Practical Advances in Artificial Intelligence and Computing
Frontier Research Publishing • https://www.frpubs.org
Vol. 4, No. 4, 2025

Learning Logical Error Classes Under Correlated Noise in a Seven Qubit Decoding Model

Ismere Valcourt[1]*, Rhovan Ellsby[1], Tavish Merren[1]
[1]University of Waterloo, Waterloo, Canada
*Corresponding author: ismerevalcourt527@outlook.com

Abstract: A learned quantum decoder can fail because of sparse training, a misspecified noise representation or an observation that cannot distinguish logical classes. We separate these mechanisms in an X-only seven-qubit CSS model with a perfect syndrome. All 128 physical error strings are enumerated under six synthetic channels. Four training budgets and 60 repetitions compare minimum weight, independently fitted physical marginals, joint syndrome-class counts and model-assisted counts against privileged exact references. At 1024 labels, pair-correlated class learning reduces failure from 0.044696 to 0.013620, while the independent fit retains the larger risk. A pair01-trained class table transferred to pair24 instead reaches 0.055457. Adding an independent undetectable logical flip raises optimal failure to 0.052531, despite preserving the best decision table. The 12240 risk records use exact test integration; variation comes from sampled training sets. Physical and logical true-channel MAP coincide for these selected channels. The study establishes transparent statistical mechanisms without claiming circuit-level thresholds, quantum hardware measurements or large-code scaling.

Keywords: quantum error correction; logical class; correlated noise; distribution shift; exact enumeration; data-driven decoding

1 The object of a decoding decision

Quantum error correction makes a statistical decision from a deliberately incomplete observation. A syndrome identifies which checks disagree with a code state; it generally does not identify the physical error uniquely. Recovery succeeds when the residual operation belongs to the stabilizer, even if the estimated physical error differs from the realized one. Consequently, physical pattern accuracy and logical recovery accuracy measure different tasks. This distinction is already present in foundational constructions of quantum codes [1] and in the general theory of correctable errors [2]. It becomes especially relevant when a decoder learns from examples of an imperfect device, because the labels and the training distribution determine which statistical decision is actually being learned.

The decoding literature offers a wide range of computational strategies. Topological memories connect syndrome defects to geometric structure [3], and their statistical formulation explains why recovery depends on classes of equivalent errors [4]. Surface code architectures place these ideas in a practical computational setting [5]. These developments do not imply that a decoder trained under one noise model will retain its decisions after a correlation changes. Nor do they imply that a more expressive predictor can recover information absent from its input. A small, exactly enumerable example can separate those questions without introducing the additional uncertainty of a large circuit simulation.

This study examines an X-only decoding problem derived from the seven-qubit CSS code. The code constructions associated with Steane and with Calderbank and Shor supply the algebraic context [6] [7],

while the interference-based account of multiple-particle error correction explains the separation between checks and encoded information [8]. Our experiment retains only one binary error sector, one perfect syndrome observation and a set of explicitly specified stochastic channels. It does not model a complete quantum processor. Its purpose is to make every probability contributing to a logical failure inspectable and to distinguish finite training error from an irreducible observation limit.

The current research context makes this distinction useful. Experimental suppression of bit or phase errors through repeated correction [9], and subsequent scaling of a surface code logical qubit [10], establish reasons to evaluate decoders at the logical level. More recent below-threshold operation [11] and low-overhead quantum memory constructions [12] show that code design, hardware noise and decoding are tightly coupled. The present synthetic numbers are not estimates for either experiment. They form a controlled accounting exercise in which the complete channel is known to the evaluator and hidden from the learned decoders.

We compare a minimum-weight rule, an independently fitted physical channel, empirical joint class counts and a model-assisted count estimator. Two privileged references use the true channel: physical maximum a posteriori decoding and logical class maximum a posteriori decoding. The latter is the exact optimum among all deterministic syndrome-only decisions for the stated loss. Training examples are sampled, but test risk is computed by summing the probabilities of all 128 possible physical error strings. Differences between repetitions therefore arise from training, rather than from finite test shots. This design supports direct statements about estimation and transfer while avoiding claims about processor latency, threshold scaling or circuit-level fault tolerance.

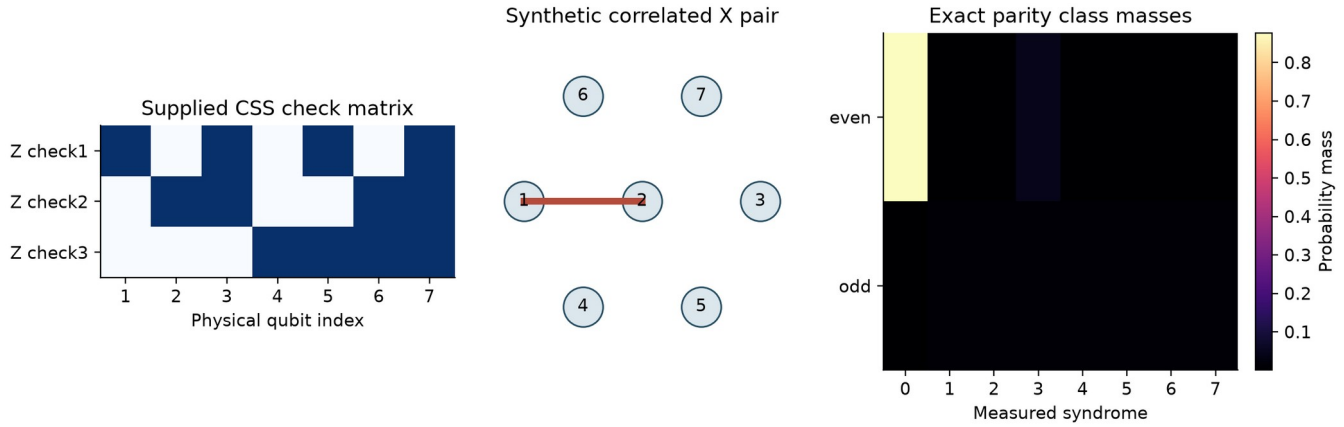


Figure 1. Algebraic observation plate: the supplied binary check matrix, a seven-qubit schematic and exact syndrome-parity masses for pair01. The drawing is code generated and is not a hardware image.

2 Algebra and the observable information

Let the physical X error be a seven-bit vector. All additions and matrix products in the syndrome calculation take place over the binary field. The three rows of the check matrix are the low, middle and high binary digits of the integers one through seven. Thus its columns are all nonzero three-bit vectors. The rank is three, and multiplying the matrix by its transpose gives the zero matrix over the binary field. The resulting kernel has dimension four. Its stabilizer subspace is the three-dimensional row space of the same matrix. These properties can be verified directly, so the subsequent class labels do not rely on an inferred property of an unspecified code.

$$s = H e \bmod 2 \tag{1}$$

Every nonzero element in the stabilizer row space has weight four, and therefore even parity. The even-parity subset of the kernel also has dimension three and equals that row space. It follows that a residual error with zero syndrome succeeds precisely when its parity is even. For any observed syndrome, choosing the parity of a correction therefore selects one of two logical classes. There are sixteen physical errors per syndrome and eight errors per syndrome-parity cell. A correction of either parity exists for every syndrome. The experiment only needs to select a class; it need not construct a particular representative in that class to calculate logical success.

$$\ell(e)=\Sigma_{i=1}^{7} e_i \bmod 2 \tag{2}$$

This use of parity is specific to the selected code and error sector. It is not a universal definition of a logical label. In a general stabilizer code, class assignment requires an appropriate reference recovery and logical operators. Here the simple parity rule is justified by the kernel and row-space calculation. Keeping that justification explicit prevents a binary classification exercise from being described as a complete decoder for arbitrary quantum codes. The check matrix, class masses and one correlated pair are displayed in Figure 1 as algebraic and probability objects; the diagram does not represent a fabricated quantum chip or an acquired hardware image.

Write the joint mass of syndrome and parity as the sum of the physical probabilities in that cell. A syndrome-only decision chooses a parity for each of the eight possible syndromes. Logical failure is the total mass assigned to the opposite parity. The optimal decision chooses the larger of the two cell masses. A syndrome with nearly equal class masses is intrinsically ambiguous under this observation, whereas a syndrome with strongly unequal masses can still be decoded poorly if its preferred class is estimated incorrectly. Those two situations require different remedies.

$$A_{s\ell}=\Sigma_{e:He=s,\ell(e)=\ell} P(e) \tag{3}$$

$$R(d)=\Sigma_{s=0}^{7} A_{s,1-d(s)} \tag{4}$$

$$d^{*}(s)=\arg\max_{\ell\in\{0,1\}} A_{s\ell} \tag{5}$$

The minimum achievable failure probability is obtained by summing the smaller mass in each syndrome row. This expression is exact for the supplied channel and loss. It does not depend on a training procedure or on how many test patterns are sampled. Degeneracy enters through the summation over physical errors before the class comparison. Work on iterative decoding of sparse quantum codes identifies the importance of this feature [13], and complexity results explain why optimal stabilizer decoding becomes difficult in general [14]. Our finite enumeration is deliberately too small to demonstrate an efficient solution to that large-code problem.

$$R^{*}=\Sigma_{s=0}^{7} \min(A_{s0},A_{s1}) \tag{6}$$

A useful diagnostic is to compare logical maximum a posteriori decoding with selection of the most probable individual physical pattern. The two rules need not agree because one compares sums and the other compares maxima. In the six channels executed here, the privileged physical rule happens to attain the same risk as the privileged logical rule. This agreement is a result for these channel choices, not evidence that degeneracy can generally be ignored. We retain the physical reference precisely to expose that limitation. No numerical improvement over physical maximum a posteriori decoding is claimed in this experiment.

Table 1. Declared channel family and observation limits. All probabilities are synthetic and every syndrome is measured perfectly.

Channel	Construction	Additional event
iid	Seven independent probabilities 0.025	None
pair01	Independent background 0.012	Qubits 1, 2; probability 0.045
pair24	Independent background 0.012	Qubits 3, 5; probability 0.045
logical	pair01 background and pair event	Qubits 1, 2, 3; probability 0.04
mixed	Background plus both pair events	Pair probabilities 0.045 and 0.03
biased	0.055, 0.025, and five entries 0.009	None

3 Channels and learned decision rules

The starting distribution is a product of seven Bernoulli error probabilities. For the independent identical channel, every probability is 0.025. For the two pair channels and the logical-mixture channel, the independent background probability is 0.012 on every qubit. A biased independent channel uses probabilities 0.055, 0.025 and then 0.009 on each of the remaining five qubits. These choices are convenient synthetic regimes. They were not fitted to gate measurements, coherence times, fabrication records or any public device data. Comparisons among them concern the relation between a supplied distribution and a learned syndrome decision.

$$P_0(e)=\Pi_{i=1}^{7} p_i^{e_i}(1-p_i)^{1-e_i} \tag{7}$$

Correlated events are introduced by XOR convolution with a fixed binary mask. The pair01 channel applies the mask on the first and second qubits with probability 0.045. The pair24 channel applies the corresponding mask on the third and fifth qubits with the same probability. These names use zero-based array positions, while the diagrams label physical qubits from one to seven. The mixed channel applies the first pair event at probability 0.045 and the second at probability 0.03, independently. If two events affect the same bit, their actions combine by XOR. This construction preserves a normalized nonnegative probability distribution and includes cancellation events automatically.

$$P_b(e)=(1-b)P_0(e)+bP_0(e\oplus m) \tag{8}$$

The logical channel starts from pair01 and adds a mask on the first three qubits with probability 0.04. Those three column labels XOR to zero, so the added event has no syndrome. Its weight is odd, so it reverses the logical parity. It creates a direct information boundary: the syndrome cannot reveal whether that independent event occurred. The purpose is not to mimic a particular cosmic-ray mechanism or correlated fault in an extraction circuit. It is to add a mathematically unobservable component while keeping every other part of the observation unchanged.

The minimum-weight decoder chooses even parity for the zero syndrome and odd parity for every nonzero syndrome. With all columns present, a nonzero syndrome can be explained by a single-qubit error, so this is the parity of a minimum-weight representative. The rule uses no training data and contains no learned channel parameters. It is a useful reference under the independent low-noise channel, but a pair event can

make an even-parity explanation more probable than the odd single-qubit explanation for the same syndrome. That change is exactly the type of distributional information a learned class rule can exploit.

The independent-fit decoder estimates each physical marginal from complete training error strings using a symmetric half-count correction. It then forms a product channel and sums that fitted channel over the same syndrome-parity cells. This decoder is not given only syndromes during training: it receives full synthetic physical errors. Its limitation is the imposed independence of the fitted distribution, rather than a shortage of labels by construction. Correlated pair events can be visible in training and still be discarded when their effect is compressed into seven marginal probabilities. More samples can make that misspecified estimate more stable without improving its decision.

$$\hat{p}_i = (n_i + 0.5)/(n + 1) \tag{9}$$

The joint-count decoder tabulates the observed syndrome and parity pairs. It adds 0.5 to each cell and chooses the larger count within each syndrome. The symmetric addition does not change a strict count comparison; it only makes the probability interpretation explicit, with ties resolved toward even parity by the array convention. A syndrome unobserved in training consequently receives an even decision. The table has sixteen cells, but rare syndrome rows can remain poorly estimated when the total training size is small. The approach is expressive enough for any binary decision on these eight syndromes, yet that expressiveness does not remove finite-sample variability.

The model-assisted decoder combines observed counts with sixteen pseudo-observations distributed according to the independent fitted joint mass. This is an empirical shrinkage construction: the reference distribution is fitted from the same training sample, and the pseudo-count strength is fixed at sixteen for all experiments. It is not an exact Bayesian posterior under an independently specified prior. The assistance is intended to stabilize uncommon rows while allowing sufficiently strong class evidence to overturn the factorized model. A favorable result therefore supports this particular regularization rule under the stated channels, rather than a general theorem about optimal prior strength.

$$d_a(s) = \mathrm{argmax}_\ell [N_{s\ell} + 16\hat{A}_{s\ell}] \tag{10}$$

Large-scale decoding methods provide context for these simple estimators. Minimum-weight matching is implemented in tools such as PyMatching [15], while Sparse Blossom addresses the speed of matching-based correction [16]. Union-find style decoding supplies another algorithmic route [17], and tensor-based methods address maximum-likelihood calculations in the surface code [18]. These are different algorithms on different problem sizes. None is run here, and none of their published timing or threshold results is placed in our synthetic comparison table. The small lookup estimators are used because their decisions can be inspected completely.

Learning-based decoders also extend far beyond a count table. Neural topological decoding [19], reinforcement learning for toric-code correction [20] and deep decoders for near-term fault-tolerant experiments [21] motivate asking what statistical target a learned system should optimize. Graph neural decoding gives a recent data-driven example [22], and high-accuracy learned decoding for quantum processors supplies another [23]. The experiment does not attribute their capabilities to the present rules. It uses a minimal learner to expose a point that remains relevant when the function approximator is much larger: a learned decision is only as transportable as the conditional relation it has estimated.

Table 2. Information available to each decoder. True-channel references are privileged diagnostic controls.

Decoder	Fitted information	Decision
Minimum weight	No examples or channel probabilities	Even at zero syndrome; odd otherwise
Independent fit	Seven marginals from physical error labels	Class MAP under product channel
Joint counts	Sixteen syndrome-parity cell counts	Largest count within syndrome
Model assisted	Counts plus sixteen fitted pseudo-counts	Largest assisted cell mass
Physical MAP	True physical channel	Parity of most probable pattern
Logical oracle	True physical channel	Largest aggregate class mass

4 Execution and accounting

All numerical experiments use the fixed master seed 152025. The program enumerates every seven-bit string, calculates its syndrome and parity, constructs each exact channel and checks probability normalization. For each of six channels, four training sizes are considered: 64, 256, 1024 and 4096 sampled error strings. Sixty independent training samples are drawn at every channel-size combination. The three learned rules share each sampled training set. The minimum-weight rule and the two true-channel references are evaluated alongside them, giving 8640 matched records. The repeated reference rows do not represent independent observations of a varying reference risk.

Transfer experiments use 1024 training patterns and sixty repetitions for each of three source channels: iid, pair01 and pair24. Every fitted decision is then evaluated on five target channels: iid, pair01, pair24, logical and mixed. Only the minimum-weight rule and three learned rules are included in this transfer grid, producing another 3600 records. The full output contains 12240 risk records. The target distributions are supplied to the evaluator for exact integration, not to the learned decoder. No adaptation, target relabeling or test-time update occurs between source fitting and target evaluation.

The matched and transfer sections use different sampled training collections generated sequentially from the same seeded random number generator. A transfer mean should therefore not be described as the very same fitted set that appears in the matched plot, even if a decision agrees in every observed repetition. Within a transfer repetition, however, the source-fitted decoder is held fixed across all five targets. This pairing is useful: a change in exact target risk can be attributed to the evaluation channel while the learned mapping remains unchanged. The supplied run ledger records the experiment type, training channel, target channel, sample size, repetition, method and risk.

For a fixed learned decoder there is no test standard error in this calculation. The channel probabilities are mathematical inputs and the failure is an exact finite sum up to floating-point arithmetic. The reported standard deviation measures variability across sixty training samples. It is not a confidence interval for a hardware logical error rate, and dividing it by a square root would answer a different question about the mean of the training procedure. Figures therefore show training-size means and distributions without presenting artificial shot-error bars. Decimal precision in the CSV exceeds the displayed precision in the manuscript tables to support independent checks of rounding.

Several execution checks guard the class accounting. The syndrome matrix is calculated directly from all error strings; the binary self-orthogonality relation is verified; each channel sums to one; and the privileged logical risk is compared with every evaluated rule. A learned rule cannot beat the exact optimum under the same channel and loss. Its excess risk must be nonnegative apart from negligible arithmetic tolerance. The equality of physical and logical references is also retained rather than suppressed. This check prevents a mathematically possible distinction from being promoted into an empirical separation that the selected experiment did not produce.

Simulation packages such as Stim enable fast stabilizer circuit studies [24], but no circuit simulator is used here. In particular, there are no extraction faults, measurement errors, repeated rounds, leakage events or correlated Z components. The quantity called a failure probability refers to one draw from the specified X-only channel followed by one perfect syndrome observation. It is not a failure per surface-code cycle. Work on decoding across the quantum LDPC landscape [25] and on integrated real-time systems with lattice surgery [26] involves additional code and execution structure that this accounting exercise deliberately leaves outside its scope.

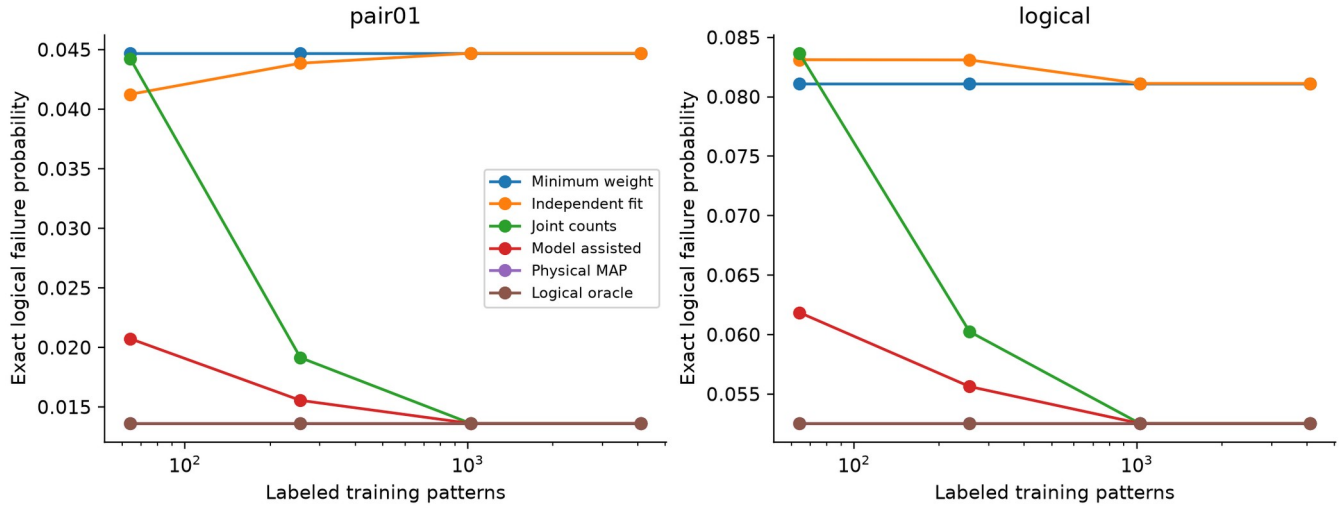


Figure 2. Matched learning curves for pair01 and the logical-mixture channel. Each point averages 60 independently sampled training sets; risks are exact sums over 128 error strings.

5 What the matched experiments establish

The independent identical channel is a useful control because the fitted independence model is correctly specified. Its optimum risk is 0.0116734, and the minimum-weight rule attains that value. At 64 training patterns, the independent-fit mean risk is 0.0126673, compared with 0.0524268 for joint counts and 0.0166433 for model assistance. The unrestricted count table can be much worse when rare syndrome rows are sparsely observed. The improvement from model assistance is consistent with shrinkage toward a useful structural approximation. It does not show that direct class estimation is always the best use of a small training set.

By 1024 training patterns, all six methods attain the same iid risk in all sixty observed repetitions. This exact equality across the sampled fits occurs because risk depends on a finite decision table. Estimated probabilities can differ while every row still chooses the same parity. A flat risk curve therefore does not mean that all probability estimates have converged numerically. It means that their errors remain on the same

side of every relevant class comparison. This distinction is important when interpreting classification learning curves: recovery risk can stabilize long before a calibrated distribution has been learned.

The pair01 channel changes that picture. Minimum weight has risk 0.0446961, whereas the exact optimum is 0.0136203. At a training size of 1024, both joint counts and model assistance attain the optimum in all sixty repetitions. The independently fitted decoder remains at 0.0446961. Its estimate has not become noisier than the count table; instead, its limiting model discards a dependency that changes the best class for a syndrome. Relative to minimum weight, the observed optimal class decision reduces exact failure by about 69.5 percent in this constructed channel. That percentage refers only to the stated synthetic distribution.

The low-data pair01 results reveal a more qualified ordering. At 64 patterns, joint counts have mean risk 0.0442498 and standard deviation 0.0125513, while model assistance has mean 0.0207322 and standard deviation 0.0124415. Independent fitting has mean 0.0412499. Thus direct class counting alone provides little average gain at the smallest budget, even though its representation can express the optimal mapping. At 256 patterns, joint counts improve to 0.0191400 and assistance to 0.0155529. The prior-shaped contribution is most useful before the decisive syndrome has been observed reliably; it becomes irrelevant to the final decision once the empirical margin is strong.

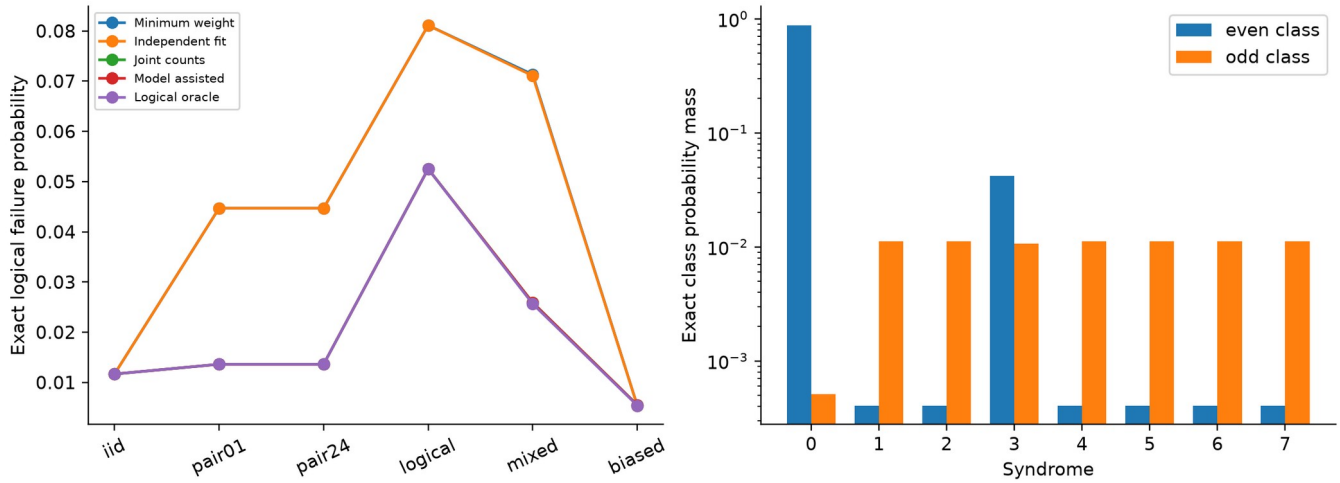


Figure 3. Matched failure across six channels at 1024 training examples (left) and exact even/odd class masses for pair01 (right). Identical decision risks can conceal different fitted probabilities.

The second pair channel has the same optimum and minimum-weight risks as the first. Relabeling which physical pair receives the burst preserves this aggregate difficulty, but it changes which syndrome needs an even decision. This symmetry is valuable because it creates two channels with equal headline risks and different conditional decision tables. A learner can perform perfectly on each channel when trained locally and still transfer badly between them. Equal source and target optimum risk therefore does not establish that the learned mapping is portable. The transfer section tests this point directly rather than inferring it from matching learning curves.

The mixed channel is harder for the learned table at 1024 patterns. Its optimum risk is 0.0256367. Joint counts and model assistance both average 0.0259013, with standard deviation 0.0020494, while independent fitting averages 0.0710678 and minimum weight remains at 0.0713324. The small residual gap for the two class estimators is a training decision error, because the exact class optimum is known and the representation can attain it. It is not an information-theoretic lower bound imposed by the syndrome. Increasing data can

address this type of gap, although the experiment does not establish a sample-complexity law from four budget choices.

The biased independent channel supplies a second control against a uniformly favorable narrative for learning correlations. Its optimum and minimum-weight risks are both 0.00542013. At 1024 patterns, independent fitting attains that value throughout the observed repetitions, while joint counts and assistance average 0.00552928. The latter difference is small, but its direction matters: a flexible class table can occasionally make an unnecessary wrong decision under a channel where a simple structural rule already works. Combining this control with the iid case shows that the benefit of a learner depends on the mismatch it corrects and the estimation variance it introduces.

Table 3. Exact matched logical failure at 1024 labeled training patterns, averaged over 60 training samples. Test risk is integrated, not estimated from test shots.

Channel	MW	Independent	Counts	Assisted	Oracle
iid	0.011673	0.011673	0.011673	0.011673	0.011673
pair01	0.044696	0.044696	0.013620	0.013620	0.013620
pair24	0.044696	0.044696	0.013620	0.013620	0.013620
logical	0.081120	0.081120	0.052531	0.052531	0.052531
mixed	0.071332	0.071068	0.025901	0.025901	0.025637
biased	0.005420	0.005420	0.005529	0.005529	0.005420

6 Transporting a fitted syndrome table

The transfer experiment freezes each source-fitted decision before changing the evaluation distribution. A table learned from pair01 attains mean risk 0.0136203 on that same channel at 1024 training patterns. When applied to pair24, its risk rises to 0.0554574 in every observed transfer repetition for both joint counts and assistance. That is worse than the untrained minimum-weight risk of 0.0446961. The transferred table confidently applies an even-parity exception to the syndrome associated with the old pair, while failing to install the exception needed by the new pair. The increase is a conditional decision shift, even though both channels have the same optimum risk.

A source-fitted exception can also be harmful on an independent target. A pair24 count table transferred to iid has risk 0.0315531, compared with the iid optimum of 0.0116734. This is a substantial increase for a decoder that was optimal in its source channel. The source rule is not malfunctioning as a computer program; its statistical assumption has ceased to match the environment. The example makes clear why random train-test splits within a stationary generator provide limited evidence about a device whose correlation structure may move after recalibration, connectivity changes or an unmodeled operating condition. Those physical causes are possible motivations, not simulated mechanisms in this study.

Training on iid creates a different transfer limitation. The learned decisions tend toward the minimum-weight table and consequently retain its elevated risk on pair channels. A larger iid training sample cannot teach an exception that the iid optimum does not require. This observation distinguishes missing training coverage from finite-sample noise. The former requires either a broader training distribution, explicit

adaptation or side information identifying the active channel. The latter can be reduced by more examples of the same relevant conditional distribution. The present experiment supplies no automated channel-change detector and no guarantee about how much target data would be needed for adaptation.

Figure 4 displays the target risks of joint counts and model assistance using the same color limits. Assistance improves low-budget matched performance, but at the transfer budget it often produces the same discrete table as counts. The resulting heatmaps are therefore similar. This is a consequence of the estimator definitions and the chosen data budget, rather than a second independent confirmation from a fundamentally different model. Reporting both is still informative: a regularizer that stabilizes a source estimate need not protect against an incorrect source assumption. Reducing source variance and increasing robustness to target shift are separate design objectives.

The mixed target gives a partial rather than complete transfer success. A table learned from pair01 reaches risk 0.0415117 on the mixed channel, below minimum weight at 0.0713324 but above the mixed-channel optimum of 0.0256367. It corrects one prevalent pair ambiguity and leaves the other. This intermediate result is easier to interpret with the exact decision table than with an aggregate accuracy alone. A single score does not reveal whether the remaining error is attributable to an unlearned exception, an intrinsically ambiguous class or a generally poor representation. The saved per-syndrome masses and decisions provide that attribution here.

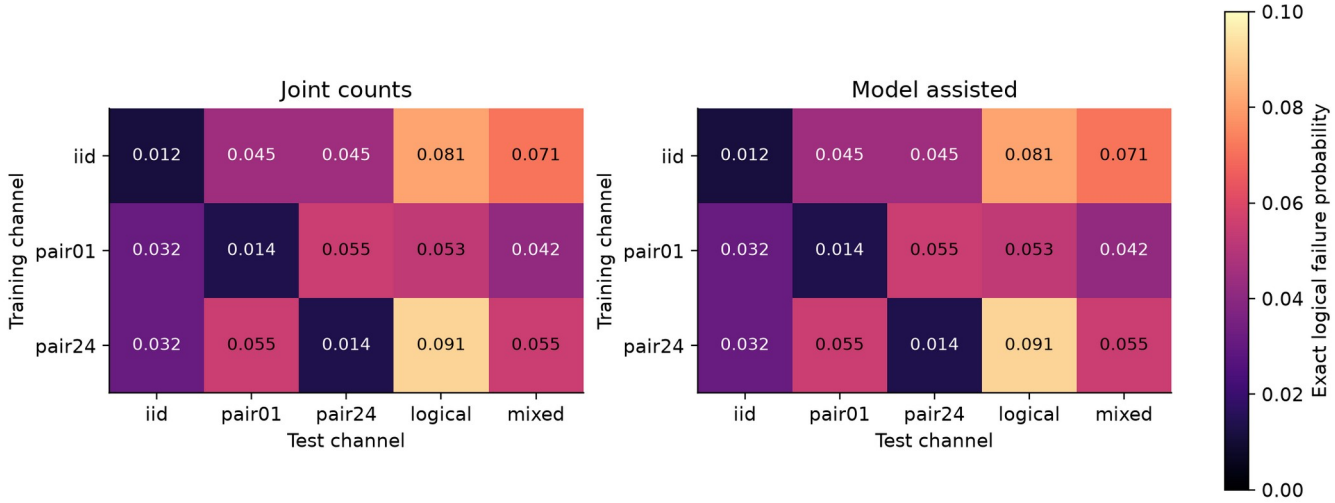


Figure 4. Transfer of source-trained class tables at 1024 patterns. The same decoder is frozen across target channels within each repetition. Color scales are shared.

Transfer risks in this small example could be made robust by assigning a supplied mixture over candidate channels and minimizing its average loss, or by optimizing a worst-case criterion over a declared family. Neither strategy is executed. They would answer different questions and require a defensible choice of mixture weights or uncertainty set. Calling the current model-assisted rule robust merely because it combines two estimates would be inaccurate. Its pseudo-counts regularize a source-fitted distribution; they do not incorporate all possible correlation changes. A practical extension should state its assumed channel family before evaluating out-of-family failures.

7 An undetectable component and its consequence

Adding the logical mask creates a limitation that more training cannot remove. The mask has zero syndrome and odd parity. If the pair channel has class mass A for one parity and B for the other at a given

syndrome, the new channel has masses 0.96A plus 0.04B and 0.96B plus 0.04A. Their difference is multiplied by 0.92. Its sign is preserved, so the optimal decision table remains unchanged. The preferred class is still preferred, but its conditional advantage is smaller. This is why a pair-trained table can transfer optimally to the logical-mixture target even while suffering a higher absolute risk.

$$R_L^* = q + (1 - 2q) R_{pair}^* \tag{11}$$

For q equal to 0.04, the exact optimum rises from 0.0136203 to 0.0525306. Substituting the pair-channel optimum into the affine expression reproduces the enumerated logical risk. The minimum-weight risk rises from 0.0446961 to 0.0811204 under the same transformation. These values distinguish avoidable decision error from unavoidable ambiguity. The minimum-weight rule leaves an avoidable gap above 0.0525306; a correctly learned class table closes that gap. No syndrome-only learner can remove the remaining error while the independent undetectable flip is present and no additional observation is supplied.

The horizontal 0.04 line in Figure 5 is a lower bound contributed by the independent logical flip, not the exact optimum of the full noisy channel. If the background channel were perfectly correctable from its syndrome, the bound would be reached. Our background channel already has risk 0.0136203, so the combined optimum is larger. This distinction prevents an appealing visual reference line from being mistaken for an attained performance floor. The exact affine relation also depends on an independent logical flip with probability below one half. It should not be applied unchanged to arbitrary coherent noise or a fault correlated with the observed syndrome.

The statement is an information result within a declared observation model. Additional information could change it. Repeated measurements, a physical sensor correlated with the event, a larger code or a different encoding might provide a different recovery problem. A second pass over the same unchanged syndrome does not. Nor does replacing the count table with a neural network supply a measurement that was absent. The experiment does not claim that practical correlated errors are always undetectable. It constructs one explicitly undetectable component so that model capacity and observational sufficiency can be separated without ambiguity.

The training distributions in Figure 5 provide the complementary finite-sample view. At 64 pair01 examples, both count-based approaches vary across training sets; their histogram records different learned tables with different exact risks. Those variations can potentially be reduced by additional relevant labels. The logical-mixture bar, in contrast, is the optimum after the channel is already known. More labels cannot lower it under the same input restriction. Putting these two displays together makes a useful methodological distinction: a learning curve estimates how a procedure approaches a decision limit, while a channel calculation identifies what that limit is.

8 Interpretation for data-driven decoding

The strongest matched result is not that a sophisticated learner defeats a strong large-scale decoder. The estimators are deliberately small, and the benchmark is an exact seven-bit distribution. The result is that preserving a syndrome-class dependency can matter more than precisely estimating physical marginals under a misspecified factorization. At 1024 pair01 examples, the independent model is stable and wrong for the recovery objective, whereas a sixteen-cell count representation is sufficient to recover the optimal decisions. This is a statement about statistical representation and loss alignment. It does not depend on claiming that a count table can replace graph-based or recurrent decoding at useful code distances.

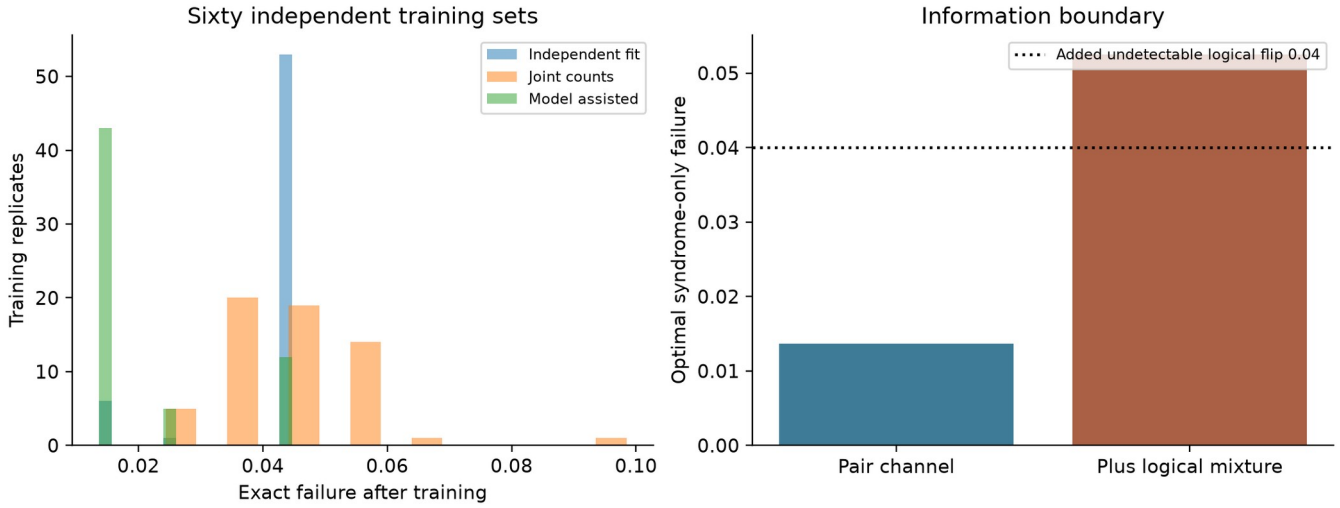


Figure 5. Training variability at 64 pair01 examples (left) and the exact information boundary after an independent undetectable logical flip (right). The 0.04 line is a lower bound, not the combined optimum.

Table 4. Evidence boundaries and what additional data could change. Physical MAP equals logical MAP for the selected channels.

Observed limitation	**Numerical evidence**	**Interpretation**
Sparse training	pair01 counts: 0.044250 at n = 64	More relevant labels can improve the table
Factorization	pair01 independent: 0.044696 at n = 1024	More labels do not restore discarded dependencies
Correlation shift	pair01 → pair24 counts: 0.055457	Source-optimal decisions need not transport
Invisible logical flip	Oracle: 0.013620 → 0.052531	More unchanged syndrome labels cannot remove ambiguity
Degeneracy comparison	Privileged physical and logical MAP coincide	No observed superiority over physical MAP claimed

The labels deserve particular attention. In a synthetic generator, the complete error string and its logical class are available at no experimental cost. A physical device does not generally reveal every actual fault pattern during ordinary operation. The learned rules here receive more detailed training information than a decoder fitted only to unlabelled syndrome logs. Even the class-count rule relies on a correct logical label for each training draw. Any future device study would need to describe how those labels are obtained, whether they are biased by the measurement protocol, and whether the labeled operating distribution matches deployment. The present data availability is a declared advantage of the numerical model.

Exact risk also creates an unusually clean evaluation environment. The evaluator knows every target probability, including rare patterns. A hardware study would need finite observations, uncertainty intervals and a protocol for detecting nonstationarity. The logical failure could drift during data collection, and pooling across that drift could conceal distinct conditional tables. The current calculations are valuable for checking mechanisms, but they cannot establish the number of experimental shots needed to resolve a given

improvement. No timing measurements or learning runtime comparisons are reported, so no real-time deployment advantage follows from the small size of the lookup computation.

The equality of the privileged physical and logical references limits what can be concluded about degeneracy. The algebra proves that class aggregation is the correct recovery objective, and the experiment evaluates that objective. Yet the selected channels do not furnish a numerical case where the largest physical pattern belongs to the wrong aggregate class. A study specifically comparing those two objectives should construct such a channel and report it as a separate controlled test. Here the physical reference is a successful negative control: it prevents the paper from attributing the learned improvement over minimum weight to a distinction that did not separate the privileged references.

The regularization comparison also has a limited scope. The pseudo-count strength of sixteen is fixed rather than tuned on a validation set. The observed benefit at small matched budgets is therefore evidence for one transparent choice. It is not a claim that sixteen is optimal, that the same strength works for larger codes, or that empirical Bayes assistance has uniform risk dominance. The iid and biased controls already show situations in which class estimation introduces unnecessary variance. A principled extension could select regularization using independent source validation, but transfer would still require an explicit assumption connecting that source to the target.

Several omitted physical processes would alter the state and observation spaces. Z errors would add another sector; Y errors could couple sectors; noisy checks would require reasoning across repeated syndromes; and circuit faults could create correlations tied to extraction schedules. Leakage and coherent errors would require still different descriptions. Enumerating the present binary channel does not approximate those mechanisms simply because the code has a quantum origin. A useful progression would introduce them one at a time, preserve exact checks where possible, and identify which conclusions survive. Such an extension should retain the separation among synthetic mechanism tests, circuit simulations and measured processor results.

The numerical artifacts are designed to make that progression auditable. The exact-channel CSV contains each physical bit string, syndrome, parity and probability. The decisions file records a representative learned mapping and the exact even and odd masses for every syndrome. The run ledger separates matched fitting from transfer and preserves all sixty repetitions. The deterministic script reconstructs the figures and tables from those objects. These records do not turn a toy model into experimental evidence, but they allow a reader to inspect the particular ambiguity responsible for each change in risk rather than relying on a visual impression of a learning curve.

9 Conclusion

An exactly enumerable seven-qubit X-error model separates three sources of decoding failure. A factorized estimate can discard a correlation that changes the optimal logical decision. A finite class table can represent that decision but estimate rare syndrome rows poorly. An undetectable logical event can preserve the observation while reversing the required class, creating a risk that persists even when the channel is known. The matched, transfer and logical-mixture experiments distinguish these mechanisms with exact probability sums. Their practical implication is to assess a learned decoder through its recovery target, its channel assumptions and the information in its observations, while keeping claims about hardware scale and circuit fault tolerance tied to evidence that this numerical example does not supply.

References

[1] Peter W. Shor (1995). Scheme for reducing decoherence in quantum computer memory. Physical Review A, 52, R2493-R2496. DOI: 10.1103/physreva.52.r2493.

[2] Emanuel Knill, Raymond Laflamme (1997). Theory of quantum error-correcting codes. Physical Review A, 55, 900-911. DOI: 10.1103/physreva.55.900.

[3] A.Yu. Kitaev (2003). Fault-tolerant quantum computation by anyons. Annals of Physics, 303, 2-30. DOI: 10.1016/s0003-4916(02)00018-0.

[4] Eric Dennis, Alexei Kitaev, Andrew Landahl, John Preskill (2002). Topological quantum memory. Journal of Mathematical Physics, 43, 4452-4505. DOI: 10.1063/1.1499754.

[5] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, Andrew N. Cleland (2012). Surface codes: Towards practical large-scale quantum computation. Physical Review A, 86, 032324. DOI: 10.1103/physreva.86.032324.

[6] A. M. Steane (1996). Error Correcting Codes in Quantum Theory. Physical Review Letters, 77, 793-797. DOI: 10.1103/physrevlett.77.793.

[7] A. R. Calderbank, Peter W. Shor (1996). Good quantum error-correcting codes exist. Physical Review A, 54, 1098-1105. DOI: 10.1103/physreva.54.1098.

[8] Andrew Steane (1996). Multiple-particle interference and quantum error correction. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 452, 2551-2577. DOI: 10.1098/rspa.1996.0136.

[9] Google Quantum AI, Zijun Chen, Kevin J. Satzinger, Juan Atalaya, Alexander N. Korotkov, Andrew Dunsworth, et al. (2021). Exponential suppression of bit or phase errors with cyclic error correction. Nature, 595, 383-387. DOI: 10.1038/s41586-021-03588-y.

[10] Google Quantum AI, Rajeev Acharya, Igor Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, et al. (2023). Suppressing quantum errors by scaling a surface code logical qubit. Nature, 614, 676-681. DOI: 10.1038/s41586-022-05434-1.

[11] Google Quantum AI and Collaborators, Rajeev Acharya, Dmitry A. Abanin, Laleh Aghababaie-Beni, Igor Aleiner, Trond I. Andersen, et al. (2025). Quantum error correction below the surface code threshold. Nature, 638, 920-926. DOI: 10.1038/s41586-024-08449-y.

[12] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, Theodore J. Yoder (2024). High-threshold and low-overhead fault-tolerant quantum memory. Nature, 627, 778-782. DOI: 10.1038/s41586-024-07107-7.

[13] D. Poulin and Y. Chung (2008). On the iterative decoding of sparse quantum codes. Quantum Information and Computation, 8(10), 987–1000. arXiv:0801.1241.

[14] P. Iyer and D. Poulin (2015). Hardness of Decoding Quantum Stabilizer Codes. IEEE Transactions on Information Theory, 61(9), 5209–5223. DOI: 10.1109/TIT.2015.2422294. Preprint arXiv:1310.3235.

[15] Oscar Higgott (2022). PyMatching: A Python Package for Decoding Quantum Codes with Minimum-Weight Perfect Matching. ACM Transactions on Quantum Computing, 3, 1-16. DOI: 10.1145/3505637.

[16] Oscar Higgott, Craig Gidney (2025). Sparse Blossom: correcting a million errors per core second with minimum-weight matching. Quantum, 9, 1600. DOI: 10.22331/q-2025-01-20-1600.

[17] Nicolas Delfosse, Naomi H. Nickerson (2021). Almost-linear time decoding algorithm for topological codes. Quantum, 5, 595. DOI: 10.22331/q-2021-12-02-595.

[18] Sergey Bravyi, Martin Suchara, Alexander Vargo (2014). Efficient algorithms for maximum likelihood decoding in the surface code. Physical Review A, 90, 032326. DOI: 10.1103/physreva.90.032326.

[19] Giacomo Torlai, Roger G. Melko (2017). Neural Decoder for Topological Codes. Physical Review Letters, 119, 030501. DOI: 10.1103/physrevlett.119.030501.

[20] Philip Andreasson, Joel Johansson, Simon Liljestrand, Mats Granath (2019). Quantum error correction for the toric code using deep reinforcement learning. Quantum, 3, 183. DOI: 10.22331/q-2019-09-02-183.

[21] Christopher Chamberland, Pooya Ronagh (2018). Deep neural decoders for near term fault-tolerant experiments. Quantum Science and Technology, 3, 044002. DOI: 10.1088/2058-9565/aad1f7.

[22] Moritz Lange, Pontus Havström, Basudha Srivastava, Isak Bengtsson, Valdemar Bergentall, Karl Hammar, et al. (2025). Data-driven decoding of quantum error correcting codes using graph neural networks. Physical Review Research, 7, 023181. DOI: 10.1103/physrevresearch.7.023181.

[23] Johannes Bausch, Andrew W. Senior, Francisco J. H. Heras, Thomas Edlich, Alex Davies, Michael Newman, et al. (2024). Learning high-accuracy error decoding for quantum processors. Nature, 635, 834-840. DOI: 10.1038/s41586-024-08148-8.

[24] Craig Gidney (2021). Stim: a fast stabilizer circuit simulator. Quantum, 5, 497. DOI: 10.22331/q-2021-07-06-497.

[25] Joschka Roffe, David R. White, Simon Burton, Earl Campbell (2020). Decoding across the quantum low-density parity-check code landscape. Physical Review Research, 2, 043423. DOI: 10.1103/physrevresearch.2.043423.

[26] Namitha Liyanage, Yue Wu, Emmet Houghton, Lin Zhong (2025). Network-Integrated Decoding System for Real-Time Quantum Error Correction with Lattice Surgery. 2025 IEEE International Conference on Quantum Computing and Engineering (QCE), 1148-1159. DOI: 10.1109/qce65121.2025.00129.