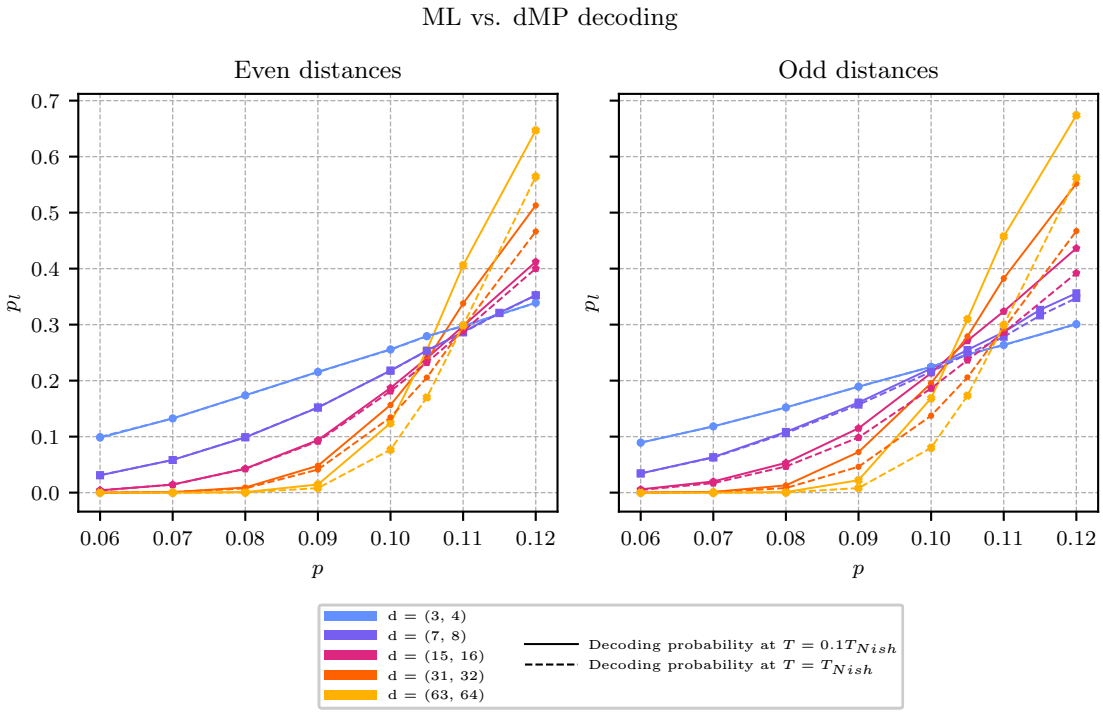


Figure 5: Comparison of dMP (estimated by decoding probability at $T = 0.1T_{\text{Nish}}$) and ML decoding (estimated by decoding probability at $T = T_{\text{Nish}}$). We see an increase of performance advantage from ML over dMP decoding with an increase of code distance.

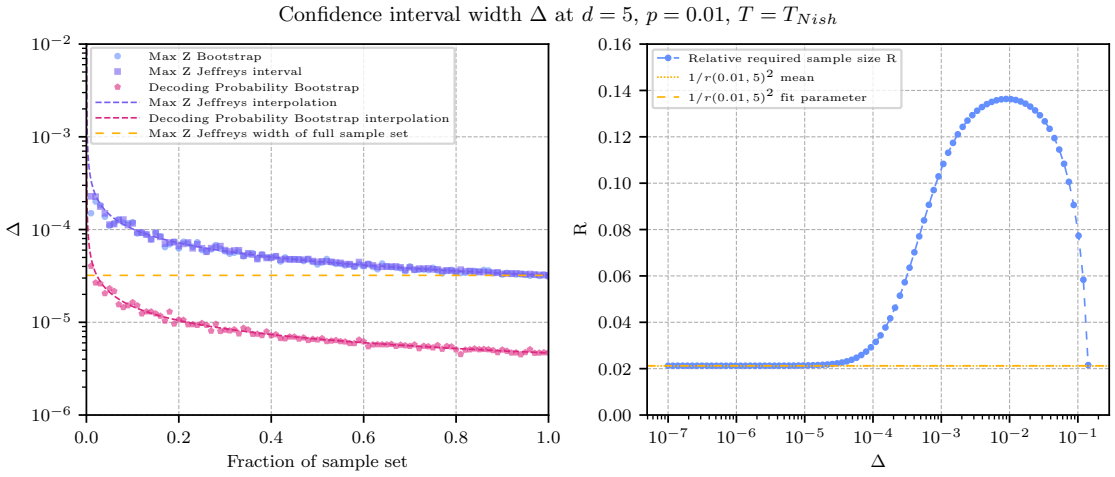


Figure 6: The left plot shows the confidence interval widths for estimates of the optimal logical error rates of the toric code under bitflip noise at $d = 5, p = 0.01$, for varying fractions of the sample set. The confidence interval widths are displayed for both the decoding probability ("ratio method") and the maximum partition function counting method. The counting method confidence interval width for the full sample set, generated by the Jeffreys interval, is shown as a horizontal line. Its intersection point with the curve corresponding to the ratio method indicates that 2.23% of the sample set is sufficient for estimating the optimal code performance to the same accuracy with the ratio method. The right plot shows the dependency of the relative required sample size R on the target confidence interval width Δ. We find that $R \to 1/r(0.01,5)^2$ as $\Delta \to 0$. The value of $r(0.01,5)$ is independent of sample size, as shown in Fig. 23, and is estimated from Fig. 23 to be $r(0.01,5) \sim 6.87$. This value is compatible with the value of $r(0.01,5) \sim 6.86$ that is obtained by fitting the ratio method data points in the left plot.

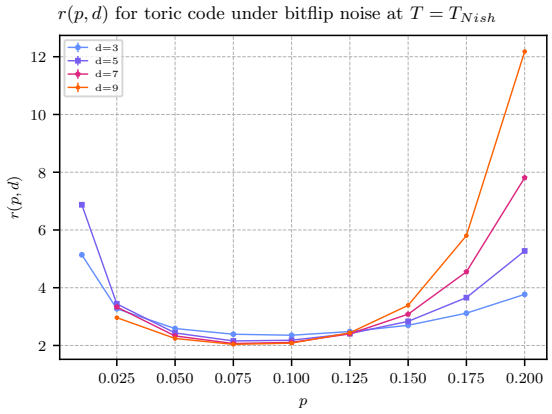


Figure 7: Scale factor $r(p, d)$ between the confidence interval widths of the counting and ratio methods for estimating optimal logical error rates, for different p, d. The scale factor is estimated as in Fig. 23. In the example of $d = 5, p = 0.01$, $r(0.01, 5) \sim 6.87$ is compatible with the observed relative required sample size of $R = 0.0223$ in Fig. 6. The plot only includes data points for p, d where at least 10^2 logical failures were observed with the counting method, excluding distances $d > 5$ at the leftmost point $p = 0.01$.

ensembling would either not be expected to lead to performance gains, or only lead to performance gains at small distances.

5.1 Matching based decoding and ensembling

Minimum-weight perfect matching (MWPM) decoding is a maximum probability decoding technique that finds a most probable correction operator consistent with stabilizer measurement outcomes. We provide a brief overview of the method in Appendix E. Implementations of MWPM decoding, such as sparse blossom [23] or fusion blossom [45], operate deterministically. That is, given a matching graph and syndrome, they will always find the exact same correction operator. Under a uniform bitflip noise model with error probability p, we employ an ensembling [20] strategy by sampling 50 matching graphs with their edge weights perturbed to

$$w_i = \log \frac{1 - (p + \xi_i)}{p + \xi_i}, \tag{41}$$

where $\xi_i \sim \mathcal{N}(0, \sigma_\xi^2)$. If a syndrome $\vec{s}$ is consistent with multiple error configurations, this small, random modification of the edge weights allows the decoder to find different correction operators for the different perturbed graphs. We group the correction operators into equivalence classes and select a representative correction operator for the equivalence class found most often as the decoding result. The expectation is that this method on average will decode to error classes with higher degeneracy, thus approximating dMP decoding, provided σ_ξ is chosen small enough to not introduce new shortest paths into the matching graph. As an alternative ensembling strategy, we also tried sampling isomorphic permutations of the matching graph $\mathcal{G}_M$, where each permutation changes the order of nodes and edges. With a deterministic decoder implementation, different permutations may result in different equivalent matchings to be found. While this method produced slight improvements on the observed logical error rate over MWPM decoding, it generally performed worse than the ensembling based on edge weight perturbations discussed here.

In the following subsections we present and discuss the results of simulations with a range of physical error rates for the toric code, as well as the unrotated and rotated planar surface code. We set $\sigma_\xi = 10^{-6}$, after optimizing this parameter as discussed in Appendix G. Our implementation of ensembling, based on `PyMatching`, is available at [46].

5.2 Comparison of MP and dMP to matching based decoding in the toric code

We have established in Propositions 1, 2 and 3 that the order probability at $T = 0$ estimates the success rate of a MP decoder while the decoding probability at $T = 0$ estimates the success rate

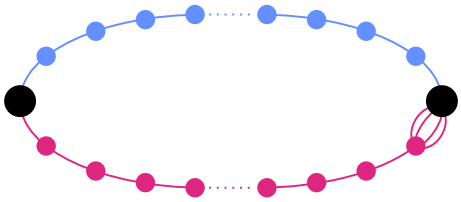


Figure 8: Depicted are five shortest paths between the black vertices, corresponding to minimum weight perfect matchings. The equivalence classes are indicated by color, with the upper (blue) path belonging to one class and the four lower (red) paths belonging to another. The red paths differ only by the last segment. A fair sampling of minimum weight perfect matchings would return the blue path 20% of the time. In weak ensembling, the weight of each segment of the above paths is perturbed by a small amount. If the perturbations add up in such a way that the blue path is the shortest, this is the matching returned. However, given a large number N_S of segments, the length difference between the blue path and any of the red paths is almost always determined by the first $N_S - 1$ segments, so that as $N_S \to \infty$ the blue path is returned 50% of the time. This demonstrates that weak ensembling does not sample fairly among matchings.

of a dMP decoder. (In practice, we use $T = 0.1T_{\text{Nish}}$ as a stand-in for $T = 0$, as discussed in Section 4). By comparison of logical failure rates from MWPM and weakly ensembled MWPM decoding to the order probability at $T = 0$ and the decoding probability at $T = 0$ respectively, one can compare potentially biased matching decoders to their unbiased counterparts. We note that the choice of normally distributed perturbations ξ_i will on rare occasions lead to sampling of higher weight matchings, but that such events are rare enough to not affect the results significantly.

The logical performance estimates under MWPM, weakly ensembled MWPM, MP and dMP decoding are shown for even and odd code distances in Fig. 9. We observe that MWPM decoding suffers from bias, which primarily affects the performance at even code distances negatively. In particular, the logical failure rates of the MWPM decoder are noticeably higher than the failure rates of the MP decoder at even distances beyond $d = 4$. In contrast, for odd distances, the logical failure rates of MWPM and MP decoding differ only slightly. This parity dependent effect of bias on MWPM decoding is closely related to the appearance of ground state degeneracies, which are more likely for even distance toric codes as discussed in Section 5.3. Meanwhile, we find that the code performance of the dMP decoder aligns closely with the performance estimates of weakly ensembled MWPM decoding. Thus, the bias of weakly ensembled MWPM decoding does not visibly affect its performance. This may be explained by the ensembling having a certain degree of robustness to bias: it requires solely the estimation of which error classes have the highest degeneracy, rather than an estimation of their exact amounts of degeneracy.

5.3 Boundary and parity dependence of performance effects from bias and weak ensembling

We have seen in Section 4 that the difference between MP and dMP depends on the parity of the distance in the toric code. For degeneracy to matter at zero temperature, there must be syndromes $\vec{s}$ such that there are at least two different error classes $\mathcal{C}_{\vec{s}}^{(1)}, \mathcal{C}_{\vec{s}}^{(2)}$ fulfilling the criteria $n_{\max}(\mathcal{C}_{\vec{s}}^{(1)}) > 0, n_{\max}(\mathcal{C}_{\vec{s}}^{(2)}) > 0$ and $n_{\max}(\mathcal{C}_{\vec{s}}^{(1)}) \neq n_{\max}(\mathcal{C}_{\vec{s}}^{(2)})$. In terms of matchings, the condition is that there are syndromes consistent with at least two classes of minimum weight perfect matchings, with one class containing more such matchings than the other. We illustrate such a scenario for the $d = 6$ toric code in Fig. 10. In addition to governing the difference between MP and dMP, effects from bias in matching based decoders can only occur when the first of these criteria is fulfilled.

Under the assumption of uniform bitflip noise, the existence of two classes such that $n_{\max}(\mathcal{C}_{\vec{s}}^{(1)}) > 0, n_{\max}(\mathcal{C}_{\vec{s}}^{(2)}) > 0$ can be related to the existence of even (Hamming) weight representatives of logical operators. Taking such a representative of weight $2w$, we may write it as the disjoint union of two errors $e^{(1)}, e^{(2)}$ with the same syndrome, with $w(e^{(1)}) = w(e^{(2)}) = w$. Conversely, taking $e^{(1)} \in \mathcal{C}_{\vec{s}}^{(1)}$ and $e^{(2)} \in \mathcal{C}_{\vec{s}}^{(2)}$ to be inequivalent maximum probability errors of weights $w(e^{(1)}) = w(e^{(2)})$, there exists a logical operator $e^{(1)}e^{(2)}$ with weight $2w(e^{(1)}) - 2w(e^{(1)} \cap e^{(2)})$, with $e^{(1)} \cap e^{(2)}$ being the overlap of the two errors. The lowest-weight even representative of a logical operator thus provides

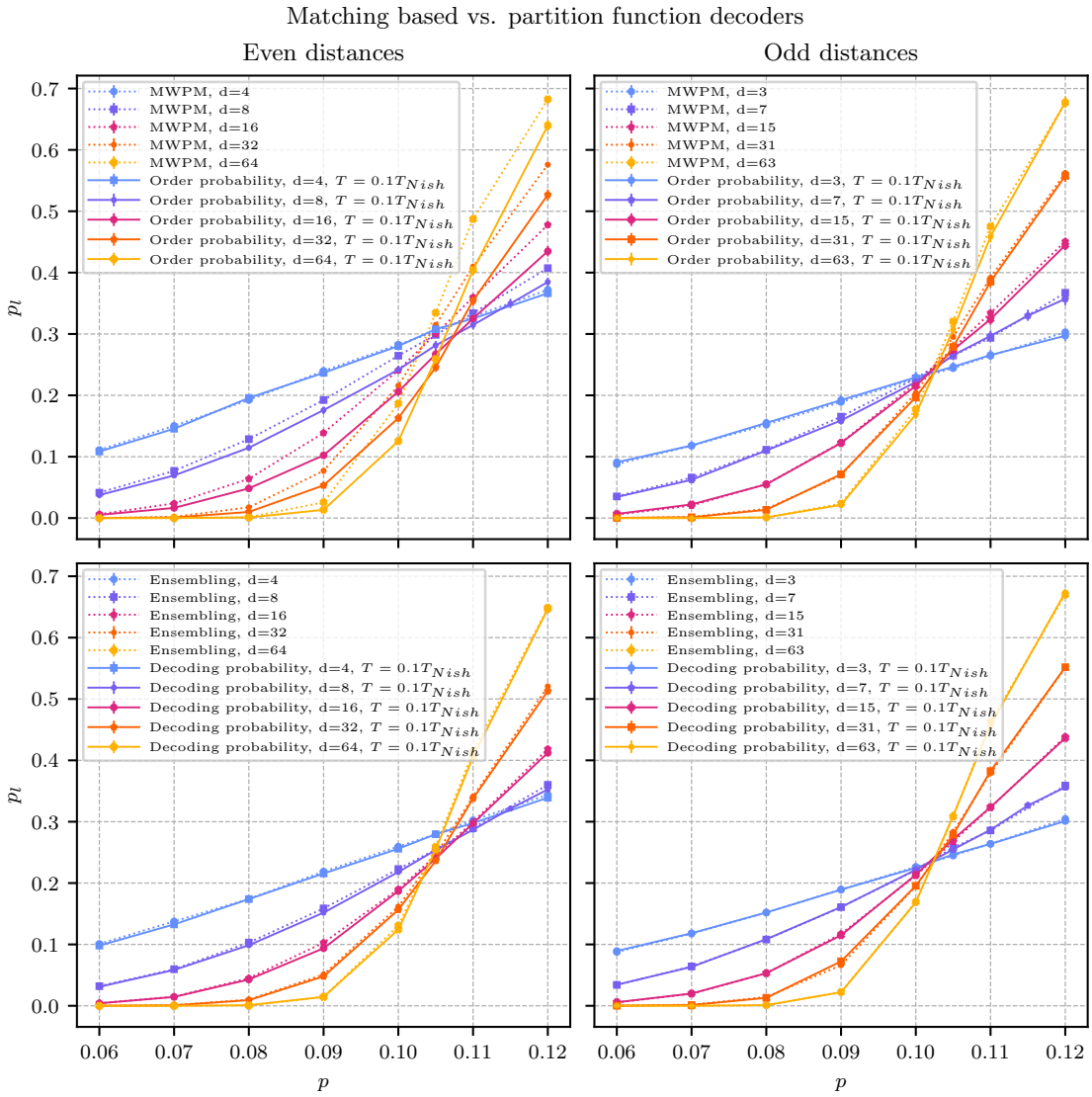
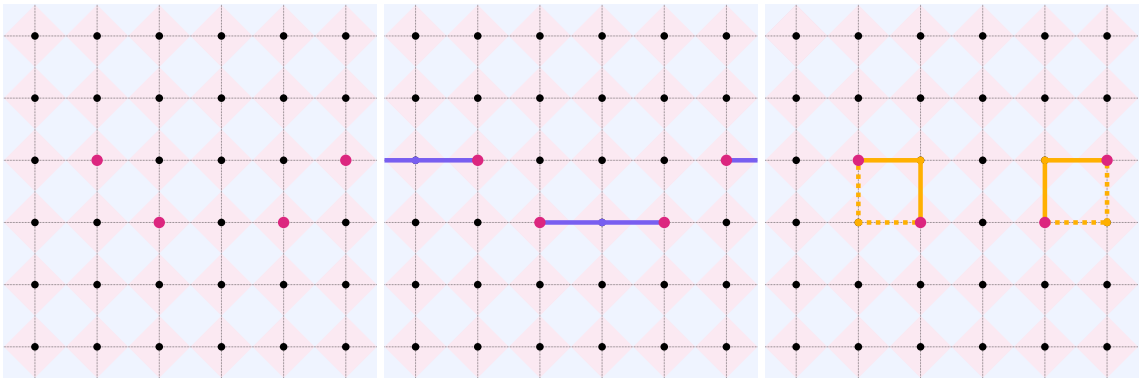


Figure 9: Comparison of potentially biased matching based decoders to their unbiased partition function decoder counterparts. In the absence of bias, an MWPM decoder is expected to act as an MP decoder while a weakly ensembled MWPM decoder is expected to behave as a dMP decoder. We notice a bias-induced deviation between decoding schemes primarily between MWPM and MP decoding at large even distances.



(a) Defect pattern. (b) Matching with degeneracy 1. (c) Matching with degeneracy 4.

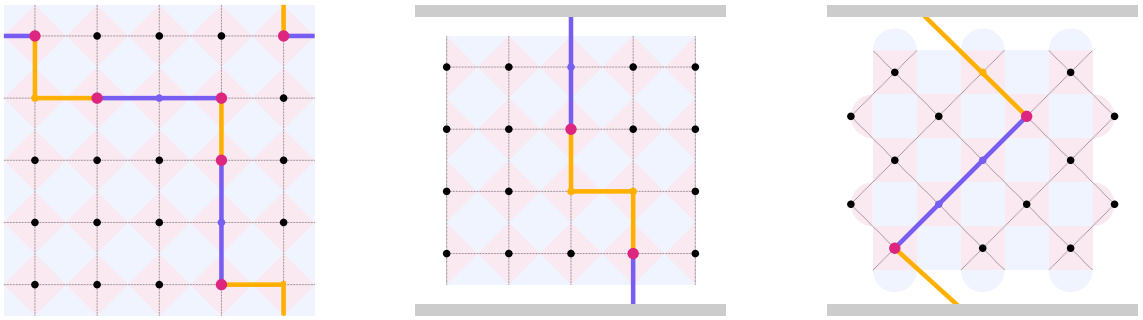
Figure 10: In Ref. [47] the authors give an example for a syndrome pattern with matchings of equal weight but different degeneracies, shown here on the 6×6 torus (a). On a larger torus, this pattern can be repeated arbitrarily often in horizontal direction. The first matching, shown in (b) corresponds to an error class with degeneracy 1. The syndrome admits another error class with degeneracy 4, shown in the matching in (c).

a lower bound on how significant the effect of degeneracy can be, as it lower-bounds the weight of errors that can give rise to syndromes $\vec{s}$ fulfilling the two criteria above.

In the toric code at even distance, this lower bound allows for effects from degeneracy in the leading term of the logical error rate, as the lowest-weight even logical representative is of weight d. Considering also the criterion $n_{\max}^{(1)} \neq n_{\max}^{(2)}$, however, shows that degeneracy can only show up in the first subleading term. A maximum probability error chain belongs to a set of equivalent error chains of equal weight ($n_{\max} > 1$) if and only if it can be deformed by stabilizer multiplication without changing its weight. On the square lattice, it is clear by inspection that this is only possible when the chain does not consist of only horizontal bonds or only vertical bonds, which excludes the leading term contribution. Degeneracy only affects the first subleading term, as illustrated in Fig. 10. In the toric code at odd distance, meanwhile, the effect of degeneracy is increasingly suppressed with d, since the lowest-weight even logical representative is of weight $2d$ (crossing the torus "diagonally"), as illustrated in Fig. 11 (a). From these considerations, we expect the performance improvement from weak ensembling to be very suppressed in the toric code at odd distance.

Extending the above considerations to the surface code, we expect improved logical performance from weak ensembling in the unrotated surface code at both even and odd distances, but only at even distances in the rotated surface code. The unrotated surface code has stabilizers of odd weight (the weight-three stabilizers at the boundary), so that both the odd and even distance code contain low-weight logical representatives of even length. In Fig. 11 (b) we illustrate a syndrome in the unrotated surface code that fulfills both of the above criteria, with leading-term effects on the logical performance. The rotated surface code contains only even-weight stabilizers, and all logical representatives have the same parity. At odd distance, we therefore expect that ensembling has no effect at all, while at even distance we again expect leading-term effects, as illustrated in Fig. 11 (c).

In Fig. 12, we show the effect of ensembling on logical performance for even and odd distance in the toric code, unrotated surface code and rotated surface code. We see that ensembling does not improve performance in the rotated surface code at odd distance, and barely differs from MWPM in the toric code at odd distance, while for the other combinations of boundaries and parities there is a noticeable (though modest) improvement. It should be noted that we have seen in Section 5.2 that bias lowers the performance of MWPM decoding compared to MP decoding. The performance gains from ensembling seen in the surface code might similarly stem from ensembling having robustness to bias, rather than from the degeneracy enhancement itself. As noted above, the effects of bias have a similar dependence on boundary conditions and the parity of the distance as the effects of degeneracy.



(a) Distance 5 toric code.

(b) Distance 5 unrotated surface code.

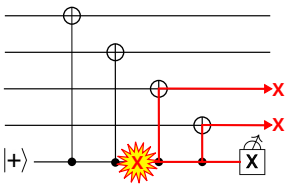
(c) Distance 6 rotated surface code.

Figure 11: Example syndrome pattern for the distance 5 toric code with a chain of two possible matchings with equal weight, wrapping around the torus in both directions (a). The matching shown in purple has degeneracy 1, whereas the matching colored orange has four equivalent alternatives. The unrotated planar surface codes admits such ambiguities as well (b), here shown with a matching of degeneracy 1 (purple) that includes the boundary nodes (gray shaded area) and a matching with degeneracy 3 (orange). For the unrotated surface code, equivalent matchings of different degeneracy can only occur for even distance (c).

5.4 Degeneracy and bias in other noise models

To explore if the effects of degeneracy and bias may be stronger in other noise models, we consider two candidates. For both the candidates we compare the performance under MWPM decoding and weakly ensembled MWPM decoding.

First, we note that the "diagonal" syndrome pattern of Fig. 10, which gives rise to a degeneracy of two each time it occurs, is precisely the one caused by an error on the ancilla qubit in the middle of a "bare ancilla" syndrome extraction circuit under a certain CNOT ordering. The circuit takes the following form, with the relevant error propagation marked in red:



(42)

We consider a noise model that only includes bitflip errors at this particular point in the bare ancilla syndrome extraction circuit for weight 4 X-type stabilizers, corresponding to pairs of bitflip errors on the adjacent data qubits. The CNOT ordering, or equivalently the possible pairs of affected qubits, is chosen such that the distance of the code is not reduced. (The pairs of qubits are aligned orthogonally to the logical bitflip operator $\bar{X}$.) We furthermore split the noise model in the construction of the decoding graph, meaning that the MWPM decoder does not take into account the correlations between the errors, but instead performs matching on a decoding graph that corresponds to regular bitflip noise. (Such splitting is performed in matching-based decoders in order to reduce a hypergraph matching problem into a graph matching problem, making it tractable.) In this setting, we find that ensembling has a significant advantage for even distances, as seen in Fig. 13. Interestingly, for $d = 4$, the observed performance difference is reversed: here, the bias in `PyMatching` apparently matches the patterns of correlated errors. We show this example separately in Appendix F, where we also compare ensembled MWPM decoding to perturbed MWPM decoding (by which we mean considering a single perturbation of the decoding graph). This example illustrates how biasing a matching-based decoder towards matchings that correspond to known correlations can improve performance.

Second, we consider the rotated surface code under a combination of bitflip and readout noise, since the 3D setting allows for more same-length paths between endpoints of error chains than the 2D setting. Here, however, we see only a modest performance difference between MWPM decoding and weakly ensembled MWPM decoding. The results are shown in Appendix F.

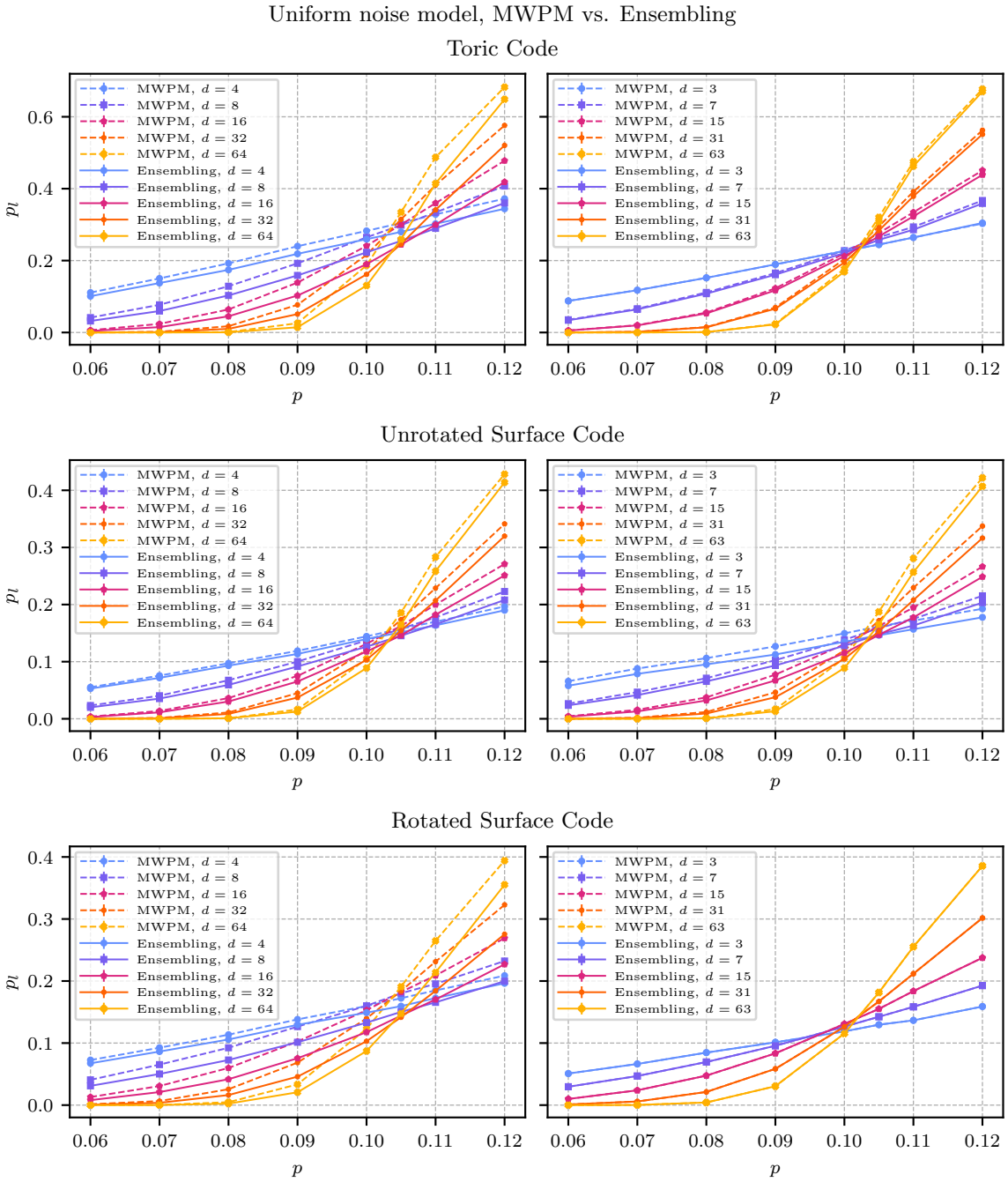


Figure 12: Comparison of MWPM decoding with ensembling using 50 random perturbations on the decoder graph for the toric code, unrotated surface code and rotated planar surface code under a uniform noise model. On the left the logical error curves for even distances are shown, on the right for odd distances. We find small improvements of the intersection of logical error curves and the physical error rates with ensembling for combinations of boundary conditions and distance parity that admit degeneracies.

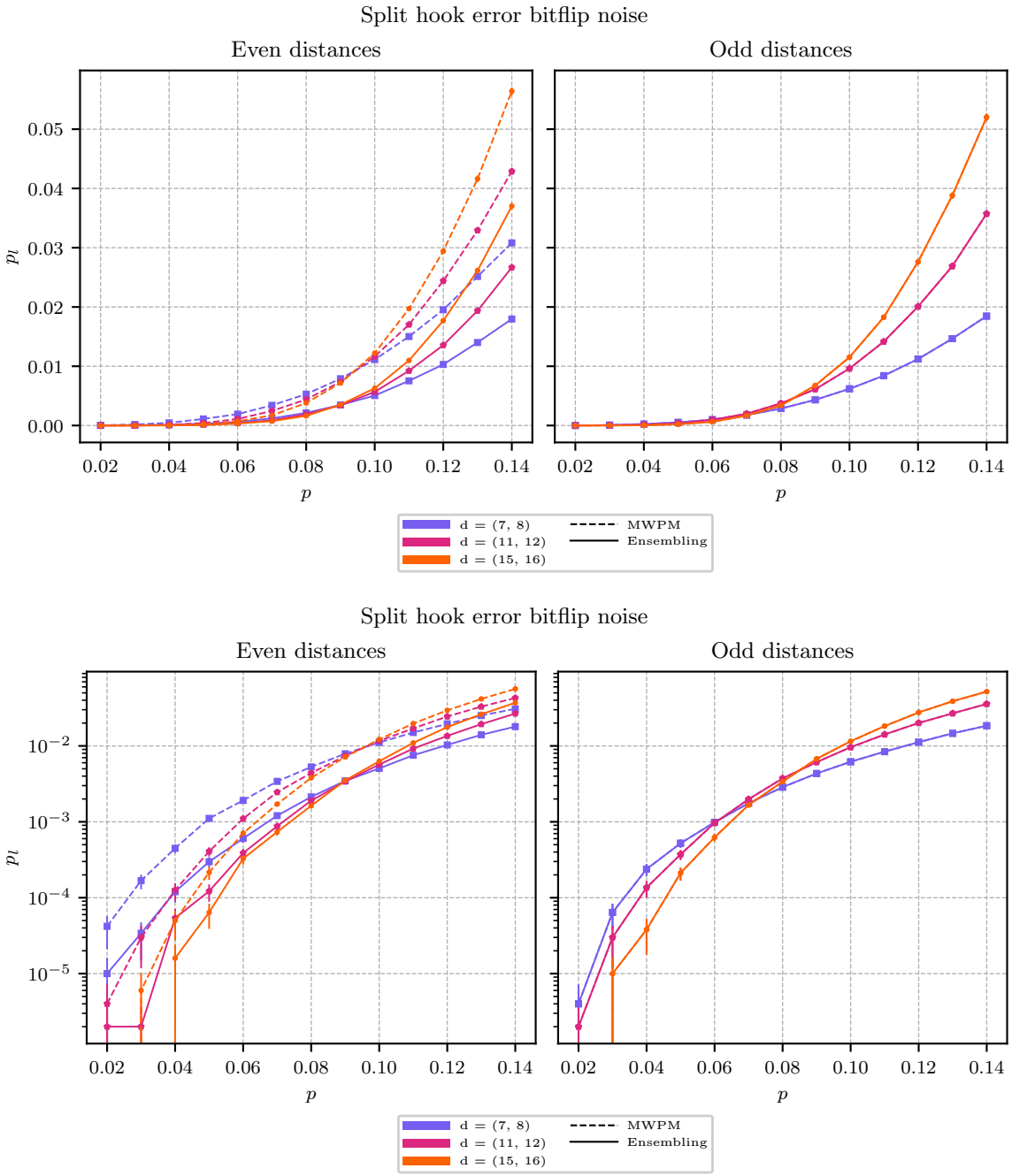


Figure 13: Top: Logical performance of the surface code under split bitflip hook errors on X-type stabilizers for even (left) and odd (right) code distances. Bottom: The same logical performance in log-log-scale, allowing the low-p behavior to be seen. The ensembling uses 50 random perturbations on the decoding graph and improves the performance of MWPM significantly for even code distances, while leaving the performance for odd code distances unchanged.

6 Strong ensembling, and performance estimates in the toric code under non-uniform bitflip noise

In the toric code under uniform bitflip noise, we have seen in Section 4 that dMP primarily provides benefits over MP for small, even code distances. The performance difference between dMP and ML decoding shows that there is room for further decoder improvements, beyond degeneracy enhancement. To deepen the analysis of decoder optimization potential, we extend our investigation to a non-uniform noise model. In this model, each qubit bitflip error rate, p_i, is independently drawn from a normal distribution with mean p and standard deviation σ_p. The error ratio samples are subsequently truncated to lie within the interval $[10^{-4}, \frac{1}{2}]$ which ensures physical plausibility. This noise model lifts the ground state degeneracies and is thus expected to reduce dMP decoding to MP decoding already for a small non-zero standard deviation.

In the following subsections, we show the results of simulations for MP, dMP, and ML decoders under the truncated Gaussian noise model for varying standard deviations $\sigma_p \in \{0.005, 0.06, 0.12\}$ with 10^4 samples of error configurations with Gaussian mean physical error rates $p \in [0.06, 0.14]$. The energy gap between ground and excited states may be arbitrarily small for non-uniform couplings introduced in Section 2.2. To suppress non groundstate contributions to performance estimates of MP and dMP decoding sufficiently, we are approximating the zero temperature limit by reduced temperature $T = 0.01 T_{\text{Nish}}$ and increased 9999 bits of precision within the FKT algorithm for non-uniform bitflip noise.

6.1 The effect of non-uniformity on the difference between MP and ML performance

Fig. 14 presents the performance estimates of MP, dMP, and ML decoders under the truncated Gaussian noise model. We observe that increasing the standard deviation generally improves overall code performance. This is shown by lower absolute logical failure rates and higher error thresholds at larger standard deviations of the noise model with respect to the same decoding strategy. This is expected as the decoding process for less uniform systems generally becomes easier when the overall error expectancy remains the same. For odd code distances, the relationship between decoding strategies remains largely unaffected by increased standard deviation: dMP decoding performs only as good as MP decoding for all investigated physical error rates, while ML decoding consistently outperforms dMP and MP decoding for larger error rates and code distances $d > 3$. Hence, weak ensembling methods are expected to produce no notable performance gain over MP decoding for odd code distances under non-uniform bitflip noise. Further, a clear performance gap to ML decoding persists across all tested standard deviations and code distances $d > 3$. In contrast, for even code distances, we observe that dMP decoding performs only as good as MP decoding already for small non uniformity in the noise. This indicates no performance enhancements by weak ensembling over standard MWPM for slightly non-uniform bitflip noise, as expected from the lifting of the ground state degeneracies. As a consequence, the ML decoder now outperforms dMP decoding at higher physical error rates already at small code distances, in contrast to the uniform case. It is important to note that for small enough physical error rates and high enough standard deviation all decoding strategies become indistinguishable, while the performance enhancement opportunities between MP/dMP toward ML decoding remain for high physical error rates within the tested range of standard deviations. The performance difference between ML and dMP/MP decoding decreases slightly with an increase of standard deviation.

6.2 Strong ensembling for non-uniform noise

We evaluated ensembling for the MWPM decoder for the toric code under the same truncated Gaussian noise model. Similar to the discussion in Section 5.1 we sample 50 perturbed matching graphs with modified edge weights:

$$w_i = \log \frac{1 - (p_i + \Xi_i)}{p_i + \Xi_i}, \tag{43}$$

where we sample $\Xi_i \sim \mathcal{N}(0, \sigma_\Xi)$. We again subjected the perturbation standard deviation σ_Ξ to optimization outlined in Appendix G and found $\sigma_\Xi = 0.025$ as an approximate optimum for the

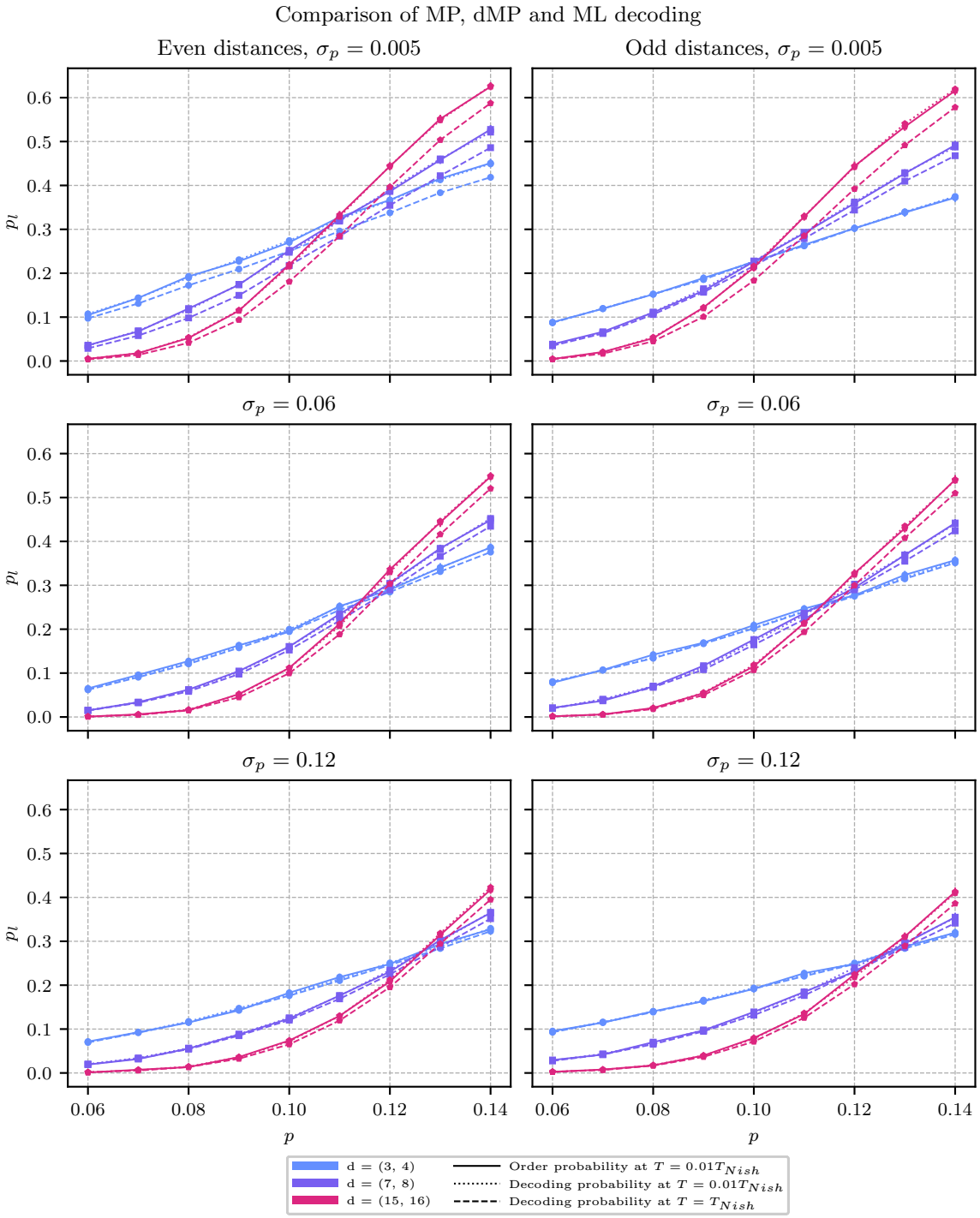


Figure 14: Comparison of MP, dMP, and ML decoding for varying standard deviations of the cut-off Gaussian bitflip noise model. We observe general performance improvement with an increase of standard deviation. We also see that the performance difference between dMP and ML decoding persists within investigated parameter range.

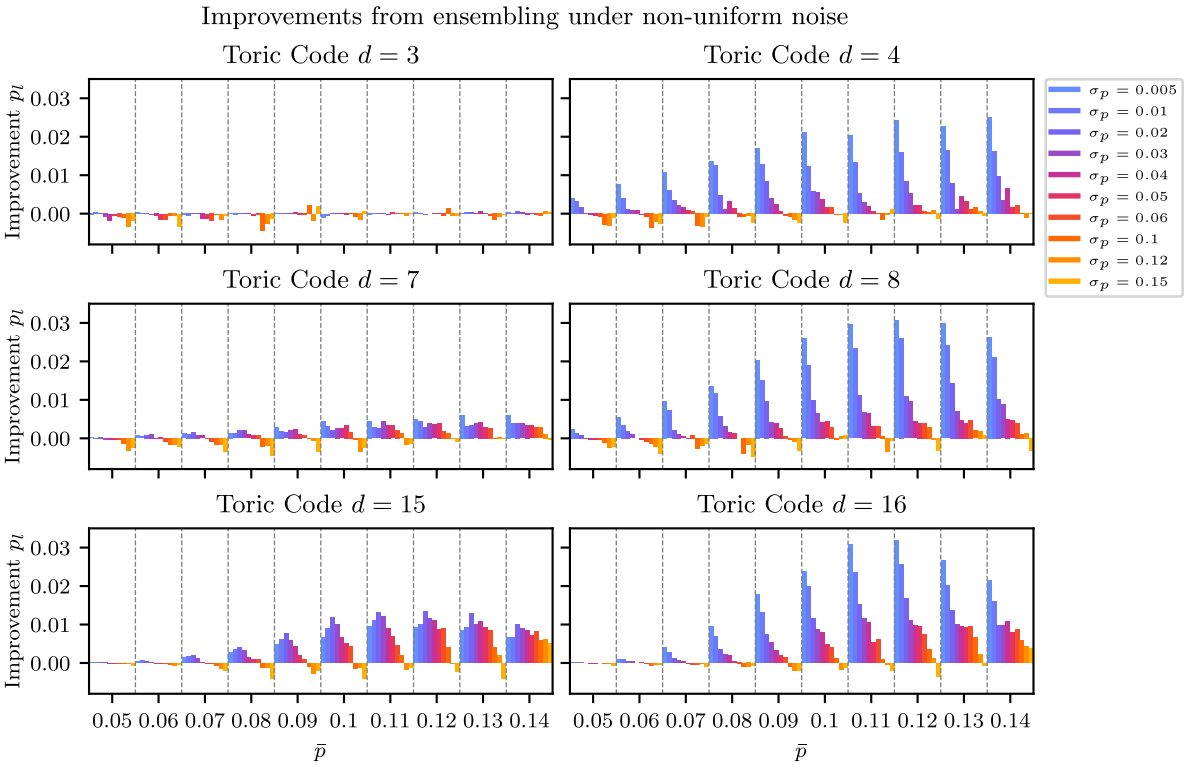


Figure 15: Improvement of the logical error rate under ensembling with $\sigma_\Xi = 0.025$ for a non-uniform noise model. For odd distances we see no or only very small improvements for $d = 15$, whereas for small standard deviation σ_p of the physical error rate and even distances, improvements up to 0.03 in p_l can be found. This effect decreases with larger standard deviations and vanishes for $\sigma_p = 0.15$. The improvements are more pronounced for physical error rates around the threshold, especially at larger distance.

configurations considered. Fig. 15 shows the improvement in the logical error rate p_l for a range of mean physical error rates, standard deviations for the non-uniform noise model, and distances $d = 3, 4, 7, 8, 15, 16$ for the toric code. For odd distances, we find no or very small improvements only, which aligns with the findings for a uniform noise model and the previous discussions about degeneracies in the toric code. For even distances, we see improvements in the logical error rate up to about 0.03, particularly for mean physical error rates around the threshold. The effect is most pronounced for small σ_p and vanishes for truncated Gaussian noise with a standard deviation for the physical error rate of $\sigma_p = 0.15$.

7 The color code under bitflip noise and depolarizing noise

As a first exploration into applying the partition function framework to other codes, we consider the color code on a hexagonal lattice under bitflip and depolarizing noise on the data qubits. In these settings, the FKT algorithm cannot be used. Given the long runtimes needed for Wang-Landau sampling in the toric code under bitflip noise, we turn instead to tensor network methods for partition function computations. Restricting to the Nishimori line, we employ the publicly available tensor network contraction package `SweepContractor.jl` [27]. This algorithm has shown reliable results for approximate maximum likelihood decoding of the color code under depolarizing noise.

The tensor network subject to the contraction constitutes of marginal error probability tensors associated with each logical qubit, connected to delta tensors for each stabilizer generator [2]. Errors are sampled according to the bitflip or depolarizing channel for varying physical error probabilities, inducing index flips in the marginal distribution tensors of the respective logical qubits. The contraction of the network sums over all error configuration within the same logical class, up to tensor truncations. The contraction acts directly on the code structure, making temperatures other than the Nishimori temperature (and thus the investigation of MP and dMP decoding) inaccessible by this method.

We observed that for given values of p and d, an insufficient maximum bond dimension $\chi_{\max}$ can introduce sign errors through truncation of the tensor network, leading to unphysical (negative) estimates of the logical class probabilities. To ensure sufficient precision while keeping the total runtime low, we adapted $\chi_{\max}$ to each combination of p and d, finding that the required values varied between $\chi_{\max} = 32$ and $\chi_{\max} = 128$ for the p and d considered, with the larger values within this range required under depolarizing noise at low p and large d.[5]

Using the partition functions computed through tensor network contraction, the maximum partition function decoder is constructed following Def. 1, and the decoding probability is evaluated in accordance with Def. 3, Eq. (24). In Fig. 16 the optimal logical error curves are shown for the color code under bitflip and depolarizing noise, computed both using decoding probability (ratio method) and by counting the number of failures of the maximum partition function decoder.

As in the case of the toric code under bitflip noise, we expect the ratio method to require fewer samples in order to achieve a given target confidence interval Δ. In Fig. 17, examples are shown of the confidence intervals achieved with the two methods. The scale factor $r(p, d)$ between the width of the confidence intervals obtained by the counting method and the ratio method again depends on p and d, as shown in Fig. 18. Example plots demonstrating the independence of $r(p, d)$ from the total sample size are shown in Appendix D. We recall from Section 4 that the relative required sample size to reach a target Δ is given by $R \to 1/r(p, d)^2$ in the large sample limit. While less dramatic efficiency gains are observed under depolarizing noise, where the increase in the required $\chi_{\max}$ limits the accessible range of p, the results point towards an increase in efficiency gain at low p under both bitflip and depolarizing noise.

[5] Methods like gauge fixing can also be applied to stabilize signs within the truncation [48], which could potentially lower the requirements on $\chi_{\max}$. We found that a minimal version of this method, which deterministically assigns the orientation of only the leading singular vector in the singular value decomposition, was insufficient to change the required $\chi_{\max}$ in our case, and did not attempt more elaborate versions of the scheme.

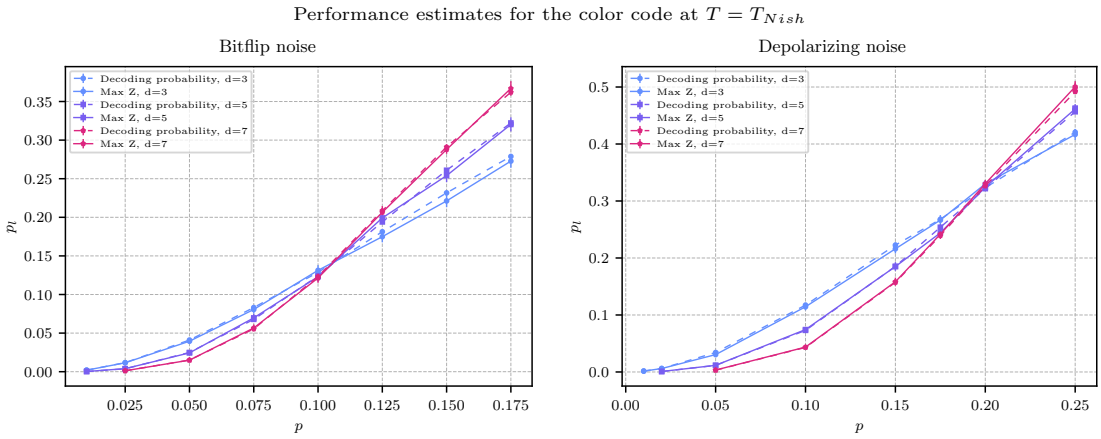


Figure 16: Estimated optimal performance for the color code under bitflip noise (left) and depolarizing noise (right). The performance estimates obtained from both the counting method ("Max Z") and ratio method are shown and yield compatible results. The corresponding threshold estimates are consistent with expected values of $p_{th} \sim 10.9\%$ for bitflip noise [49] and $p_{th} \sim 18.9\%$ for depolarizing noise [12].

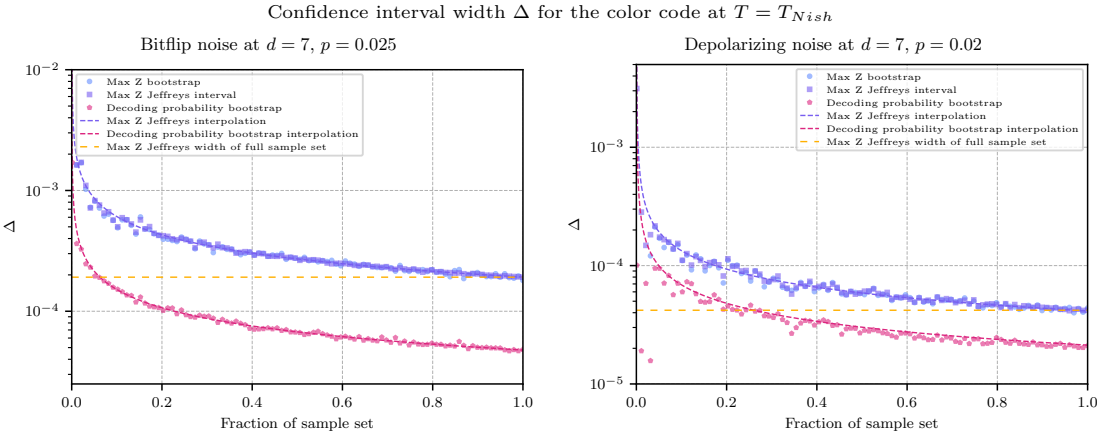


Figure 17: Confidence interval widths for estimates of the optimal logical error rates of the color code under bitflip noise at $d = 7, p = 0.025$ (left) and under depolarizing noise at $d = 7, p = 0.02$ (right), for varying fractions of the sample sets. The confidence interval widths are displayed for both the decoding probability ("ratio method") and the maximum partition function counting method. The counting method confidence interval widths for the full sample set, generated by the Jeffreys interval, are shown as horizontal lines. Their intersection points with the curves corresponding to the ratio method indicates that under bitflip noise at $d = 7, p = 0.025$, 6.3% of the sample set is sufficient for estimating the optimal code performance to the same accuracy with the ratio method, while the corresponding number under depolarizing noise at $d = 7, p = 0.02$ is 26%. The full sample sizes are 5×10^5 and 8×10^5, respectively.

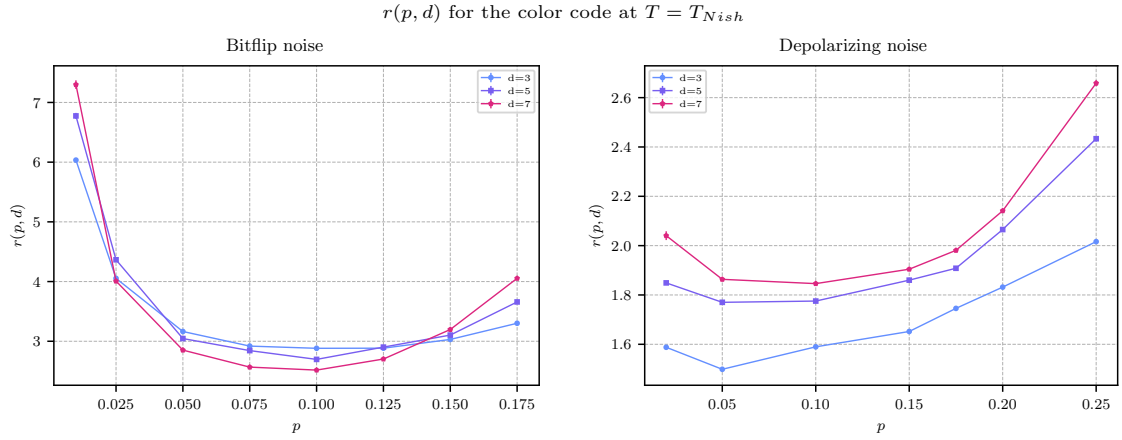


Figure 18: Scale factor $r(p,d)$ between the confidence interval widths of the counting and ratio methods for estimating optimal logical error rates of the color code, for different p, d. Examples of scale factor estimation for the color code are shown in Fig. 24. The results are shown for the color code under bitflip noise (left) and depolarizing noise (right). Under both noise models, the ratio method exhibits an increasing advantage in the low-p regime.

8 Discussion

In this paper, we have presented a framework for estimating logical performance in stabilizer codes under different decoding approaches. The framework relies on estimating partition functions in the corresponding statistical mechanics models. While the numerical results mainly focus on the toric code under bitflip noise, where the partition functions can be computed with Pfaffian methods, the method carries over to more general settings, such as more general qLDPC codes[6] and circuit level noise, where partition functions can be estimated through other means, using e.g., tensor network methods. We showcase such an application of our framework on top of tensor network methods by considering the color code under bitflip and depolarizing noise, although the distances considered are more limited.

The feasibility of extending to more general codes and noise models depends on the feasibility of computing or approximating partition functions – or, if restricting to the Nishimori line – logical error class probabilities. Tensor network methods have shown great promise for 2D problems, while (so far) being far more limited in 3D: in [27], distances up to 96 are reached for the surface code under bitflip noise, while in [4], where circuit level noise is considered for the surface code, the distances are far more modest (up to d=7). Outside of tensor network methods, the Wang-Landau method can also be applied to general settings. It is possible that there are regimes in which Wang-Landau could be preferable; we have not seen any systematic findings in the literature on which regimes would be the most interesting, but as far as we can tell it is not known to depend strongly on dimensionality and structure in the same way tensor network methods do, and it would be of interest to explore whether Wang-Landau may outperform tensor network methods for qLDPC codes that lack these features.

Compared to tensor network methods and Wang-Landau, the FKT algorithm used in this work is less likely to be useful for any realistic noise settings. However, in [50], a sampling-based algorithm known as the "worm" makes it possible to efficiently approximate logical error class probabilities (and, more generally, partition functions at different temperatures) for a slightly broader class of problems: those with a "matchable" structure. While still restrictive, this broader class includes the surface code under both bitflip and readout noise. The authors demonstrate how this method can be combined with correlated decoding to improve the surface code depolarizing threshold over correlated matching, exemplifying how methods that can handle only part of the decoding problem optimally may still improve the overall performance. A different matching-based

[6] See e.g. Ref. [16], which leverages the statistical mechanics mapping to explore the decoding transition in non-local, finite-rate qLDPC codes.

method for estimating logical error class probabilities, by finding the first K minimum weight matchings from the decoding graph, is given in [51].

The partition function framework that we define allows for apples-to-apples comparisons of the logical error curves of different codes, avoiding confounding factors from code-specific decoder implementations. One example of a confounding factor in the comparison of codes and decoders is bias within matching implementations. As was pointed out in [44], this bias can also lead to inaccurate estimates of the location of the phase boundary of the corresponding statistical mechanics model. In Fig. 12, ensembling is seen to outperform MWPM for all distances, while in Fig. 4 MP and dMP are seen to agree in the toric code at large distance. This shows that the improvement in Fig. 12 is not due to an inherent difference between MP and dMP at large distance, but rather due to ensembling suffering less from bias in the `PyMatching` implementation (as is also seen in Fig. 9). Furthermore, the framework is sample-efficient compared to conventional sampling of decoder failures. The sample efficiency depends on p and d, and the results indicate that estimation through ratios of partition functions may be particularly useful at low p and large d, meaning at low logical error rates, where conventional sampling struggles. Probing the regime of low logical error rates is important, as this is the regime relevant for large-scale quantum computing. Other recent work for estimating or bounding the behavior in this regime includes Ref. [52].

The bitflip noise model is far from realistic, and non-uniform noise is furthermore unlikely to be normally distributed. Future work would be needed to establish which of the qualitative features seen above would generalize to more realistic settings. For instance, we see that the performance improves for non-uniform noise when the decoder is given the individual qubit fidelities. For correlated noise such as circuit noise there could be an overall performance reduction compared to uniform noise. Similarly, qualitative observations concerning the performance gains from ensembling (that they are most notable around the threshold, and that they decrease for strongly non-uniform noise) may change in the presence of correlations.

The role of degeneracy would also merit further work. For bitflip noise, it was demonstrated in [53] how the increased amount of degeneracy in the rotated surface code can lead to worse logical performance than that of the unrotated surface code in certain regimes, despite having a distance that is larger by a factor of $\sqrt{2}$ for the same number of physical qubits. In [44] the parity of the distance was seen to clearly influence the threshold estimates when matching degeneracy is taken into account, in the setting of the toric code under bitflip noise. These observations of the effects of boundaries and parity agree with those of the present work. Future work would be needed to quantify the effect of degeneracy and its dependence on boundary conditions and parity in more general settings. In stabilizer codes or noise models with stronger effects from degeneracy, methods such as ensembling could lead to larger gains. We see an illustrative example of this in the case of the rotated surface code under a split hook error noise model.

The numerical simulations in the present work have focused on $T = T_{\text{Nish}}$ and the zero temperature limit, as these are the temperatures relevant for ML, MP and dMP decoding. Here, we find that the threshold estimates for maximum partition function decoding and probabilistic partition function decoding agree. This is consistent with a maximum partition function decodability boundary that is the same as the phase boundary (although the possibility remains that they may disagree at other temperatures). We leave as an open question whether there are statistical mechanics models where the maximum partition function decodability boundary differs from the phase boundary, or whether it is possible to prove that these boundaries must agree.

9 Author contributions

LW performed the FKT simulations and tensor network simulations, and HH performed the `PyMatching` simulations. LW and HH also analyzed the results. LGS conceived the project and developed the theory. EM developed the initial CUDA code then LW took over. EM and LGS guided the direction of the project, and supervised the numerical simulations. All authors contributed to the discussion of the results and the writing of the manuscript.

10 Acknowledgments

LGS is supported through a Leverhulme-Peierls Fellowship at the University of Oxford, funded by grant no. LIP-2020-014. LW, HH and EM acknowledge funding by the German Ministry of Economic Affairs and Climate Action (BMWK) and the German Aerospace Center (DLR) in project QuDA-KI under grant no. 50RA2206.

We thank Matthew Steinberg for related discussions and collaboration, Marius Beuerle for initial collaboration, Benedikt Placke for discussion, and Bela Bauer and Christina Knapp for discussions about non-uniform noise. We than Steven Simon and Matthew Steinberg for feedback on a draft version of the manuscript.

A Zero Temperature limit

In order to determine a temperature scale sufficient to estimate decoder performance in the zero temperature limit with the FKT algorithm, we performed simulations with varying temperature and bits of precision parameters. The performance estimates for a selection of these simulations are shown in Fig. 19 and Fig. 20. Fig. 19 shows a comparison of FKT performance estimates of maximum partition function decoding at 9192 bits of precision for $T = 0.01T_{\text{Nish}}$ and 4096 bits of precision for $T = 0.1T_{\text{Nish}}$. The performance estimates are matching within error bars. Additionally, Fig. 20 shows the WL performance estimates of maximum partition function decoding at $T = 0.1T_{\text{Nish}}$ and WL performance estimates of maximum partition function decoding at $T = 0$. The latter reduces to maximizing $g(E_{\min}(\vec{s}))$, a quantity that WL gives direct access to. The figure shows very close matching of performance estimates. Fig. 2 depicts a further comparison between WL performance estimates of maximum partition function decoding at $T = 0$ and FKT results of maximum partition function decoding at $T = 0.1T_{\text{Nish}}$ with 4096 bits of precision. As the performance estimates are matching within error bars, we determine $T = 0.1T_{\text{Nish}}$ and 4096 bits of precision to be sufficient parameters to estimate the low temperature performance with the FKT algorithm.

B FKT algorithm

The toric code under bitflip noise maps under the statistical mechanics mapping to a two-dimensional Random Bond Ising Model, as shown in Eq. (19). The Ising lattice defines a graph $G = (V, E)$ by associating Ising spins with vertices and interactions between spins with edges. The dual-lattice $G^\star$ forms an $m \times n$ square lattice with spin degrees of freedom associated with faces, interactions transmitted via edges between neighboring faces, and periodic boundary conditions. A spin configuration on the Ising lattice is described by a face value assignment $\{S_i | S_i \in \pm 1 \wedge 0 \leq i < m \times n\}$ on $G^\star$. The correspondence between spin configurations on $G^\star$ and relative domain walls with additional fixing of a single spin value is depicted in Fig. 21. Furthermore, by decorating each vertex in $G^\star$ with a Kasteleyn city as depicted in Fig. 22, one realizes a map between perfect matchings in the decorated graph $\tilde{G}^\star$ and spin configurations of the Ising model. We note that this map is not one-to-one. We endow $\tilde{G}^\star$ with an orientation D and according edge weights by $\omega((i,j)) = 0$ for $i, j \in$ vertices of same Kasteleyn city and $\omega((i,j)) = J_{ij}$ for $(i,j) \in$ D and i, j from different cities. We define the Kasteleyn matrix $K = (K_{ij})_{i,j \in V}$ as the adjacency matrix of $\tilde{G}^\star$, which carries information on connectivity, weights and the orientation of the graph as follows: $K_{ij} = e^{-2\beta\omega((i,j))}$ if the edge (i,j) is an oriented edge in D, $K_{ij} = -e^{-2\beta\omega((j,i))}$ if (j,i) is an oriented edge in D, and

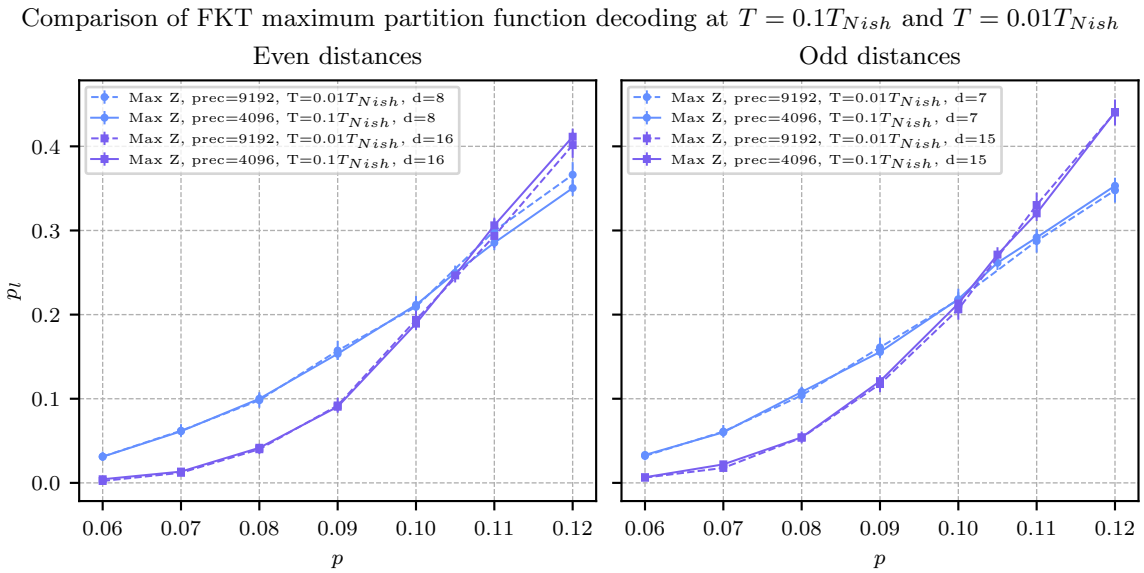


Figure 19: Comparison of FKT performance estimates of maximum partition function decoding at $T = 0.1T_{\text{Nish}}$ and $T = 0.01T_{\text{Nish}}$ for respective number of bits of precision 4096 and 9192.

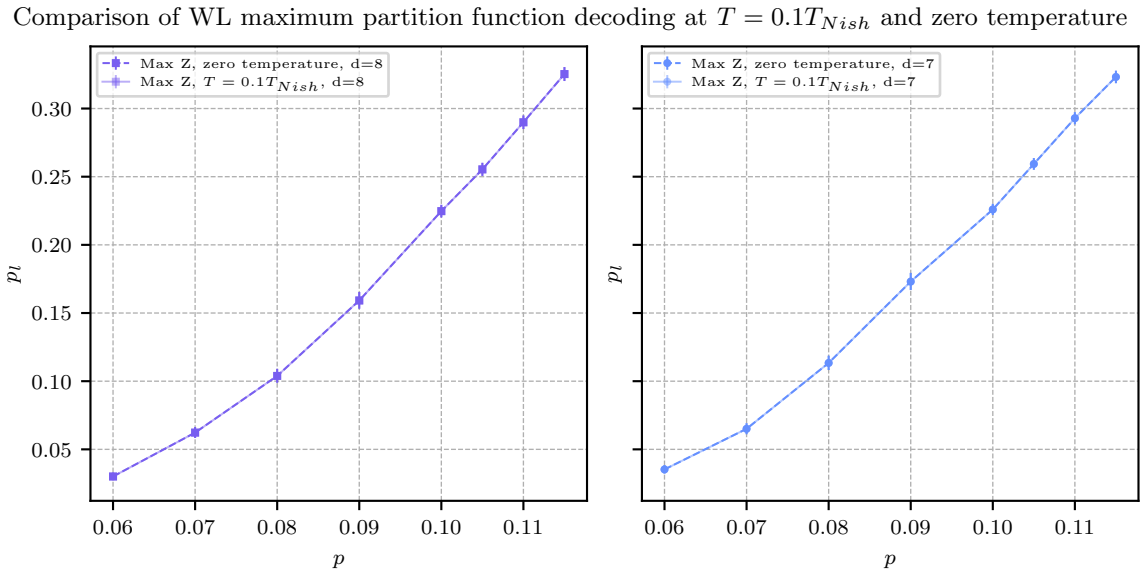


Figure 20: Comparison of WL performance estimates of maximum partition function decoding at $T = 0.1T_{\text{Nishimori}}$ and $T = 0$.

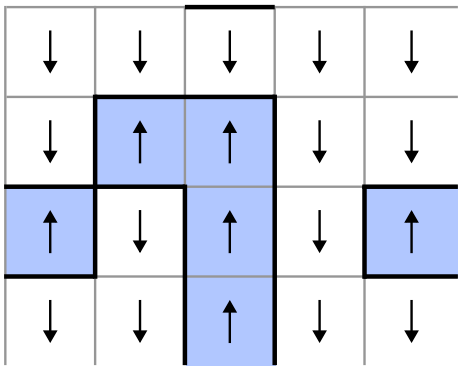


Figure 21: Relative domain walls of a spin configuration on $G^\star$. To describe a spin configuration uniquely we have to define relative domain walls and additionally the spin value on a single face.

$K_{ij} = 0$ otherwise. The skew-symmetric weighted adjacency matrix $K = (K_{ij})_{i,j}$ has dimension $4nm \times 4nm$. The Pfaffian of a $2N \times 2N$ skew-symmetric matrix K is defined by:

$$Pf(K) = \frac{1}{n!2^n} \sum_{\pi \in S_{2n}} \sigma(\pi) \prod_{j=1}^{n} K_{\pi(2j-1),\pi(2j)} \tag{44}$$

with $\sigma(\pi)$ the sign of the permutation. All non vanishing terms in the Pfaffian correspond to products of edge weights of a perfect matching on the related graph with signs depending on the orientation. For all planar graphs there exists an orientation called Pfaffian orientation which ensures that all perfect matchings contribute with the same sign to the Pfaffian. One can show for planar graphs that $Z = Pf(K_{\mathrm{Pf}})$ with the Kasteleyn matrix K_{Pf} corresponding to a Pfaffian orientation of $\tilde{G}^\star$. Notice that in the non planar case not all perfect matchings on $\tilde{G}^\star$ are associated with physical spin configurations; domain walls which form non trivial loops around the torus must come in pairs while dimer configurations can come in four distinct ways: (o,o), (o,e), (e,o), (e,e). The tuples denote even or odd wrapping number along the two dimensions of the torus. Only the (e,e) component is related to physical spin configurations. To relate the partition function of an Ising model with periodic boundary conditions to the calculation of Pfaffians one chooses four different orientations on $\tilde{G}^\star$. The four orientations differ only on edges at the boundary (the edges which wrap around the surface) and are equal to a specific Pfaffian orientation on the bulk. Specifically, one assigns uniformly for the first row and column either $K_{ij} = e^{-2\beta\omega((j,i))}$ or $K_{ij} = -e^{-2\beta\omega((j,i))}$. Thus, the boundary orientation defines four different Kasteleyn matrices which are labelled by the chosen signs K^{++}, K^{+-}, K^{-+} and K^{--}. The Pfaffian of each of these Kasteleyn matrices contains summands which correspond to non trivial domain wall loops which can not be related to spin configurations on the Ising model. These summands cancel if summed over all boundary orientations. Only the (e,e) components participate which leads to: $2Z = Pf(K^{++}) + Pf(K^{+-}) + Pf(K^{-+}) + Pf(K^{--})$.

C Replica-exchange Wang-Landau algorithm

The replica-exchange Wang-Landau algorithm [36, 39], is a Monte Carlo algorithm to estimate the density of states $g(E_i)$ over an energy spectrum divided into bins E_i of a classical spin system described by the Hamiltonian H. We summarize the algorithm in Alg. 1.

The algorithm relies on the fact that a random walk over the energy spectrum $S = \{E_i\}$ with probability proportional to $\frac{1}{g(E_i)}$ will produce flat histograms with support on the energy spectrum of H. Access to the density of states enables the calculation of the partition function for arbitrary temperatures, including the zero-temperature limit, in contrast to the finite-temperature restriction of the FKT algorithm. We start by explaining the general procedure of the Wang-Landau (WL) algorithm and continue by explaining the chosen replica exchange (RE) parallelization scheme. The WL algorithm starts by initializing an array $\log(g(E_i)) = 0 \ \forall \ E_i \in S$, which holds the density of states estimate after completing the algorithm and a histogram $h(E_i) = 0 \ \forall \ E_i \in S$. In

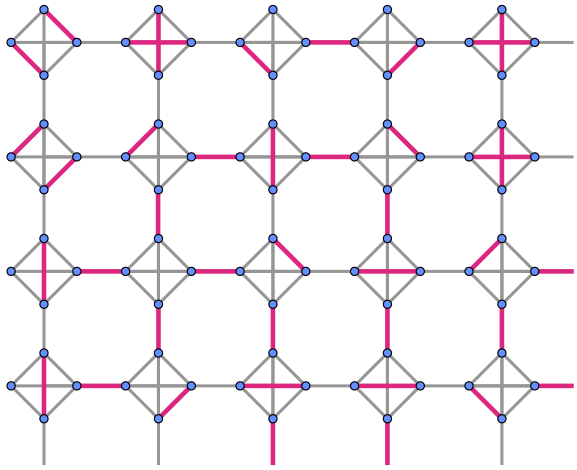


Figure 22: Mapping of domain walls to a dimer cover of $\tilde{G}^\star$ with Kasteleyn city decoration on original vertices in $G^\star$. Observe that a dimer configuration is unique at cities with inter city connections determined by the respective domain wall. Vertices not participating in a domain wall can be covered in three different ways by dimers. These three possibilities are shown on the three cities on the top left corner. Thus, the map is not one to one between domain walls and dimer coverings. Furthermore, the map is invariant under global spin flip and thus not injective.

addition, one initializes an update factor $f = e$. Subsequently, random spin flips are performed. The move is accepted with transition probability $p(E_1 \to E_2) = \min\{\frac{g(E_1)}{g(E_2)}, 1\}$, with energy of the spin configuration E_1 before the spin flip and E_2 after the flip. For each update step, a potentially new configuration with energy E is realized. The histogram and density arrays are updated accordingly: $h(E) \mapsto h(E) + 1$ and $\log(g(E)) \mapsto \log(g(E)) + \log(f)$. The random walk continues until the histogram satisfies the flatness condition determined by a flatness parameter α: $\min(h(E_i)) \geq \alpha \cdot \text{mean}(h(E_i))$. If the histogram is flat under this condition, the update parameter is modified by $f \mapsto \sqrt{f}$ and the histogram is reset $h(E_i) = 0 \; \forall \; E_i \in S$. The random walk continues as long as the update factor f is not below a threshold determined by a run parameter β: $f \geq e^\beta$. The parameter β controls the precision of the simulation and thus the total number of random spin flips. The resulting $\log(g(E_i))$ contains relative degeneracy factors that must be rescaled to match physical degrees of freedom. We are using the total number of spin configurations to rescale the degeneracy factors by $\sum_{E_i \in S} g(E_i) \stackrel{!}{=} 2^{\#\text{spins}}$. The algorithm requires knowledge on the energy spectrum, which is estimated by performing random walks over the energy space with the WL transition probability over a fixed number of steps within our investigation.

Algorithm 1: Wang-Landau Algorithm

Input: $\{E_i\}$: Energy spectrum, H: Hamiltonian, α: Flatness condition, β: Minimal update factor, N: Number of MC steps per iteration

Output: Density of states $\log g(E_i)$

Initialize density estimates $\log g(E_i) = 0 \; \forall \; E_i$
Initialize histogram $h(E_i) = 0 \; \forall \; E_i$
Initialize update factor $f = e$
Initialize spin system S
Calculate energy E of S
while $f \geq e^{\beta}$ **do**
  for N *times* **do**
    Random spin flip Calculate energy after flip $\tilde{E}$
    Accept spin flip with transition probability $p = \min\left\{\frac{g(E)}{g(\tilde{E})}, 1\right\}$
    Store new E
    $h(E) += 1$
    $\log(g(E)) += \log(f)$
  end for
  if $\min(h) \geq \alpha \cdot mean(h)$ **then**
    Update $f = \sqrt{f}$
    Reset $h(E_i) = 0 \; \forall \; E_i$
  end if
end while

D Examples of confidence interval ratios

Examples of ratios of confidence interval widths for estimation of logical error rates through counting failures of a maximum partition function ("counting method) and through the decoding probability ("ratio method) are shown in Fig. 23 (toric code under bitflip noise) and Fig. 24 (color code under bitflip and depolarizing noise). The confidence interval widths for the counting method are displayed both using bootstrapping with 1000 resamples and using Jeffreys interval. We find that the confidence interval widths between the counting method and the ratio method differ by a scale factor $r(p, d)$ that is independent of the total number of samples, with sample-dependent fluctuations that vanish with increased sample size. The values of $r(p, d)$ shown in Figs. 7 and 18 are obtained by taking the mean over the values obtained from subsets of the sample that include more than 0.1 of the full sample size. The error bars on $r(p, d)$ are obtained by the corresponding standard deviation.

The independence of $r(p, d)$ from the sample size is the basis of the ansatz used in the interpolation of the confidence intervals of the ratio method in Figs. 6 and 17, and to estimate the relative required sample sizes R to reach a given target confidence interval Δ, as discussed in Section 4.

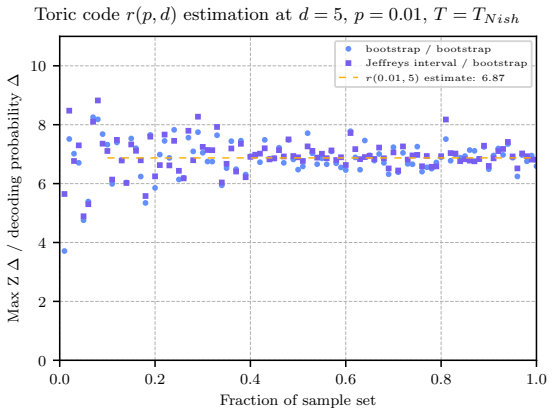


Figure 23: Ratios of confidence interval widths Δ for estimates of optimal logical error rates by counting method and by ratio method, for the toric code under bitflip noise, using subsamples containing different fractions of the full sample size of just below two million physical error configurations. The values of Δ for the counting method are generated using both Jeffreys interval and bootstrapping, while the values of Δ for the ratio method are generated using bootstrapping only. The labels "bootstrap/bootstrap" and "Jeffreys interval/bootstrap" indicate what method was used for counting method/ratio method to generate a given data point. We derive an estimate of the scale factor $r(0.01, 5)$ by taking the mean of the data points (yellow line). When computing the mean we exclude data points at sample set fractions below 0.1, since these contain very few logical failures and lead to a large uncertainty in the counting method. Estimates of $r(p, d)$ for varying p, d, derived in the same manner, are displayed in Fig. 7.

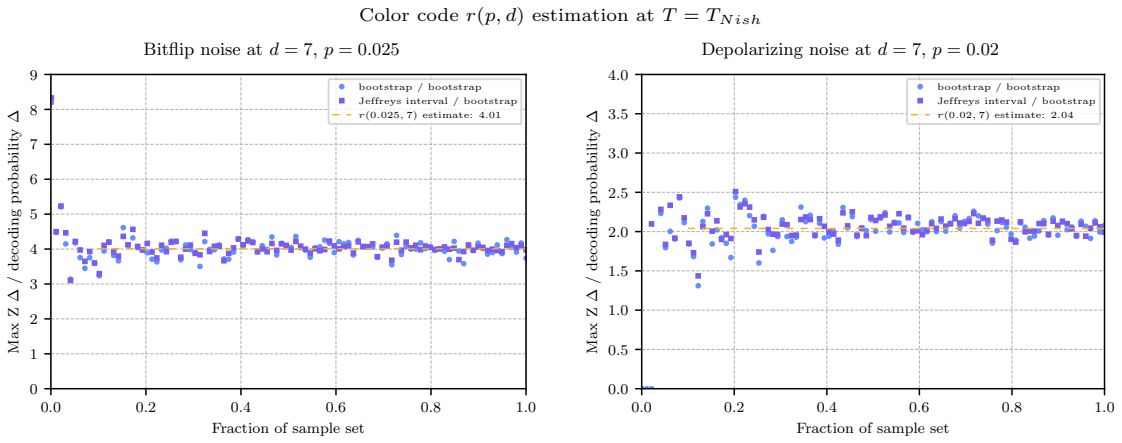


Figure 24: Ratios of confidence interval widths Δ for estimates of optimal logical error rates by counting method and by ratio method, for the color code under bitflip noise (left) and depolarizing noise (right), using subsamples containing different fractions of the full sample sizes, which are 5×10^5 and 8×10^5, respectively. Same method is used as in Fig. 23 to estimate the corresponding values of $r(p, d)$. Estimates of $r(p, d)$ for varying p, d, derived in the same manner, are displayed in Fig. 18.

E Minimum-Weight Perfect Matching Decoder

A Minimum-weight perfect matching (MWPM) decoder is a maximum probability decoder, that finds a correction operator $e(\vec{s})$ consistent with a syndrome $\vec{s}$, that is

$$P(e(\vec{s})) \geq P(e'(\vec{s})), \quad \forall e'(\vec{s}). \tag{45}$$

We limit our discussion to the toric code as well as the unrotated and rotated planar surface codes, though MWPM decoding is applicable to a considerable number of quantum error correction codes [54]. Decoding is generally performed independently for Pauli-X and Pauli-Z errors, by the same means. Given a Pauli-Z error of the form $e \in \{Z, I\}^m$, we denote by $\vec{e} \in \mathbb{Z}_2^m$ the binary vector with $\vec{e}_i = 1$ if an error occurred at qubit i and $\vec{e}_i = 0$ otherwise. The error models is fully described by a three objects. The detector check matrix $H \in \mathbb{Z}_2^{n \times m}$ has a row for each detector measurement and a column for each error mechanism, with $H_{ij} = 1$ if detector i is flipped by error mechanism j and $H_{ij} = 0$ otherwise. Each error mechanism i occurs with probability p_i, described by $\vec{p} \in [0, 1]^m$. Furthermore, the effect of errors on the logical observables is captured in $O \in \mathbb{Z}_2^{n_l \times m}$, with $O_{ij} = 1$ if the logical observable i is flipped by error j and $O_{ij} = 0$ otherwise. Such a general description of an error model by H, O and $\vec{p}$ is readily available from e.g., a stabilizer circuit simulator such as `Stim` [55] by forward propagating Pauli errors through the circuit. The steps of the decoding procedure are shown in Fig. 25. The error model induces the matching graph $\mathcal{G}_M$ with incidence matrix H, where each stabilizer measurement is a node and each error mechanism an edge. For codes that admit error chains with a single defect, such as the rotated and unrotated planar surface code for error chains ending at the edge of the lattice, an additional boundary node is introduced for each such stabilizer measurement, with all boundary nodes connected by edges of weight 0. The remaining edges in the graph are weighted by [1]

$$w_i = \log \frac{1 - p_i}{p_i}, \tag{46}$$

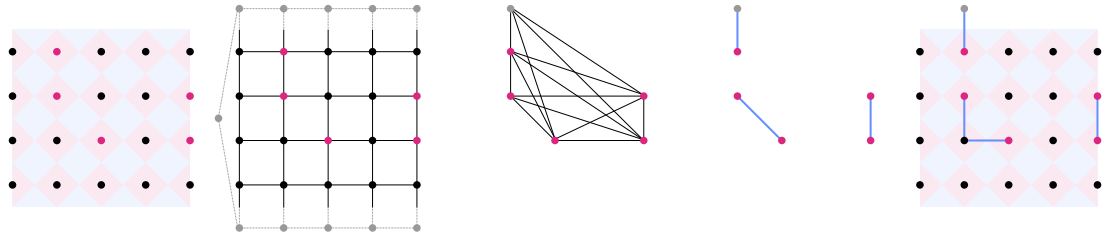
such that edges corresponding to more probable errors have lower weights. Every error $\vec{e} \in \mathbb{Z}_2^n$ produces a syndrome $\vec{s} = H\vec{e}$ which needs to be decoded into a correction operator $\vec{g} \in \mathbb{Z}_2^m$ with $\vec{g} \equiv e(\vec{s})$ fulfilling Eq. (45). From the defect nodes in $\vec{s}$ and the matching graph $\mathcal{G}_M$ the syndrome graph $\mathcal{G}_{\vec{s}}$ shown in Fig. 25c is constructed. It is a complete graph of all the defect nodes and the shortest paths between them in $\mathcal{G}_M$ as edges. A matching in $\mathcal{G}_{\vec{s}}$ is a set of edges in $\mathcal{G}_{\vec{s}}$, such that no two edges are incident to the same node. A perfect matching is a matching containing all nodes of the syndrome graph, it also is a minimum-weight perfect matching, if it fulfills these conditions and the sum of the edge weights contained in the matching is minimal. Any minumum-weight perfect matching in $\mathcal{G}_{\vec{s}}$ corresponds to a correction operator $\vec{g} \in \mathbb{Z}_2^m$ that is consistent with the syndrome and furthermore fulfills Eq. (45). The decoding is successful, if $\vec{g}$ and the (unknown) error $\vec{e}$ that caused syndrome $\vec{s}$ have the same effect on O, that is

$$O(\vec{e} \oplus \vec{g}) = 0. \tag{47}$$

Otherwise a logical error occurs. For the efficient computation of a minimum-weight perfect matching, Edmonds' blossom algorithm [56] is the common method, of which various high-quality implementations and variants exist [23, 45, 54, 57, 58]. Here we conduct all simulations involving minimum-weight perfect matching decoding with the `PyMatching 2` software package, implementing the sparse blossom algorithm [23].

F Further results for MWPM Decoding and Ensembling under other noise models

This Appendix contains further plots for Section 5.4. Fig. 26 shows the effect of ensembling on the rotated surface code under split hook error bitflip noise at $d = 4$. The results for $d = 3$ are not shown since there are only two error locations at this distance, and no combination of errors on these two locations leads to a logical error. Fig. 27 shows the effect of ensembling on the rotated surface code under combined bitflip and readout noise, with d rounds of stabilizer measurements. The readout noise is applied to Z-type stabilizers only.



(a) Surface Code. (b) Matching Graph. (c) Syndrome Graph. (d) Matching. (e) Correction.

Figure 25: The Pauli-X stabilizer measurements of the distance 5 surface code (a), induce a matching graph (b) which is extended by boundary nodes. From the matching graph, the syndrome graph is constructed as a complete graph of the defect nodes (red) and the shortest paths between them (c). A minimum-weight perfect matching in the syndrome graph (d) corresponds to a most probable correction operator consistent with the syndrome (e).

G Optimization of Ensembling Parameters

Ensembling by perturbing the matching graph of the MWPM decoder as discussed for a uniform noise model in Section 5.1 and a non-uniform noise model in Section 6.2, requires setting the standard deviation for the edge weight perturbations σ_ξ and σ_Ξ. We optimized both values using a simple coarse grained global optimization scheme [59] within the interval $[0, 0.5]$ for both standard deviations and furthermore tested multiple values within the neighborhood of the determined approximate optimum. Due to the considerable computational cost of this optimization, we limited the search to the toric code with distances $d = 8$ and $d = 16$, error rates of $p \in \{0.05, 0.07, 0.1, 0.11\}$, standard deviations for the non-uniform noise model of $\sigma_p \in \{0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.15\}$, and 10,000 sampled error configurations for each optimization step. For the uniform noise model, we consistently find the best results for $\sigma_\xi \in [10^{-6}, 10^{-2}]$, for larger values the observed logical error rate increases monotonously, considerably smaller values lead to slight increases. The optimization results for the uniform noise model are shown in Fig. 28. With a truncated Gaussian noise model, across the tested error rate standard deviations, an approximate optimum for all tested configurations can be observed at $\sigma_\Xi \approx 0.025$, which we set for all ensembling simulations with the MWPM decoder and non-uniform noise-models. The optimization results for the toric code with $d = 8$ and $d = 16$ are shown in Fig. 29 and Fig. 30.

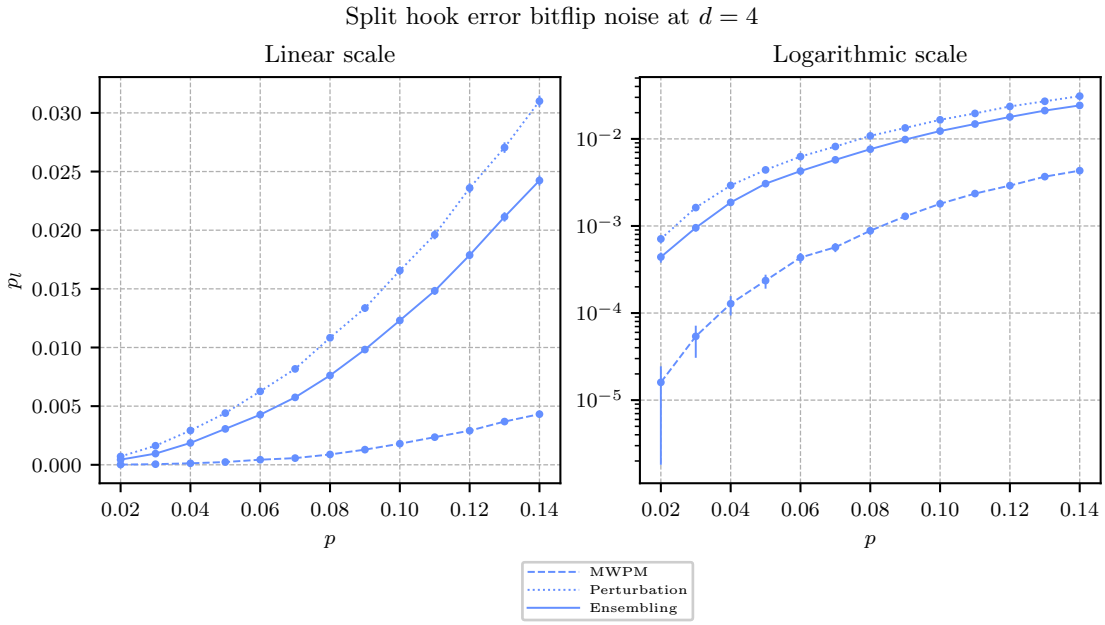


Figure 26: Left: Logical performance of the surface code under split bitflip hook errors on X-type stabilizers for $d = 4$. Right: The same logical performance in log-log-scale, allowing the low-p behavior to be seen. The ensembling uses 50 random perturbations. The results labeled "Perturbation" were generated using one single perturbation. MWPM significantly outperforms both ensembled MWPM and perturbed MPWM, indicating that the bias in `PyMatching` aligns with the split hook errors at $d = 4$.

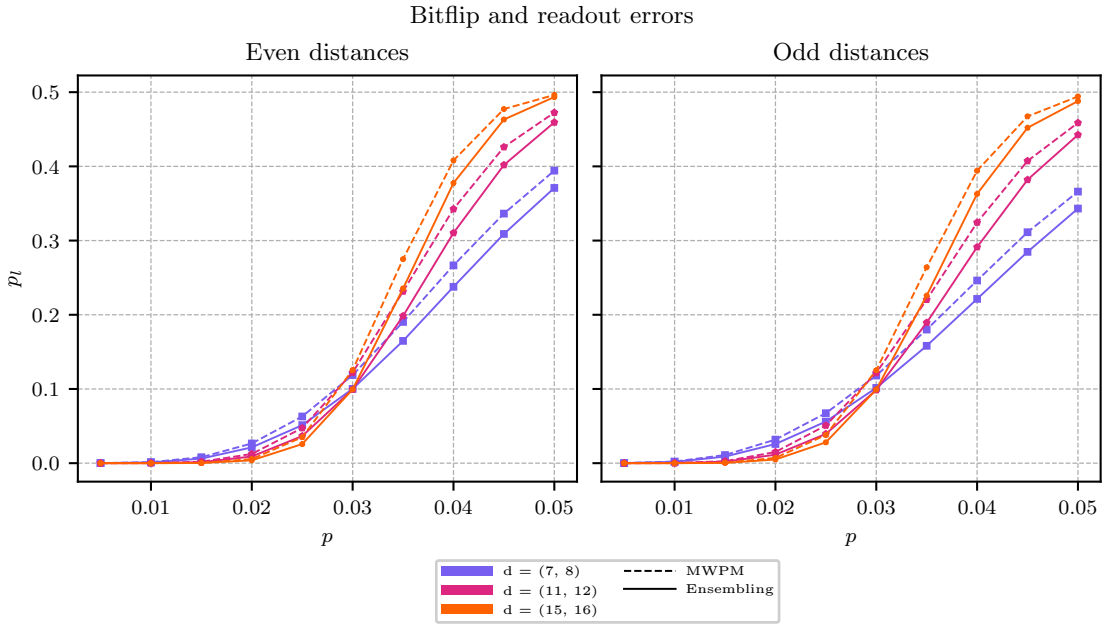


Figure 27: Logical performance of the surface code under combined bitflip and readout noise for even (left) and odd (right) code distances. The ensembling uses 50 random perturbations on the decoding graph. Both even and odd distances show modest performance improvement under ensembling. The observed threshold is compatible with the known value of $p_{th} \sim 2.9\%$ [10].

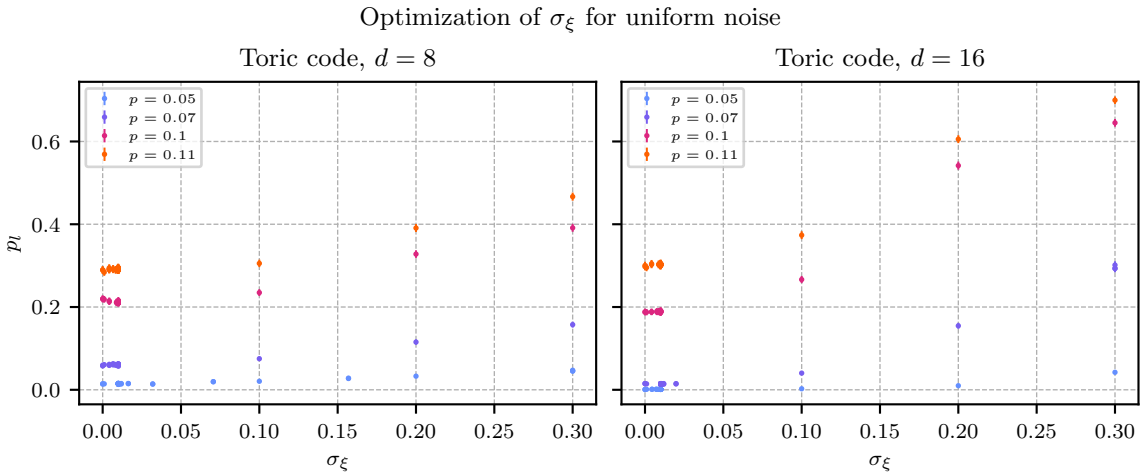


Figure 28: Optimization results for σ_ξ under a uniform noise model and the toric code with distance $d = 8$ (left) and $d = 16$ (right). Across four different error rates p we find the best results for $\sigma_\xi \in [10^{-6}, 10^{-2}]$.

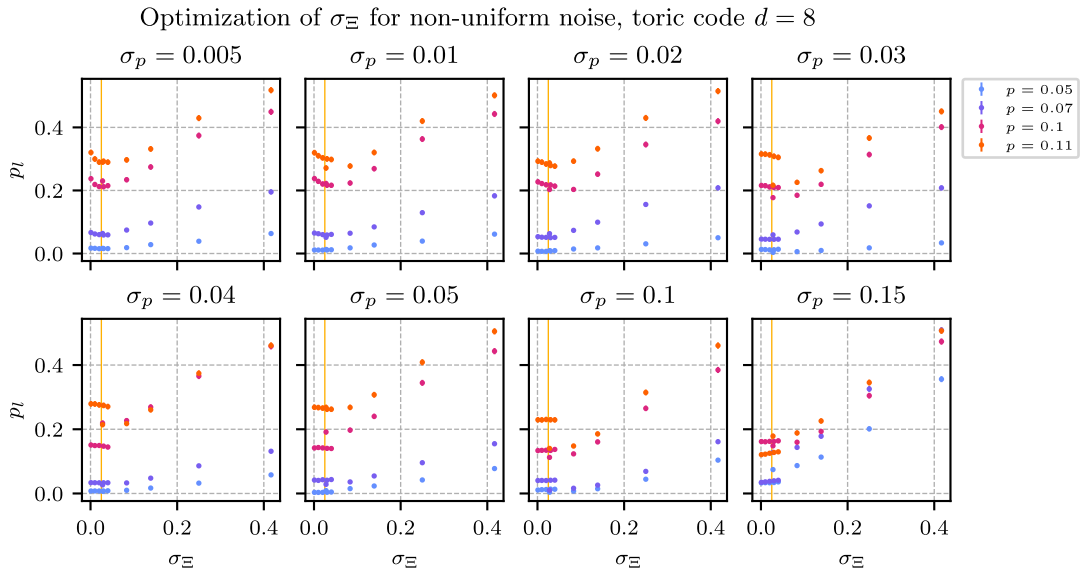


Figure 29: Optimization results for σ_Ξ under a non-uniform noise model with various noise standard deviations σ_p and the toric code with distance $d = 8$. The approximate optimum of $\sigma_\Xi = 0.025$ is marked with the vertical orange line.

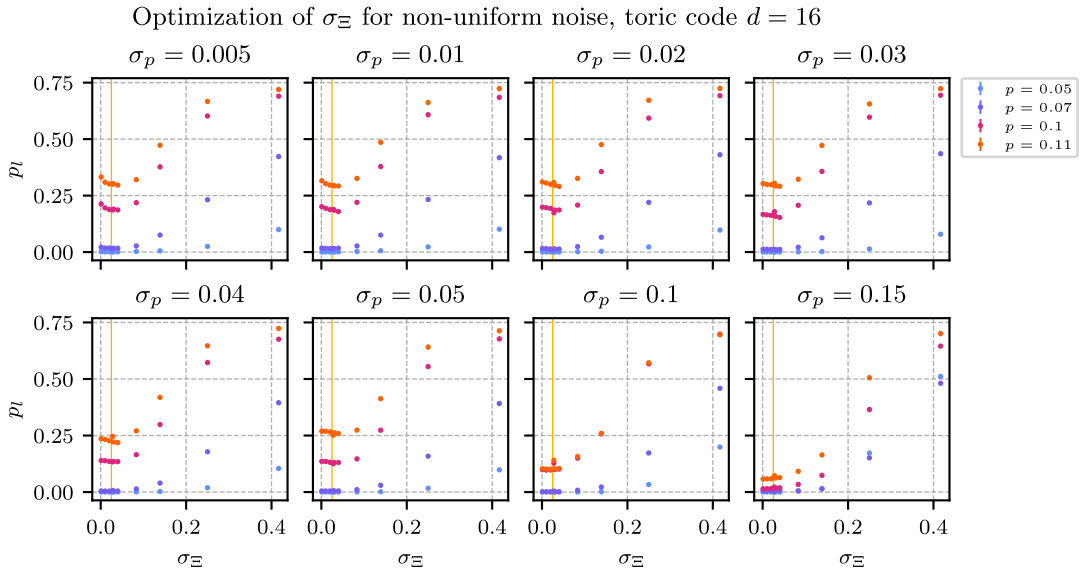


Figure 30: Optimization results for σ_Ξ under a non-uniform noise model with various noise standard deviations σ_p and the toric code with distance $d = 16$. The approximate optimum of $\sigma_\Xi = 0.025$ is marked with the vertical orange line.

References

[1] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. "Topological quantum memory". Journal of Mathematical Physics **43**, 4452–4505 (2002).

[2] Christopher T. Chubb and Steven T. Flammia. "Statistical mechanical models for quantum codes with correlated noise". Annales de l'Institut Henri Poincaré D **8**, 269–321 (2021).

[3] Manuel Rispler, Davide Vodola, Markus Müller, and Seyong Kim. "The random coupled-plaquette gauge model and the surface code under circuit-level noise" (2024). arXiv:2412.14004.

[4] Christophe Piveteau, Christopher T. Chubb, and Joseph M. Renes. "Tensor Network Decoding Beyond 2D". PRX Quantum **5**, 040303 (2024).

[5] Jan Behrends and Benjamin Béri. "Statistical mechanical mapping and maximum-likelihood thresholds for the surface code under generic single-qubit coherent errors". PRX Quantum **6**, 040305 (2025).

[6] Jan Behrends and Benjamin Béri. "The surface code beyond Pauli channels: Logical noise coherence, information-theoretic measures, and errorfield-double phenomenology". PRX Quantum **6**, 040350 (2025).

[7] Davide Vodola, Manuel Rispler, Seyong Kim, and Markus Müller. "Fundamental thresholds of realistic quantum error correction circuits from classical spin models". Quantum **6**, 618 (2022).

[8] Florian Venn, Jan Behrends, and Benjamin Béri. "Coherent-Error Threshold for Surface Codes from Majorana Delocalization". Physical Review Letters **131**, 060603 (2023).

[9] Yinzi Xiao, Basudha Srivastava, and Mats Granath. "Exact results on finite size corrections for surface codes tailored to biased noise". Quantum **8**, 1468 (2024).

[10] Chenyang Wang, Jim Harrington, and John Preskill. "Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory". Annals of Physics **303**, 31–58 (2003).

[11] Creighton K. Thomas and Helmut G. Katzgraber. "Simplest model to study reentrance in physical systems". Physical Review E **84**, 040101 (2011).

[12] H. Bombin, Ruben S. Andrist, Masayuki Ohzeki, Helmut G. Katzgraber, and M. A. Martin-Delgado. "Strong Resilience of Topological Codes to Depolarization". Physical Review X **2**, 021004 (2012).

[13] Ruben S. Andrist, H. Bombin, Helmut G. Katzgraber, and M. A. Martin-Delgado. "Optimal error correction in topological subsystem codes". Physical Review A **85**, 050302 (2012).

[14] Hao Song, Janik Schönmeier-Kromer, Ke Liu, Oscar Viyuela, Lode Pollet, and M. A. Martin-Delgado. "Optimal Thresholds for Fracton Codes and Random Spin Models with Subsystem Symmetry". Physical Review Letters **129**, 230502 (2022).

[15] Helmut G. Katzgraber, H. Bombin, Ruben S. Andrist, and M. A. Martin-Delgado. "Topological color codes on Union Jack lattices: A stable implementation of the whole Clifford group". Physical Review A **81**, 012319 (2010).

[16] Alexey A. Kovalev and Leonid P. Pryadko. "Spin glass reflection of the decoding transition for quantum error correcting codes". Quant. Inf. Comput. **15**, 0825–0852 (2015).

[17] Sergey Bravyi, Martin Suchara, and Alexander Vargo. "Efficient Algorithms for Maximum Likelihood Decoding in the Surface Code". Physical Review A **90**, 032326 (2014).

[18] Arshpreet Singh Maan and Alexandru Paler. "Testing the Accuracy of Surface Code Decoders". In 2023 IEEE International Conference on Rebooting Computing (ICRC). Pages 1–5. (2023).

[19] Lucas H. English, Sam Roberts, Stephen D. Bartlett, Andrew C. Doherty, and Dominic J. Williamson. "Ising on the donut: Regimes of topological quantum error correction from statistical mechanics" (2025). arXiv:2512.10399.

[20] Noah Shutty, Michael Newman, and Benjamin Villalonga. "Efficient near-optimal decoding of the surface code through ensembling". Phys. Rev. Lett. **136**, 070603 (2026).

[21] Hidetoshi Nishimori. "Geometry-Induced Phase Transition in the ±J Ising Model". Journal of the Physical Society of Japan **55**, 3305–3307 (1986).

[22] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth,

Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. "Quantum supremacy using a programmable superconducting processor". Nature **574**, 505–510 (2019).

[23] Oscar Higgott and Craig Gidney. "Sparse Blossom: correcting a million errors per core second with minimum-weight matching". Quantum **9**, 1600 (2025).

[24] Michael A. Nielsen and Isaac L. Chuang. "Quantum computation and quantum information: 10th anniversary edition". Cambridge University Press. (2010).

[25] Creighton K. Thomas and A. Alan Middleton. "Exact algorithm for sampling the two-dimensional Ising spin glass". Physical Review E **80**, 046708 (2009).

[26] Fugao Wang and D. P. Landau. "Efficient, multiple-range random walk algorithm to calculate the density of states". Physical Review Letters **86**, 2050–2053 (2001).

[27] Christopher T. Chubb. "General tensor network decoding of 2D Pauli codes" (2021). arXiv:2101.04125.

[28] Adithya Sriram, Nicholas O'Dea, Yaodong Li, Tibor Rakovszky, and Vedika Khemani. "Rare events and Griffiths phases in topological quantum error correction". Phys. Rev. B **111**, 224208 (2025).

[29] Leonid P. Pryadko. "On maximum-likelihood decoding with circuit-level errors". Quantum **4**, 304 (2020).

[30] Creighton K. Thomas and A. Alan Middleton. "Numerically exact correlations and sampling in the two-dimensional Ising spin glass". Physical Review E **87**, 043303 (2013).

[31] M Picco, A Honecker, and P Pujol. "Strong disorder fixed points in the two-dimensional random-bond Ising model". Journal of Statistical Mechanics: Theory and Experiment **2006**, P09006 (2006).

[32] Luis Colmenarez, Seyong Kim, and Markus Müller. "Fundamental thresholds for computational and erasure errors via the coherent information". PRX Quantum **6**, 040327 (2025).

[33] Luis Colmenarez, Ze-Min Huang, Sebastian Diehl, and Markus Müller. "Accurate optimal quantum error correction thresholds from coherent information". Physical Review Research **6**, L042014 (2024).

[34] Ryotaro Niwa and Jong Yeon Lee. "Coherent information for Calderbank-Shor-Steane codes under decoherence". Physical Review A**111** (2025).

[35] Sun Woo P. Kim. "Optimal recovery for quantum error correction" (2026). arXiv:2603.06520.

[36] Thomas Vogel, Ying Wai Li, Thomas Wüst, and David P. Landau. "Generic, hierarchical framework for massively parallel Wang-Landau sampling". Physical Review Letters **110**, 210603 (2013).

[37] P.W. Kasteleyn. "The statistics of dimers on a lattice". Physica **27**, 1209–1225 (1961).

[38] H. N. V. Temperley and Michael E. Fisher. "Dimer problem in statistical mechanics-an exact result". Philosophical Magazine **6**, 1061–1063 (1961).

[39] Thomas Vogel, Ying Wai Li, and David P Landau. "A practical guide to replica-exchange Wang—Landau simulations". Journal of Physics: Conference Series **1012**, 012003 (2018).

[40] Leon Wichette, Hans Hohenfeld, Elie Mounzer, and Linnea Grans-Samuelsson. "Github repository: qec_pf_fkt". url: `https://github.com/dfki-ric-quantum/qec_pf_fkt`.

[41] Leon Wichette, Hans Hohenfeld, Elie Mounzer, and Linnea Grans-Samuelsson. "Github repository: qec_pf_wang_landau". url: `https://github.com/dfki-ric-quantum/qec_pf_wang_landau`.

[42] Leon Wichette, Hans Hohenfeld, Elie Mounzer, and Linnea Grans-Samuelsson. "Github repository: qec_pf_post_processing". url: https://github.com/dfki-ric-quantum/qec_pf_post_processing.
[43] Bradley Efron. "Better bootstrap confidence intervals". Journal of the American Statistical Association **82**, 171–185 (1987).
[44] Thomas M. Stace and Sean D. Barrett. "Error correction and degeneracy in surface codes suffering loss". Physical Review A **81**, 022317 (2010).
[45] Yue Wu and Lin Zhong. " Fusion Blossom: Fast MWPM Decoders for QEC ". In 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). Pages 928–938. Los Alamitos, CA, USA (2023). IEEE Computer Society.
[46] Leon Wichette, Hans Hohenfeld, Elie Mounzer, and Linnea Grans-Samuelsson. "Github repository: qec_pf_ensembling_mwpm". url: https://github.com/dfki-ric-quantum/qec_pf_ensembling_mwpm.
[47] Guillaume Duclos-Cianci and David Poulin. "Fast decoders for topological quantum codes". Physical review letters **104**, 050504 (2010).
[48] Rasmus Bro, Evrim Acar, and Tamara Kolda. "Resolving the sign ambiguity in the singular value decomposition". Journal of Chemometrics **22**, 135 – 140 (2008).
[49] Helmut G. Katzgraber, H. Bombin, and M. A. Martin-Delgado. "Error threshold for color codes and random three-body Ising models". Phys. Rev. Lett. **103**, 090501 (2009).
[50] Zac Tobias, Nikolas P. Breuckmann, and Benedikt Placke. "Optimal decoding with the worm" (2026). arXiv:2603.05428.
[51] Mao Lin. "Approximate maximum-likelihood decoding with k minimum weight matchings". Phys. Rev. A **112**, 042436 (2025).
[52] Michael E. Beverland, Malcolm Carroll, Andrew W. Cross, and Theodore J. Yoder. "Fail fast: techniques to probe rare events in quantum error correction" (2025). arXiv:2511.15177.
[53] Michael E Beverland, Benjamin J Brown, Michael J Kastoryano, and Quentin Marolleau. "The role of entropy in topological quantum error correction". Journal of Statistical Mechanics: Theory and Experiment **2019**, 073404 (2019).
[54] Oscar Higgott. "Pymatching: A python package for decoding quantum codes with minimum-weight perfect matching". ACM Transactions on Quantum Computing**3** (2022).
[55] Craig Gidney. "Stim: a fast stabilizer circuit simulator". Quantum **5**, 497 (2021).
[56] Jack Edmonds. "Paths, trees, and flowers". Canadian Journal of mathematics **17**, 449–467 (1965).
[57] Vladimir Kolmogorov. "Blossom V: a new implementation of a minimum cost perfect matching algorithm". Mathematical Programming Computation **1**, 43–67 (2009).
[58] Balázs Dezső, Alpár Jüttner, and Péter Kovács. "LEMON–an open source C++ graph template library". Electronic notes in theoretical computer science **264**, 23–45 (2011).
[59] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. "Lipschitzian optimization without the Lipschitz constant". Journal of optimization Theory and Applications **79**, 157–181 (1993).