CONDITIONAL DIFFUSION DECODERS FOR ERROR-CORRECTING CODES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a diffusion framework for neural decoding of error-correcting codes (ECCs). We interpret decoding as a conditional diffusion process that progressively predicts codeword bits while treating the physical channel observation as fixed side information. At each stage, a neural model estimates bitwise posteriors conditioned on the physical observation and the current noisy representation of the codeword. We focus particularly on masked diffusion, for which the reverse process naturally induces a list decoder inspired by the Maxwell construction of Measson et al. Rather than committing to a single uncertain decision, the decoder maintains multiple candidate trajectories, removes parity-infeasible candidates, and ranks the remaining candidates using accumulated reverse-transition log-probabilities. We evaluate the framework across multiple code families on binary phase-shift keying (BPSK) over additive white Gaussian noise (AWGN). Experiments show competitive decoding performance across Polar, low-density parity-check (LDPC), and Bose–Chaudhuri–Hocquenghem (BCH) codes. The list-augmented decoder consistently improves upon the corresponding single-trajectory diffusion decoder and, in most evaluated settings, improves upon available neural decoding results.

1 INTRODUCTION

Reliable communication requires recovering transmitted information from noisy channel observations with low error and acceptable decoding latency. For a uniform prior over codewords, the optimal decision under frame-error loss is maximum-likelihood (ML) decoding. It is well-known that finding an ML codeword for a general linear code is NP-hard (Berlekamp et al., 1978). To avoid the cost of exact ML computation, practical decoders exploit the structure of specific code families. For low-density parity-check (LDPC) codes, belief propagation (BP) passes messages on a sparse factor graph and the min-sum approximation simplifies the message updates (Kschischang et al., 2001; Fossorier et al., 1999). This complexity reduction comes with tradeoffs: BP loses accuracy on graphs with cycles, and min-sum introduces additional error (Yedidia et al., 2005; Fossorier et al., 1999). One way to address these limitations is to learn elements of the decoder from data rather than using an update rule derived from theory. Indeed, on selected codes, learned decoders improve classical message-passing performance (Nachmani et al., 2016; Nachmani & Wolf, 2021; Park et al., 2025).

Despite the success of BP-based neural decoders for sparse graph codes, polar codes are commonly decoded using successive cancellation (SC), which decides codeword bits sequentially (Arıkan, 2009). An incorrect early decision cannot be revisited, which yields suboptimal performance. Successive-cancellation list (SCL) decoding (Tal & Vardy, 2015) addresses this limitation by branching at bit decisions, retaining multiple candidate paths, and selecting a codeword only after all bits have been processed using a metric that corresponds to their likelihood. This guessing mechanism turns the suboptimal SC decoder into a competitive decoder. For Polar$(64, 32)$, SCL with list size 32 already matches the best neural decoders we evaluate (Table 3). Its main drawback is the sequential dependence inherited from SC, which limits parallelism and can increase latency. The benefits of list decoding for SC decoding for polar codes thus lead to the central question of this paper: Can

Code in `https://anonymous.4open.science/r/conditional-diffusion-ecc-B6FA`
List paths can be processed in parallel, but the bit decisions along each SC path depend on earlier decisions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

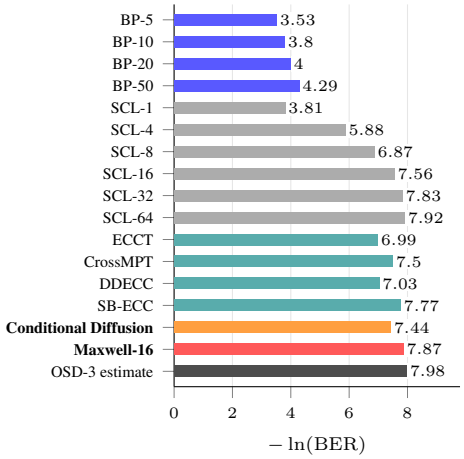


Figure 1: Bit error rate (BER) comparison on Polar$(64, 32)$ at 4 dB; higher is better. Order-3 ordered-statistics decoding (OSD-3) provides an approximate reference. Values and sources are given in Table 3.

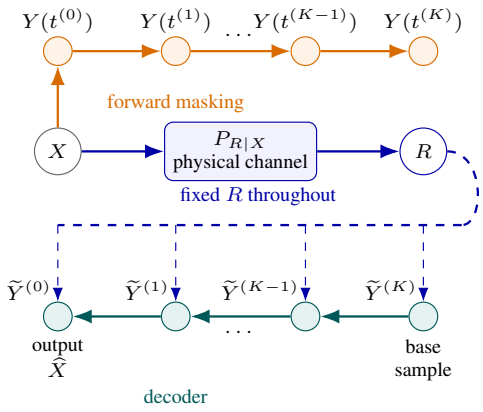


Figure 2: Conditional masking-diffusion decoding. The observation R remains fixed while an auxiliary process masks X; the decoder reverses masking to estimate $\widehat{X}$.

we devise a principled guessing mechanism that improves neural decoders by spending additional computation in parallel across candidates, rather than in additional sequential steps?

Diffusion-based decoders provide a natural starting point, having shown competitive error-correction performance through iterative denoising and score-based inference (Choukroun & Wolf, 2023; Helvits & Nachmani, 2026). We first model the decoder as a conditional diffusion process based on conservation laws developed for coding (Méasson, 2006) and extended to diffusion under model mismatch (Aharoni & Pfister, 2026). In our formulation, the physical-channel observation remains fixed as side information while a separate auxiliary process corrupts the codeword with a broad class of memoryless auxiliary noise processes. The decoder then follows the reverse of this auxiliary process to recover the codeword having the physical observation as a fixed condition, as illustrated in Figure 2. Formally, this process should generate a sample from the posterior distribution of transmitted codeword given the physical observation. Since the posterior concentrates on the correct answer in typical communication scenarios, the challenge is learning a strong enough denoiser to recover it. For the purpose of developing a list decoder, we choose masking noise as the auxiliary process, even though the conservation laws in (Méasson, 2006; Aharoni & Pfister, 2026) accommodate a broad class of memoryless auxiliary noise processes.

Having established the conditional diffusion framework, we then augment the masking-diffusion decoder with guessing, inspired by the Maxwell construction of Méasson et al. (2008). Specifically, at every reverse decision, the decoder branches on multiple bit assignments and scores each path using the log probabilities of the reverse transitions along that path. When the number of paths exceeds the list size, the decoder prunes infeasible candidates and the least likely paths based on the accumulated reverse-transition log probabilities. The final codeword decision is made after all bits have been predicted, and the decoder selects the highest-scoring path among the remaining candidates satisfying the parity constraints. Unlike SCL, whose sequential depth grows with the block length, our decoder's depth is the fixed number of reverse steps K and the list adds only batched work at each step.

Our main experiments show that the conditional diffusion decoder and its Maxwell-inspired list extension achieve competitive performance across several code families on BPSK-AWGN. Figure 1 compares the BER of our methods with classical and neural decoders for Polar$(64, 32)$ at 4 dB.

Among the code families evaluated, the list decoder improves on the corresponding single-trajectory diffusion decoder and, in most evaluated settings, on the available neural decoders.

2 Related Work

Neural decoders. The central line of neural error-correcting code (ECC) decoding consists of iterative, parity-aware methods. Learned BP decoders explicitly unroll Tanner-graph message passing while training message weights or update rules (Nachmani et al., 2016; Nachmani & Wolf, 2021). The Error Correction Code Transformer (ECCT) and Cross-Attention Message-Passing Transformer (CrossMPT) likewise retain the parity-check structure while replacing BP updates with learned Transformer updates (Choukroun & Wolf, 2022; 2024; Park et al., 2025). Neural polar decoders (NPDs) offer an example of a learned decoder augmented with list decoding, and were proven to be effective in various scenarios (Aharoni et al., 2024; Hirsch et al., 2026). However, their sequential nature limits parallelism and increases latency, motivating our search for a parallelizable guessing mechanism that can be applied to a broad range of codes.

Diffusion-based ECC decoders. Denoising Diffusion Error Correction Codes (DDECC) and the Score-Based Error-Correcting Code decoder (SB-ECC) are recent diffusion-based neural decoders that have achieved competitive performance across a variety of codes (Choukroun & Wolf, 2023; Helvits & Nachmani, 2026). Both model decoding as the reversal of channel noise and assume a Gaussian channel. Our conditional formulation separates physical-channel noise from the auxiliary diffusion process, allowing us to choose masking noise and construct a list decoder.

EXIT, GEXIT, and conservation laws. Extrinsic information transfer (EXIT) charts were developed to analyze the behavior of iterative decoders through their extrinsic-information transfer curves, whose area properties connect decoding behavior to code rate (ten Brink, 2001; Ashikhmin et al., 2002). Generalized EXIT (GEXIT) functions and coding conservation laws extend this analysis to more general channel families (Méasson, 2006; Méasson et al., 2009). Recently, these conservation laws were extended to diffusion models under posterior mismatch: the data–model cross-entropy, or expected negative log-likelihood, is characterized by an integral along the noise path whose local terms depend on marginal bitwise posteriors (Aharoni & Pfister, 2026).

Masked diffusion and search. Discrete diffusion with an absorbing mask state was studied by Austin et al. (2021), and later work showed that its training objective reduces to a weighted masked cross-entropy (Sahoo et al., 2024; Shi et al., 2024). We use the same masking process, but condition it on a fixed physical observation and interpret its loss through coding conservation laws. Search over diffusion trajectories has been explored for generation (Jain et al., 2025; Cao et al., 2026); our decoder instead exploits hard parity constraints to prune candidates. The list construction is also related to reliability-based decoders such as Chase decoding (Chase, 1972) and ordered-statistics decoding (Fossorier & Lin, 1995), which perturb hard decisions according to fixed channel reliabilities. Maxwell instead branches on posterior-ranked positions, recomputing rankings after each decision and parity propagation.

3 Preliminaries and Background

3.1 Error-correcting codes

We consider the standard coded transmission setup using a binary linear block code $\mathcal{C}$. The code is specified by a generator matrix $G \in \{0,1\}^{n\times k}$ and a parity-check matrix $H \in \{0,1\}^{r\times n}$ such that $HG = \mathbf{0}_{r\times k}$ over $\mathbb{F}_2$ (Richardson & Urbanke, 2008). The symbol $\mathbf{0}_{r\times k}$ denotes an all-zero matrix of the indicated dimensions. For an (n,k) code, $\mathrm{rank}(H) = n-k$ and the number r of listed parity checks satisfies $r \geq n-k$, with equality when the checks are linearly independent. Given a message $m \in \{0,1\}^k$, the encoder produces the length-n codeword

$$x = Gm \in \mathcal{C} \subseteq \{0,1\}^n, \qquad Hx = 0, \tag{1}$$

and the code rate is k/n. For transmission, we map the codeword to its binary phase-shift-keying (BPSK) representation $x_{\mathrm{s}} = \mathbf{1} - 2x \in \{\pm 1\}^n$. We use uppercase X for the random transmitted codeword and lowercase x for one realization.

3.2 PHYSICAL OBSERVATION AND AUXILIARY NOISE

We distinguish two channels because they play different roles in the decoder. The physical channel produces the observation R, which remains fixed and conditions every decoding decision. The auxiliary channel instead governs the evolving state $Y(t)$ and serves as an internal mechanism for taking decisions about the transmitted codeword.

Physical observation. The physical channel is described by the conditional distribution

$$P_{R|X}(r \mid x), \tag{2}$$

from a transmitted codeword $x \in \mathcal{C}$ to a received observation R. This distribution is set by the communication medium. The realization $R = r$ is the physical information available to the decoder and remains fixed throughout decoding.

Auxiliary noise. Separately, the decoding process is designed by choosing an auxiliary scalar noising family $\mathcal{W} = \{W_t\}_{t\in\mathcal{T}}$, where $\mathcal{T} \subseteq \mathbb{R}_{\geq 0}$ is the scalar noise-parameter domain. This family facilitates the decision process of the decoder. We use only the homogeneous auxiliary channel: at noise level t, every coordinate is passed through the same scalar channel. Thus $Y(t) = (Y_1(t), \ldots, Y_n(t))$ has conditional distribution

$$W_t^{\otimes n}(y \mid x) = \prod_{i=1}^{n} W_t(y_i \mid x_i), \qquad t \in \mathcal{T}. \tag{3}$$

The family is ordered so that increasing t cannot add information. Assume that $\mathcal{T}$ has a clean endpoint t^{cln} and an uninformative endpoint t^{ind}, for which $Y(t^{\mathrm{cln}}) = X$ almost surely and $Y(t^{\mathrm{ind}}) \perp X$. The scalar parameter t therefore indexes the decoder trajectory from clean to uninformative. Our formulation requires $\mathcal{W}$ to be a smooth, degraded memoryless channel family under the standard GEXIT regularity conditions (Méasson et al., 2009; Aharoni & Pfister, 2026); Appendix A.4.1 lists the precise assumptions for clarity. Conditional on X, this auxiliary randomness is independent of the physical channel, so

$$R \longleftrightarrow X \longleftrightarrow Y(t) \tag{4}$$

is a Markov chain.

4 DECODING ERROR-CORRECTING CODES AS CONDITIONAL DIFFUSION

This section commences with a brief background of the conservation laws for masked diffusion, which connect the local bitwise posterior quantities to the global conditional entropy of the codeword given the physical observation. Then, it describes how these posterior marginals define the diffusion decoder and its Maxwell-inspired list extension.

4.1 CONSERVATION LAW FOR MASKED DIFFUSION

We extend the masking conservation law of Aharoni & Pfister (2026) to the setting with additional physical-channel side information R, which remains fixed along the auxiliary-noise path. At mask probability $t \in [0, 1]$, each bit X_i is either revealed or replaced by the erasure symbol $*$, independently of the other bits. The vector $Y_{-i}(t)$ contains the auxiliary observations of all bits except i.

The masking result of Aharoni & Pfister (2026) expresses joint entropy as an integral of bitwise prediction losses along the mask-probability path. Conditioning every term on the fixed observation R yields the following conditional version. The conservation law follows two steps: first, the derivative of $H(X \mid R, Y(t))$ with respect to t decomposes into a sum of terms that depend only on the local bitwise posterior $P_{X_i|R,Y_{-i}(t)}$. In particular, for masking noise we have that

$$\frac{d}{dt} H(X \mid R, Y(t)) = \sum_{i=1}^{n} H(X_i \mid R, Y_{-i}(t)). \tag{5}$$

Thus, integrating the derivative along the path from $t = 0$ to $t = 1$ connects the local bitwise posteriors to the global conditional entropy $H(X \mid R)$ by

$$H(X \mid R, Y(1)) - H(X \mid R, Y(0)) = \int_0^1 \sum_{i=1}^{n} H(X_i \mid R, Y_{-i}(t))\, dt, \tag{6}$$

which equals $H(X \mid R)$.

For the mismatched case, let $Q_{X|R}$ be a model of $P_{X|R}$. Pass both the true and model distributions through the same masking channel W_t. Write $Q_{X_i|R,Y_{-i}(t)}$ for the resulting model posterior of bit i given R and the other masked coordinates. The masked conservation law then gives

$$\mathbb{E}_R\big[\mathrm{CE}\big(P_{X|R}, Q_{X|R}\big)\big] = \int_0^1 \sum_{i=1}^{n} \mathbb{E}_{R,Y_{-i}(t)}\big[\mathrm{CE}\big(P_{X_i|R,Y_{-i}(t)}, Q_{X_i|R,Y_{-i}(t)}\big)\big]\ dt. \tag{7}$$

where $\mathrm{CE}(P_{X_i|R,Y_{-i}(t)}, Q_{X_i|R,Y_{-i}(t)}) = -\mathbb{E}_{P_{X_i|R,Y_{-i}(t)}}\big[\log Q_{X_i|R,Y_{-i}(t)}\big]$. The extension to additional side information R is detailed in Appendix A.

4.2 Learning posterior marginals

Let $Q^{\theta}_{X_i|R,Y(t)}$ be posterior marginals modeled using a neural network with parameters θ. To ensure consistency with Bayes' rule,

$$Q^{\theta}_{X_i|R,Y(t)}(x_i \mid r, y) = \frac{Q^{\theta}_{X_i|R,Y_{-i}(t)}(x_i \mid r, y_{-i}) W_t(y_i|x_i)}{\sum_{x'} Q^{\theta}_{X_i|R,Y_{-i}(t)}(x' \mid r, y_{-i}) W_t(y_i|x')}, \tag{8}$$

we enforce

$$Q^{\theta}_{X_i|R,Y(t)}(x_i \mid r, y) = \begin{cases} Q^{\theta}_{X_i|R,Y_{-i}(t)}(x_i \mid r, y_{-i}) & y_i = *, \\ \mathbf{1}\{x_i = y_i\}, & y_i \in \{0, 1\} \end{cases}. \tag{9}$$

Thus, the model predicts posteriors at masked positions and is deterministic at revealed positions.

Each training step begins by sampling a codeword and one physical channel observation, $(X, R) \sim P_X P_{R|X}$. After observing $R = r$, we hold this realization fixed. We then sample a mask probability $t \sim \mathrm{Unif}(0, 1)$ and generate only the auxiliary state

$$Y(t) \sim W_t^{\otimes n}(\cdot \mid X). \tag{10}$$

The parameters θ are trained to minimize the loss

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{t\sim\mathrm{Unif}(0,1), X\sim P_X \\ R\sim P_{R|X}, Y(t)\sim W_t^{\otimes n}}} \left[-\sum_{i\in\mathcal{M}} \log Q^{\theta}_{X_i|R,Y_{-i}(t)}(X_i \mid R, Y_{-i}(t)) \right], \tag{11}$$

where $\mathcal{M} = \{i : Y_i(t) = *\}$ with the requirement that $|\mathcal{M}| \geq 1$.

4.3 Inference

During inference, the learned bitwise posterior field defines the reverse transitions. We describe conditional masked diffusion in Section 4.3.1 and its list-decoding extension in Section 4.3.2.

Algorithm 1 Confidence-scheduled decoding

Require: r; counts $(b_K, \ldots, b_1)$ with $\sum_k b_k = n$
1: $\widetilde{Y}^{(K)} \leftarrow *^n$
2: **for** $k = K, K-1, \ldots, 1$ **do**
3: $\quad t^{(k)} \leftarrow$ current masked fraction
4: $\quad$ Compute $\lambda_i^{(k)}$ with (12)
5: $\quad \mathcal{G}_k \leftarrow b_k$ most-confident masked indices
6: $\quad \widetilde{Y}_i^{(k-1)} \leftarrow \mathbf{1}\{\lambda_i^{(k)} \geq 0\}$ for $i \in \mathcal{G}_k$
7: **end for**
8: **return** $\widehat{X} = \widetilde{Y}^{(0)}$

4.3.1 Conditional-diffusion decoding.

Before decoding, we specify the number of bits $b_K, \ldots, b_1$ to reveal at each of K steps, with $\sum_{k=1}^{K} b_k = n$. The decoder starts from the all-erasure word $\widetilde{y}^{(K)} = *^n$ and keeps the physical observation $R = r$ fixed. At step k, the current word is $\widetilde{y}^{(k)}$ and the noise level $t^{(k)} = (b_k + \cdots + b_1)/n$

During training, if none of the symbols are masked, then one random symbol in the codeword is masked.

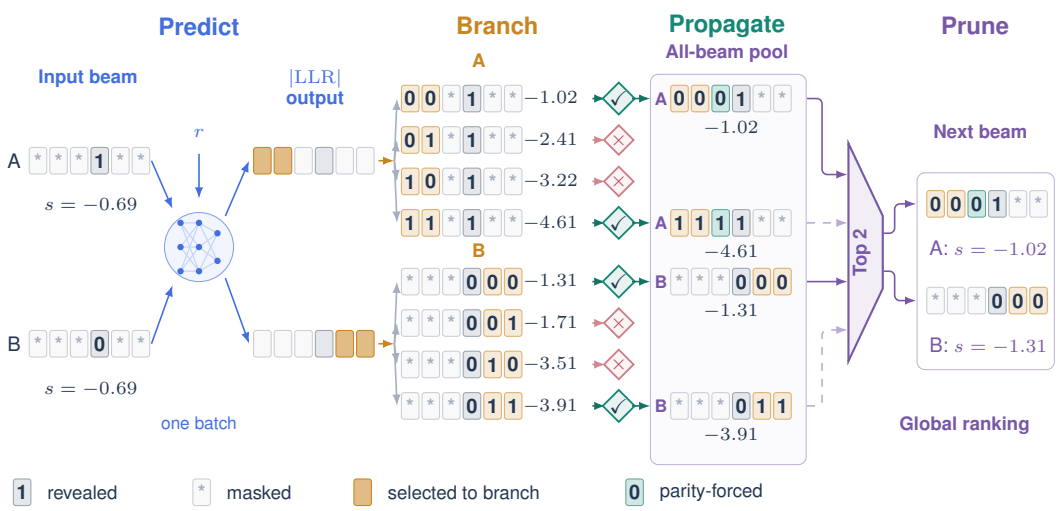


Figure 3: One decoding round with beam-width two and two branching bits per candidate. A batched prediction selects the least-confident masked positions, those with smallest absolute LLRs (amber): $(1, 2)$ for A and $(5, 6)$ for B. Each candidate produces four assignments, adding their log probabilities to its accumulated score. Parity propagation rejects inconsistent children and fills forced bits without changing scores. Global top-2 retains the two best survivors across both candidates.

is the fraction of bits still masked. One forward pass evaluates the posterior marginals in (9) for this state. For each still-masked bit i, we compute the predicted log-likelihood ratio (LLR)

$$\lambda_i^{(k)} = \log \frac{Q^{\theta}_{X_i|R,Y(t^{(k)})}(1 \mid r, \widetilde{y}^{(k)})}{Q^{\theta}_{X_i|R,Y(t^{(k)})}(0 \mid r, \widetilde{y}^{(k)})}, \tag{12}$$

and $|\lambda_i^{(k)}|$ measures confidence. We select the b_k still-masked positions with the largest $|\lambda_i^{(k)}|$ to form $\mathcal{G}_k$, then reveal them using the LLR hard decision induced by the marginal backward kernel

$$Q^{\theta}_{\widetilde{Y}_i^{(k-1)}|R,\widetilde{Y}^{(k)}}(\widetilde{y}_i^{(k-1)} \mid r, \widetilde{y}^{(k)}) = \begin{cases} Q^{\theta}_{X_i|R,Y(t^{(k)})}(\widetilde{y}_i^{(k-1)} \mid r, \widetilde{y}^{(k)}), & i \in \mathcal{G}_k, \\ \mathbf{1}\{\widetilde{y}_i^{(k-1)} = \widetilde{y}_i^{(k)}\}, & i \notin \mathcal{G}_k. \end{cases} \tag{13}$$

For $i \in \mathcal{G}_k$, the new value is $\widetilde{y}_i^{(k-1)} = \mathbf{1}\{\lambda_i^{(k)} \geq 0\}$; every other coordinate keeps its current value. After K steps all bits are revealed and the noise level has decreased from $t^{(K)} = 1$ to $t^{(0)} = 0$. Algorithm 1 summarizes the procedure. The next section describes the Maxwell beam decoder, which augments Algorithm 1 with a list of candidate paths.

4.3.2 Maxwell beam decoder

The Maxwell decoder was introduced for erasure decoding to bridge belief propagation (BP) and maximum a posteriori (MAP) decoding (Méasson et al., 2008). When BP cannot resolve an erased bit, it branches on both possible values and uses parity constraints to eliminate inconsistent hypotheses. Our formulation likewise recovers erased bits, with the physical observation R providing additional side information. We adapt this branching mechanism by using the learned neural posterior marginals in place of BP predictions and retaining a finite beam of candidate decoding paths.

Initialization. The decoder takes the observation r, parity-check matrix H, beam width L, a predefined schedule of K steps with branch counts b_k, and a decision rule $\mathcal{D}$. It starts with $\mathcal{B}_K = \{(*^n, 0)\}$: one all-erasure word with path score zero. At step k, the beam contains $L_k = |\mathcal{B}_k| \leq L$ candidates $(\widetilde{y}_\ell^{(k)}, s_\ell^{(k)})$, indexed by $\ell = 1, \ldots, L_k$.

Main algorithm. Each round uses one batched forward pass to evaluate the posterior marginals for all current candidates, with candidate-specific masked fraction $t_\ell^{(k)} = n^{-1} \sum_i \mathbf{1}\{\widetilde{y}_{\ell,i}^{(k)} = *\}$. The

predicted LLRs $\lambda_{\ell,i}^{(k)}$ from (12) are evaluated at $(r, \widetilde{y}_\ell^{(k)}, t_\ell^{(k)})$. For $k > 1$, candidate ℓ selects up to b_k still-masked positions with the smallest absolute LLRs, the least-confident unresolved bits, which we denote by $\mathcal{G}_{\ell,k} \subseteq \{1, \ldots, n\}$. It creates $2^{|\mathcal{G}_{\ell,k}|}$ child words $\bar{y}_{\ell,j}^{(k-1)}$, one per binary assignment, indexed by $j = 1, \ldots, 2^{|\mathcal{G}_{\ell,k}|}$. Child j copies its parent bit values outside $\mathcal{G}_{\ell,k}$ and assigns the jth binary combination inside it. Its accumulated path score uses the marginal backward kernel in (13):

$$s_{\ell,j}^{(k-1)} = s_\ell^{(k)} + \sum_{i \in \mathcal{G}_{\ell,k}} \log Q^\theta_{\widetilde{Y}_i^{(k-1)} | R, \widetilde{Y}^{(k)}} \left(\bar{y}_{\ell,j,i}^{(k-1)} \mid r, \widetilde{y}_\ell^{(k)} \right). \tag{14}$$

The decoder then applies parity propagation to each child, filling bits uniquely determined by the parity checks in H and discarding any child for which a parity contradiction is detected. These forced assignments contribute zero to the path score because they have conditional probability one under the code constraints. The decoder pools the surviving child–score pairs from all parents into $\mathcal{S}_k$ and retains up to L candidates with the highest path scores to form the next beam $\mathcal{B}_{k-1}$. Figure 3 illustrates a branching round, and Algorithm 2 in Appendix H summarizes the procedure.

Decision rules. At the final step $k = 1$, $\mathcal{G}_{\ell,1}$ contains all remaining masked indices. For each candidate ℓ, the decoder forms a single child using the LLR decision rule

$$\bar{y}_{\ell,1,i}^{(0)} = \begin{cases} \mathbf{1}\{\lambda_{\ell,i}^{(1)} \geq 0\}, & i \in \mathcal{G}_{\ell,1}, \\ \widetilde{y}_{\ell,i}^{(1)}, & i \notin \mathcal{G}_{\ell,1}. \end{cases} \tag{15}$$

The completed child's score $s_{\ell,1}^{(0)}$ is computed by (14) with $k = 1$ and $j = 1$: to the parent score $s_\ell^{(1)}$, we add the log marginal probability of each bit assigned by (15) at indices $i \in \mathcal{G}_{\ell,1}$, all conditioned on the same parent state $\widetilde{y}_\ell^{(1)}$ and observation r. Only completed children satisfying $H\bar{y}_{\ell,1}^{(0)} = 0$ are retained. Let $\mathcal{L}$ be the final list of complete, parity-valid codewords, with $s(x)$ denoting the largest retained path score for codeword x.

Let $\mathcal{D} \in \{\mathcal{D}_{\text{model}}, \mathcal{D}_{\text{dist}}\}$ be the final decision rule, with output $\widehat{x} = \mathcal{D}(\mathcal{L}, r)$. The model rule selects the largest path score:

$$\mathcal{D}_{\text{model}}(\mathcal{L}, r) = \arg\max_{x \in \mathcal{L}} s(x). \tag{16}$$

The minimum-distance rule selects the most likely codeword under BPSK-AWGN, equivalently the one with minimum squared Euclidean distance:

$$\mathcal{D}_{\text{dist}}(\mathcal{L}, r) = \arg\min_{x \in \mathcal{L}} \|r - (\mathbf{1} - 2x)\|_2^2. \tag{17}$$

Path metric justification. If every non-forced bit is revealed individually using exact posterior marginals, the chain rule of probability gives that the score equals $\log P_{X|R}(x \mid r)$. Parity-forced bits have conditional probability one and contribute zero log probability. When several bits are assigned in parallel, summing their log marginal probabilities ignores conditional dependencies. In expectation, the resulting score lower-bounds the log posterior: its negative exceeds $H(X \mid R = r)$ by the total correlation of the jointly revealed bits, which is nonnegative. See Appendix B for a proof and extension for learned marginals.

5 Experiments

We evaluate conditional diffusion and Maxwell list decoding on standard linear block codes over BPSK-AWGN. We then examine adaptation to a binary symmetric channel by updating the physical-observation input modules, and study the effect of the beam width.

5.1 Experimental setup

Codes, channels, and model. We consider LDPC (Gallager, 1962), Polar (Arıkan, 2009), and Bose–Chaudhuri–Hocquenghem (BCH) codes (Hocquenghem, 1959; Bose & Ray-Chaudhuri,

A parity check with exactly one masked bit fixes that bit to satisfy the even-parity constraint. Propagation repeats until no such checks remain or a contradiction is detected.

Table 1: **BER comparison over BPSK-AWGN.** Values are $-\ln(\mathrm{BER})$ (higher is better). Bold and underline mark the best and second-best. Maxwell selects minimum Euclidean distance from list.

Architecture		Non-neural decoders			Neural decoder			Diffusion-based neural decoders						Our decoders					
Family	Code	BP			CrossMPT			DDECC			SB-ECC			Diffusion			Maxwell		
		4	5	6	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
Polar	$(64, 32)$	4.26	5.38	6.50	7.50	9.97	13.31	7.03	9.69	12.97	7.77	10.30	**13.78**	7.44	10.06	13.21	**7.87**	**10.52**	13.62
Polar	$(64, 48)$	4.74	5.94	7.42	6.51	**8.70**	11.31	6.07	8.40	10.90	6.63	8.64	11.27	6.52	8.60	11.10	**6.65**	8.63	**11.45**
Polar	$(128, 64)$	4.10	5.11	6.15	7.52	11.21	14.76	8.16	12.04	16.27	**9.03**	**13.13**	**16.94**	7.74	11.19	13.86	8.91	13.04	16.09
Polar	$(128, 86)$	4.49	5.65	6.97	7.51	10.83	15.24	7.92	11.34	15.61	8.03	11.48	16.10	7.63	10.95	13.81	**8.21**	**11.88**	**16.48**
Polar	$(128, 96)$	4.61	5.79	7.08	7.15	10.15	13.13	7.24	10.32	13.26	7.64	10.32	**13.37**	7.38	10.17	12.92	**7.94**	**10.63**	13.24
LDPC	$(49, 24)$	6.23	8.19	11.72	6.68	9.52	13.19	6.26	9.07	12.71	7.74	10.88	14.63	7.12	10.11	14.20	**8.46**	**11.67**	**15.93**
LDPC	$(121, 60)$	5.64	8.88	13.33	5.74	9.26	14.78	6.22	10.18	15.89	6.33	10.38	16.38	5.81	9.87	15.81	**7.09**	**11.87**	**17.64**
LDPC	$(121, 70)$	6.91	10.66	15.62	7.06	11.39	17.52	7.64	12.30	17.98	7.75	12.69	19.24	7.38	12.18	18.88	**8.86**	**14.34**	**20.16**
LDPC	$(121, 80)$	7.84	11.86	17.33	7.99	12.75	18.15	8.62	13.55	18.26	8.84	14.02	20.42	8.41	13.49	20.24	**10.06**	**15.80**	**21.75**
BCH	$(63, 36)$	4.03	5.42	7.26	5.03	6.91	9.37	5.30	7.32	10.25	5.74	8.12	11.20	5.01	6.98	9.60	**5.83**	**8.28**	**11.67**
BCH	$(63, 45)$	4.36	5.55	7.26	5.90	8.20	11.62	5.67	8.07	11.41	6.58	9.48	13.17	5.81	8.24	11.59	**7.43**	**10.52**	**14.38**
BCH	$(63, 51)$	4.50	5.82	7.42	5.78	8.08	11.41	5.37	7.48	10.51	6.19	**8.82**	**12.45**	5.64	7.99	11.40	**6.20**	8.77	12.36
MacKay	$(96, 48)$	6.84	9.40	12.57	7.97	11.77	15.52	8.58	12.48	16.04	8.97	12.82	16.25	8.71	12.50	15.82	**10.18**	**14.02**	**17.62**
CCSDS	$(128, 64)$	8.00	12.11	17.17	7.68	11.88	17.50	8.40	13.18	17.47	8.60	13.43	19.20	8.21	12.63	18.96	**9.28**	**14.38**	**20.21**

1960), including LDPC codes from MacKay's collection (MacKay, 1999) and the Consultative Committee for Space Data Systems (CCSDS) short-block family (CCSDS, 2015). For the physical channel, we consider BPSK over AWGN and a binary symmetric channel (BSC) with crossover probability $p = 0.95$ for the adaptation experiment.

Our BPSK-AWGN evaluations cover linear block codes at $E_b/N_0 \in \{4, 5, 6\}$ dB. We use a CrossMPT (Park et al., 2025) architecture with 6 attention layers, width 128, and 8 attention heads to model Q^θ; additional implementation details are in Appendix C.

Training and evaluation. Training examples are generated on the fly from random codewords, with E_b/N_0 drawn uniformly from the discrete grid $\{2, 3, 4, 5, 6, 7\}$ dB. We train the posterior marginals with the objective in (11) for 3 million iterations. We use AdamW with zero weight decay and a batch size of 256 for all codes. The learning rate is cosine-annealed from a peak of 5×10^{-4} to 5×10^{-6}. At evaluation, conditional diffusion uses 16 reverse steps with equal-sized groups, up to integer rounding, revealing the currently most confident masked coordinates first. We evaluate using an exponential moving average (EMA) of the model parameters, stopping each evaluation after 500 frame errors or 10^8 frames.

Maxwell decoder. We use a beam width of $L = 16$ and a predefined schedule of initial single-bit branching rounds ($b_k = 1$), followed by one final step that completes all remaining masked positions in parallel for each candidate. Unless otherwise stated, we select the final codeword using the minimum Euclidean-distance selection (17) after fifteen single-bit branching rounds and one parallel completion step. These evaluations stop when observing 500 frame errors or 10^8 frames.

Baselines. For theoretical decoding baselines, we use belief propagation (BP) (Kschischang et al., 2001), successive cancellation (SC) (Arıkan, 2009), and successive-cancellation list (SCL) decoding (Tal & Vardy, 2015). The neural methods compared in Table 1 are CrossMPT (Park et al., 2025), DDECC (Choukroun & Wolf, 2023), and SB-ECC (Helvits & Nachmani, 2026).

5.2 Results

Table 1 summarizes BER performance over BPSK-AWGN. SB-ECC has lower reported BER than our conditional diffusion decoder for every evaluated code and signal-to-noise ratio (SNR). One possible explanation is that DDECC and SB-ECC are formulated as reversing the channel noise, which provides an inductive bias suited to AWGN. Augmenting our decoder with Maxwell list decoding yields the lowest reported BER in 34 of the 42 cells.

Figure 4 complements these results with block error rate (BLER) curves for three representative codes at $E_b/N_0 \in \{2, 3, 4, 5\}$ dB, comparing our methods with SB-ECC and the ML estimates of Helmling et al. (2025). We use Maxwell list sizes $L \in \{8, 16, 32\}$ to study beam-width effects.

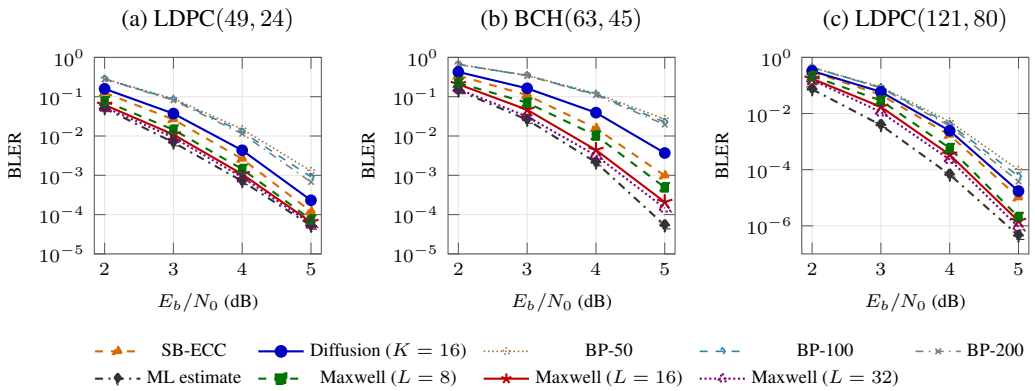


Figure 4: BLER versus E_b/N_0 for (a) LDPC$(49, 24)$, (b) BCH$(63, 45)$, and (c) LDPC$(121, 80)$. Maxwell uses minimum-distance selection.

5.3 Adaptation to other channels

We adapt the LDPC$(49, 24)$ decoder from BPSK-AWGN to a BSC with crossover probability $p = 0.95$, without inverting the received bits. We fine-tune only the 33,536 parameters of the physical-observation projection and conditioner for 50k updates, keeping all other decoder parameters frozen. Both unadapted neural decoders fail on every evaluated frame. On the same 3,616 frames, the adapted Maxwell decoder achieves $-\ln(\text{BLER}) = 2.652$, compared with 1.871 for adapted diffusion and 3.588 for exact codeword MAP (Table 2). This tests adaptation by updating only physical-observation beliefs while fixing the inference mechanism; see Appendix F.1.

Table 2: LDPC$(49, 24)$ on BSC$(p = 0.95)$ before and after adaptation. Entries are $-\ln(\text{error rate})$.

Decoder	BLER	BER
Unadapted		
Diffusion, $K = 16$	0.000	0.236
Maxwell, $L = 16$	0.000	0.198
Adapted		
Diffusion, $K = 16$	1.871	3.710
Maxwell, $L = 16$	2.652	4.157
Exact codeword MAP	3.588	5.224

Additional experiments and analysis. Beyond the short-code comparisons above, we also evaluate longer LDPC codes in Appendix E. Appendix F studies channel adaptation and generalization, and Appendix G reports computational complexity and measured decoding latency.

6 Conclusion

We develop a conditional masked diffusion framework for error-correcting codes in which the received word remains fixed side information. Learned posterior marginals guide the reverse process, motivated by an information-theoretic conservation law. The masked formulation enables a beam search mechanism inspired by the Maxwell decoder of Méasson et al. (2008). Our experiments show that augmenting conditional diffusion with Maxwell beam search yields competitive decoding performance across the evaluated settings. They also demonstrate adaptation from AWGN to a BSC by fine-tuning only the physical-observation modules.

Reliable ECC decoding is important for communication and storage, and offers a controlled setting for studying learned inference under exact constraints. We believe that the Maxwell mechanism may also benefit diffusion-based generation more broadly by retaining alternative partial solutions before committing to uncertain decisions. Related uses of search during generation are already being explored in SOAR (Cao et al., 2026) and Diffusion Tree Sampling (Jain et al., 2025), suggesting a promising direction for extending these ideas beyond ECC.

AI Use Statement

We used OpenAI Codex to assist with finding and summarizing related work; evaluating methodological and experimental options; implementing, debugging, and analyzing research code and eval-

uations; and drafting, organizing, and editing portions of the manuscript. We treated these outputs as suggestions and reviewed AI-assisted text, code, citations, and results before incorporating them. The authors take responsibility for all claims, results, and content of this paper.

REFERENCES

Ziv Aharoni and Henry D. Pfister. Conservation Laws for Diffusion Models, 2026. URL https://arxiv.org/abs/2607.10067. arXiv:2607.10067.

Ziv Aharoni, Bashar Huleihel, Henry D. Pfister, and Haim H. Permuter. Data-Driven Neural Polar Decoders for Unknown Channels With and Without Memory. *IEEE Transactions on Information Theory*, 70(12):8495–8510, December 2024. doi: 10.1109/TIT.2024.3476681. URL https://doi.org/10.1109/TIT.2024.3476681.

Erdal Arıkan. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Transactions on Information Theory*, 55 (7):3051–3073, 2009. doi: 10.1109/TIT.2009.2021379.

Alexei Ashikhmin, Gerhard Kramer, and Stephan ten Brink. Code rate and the area under extrinsic information transfer curves. In *Proceedings IEEE International Symposium on Information Theory*, pp. 115, June 2002. doi: 10.1109/ISIT.2002.1023387. URL https://ieeexplore.ieee.org/abstract/document/1023387.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the Inherent Intractability of Certain Coding Problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978. ISSN 0018-9448. doi: 10.1109/TIT.1978.1055873.

R. C. Bose and D. K. Ray-Chaudhuri. On a Class of Error Correcting Binary Group Codes. *Information and Control*, 3(1):68–79, 1960. doi: 10.1016/S0019-9958(60)90287-4.

Mingyu Cao, Alvaro H. C. Correia, Christos Louizos, Shiwei Liu, and Lu Yin. Search or Accelerate: Confidence-Switched Position Beam Search for Diffusion Language Models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65806.

CCSDS. Short Block Length LDPC Codes for TC Synchronization and Channel Coding. Experimental Specification CCSDS 231.1-O-1, Consultative Committee for Space Data Systems, April 2015. URL https://ccsds.org/Pubs/231x1o1s.pdf.

D. Chase. A class of algorithms for decoding block codes with channel measurement information. *IEEE Transactions on Information Theory*, 18(1):170–182, 1972. doi: 10.1109/TIT.1972.1054746.

Yoni Choukroun and Lior Wolf. Error Correction Code Transformer. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/file/fcd3909db30887ce1da519c4468db668-Paper-Conference.pdf.

Yoni Choukroun and Lior Wolf. Denoising Diffusion Error Correction Codes. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=rLwC0_MG-4w.

Yoni Choukroun and Lior Wolf. A Foundation Model for Error Correction Codes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/093b793f0af498e8fa218eb5445daeb0-Paper-Conference.pdf.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, December 1934. doi: 10.1093/biomet/26.4.404. URL https://doi.org/10.1093/biomet/26.4.404.

Marc P. C. Fossorier and Shu Lin. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Transactions on Information Theory*, 41(5):1379–1396, September 1995. doi: 10.1109/18.412683. URL https://doi.org/10.1109/18.412683.

Marc P. C. Fossorier, Miodrag Mihaljević, and Hideki Imai. Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. *IEEE Transactions on Communications*, 47(5):673–680, 1999. doi: 10.1109/26.768759.

Robert G. Gallager. Low-Density Parity-Check Codes. *IRE Transactions on Information Theory*, 8 (1):21–28, 1962. doi: 10.1109/TIT.1962.1057683.

Michael Helmling, Stefan Scholl, Florian Gensheimer, Tobias Dietz, Kira Kraft, Oliver Griebel, Stefan Ruzika, and Norbert Wehn. Database of Channel Codes and ML Simulation Results, 2025. URL https://rptu.de/channel-codes.

Alon Helvits and Eliya Nachmani. Score Based Error Correcting Code Decoder. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://openreview.net/forum?id=uQiLoAF0bM.

Rom Hirsch, Ziv Aharoni, Henry D. Pfister, and Haim H. Permuter. Neural Polar Decoders for Receivers in Wireless Communication. *IEEE Transactions on Communications*, 74:6889–6903, 2026. doi: 10.1109/TCOMM.2026.3679226. URL https://ieeexplore.ieee.org/document/11457977/.

Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, pp. 146–156, 1959. URL https://db.aconit.org/dbmedia_0/pdf_8/8514.pdf.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion Tree Sampling: Scalable Inference-Time Alignment of Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 145576–145615. Curran Associates, Inc., 2025. doi: 10.52202/085713-4871. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d6484394c4cb5e1f4ecad8d90b912025-Abstract-Conference.html.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. doi: 10.1109/18.910572. URL https://web.stanford.edu/~montanar/TEACHING/Stat375/papers/sumprod.pdf.

David J. C. MacKay. Encyclopedia of Sparse Graph Codes. Hypertext archive, 1999. URL https://www.inference.org.uk/mackay/codes/data.html.

Cyril Méasson. *Conservation Laws for Coding*. PhD thesis, EPFL, 2006. URL https://infoscience.epfl.ch/handle/20.500.14299/221943.

Cyril Méasson, Andrea Montanari, and Rüdiger Urbanke. Maxwell Construction: The Hidden Bridge Between Iterative and Maximum a Posteriori Decoding. *IEEE Transactions on Information Theory*, 54(12):5277–5307, December 2008. ISSN 1557-9654. doi: 10.1109/TIT.2008.2006466. URL https://ieeexplore.ieee.org/document/4675720.

Cyril Méasson, Andrea Montanari, Thomas J. Richardson, and Rüdiger Urbanke. The Generalized Area Theorem and Some of its Consequences. *IEEE Transactions on Information Theory*, 55 (11):4793–4821, November 2009. ISSN 1557-9654. doi: 10.1109/TIT.2009.2030457. URL https://ieeexplore.ieee.org/document/5290273.

Eliya Nachmani and Lior Wolf. Autoregressive Belief Propagation for Decoding Block Codes, 2021. URL https://arxiv.org/abs/2103.11780. arXiv:2103.11780.

Eliya Nachmani, Yair Be'ery, and David Burshtein. Learning to decode linear codes using deep learning. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346, September 2016. doi: 10.1109/ALLERTON.2016.7852251. URL https://ieeexplore.ieee.org/abstract/document/7852251.

Seong-Joon Park, Hee-Youl Kwak, Sang-Hyo Kim, Yongjune Kim, and Jong-Seon No. CrossMPT: Cross-Attention Message-Passing Transformer for Error Correcting Codes. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8f0942c43fcfba4cc66a859b9fcb1bba-Paper-Conference.pdf.

Tom Richardson and Rüdiger Urbanke. *Modern Coding Theory*. Cambridge University Press, 1st edition, March 2008. ISBN 978-0-521-85229-6 978-0-511-79133-8. doi: 10.1017/CBO9780511791338. URL https://www.cambridge.org/core/product/identifier/9780511791338/type/book.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167. Curran Associates, Inc., 2024. doi: 10.52202/079017-3277. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Ido Tal and Alexander Vardy. List Decoding of Polar Codes. *IEEE Transactions on Information Theory*, 61(5):2213–2226, May 2015. doi: 10.1109/TIT.2015.2410251.

Stephan ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Transactions on Communications*, 49(10):1727–1737, October 2001. ISSN 1558-0857. doi: 10.1109/26.957394. URL https://ieeexplore.ieee.org/document/957394.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51 (7):2282–2312, July 2005. ISSN 1557-9654. doi: 10.1109/TIT.2005.850085. URL https://ieeexplore.ieee.org/document/1459044/.

APPENDIX CONTENTS

Appendix A develops the conditional conservation laws and proves the local GEXIT decomposition. Appendix B relates the expected parallel decoding score to conditional dependence and posterior approximation error. Appendix C describes the neural posterior estimator and its conditioning on physical and auxiliary observations.

Appendix D documents evaluation protocols, sample sizes, and reference results. Appendix E presents longer-code results. Appendix F studies adaptation to a binary symmetric channel and generalization to Rayleigh fading. Appendix G analyzes computational complexity and measured decoding latency. Appendix H gives the Maxwell beam-decoder pseudocode.

A Conditional Conservation Laws . . . 13
B Parallel Sampling, Total Correlation, and Path Scores . . . 15
C Posterior-Estimator Implementation . . . 16
D Evaluation Protocols and Reference Results . . . 16
E Longer-Code Results . . . 18
F Adaptation and Generalization across Physical Channels . . . 19
G Computational Complexity and Measured Latency . . . 20
H Maxwell Decoder Algorithm . . . 21

A CONDITIONAL CONSERVATION LAWS

This appendix develops the information-theoretic basis for the decoder in Section 4. We derive a local decomposition of the conditional-entropy derivative and state the resulting conservation laws for exact and learned posterior models. These results follow Aharoni & Pfister (2026), who extended the coding conservation laws of Méasson (2006) to diffusion under model mismatch.

A.1 LOCAL CONDITIONAL GEXIT DECOMPOSITION

Condition on the fixed physical observation R and let the auxiliary channel move from its clean endpoint at $t = 0$ to its uninformative endpoint at $t = 1$. Define the joint conditional entropy

$$h_R(t) = H(X \mid R, Y(t)), \tag{18}$$

which satisfies the endpoint conditions $h_R(0) = 0$ and $h_R(1) = H(X \mid R)$. A homogeneous path assigns the same noise parameter to every coordinate:

$$\gamma(t) = (t_1(t), \dots, t_n(t)) = (t, \dots, t), \qquad t \in [0, 1]. \tag{19}$$

Under the regularity conditions in Appendix A.4.1, the memoryless auxiliary channel yields the following decomposition along this path:

$$\frac{d}{dt} h_R(t) = \sum_{i=1}^{n} \frac{d}{ds} H(X_i \mid R, Y_{-i}(t), Y_i(s)) \Big|_{s=t}. \tag{20}$$

Here $Y_{-i}(t)$ denotes the auxiliary observations in all coordinates other than i at noise level t, while $Y_i(s)$ is the observation in coordinate i at noise level s. The decomposition is local: each derivative varies only s, with every other noise level fixed at t.

Denote the ith local derivative in (20) by $g_{i|R}(t)$. Under the same regularity conditions, it admits the score-weighted likelihood form in (22) (Méasson et al., 2009; Aharoni & Pfister, 2026). Define the scalar auxiliary-channel score

$$S_t(x, y) = \frac{\partial}{\partial t} \log W_t(y \mid x). \tag{21}$$

The local term then has the posterior-marginal representation

$$g_{i|R}(t) = -\mathbb{E}\big[S_t(X_i, Y_i(t)) \log P_{X_i|R,Y(t)}(X_i \mid R, Y(t))\big]. \tag{22}$$

This representation expresses the entropy derivative in terms of the scalar auxiliary kernel and coordinate posterior marginals. We focus on the masking auxiliary channel, although this formulation applies to any memoryless auxiliary-channel family satisfying the regularity, conditional-independence, and endpoint assumptions in Appendix A.4.1.

A.2 PERFECT-MODEL CONSERVATION LAW

Assume that we have access to the true bitwise MAP marginals $P_{X_i|R,Y(t)}$ along the path. The conditional conservation law is

$$H(X \mid R) = \int_0^1 \sum_{i=1}^{n} g_{i|R}(t)\, dt. \tag{23}$$

For a uniformly distributed (n, k) codeword, this area also gives the joint information carried by the physical observation,

$$I(X; R) = k \log 2 - H(X \mid R). \tag{24}$$

A.3 LEARNED POSTERIOR AND MISMATCHED CONSERVATION

For each physical observation $R = r$, consider two distributions on codewords: the true posterior $P_{X|R=r}$ and a model $Q^{\theta}_{X|R=r}$. Pass both distributions through the same memoryless auxiliary channel $W_t^{\otimes n}$. Our aim is to express their conditional cross-entropy,

$$\mathbb{E}_R[\mathrm{CE}(P_{X|R}, Q^{\theta}_{X|R})] = -\mathbb{E}_{P_{R,X}}[\log Q^{\theta}_{X|R}(X \mid R)],$$

as an integral of local posterior quantities, with the average over $R \sim P_R$.

Let $Q^{\theta}_{X_i|R,Y(t)}$ denote the marginal posterior induced by Q^{θ} and the shared channel. It obeys the Bayes factorization

$$Q^{\theta}_{X_i|R,Y(t)}(x_i \mid r, y) = \frac{Q^{\theta}_{X_i|R,Y_{-i}(t)}(x_i \mid r, y_{-i}) W_t(y_i \mid x_i)}{\sum_{u\in\{0,1\}} Q^{\theta}_{X_i|R,Y_{-i}(t)}(u \mid r, y_{-i}) W_t(y_i \mid u)}. \tag{25}$$

The true posterior has the same form with Q^{θ} replaced by P. Thus, posterior approximation is confined to the extrinsic term, while the known local channel likelihood is applied exactly.

The cross-entropy conservation law of Aharoni & Pfister (2026), applied conditionally on R, uses the local field

$$\begin{aligned} g^{Q^{\theta}}_{i|R}(t) = \mathbb{E}_P\Big[&- S_t(X_i, Y_i(t)) \log Q^{\theta}_{X_i|R,Y(t)}(X_i \mid R, Y(t)) \\ &+ \mathbb{E}_{X_i' \sim Q^{\theta}_{X_i|R,Y(t)}}[S_t(X_i', Y_i(t)) \mid R, Y(t)]\Big]. \end{aligned} \tag{26}$$

The outer expectation is under the true distribution and the shared auxiliary channel. Under the regularity conditions in Appendix A.4.1, with finite log loss, model support covering the true support, and the corresponding cross-entropy endpoint limits, integration yields

$$\mathbb{E}_R\Big[\mathrm{CE}\Big(P_{X|R}, Q^{\theta}_{X|R}\Big)\Big] = \int_0^1 \sum_{i=1}^{n} g^{Q^{\theta}}_{i|R}(t)\, dt. \tag{27}$$

This has the same conservation form as (23), with cross-entropy replacing entropy. Its excess over the perfect-model area is $\mathbb{E}_R[D_{\mathrm{KL}}(P_{X|R} \| Q^{\theta}_{X|R})]$. For masking, the factorization in (25) gives the posterior in (9) and yields the masked cross-entropy law in (7).

A.4 PROOF OF THE CONDITIONAL GEXIT DECOMPOSITION

We prove (20) and (22) with side information R, adapting the arguments of Méasson et al. (2009) and Aharoni & Pfister (2026, Appendix A.1–A.2).

A.4.1 AUXILIARY-CHANNEL REGULARITY ASSUMPTIONS

We use the degraded memoryless auxiliary channel and the conditional independence from R specified in Section 3.2. The following conditions restate the smoothness conventions of Aharoni & Pfister (2026, Appendix A) using the notation of Appendices A.1–A.3.

(C1) **Channel regularity.** For each input x and almost every output y, $W_t(y \mid x)$ is locally absolutely continuous in $0 < t < 1$. Its derivative is bounded in absolute value by an integrable function of y, uniformly on compact interior intervals. Summing this derivative over discrete outputs, or integrating it over continuous outputs, gives zero. The score S_t in (21) is evaluated only on the channel support.

(C2) **Cross-entropy domination.** Fix coordinate i and write $Y = (Y_{-i}(t), Y_i(s))$, as in (20). The model posterior $Q^{\theta}_{X_i|R,Y}$ is positive on the true support and has finite expected log loss. The derivative

$$\frac{\partial}{\partial s}\Big[P_{X_i,Y|R}(x_i, y \mid r) \log Q^{\theta}_{X_i|R,Y}(x_i \mid r, y)\Big]$$

is bounded in absolute value by an integrable function, uniformly in compact neighborhoods of the interior homogeneous path.

A.4.2 Local decomposition

Fix t and coordinate i, and abbreviate $Y = (Y_{-i}(t), Y_i(s))$ as in (20). The product auxiliary channel and its conditional independence from R imply that $Y_i(s)$ is independent of $(X_{-i}, R, Y_{-i}(t))$ given X_i. The entropy chain rule therefore gives

$$H(X \mid R, Y) = H(X_i \mid R, Y) + H(X_{-i} \mid X_i, R, Y_{-i}(t)). \tag{28}$$

The second term does not depend on s. Differentiating thus equates the local derivatives of the joint and marginal conditional entropies. Evaluating at $s = t$ and summing along $\gamma(t)$ establishes (20).

A.4.3 Score-weighted likelihood form

Write the marginal conditional entropy as $H(X_i \mid R, Y) = -\mathbb{E}[\log P_{X_i|R,Y}(X_i \mid R, Y)]$. Only the factor $W_s(Y_i \mid X_i)$ in the joint law depends on s. Using the score defined in (21), differentiation gives

$$\begin{aligned}\frac{d}{ds}H(X_i \mid R, Y) = &-\mathbb{E}\big[S_s(X_i, Y_i) \log P_{X_i|R,Y}(X_i \mid R, Y)\big] \\ &-\mathbb{E}\left[\frac{d}{ds}\log P_{X_i|R,Y}(X_i \mid R, Y)\right].\end{aligned} \tag{29}$$

Apply the Bayes factorization (25). The extrinsic factor depends only on $Y_{-i}(t)$, so differentiating the log posterior yields

$$\begin{aligned}\frac{d}{ds}\log P_{X_i|R,Y}(x_i \mid r, y) = &\, S_s(x_i, y_i) \\ &- \mathbb{E}_{X_i' \sim P_{X_i|R,Y}(\cdot|r,y)}[S_s(X_i', y_i)]\,.\end{aligned} \tag{30}$$

At $(x_i, r, y) = (X_i, R, Y)$, the two terms have equal expectations by the tower property. The second expectation in (29) therefore vanishes, proving (22) at $s = t$. The corresponding score form for the cross-entropy derivative is derived by Aharoni & Pfister (2026, Theorem 3.1 and Appendix A.3); conditioning on R gives (26).

B Parallel Sampling, Total Correlation, and Path Scores

Independent sampling replaces the joint posterior of the selected bits by a product of marginals. Total correlation (TC) measures the resulting expected log-score penalty.

Penalty for a parallel step. Fix $R = r$ and the decoder state $\widetilde{y}^{(k)}$. Select $\mathcal{G} = \mathcal{G}_k$ before assigning its bit values, and write

$$\pi(x_{\mathcal{G}}) = P_{X_{\mathcal{G}}|R,Y(t^{(k)})}(x_{\mathcal{G}} \mid r, \widetilde{y}^{(k)}), \qquad \pi_i(x_i) = P_{X_i|R,Y(t^{(k)})}(x_i \mid r, \widetilde{y}^{(k)}).$$

The expected difference between the joint log probability and the sum of marginal log probabilities is

$$\begin{aligned}\mathrm{TC}(X_{\mathcal{G}} \mid r, \widetilde{y}^{(k)}) &:= D_{\mathrm{KL}}\left(\pi \,\middle\|\, \prod_{i\in\mathcal{G}} \pi_i\right) \\ &= \mathbb{E}_{\pi}\left[\log \pi(X_{\mathcal{G}}) - \sum_{i\in\mathcal{G}} \log \pi_i(X_i)\right] \geq 0. \end{aligned} \tag{31}$$

TC vanishes for conditionally independent selected bits, and automatically for a singleton.

Accumulated path score. Let $s_{\mathrm{par}}(x; r)$ be the score in (14) with exact posterior marginals along a complete reveal path. Each group is selected using r and the previously revealed bits. Predicting and revealing one bit at a time takes $K = n$ steps for a length-n codeword. Updating the exact posterior after each bit, the probability chain rule gives $s_{\mathrm{par}}(x; r) = \log P_{X|R}(x \mid r)$. When $K < n$, some steps reveal several bits in parallel; summing their local TC penalties gives

$$\begin{aligned}\mathbb{E}[-s_{\mathrm{par}}(X; r) \mid R = r] &= H(X \mid R = r) \\ &\quad + \sum_{k=1}^{K} \mathbb{E}[\mathrm{TC}(X_{\mathcal{G}_k} \mid r, \widetilde{y}^{(k)}) \mid R = r]. \end{aligned} \tag{32}$$

Thus the expected negative path score upper-bounds posterior entropy.

Learned marginals. Let q_i denote $Q^{\theta}_{X_i|R,Y(t^{(k)})}(\cdot \mid r, \widetilde{y}^{(k)})$. For finite log loss and model support covering the true support,

$$\mathbb{E}_{\pi}\left[\log \pi(X_{\mathcal{G}}) - \sum_{i\in\mathcal{G}} \log q_i(X_i)\right] = \mathrm{TC}(X_{\mathcal{G}} \mid r, \widetilde{y}^{(k)}) + \sum_{i\in\mathcal{G}} D_{\mathrm{KL}}(\pi_i \| q_i). \tag{33}$$

Marginal prediction error therefore adds to the dependence penalty.

LDPC intuition. Early sequential decisions and parity propagation can break Tanner-graph cycles. For locally tree-like codes, delaying parallel completion until residual dependencies weaken can improve the path score's approximation to the log posterior.

C Posterior-Estimator Implementation

We implement the posterior estimator of Section 4.2 with a CrossMPT backbone (Park et al., 2025). Code-bit and parity-check tokens use alternating H- and H^{T}-masked cross-attention.

At decoding step k, an auxiliary encoder represents the current state and mask fraction $(\widetilde{y}^{(k)}, t^{(k)})$. A separate physical encoder represents the fixed observation r and any available physical-channel parameters. Both encoders condition every CrossMPT layer through adaptive layer normalization: their scale contributions are added, and their shift contributions are added. The alternating cross-attention blocks remain unchanged.

For each masked bit i, the output head predicts logits for the extrinsic posterior

$$Q^{\theta}_{X_i|R,Y_{-i}(t^{(k)})}\left(x_i \mid r, \widetilde{y}^{(k)}_{-i}\right), \qquad x_i \in \{0, 1\}. \tag{34}$$

The known local auxiliary-channel likelihood is applied outside the network using (9). Thus, $Q^{\theta}_{X_i|R,Y(t^{(k)})}(x_i \mid r, \widetilde{y}^{(k)})$ equals the predicted extrinsic posterior at masked positions and $\mathbf{1}\{x_i = \widetilde{y}^{(k)}_i\}$ at revealed positions. These marginals drive both the diffusion and Maxwell decoders.

D Evaluation Protocols and Reference Results

This appendix describes the training procedure, simulation protocol, reference sources, decoder configurations, sample sizes, and stopping criteria used in the experimental comparisons. Comparisons with published methods use independently generated samples and may differ in training objective, inference procedure, and evaluation budget.

D.1 BASELINES EVALUATIONS

The BP, CrossMPT, DDECC, and SB-ECC BER values in Table 1 are transcribed from Helvits & Nachmani (2026, Table 1). BP uses 50 decoding iterations, and DDECC and SB-ECC share the CrossMPT backbone in that comparison.

For LDPC$(49, 24)$, DDECC reports at least 10^5 frames and at least 500 erroneous frames per SNR; SB-ECC stops after 500 frame errors or 10^9 evaluated LDPC frames; CrossMPT reports at least 500 erroneous frames without specifying the total frame count. At 4 dB, Park et al. (2025, Appendix Table 3) report $-\ln(\mathrm{BLER}) = 4.47$ for CrossMPT, where BLER denotes the frame-error rate. For the longer-code comparison in Table 4, the SB-ECC values for LDPC$(204, 102)$ and LDPC$(529, 440)$ are transcribed from Helvits & Nachmani (2026, Table 6).

D.2 TRAINING AND SIMULATION DETAILS

AWGN training. We train a separate posterior estimator for each code. Training samples are generated online from random codewords, with E_b/N_0 drawn uniformly from the discrete grid $\{2, 3, 4, 5, 6, 7\}$ dB. For every sample, the codeword is mapped to BPSK and passed through AWGN to obtain the fixed physical observation. We then draw $t \sim \mathrm{Unif}(0, 1)$, mask coordinates independently with probability t, and condition on masking at least one coordinate. The model minimizes the masked-bit negative log-likelihood in (11) for three million iterations. We use batches of 256 codewords and AdamW with zero weight decay; the learning rate is cosine-annealed from 5×10^{-4} to 5×10^{-6}. We maintain both raw and exponential-moving-average (EMA) model parameters during training and use the EMA weights for the reported AWGN diffusion and Maxwell evaluations. The posterior-estimator architecture is specified in Appendix C.

Monte Carlo simulation. For a rate-$R_c = k/n$ code at a specified E_b/N_0, each frame independently samples a random codeword and Gaussian noise according to

$$r_i = 1 - 2x_i + z_i, \qquad z_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2), \qquad \sigma^2 = \frac{1}{2R_c 10^{(E_b/N_0)_{\mathrm{dB}}/10}}. \tag{35}$$

Unless stated otherwise, conditional diffusion uses $K = 16$ confidence-balanced reverse steps. Maxwell uses beam width $L = 16$, fifteen single-bit branches at the least-confident unresolved positions, parity peeling, and one final parallel completion. The model-path and minimum-distance decision rules are evaluated on the same frames and candidate lists. Diffusion evaluations stop after 500 frame errors or 10^8 frames.

Compute. The base models were trained on NVIDIA H200 graphics processing units (GPUs). The short-code simulations (codes with $n \leq 128$) were evaluated on NVIDIA RTX 2080 Ti GPUs, whereas the longer-code experiments in Appendix E and the BSC adaptation experiment in Appendix F.1 used H200 GPUs. These hardware statements describe where our runs were executed; the comparisons with published baselines are not compute matched.

D.3 POLAR-CODE BER AND BLER MEASUREMENTS

Table 3 compares coded-bit BER and BLER for Polar$(64, 32)$ at 4 dB; Figure 1 uses the BER column. For ECCT and CrossMPT, we use BER from Table 4 and BLER from Table 3 of Park et al. (2025); DDECC and SB-ECC BER values follow Table 1.

BP and Polar SCL baselines. We evaluate sum-product BP on the Tanner graph of the code's parity-check matrix, using 5, 10, 20, or 50 iterations (Kschischang et al., 2001). The Polar baseline uses successive-cancellation list (SCL) decoding without a cyclic redundancy check (CRC), with list sizes $L \in \{1, 4, 8, 16, 32, 64\}$ (Tal & Vardy, 2015). SCL selects a candidate using its path metric and re-encodes the decoded information vector to measure BER over all 64 codeword positions. Both baselines use the same parity-check matrix defining the Polar$(64, 32)$ code.

Table 3: Coded-bit BER and BLER for Polar$(64, 32)$ at 4 dB. BER values are used in Figure 1; dashes denote unavailable results.

Decoder	$-\ln(\mathrm{BER})$	$-\ln(\mathrm{BLER})$
BP-5	3.5265	0.9464
BP-10	3.8011	1.2577
BP-20	3.9960	1.5170
BP-50	4.2921	1.8970
SCL-1	3.8083	2.1005
SCL-4	5.8792	3.9945
SCL-8	6.8695	4.9123
SCL-16	7.5590	5.5211
SCL-32	7.8346	5.7944
SCL-64	7.9150	5.8578
ECCT	6.9900	4.18
CrossMPT	7.5000	4.83
DDECC	7.0300	–
SB-ECC	7.7700	–
Diffusion	7.4360	5.260
Maxwell + min. distance	7.8670	5.829
OSD-3 estimate	7.9783	5.9313

Codeword recovery. Both our diffusion decoder and its Maxwell variant achieve lower BLER than BP and the non-diffusion neural decoders ECCT and CrossMPT in this comparison. They therefore recover the transmitted codeword more frequently. Relative to CrossMPT, diffusion achieves lower BLER at the expense of slightly higher BER. Maxwell further improves both metrics.

OSD reference estimate. Ordered-statistics decoding (OSD) selects a most-reliable information set and puts the generator matrix in systematic form on those positions (Fossorier & Lin, 1995). OSD-3 re-encodes the hard decisions on this set after every perturbation of weight at most three and selects the candidate with minimum Euclidean distance to the received word. Over 400,000 frames at 4 dB, we observed 8,776 coded-bit errors and 1,062 block errors, all of which yielded a codeword with strictly higher channel likelihood than the transmitted word. Under a uniform codeword prior, the probability of such likelihood witnesses lower-bounds the exact-ML BLER, while the OSD-3 BLER upper-bounds it. Applying one-sided 97.5% Clopper–Pearson limits (Clopper & Pearson, 1934) to these two probabilities gives a 95% confidence bracket of $[2.4978, 2.8194] \times 10^{-3}$ for ML BLER, or $-\ln(\mathrm{BLER}_{\mathrm{ML}}) \in [5.8712, 5.9924]$; the BER entry remains an OSD-3 estimate.

E LONGER-CODE RESULTS

This appendix evaluates decoding performance for two longer LDPC codes over BPSK–AWGN at 4 and 5 dB. The models use the training and simulation protocol in Appendix D.2. We report $-\ln(\mathrm{BER})$. Overall, diffusion remains competitive with SB-ECC on both longer codes, whereas Maxwell yields the strongest estimates across the reported SNRs.

Table 4: Negative log BER for long-block LDPC codes over BPSK–AWGN at 4 and 5 dB. SB-ECC results are reproduced from Helvits & Nachmani (2026, Table 6). Boldface and underlining mark the best and second-best distinct values.

Code	SB-ECC		Diffusion		Maxwell	
E_b/N_0 (dB)	4	5	4	5	4	5
LDPC$(204, 102)$	11.20	<u>15.78</u>	<u>11.30</u>	14.74	**12.95**	**16.29**
LDPC$(529, 440)$	7.93	15.23	<u>8.06</u>	<u>15.56</u>	**8.85**	**17.71**

F Adaptation and Generalization across Physical Channels

We evaluate two forms of transfer from BPSK-AWGN training: adaptation to a binary symmetric channel by updating only the physical-observation modules, and generalization to memoryless Rayleigh fading without fine-tuning.

F.1 Adaptation to a binary symmetric channel

Channel and adaptation objective. For LDPC$(49, 24)$, we use signed code symbols $X_i \in \{+1, -1\}$ and channel outputs

$$R_i = X_i Z_i, \qquad \Pr(Z_i = -1) = 0.95, \qquad \Pr(Z_i = +1) = 0.05, \tag{36}$$

with independent channel signs Z_i. This BSC is equivalent to BSC(0.05) after deterministic output inversion and therefore has the same capacity.

Training. We initialize from the EMA weights of the 3-million-update CrossMPT masking model pretrained on BPSK-AWGN. Only the physical-observation projection and conditioner are updated, totaling 33,536 existing parameters; the remaining decoder backbone is frozen, and no parameters are added. Training uses one seed, batches of 256 codewords, and AdamW with zero weight decay, peak learning rate 10^{-4}, and floor 5×10^{-6}. Training runs for 50k updates with 1k warm-up updates. All adapted results below use the raw weights after 50k updates, rather than adapter EMA weights.

Decoders and paired evaluation. Diffusion uses 16 confidence-balanced reverse steps. Maxwell uses beam width 16, with 15 sequential branching rounds at the least-confident unresolved bit followed by one parallel completion round. The classical baselines are sum-product BP with 50 and 100 iterations. Exact bitwise MAP and exact codeword MAP enumerate all 2^{24} codewords. The six adapted and reference decoders share a deterministic channel stream.

Table 5: LDPC$(49, 24)$ results on BSC$(p = 0.95)$, using the same 3,616 frames for every decoder. Entries headed BLER and BER are $-\ln$(error rate). Valid is the parity-valid output fraction.

Decoder	Updates	Bit errors	Frame errors	BLER	BER	Valid
Diffusion, $K = 16$	None	139,975	3,616	0.000	0.236	0.3465
Maxwell, $L = 16$	None	145,325	3,616	0.000	0.198	0.9621
Diffusion, $K = 16$	50k	4,335	557	1.871	3.710	0.8509
Maxwell, $L = 16$	50k	2,775	255	2.652	4.157	0.9900
BP(50)	–	2,690	431	2.127	4.188	0.8822
BP(100)	–	2,568	424	2.143	4.234	0.8850
Exact bitwise MAP	–	824	164	3.093	5.371	0.9552
Exact codeword MAP	–	954	100	3.588	5.224	1.0000

Results and scope. Both unadapted neural decoders fail on every frame, with BER 0.79000 for diffusion and 0.82019 for Maxwell. Adaptation reduces these BERs to 0.02447 and 0.01566. Maxwell achieves BLER 0.07052, compared with 0.11726 for BP(100), a 39.9% relative reduction, while its BER is 8.1% higher than BP(100)'s 0.01449. Exact codeword MAP achieves BLER 0.02765, so a gap to optimal frame decoding remains. Bitwise MAP minimizes expected bit error and has the lowest observed BER; it can return a non-codeword.

F.2 Generalization to Rayleigh fading

Following the channel setting in Helvits & Nachmani (2026, Section 7.4 and Table 7), we consider

$$R_i = h_i(1 - 2X_i) + Z_i, \qquad h_i \overset{\text{i.i.d.}}{\sim} \text{Rayleigh}(1), \qquad Z_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2). \tag{37}$$

The fading amplitudes are independent across coordinates, so this is a memoryless fading channel. We use $\sigma^2 = [2(k/n)10^{(E_b/N_0)_{\text{dB}}/10}]^{-1}$; the scale-one Rayleigh distribution has $\mathbb{E}[h_i^2] = 2$. The receiver is given the channel observations without the fading amplitudes.

We evaluate AWGN-trained EMA checkpoints without fine-tuning, using 16 confidence-balanced diffusion steps and Maxwell decoding with beam width $L = 16$. Maxwell performs fifteen single-bit branching rounds followed by parallel completion. We compare selection by the highest accumulated model-path score and with selection by minimum distance to the received signal. All three decoder decisions are evaluated on the same frames, with a stopping target of 500 frame errors for minimum-distance selection or 10^8 frames.

Table 6: Rayleigh-fading results for AWGN-trained decoders. Entries are $-\ln(\mathrm{BER})$; higher is better. Baselines are from Helvits & Nachmani (2026, Table 7). Bold marks the best observed values among our decoders in the unstarred rows.

Code	E_b/N_0	Published baselines				Ours		
	(dB)	BP-50	BP-200	CrossMPT	SB-ECC	Diffusion	Maxwell	
						16 steps	Min. distance	Highest score
LDPC$(121, 70)$	4	4.06	4.26	4.25	4.62	4.330	**4.999**	**4.999**
LDPC$(121, 70)$	5	5.10	5.48	5.53	6.02	5.702	**6.600**	6.597
LDPC$(121, 70)$	6	6.34	6.89	7.11	7.92	7.449	**8.769**	8.761
CCSDS$(128, 64)$	4	5.25	5.59	5.25	5.77	5.610	**6.191**	**6.191**
CCSDS$(128, 64)$	5	6.78	7.37	6.94	7.70	7.485	**8.372**	**8.372**
CCSDS$(128, 64)$	6	8.72	9.64	8.92	10.08	9.795	10.737	10.737

Table 6 compares these results with BP-50, BP-200, CrossMPT, and SB-ECC values from Helvits & Nachmani (2026, Table 7). Both Maxwell selectors have lower observed BER than 16-step diffusion at the displayed points and yield similar BER to one another. The published baselines use separate evaluations, and the comparison does not control for code construction or SNR normalization across implementations. These values therefore provide a cross-study reference.

G Computational Complexity and Measured Latency

Computational cost differs across decoder families. A single-pass decoder requires one neural evaluation, whereas diffusion performs K sequential evaluations. Maxwell also performs K sequential rounds, each batching at most L beam candidates, for at most KL candidate evaluations and $O(Ln)$ beam storage. Table 7 reports latency for Polar$(64, 32)$ on an RTX 2080 Ti. Each row comprises 300 timed requests following 30 warm-up requests. Maxwell uses $L = 16$, single-bit branching, and a final parallel-completion round.

Table 7: Measured latency for single-frame Polar$(64, 32)$ decoding.

Decoder	Steps	Median (ms)	percentile 95 (ms)
CrossMPT	1	6.632	6.712
Diffusion	10	65.786	68.130
Diffusion	16	104.639	105.920
Maxwell, peeling	10	126.754	130.000
Maxwell, peeling	16	182.647	186.975
SB-ECC, Euler	10	76.000	76.990

H MAXWELL DECODER ALGORITHM

Algorithm 2 summarizes the Maxwell beam decoder, including parity propagation and the final model-score or minimum-distance decision rule.

Algorithm 2 Maxwell beam search

Require: r, H, L, K; branch counts $b_K, \dots, b_2$; $\mathcal{D} \in \{\mathcal{D}_{\text{model}}, \mathcal{D}_{\text{dist}}\}$

$(\widetilde{y}_1^{(K)}, s_1^{(K)}) \leftarrow (*^n, 0)$; $\mathcal{B}_K \leftarrow \{(\widetilde{y}_1^{(K)}, s_1^{(K)})\}$
for $k = K, K-1, \dots, 1$ **do**
  $L_k \leftarrow |\mathcal{B}_k|$; $\mathcal{S}_k \leftarrow \varnothing$
  $t_\ell^{(k)} \leftarrow n^{-1} \sum_i \mathbf{1}\{\widetilde{y}_{\ell,i}^{(k)} = *\}$ for $\ell = 1, \dots, L_k$
  Predict $Q^\theta_{X_i \mid R, Y(t_\ell^{(k)})}(\cdot \mid r, \widetilde{y}_\ell^{(k)})$ for all ℓ, i in one batched pass
  Compute $\lambda_{\ell,i}^{(k)}$ by (12) for each candidate ℓ
  for all $(\widetilde{y}_\ell^{(k)}, s_\ell^{(k)}) \in \mathcal{B}_k$ in parallel **do**
    if $k > 1$ **then**
      $\mathcal{G}_{\ell,k} \leftarrow$ up to b_k masked indices with smallest $|\lambda_{\ell,i}^{(k)}|$ (least confident)
      Form $\bar{y}_{\ell,j}^{(k-1)}$, $j = 1, \dots, 2^{|\mathcal{G}_{\ell,k}|}$, by all binary assignments on $\mathcal{G}_{\ell,k}$
      Set $\bar{y}_{\ell,j,i}^{(k-1)} \leftarrow \widetilde{y}_{\ell,i}^{(k)}$ for $i \notin \mathcal{G}_{\ell,k}$
    else
      $\mathcal{G}_{\ell,1} \leftarrow \{i : \widetilde{y}_{\ell,i}^{(1)} = *\}$
      Form $\bar{y}_{\ell,1}^{(0)}$ by (15), using $\mathbf{1}\{\lambda_{\ell,i}^{(1)} \geq 0\}$ on $\mathcal{G}_{\ell,1}$
    end if
    for all constructed children $\bar{y}_{\ell,j}^{(k-1)}$ **do**
      Compute $s_{\ell,j}^{(k-1)}$ by (14) from $(\widetilde{y}_\ell^{(k)}, s_\ell^{(k)})$
      Propagate parity checks in H on $\bar{y}_{\ell,j}^{(k-1)}$; discard it if a contradiction is detected
      Append the surviving pair $(\bar{y}_{\ell,j}^{(k-1)}, s_{\ell,j}^{(k-1)})$ to $\mathcal{S}_k$
    end for
  end for
  $\mathcal{B}_{k-1} \leftarrow \mathrm{Top}_L(\mathcal{S}_k)$ by path score
end for
$\mathcal{L} \leftarrow \{x : (x, s) \in \mathcal{B}_0,\ x \in \{0,1\}^n,\ Hx = 0\}$, retaining the largest score $s(x)$
return $\widehat{x} = \mathcal{D}(\mathcal{L}, r)$; report failure if $\mathcal{L} = \varnothing$