How Good Are Masked Discrete Diffusion Models at Constrained Sampling?

Anonymous authors
Paper under double-blind review

Abstract

We study constrained sampling with masked discrete diffusion models in combinatorial domains with sparse constraint dependency graphs. Adopting the masking policy of partial rejection sampling (PRS), we extend its classical repair guarantees from fixed product distributions of independent variables to an adaptive repair process driven by the conditional distributions of masked diffusion. We obtain Lovász local lemma–like conditions under which masked diffusion efficiently reaches a feasible assignment. Under the symmetric conditions, the expected number of repair rounds is at most logarithmic in the number of constraints, and the expected total number of constraints selected across rounds is at most linear, matching the classical PRS bounds for product distributions. We also give conditions for tractable exact sampling from the data distribution conditioned on feasibility. We test the distributional fidelity of the model for hypergraph coloring, uniform SAT, and rooted spanning trees, and show strong empirical results against state of the art diffusion solvers on 3-SAT.

1 Introduction

Constrained generation seeks to combine a probabilistic model with explicit requirements on the validity of its outputs. Such requirements range from schema and syntax constraints on function calls, queries, and symbolic expressions (Mündler et al., 2026; Dang & Ermon, 2026) to chemical valence constraints in molecular generation (Liu et al., 2018) and non-overlap and boundary constraints in chip design (Lee et al., 2025b). We study this problem for masked diffusion models, which generate discrete sequences by iteratively predicting missing tokens (Sahoo et al., 2024; Shi et al., 2024). For diffusion models more broadly, recent theory relates score accuracy to approximate sampling guarantees for broad classes of non-log-concave continuous distributions (Chen et al., 2023) and arbitrary distributions on the Boolean hypercube (Chen & Ying, 2025). Recent work on random constraint satisfaction problems shows how their mathematical structure can explain diffusion sampling performance and how variable ordering can improve it (Bhatt et al., 2026). Together with recent advances in constrained decoding (Suresh et al., 2025; Dang & Ermon, 2026), these results lead us to ask under what conditions diffusion models can efficiently generate feasible samples. More strongly, when can we certify sampling from a prescribed target distribution conditioned on feasibility? We investigate these questions for constrained combinatorial distributions on the Boolean hypercube, which arise naturally in optimization and design problems.

Establishing such guarantees for neural methods is challenging. One approach is *algorithmic alignment*, which relates a network's internal computations to the steps of an algorithm with known guarantees (Xu et al., 2020; Veličković & Blundell, 2021; Wittig et al., 2026). Examples include graph neural networks (GNNs) trained to imitate Bellman–Ford updates (Veličković et al., 2020), with guarantees under specific training conditions (Nerem et al., 2026), and GNN architectures derived from solvers for semidefinite relaxations of combinatorial optimization problems (Yau et al., 2024). We follow this principle at the level of the sampling procedure, seeking a classical algorithm whose resampling steps can be implemented by masked diffusion.

Rejection sampling is a natural starting point, but its acceptance probability can be too small for efficient sampling in large constrained spaces. Partial rejection sampling (PRS) offers an appealing alternative by iteratively resampling selected subsets of variables while keeping the rest. Intuitively, this should work well when violations are not too likely and dependencies between constraints are

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.

local and sparse, so that a repair is unlikely to trigger a growing chain of new violations. The Lovász local lemma, a classical tool from probabilistic combinatorics, formalizes this balance between violation probabilities and dependencies and, in its algorithmic forms, gives termination guarantees for local resampling procedures (Erdős & Lovász, 1975; Moser & Tardos, 2010; Achlioptas et al., 2019). Under suitable local-lemma–type conditions, general PRS samples exactly from a fixed product distribution of independent variables conditioned on feasibility (Guo et al., 2019). These guarantees make PRS a natural reference algorithm for sparse combinatorial distributions.

To align masked diffusion with PRS, we implement the resampling policy of *general PRS* as a masking policy. Starting from violated constraints, the PRS rule adds selected neighbors in the dependency graph, whose edges connect constraints sharing a variable. The model then resamples the variables in the selected constraints conditional on the values retained outside the mask. This preserves the PRS choice of variables while allowing a learned conditional distribution to supply their replacement values.

Transferring these guarantees to a learned sampler raises two challenges. First, its replacement distribution depends on the values retained outside the mask and may include dependencies among the resampled variables, whereas the PRS analysis relies on a fixed product distribution. Hence, the PRS result does not apply unchanged. Second, reaching some feasible sample – the goal of PRS – does not ensure the correct probabilities among feasible outputs – the goal of diffusion, even when each repair uses exact data conditionals. We address the first challenge by bounding how violations and additions to the repair set propagate through shared variables. We address the second with a correction that restores the target distribution under additional sampling and acceptance assumptions.

Contributions. Our contributions connect masked diffusion with PRS, from guarantees in sparse regimes to empirical performance beyond them.

- We extend the PRS conditions to adaptive conditional distributions and show that, under these conditions, a masked diffusion model reaches feasibility almost surely, with bounds on the expected number of repair rounds and the total number of constraints selected for repair across rounds. We recover the classical PRS guarantees for fixed product distributions as a special case.
- We provide a simple algorithm for sampling from p_{data} conditioned on feasibility with polynomial expected overhead under the symmetric PRS conditions and additional sampling and acceptance assumptions.
- We test sampling fidelity on several constrained combinatorial distributions against neural and classical baselines. We also show that the PRS repair blueprint can yield an empirically competitive algorithm for finding satisfying assignments in harder SAT regimes beyond those we can readily certify.

2 Background

2.1 Masked diffusion models

Masked diffusion models (MDMs) generate discrete assignments by predicting variables hidden by a mask (Sahoo et al., 2024; Shi et al., 2024). Let p_{data} be a distribution on $\Omega = \prod_{v=1}^{n} \Sigma_v$, with each Σ_v finite. For $B \subseteq [n]$, $M_B x$ masks the variables in B and retains the *visible assignment* $z = x_{B^c}$, where $B^c = [n] \setminus B$. A denoiser predicts a distribution $\nu_{B,z,v}$ for each masked variable $v \in B$. With exact predictions, revealing masked variables one at a time draws from $p_{\mathrm{data}}(X_B \mid X_{B^c} = z)$ by the chain rule. The masking policy decides which values to reveal or revisit. Appendix J.1 gives the masked cross-entropy training objective and describes mask-selection policies.

Remasking lets an MDM revisit generated values through scheduled or quality-based masks (Wang et al., 2025; Kim et al., 2026). Iterative Partial Refinement similarly resamples selected regions of a completed sample and improves constraint satisfaction in tasks such as Sudoku (Kang et al., 2026).

2.2 PARTIAL REJECTION SAMPLING

We first review how shared variables affect the number of constraints selected during fixed-product PRS, then turn to model-dependent resampling in Section 3.

Consider m constraints on the n variables above. For each constraint $i \in [m]$, the *bad event* $A_i \subseteq \Omega$ consists of the assignments that violate it. Its *scope* $V_i \subseteq [n]$ lists the variables it involves. The set of feasible assignments $\mathcal{F}$ and the constraints violated by assignment x are

$$\mathcal{F} = \bigcap_{i=1}^{m} A_i^c, \qquad \mathrm{Bad}(x) = \{i \mid x \in A_i\}.$$

Resampling a variable can affect only the constraints that contain it. Thus, the possible propagation of a repair is determined by which constraints share variables. We record this variable sharing in the *dependency graph*, whose vertices are constraints and whose edges connect $i \neq j$ when $V_i \cap V_j \neq \varnothing$. For a set S of constraints, $V(S) = \bigcup_{i \in S} V_i$ lists all variables they involve. Write $\Gamma(S)$ for their neighbors outside S and $\Gamma^+(S) = S \cup \Gamma(S)$. We write Δ for the maximum degree of the dependency graph, the largest number of neighbors of any constraint. When variables are drawn independently, constraints with disjoint scopes are independent.[1]

We summarize the key elements of general PRS (Guo et al., 2019) through a repair template that separates the choice of variables from their resampling distribution. Starting from a complete assignment, the template selects violated constraints and certain neighbors, resamples their variables while retaining all other values, and repeats until every constraint is satisfied.

The PRS repair mask. The *PRS repair set* is the set of constraints whose variables will be resampled in the current round. It includes every violated constraint and may also include satisfied neighboring constraints. To construct it, start with $S = \mathrm{Bad}(x)$ and examine neighboring constraints that have not yet been tested. For a candidate neighboring constraint j, keep the values assigned to variables in $V_j \cap V(S)$ fixed and ask whether some assignment to the remaining variables of V_j would violate j. Equivalently, some assignment $y \in A_j$ must agree with x on $V_j \cap V(S)$. If so, add j to S. Otherwise, exclude it for the rest of the round. Candidates are tested in fixed-order batches until no unexamined neighbor remains. The assignment x stays unchanged during selection. Write $\mathrm{Res}(x)$ for the final repair set. Its variables $V(\mathrm{Res}(x))$ form the ***PRS repair mask***, the coordinates to resample. Algorithm 2 in the appendix gives the full construction, and Algorithm 1 combines it with a resampling rule that draws replacement values conditional on the retained values. Figure 1 illustrates this construction for three clauses and contrasts fixed-product resampling with the masked-diffusion resampling introduced in Section 3.

Algorithm 1 The PRS repair template

Require A complete initial proposal x, the constraints, and a resampling rule
loop
  Find all violated constraints
  if none are violated **then**
    return x
  end if
  Construct the repair set $S = \mathrm{Res}(x)$ using the PRS rule
  Form the PRS repair mask $B = V(S)$
  Draw replacement values for B using the resampling rule
  Update x with these values, keeping all variables outside B fixed
end loop

General PRS instantiates this template with a resampling rule based on a fixed product distribution $\mu = \bigotimes_{v=1}^{n} \mu_v$. It initializes $x \sim \mu$ and independently resamples each selected variable v from its original distribution μ_v. If this product-distribution instance stops with probability one, its output distribution is exactly $\mu(\cdot \mid \mathcal{F})$ (Guo et al., 2019).

[1] For correlated data, sharing variables and statistical dependence are different. Our graph records variable sharing.

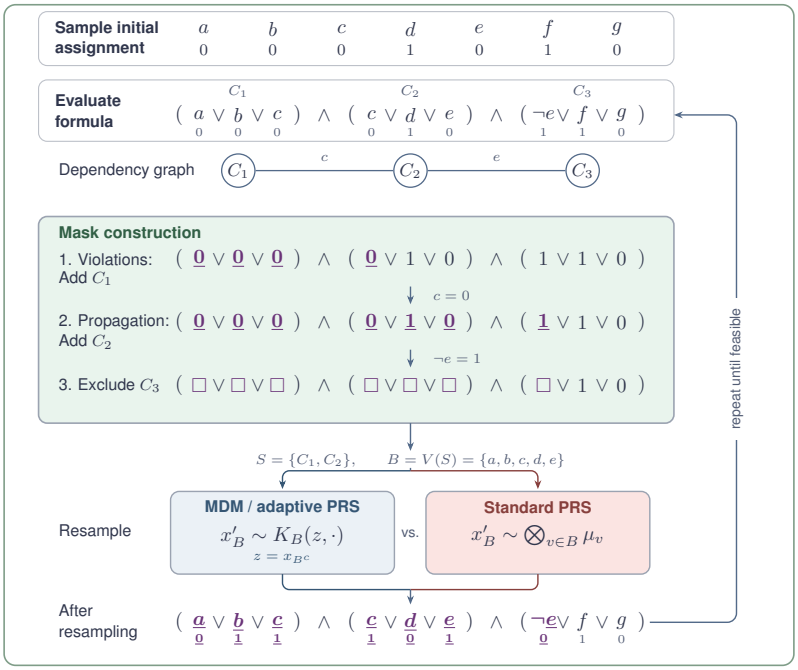

Figure 1: A repair round with the PRS template. The initial assignment violates only C_1, which enters the PRS repair set (purple, underlined variables). The rule then adds C_2 because the current assignment with $c = 0$ can be extended with $d = e = 0$ to violate C_2. It excludes C_3 because $e = 0$ satisfies C_3. Bold, underlined violet values in the repair-set construction mark the selected variables. □ denotes a masked value. The two branches are alternative resampling rules. Standard PRS draws independently from fixed marginals μ_v while adaptive PRS uses the model's joint distribution $K_B(z, \cdot)$ conditional on the unmasked values $z = x_{B^c}$.

The local lemma and symmetric PRS conditions. Exactness of the output distribution alone does not tell us whether general PRS reaches feasibility efficiently. The Lovász local lemma provides a starting point by relating the probability of a violation to the number of constraints that depend on it (Erdős & Lovász, 1975). If every bad event has probability $\mu(A_i) \le p_\star$, its symmetric form gives the sufficient condition

$$e\, p_\star(\Delta + 1) \le 1 \tag{1}$$

This ensures $\mu(\mathcal{F}) > 0$, so feasible assignments exist in the support of μ. Moser–Tardos resampling turns local-lemma conditions into an efficient procedure for finding a feasible assignment (Moser & Tardos, 2010). Its output need not follow $\mu(\cdot \mid \mathcal{F})$, whereas terminating general PRS returns an exact sample from that distribution. To bound PRS repair rounds and the total number of constraints selected, we must also control how far the repair set can expand.

The number of rounds and selected constraints before repair terminates depends on two effects. Resampling can create new violations, and the construction of the next repair set can add further neighbors. We bound the first through the violation probabilities $\mu(A_i)$. For the second, consider an ordered pair of neighboring constraints i, j and define

$$U_{ij} = \{x \mid \text{some } y \in A_j \text{ agrees with } x \text{ on } V_i \cap V_j\}. \tag{2}$$

In words, U_{ij} holds if we can assign values to the variables of j outside V_i so that, together with the current values on the variables shared by i and j, constraint j is violated. If i is already selected, the repair rule can add j only if this compatibility holds. The full test may also fix values shared with other selected constraints. With $p_\star = \max_i \mu(A_i)$ and $r_\star = \max_{i \sim j} \mu(U_{ij})$, one sufficient set of symmetric PRS conditions is

$$6e\, p_\star \Delta^2 \le 1, \qquad 3e\, r_\star \Delta \le 1 \quad (\Delta \ge 2). \tag{3}$$

The first inequality limits new violations. The second is an overlap condition that limits growth of the repair set. A *repair round* draws replacement values for the entire selected mask. We count each

constraint in the repair set once per round, so a set S contributes $|S|$ to the total even when variable scopes overlap. Under these conditions, the expected total number of constraints selected across rounds is $O(m)$, and the expected number of repair rounds is $O(\log m)$ (Guo et al., 2019).

The quadratic dependence on Δ in (3) is close to a hardness barrier: unless NP $=$ RP, no general polynomial-time approximately uniform sampler can be guaranteed using only a bound on $p_\star \Delta^{2-\varepsilon}$ for any fixed $\varepsilon \in (0, 2)$ (see Proposition C.10 in Appendix C.9 for the hardness construction and proof). This does not establish optimality of both PRS conditions, since the hardness result does not include the overlap condition on $r_\star$. Next, we investigate whether similar guarantees hold when a trained model supplies a conditional resampling distribution that adapts to the current mask and visible values. Masked diffusion is a natural candidate due to its masking policy with sampling conditioned on the visible variables.

3 Masked diffusion meets PRS

The masking policy determines which variables to reveal or reconsider. These choices affect generation quality at a fixed sampling budget. For example, confidence-based selection is known to improve task performance (Nie et al., 2025). We use the PRS repair mask to choose the variables to resample. Classical product PRS draws their replacement values independently from fixed one-variable distributions, whereas a masked diffusion model conditions its predictions on the values left visible. It may draw all masked values independently in parallel or form a joint block draw through sequential predictions. Because these conditional predictions can change from one repair to the next, the termination bounds for classical fixed-product PRS need to be extended.

Figure 1 illustrates how we instantiate Algorithm 1. Starting from a complete assignment $X_0 \sim q_0$, PRS constructs the repair set $R_t = \mathrm{Res}(X_t)$ and its mask $B_t = V(R_t)$. The values of X_t outside B_t form the visible, or retained, assignment. They stay fixed while the model supplies replacement values for B_t. The model uses the same resampling rule each round, but its replacement distribution can depend on both the current mask and these retained values. To certify termination, we must therefore control the probability of new violations under this changing conditional distribution. For a mask B, let x_B denote the values of x on B. The retained assignment $z = x_{B^c}$ consists of the unmasked values, held fixed during the current repair round. The model's *resampling kernel* $K_B(z, \cdot)$ is the joint distribution of replacement values for all variables in B, allowing dependence within the block. The model uses this fixed rule with fresh randomness each round. For $S = \mathrm{Res}(x)$ and $B = V(S)$, let Q_x be the distribution of the complete assignment after one repair from x. It combines new values on B with the retained values z outside B. Thus

$$Q_x(y) = K_B(z, y_B)\mathbf{1}\{y_{B^c} = z\}. \tag{4}$$

In parallel resampling, $\nu_{B,z,v}$ is the model's predicted distribution for variable v, and independent draws given B, z yield $K_B(z, u) = \prod_{v \in B} \nu_{B,z,v}(u_v)$. Joint resampling may instead use several internal denoising steps or predict tokens sequentially. Both retain the outside values throughout the round. After stopping, we keep the assignment fixed.

3.1 A termination certificate

A repair can remove existing violations while creating new ones and adding constraints to the next repair set. We extend the probability bounds of the PRS analysis to the learned sampler's conditional distributions. Our criterion makes the expected repair weight decrease, giving almost-sure termination and bounds on expected repair rounds and the total number of constraints selected across rounds. To measure this, assign each constraint a fixed positive weight w_i and write $W(S) = \sum_{i \in S} w_i$. With unit weights, $W(S)$ simply counts the constraints in S.

The same repair set can occur with different retained assignments , so it does not determine a unique resampling distribution. We therefore average over current assignments reached by the actual sampler, conditional on $R_t = S$ at round t. We call S *attainable at round* t when $\mathbb{P}(R_t = S) > 0$. For an event D of completed assignments, such as a violation A_i, its probability after repairing an attainable nonempty S is

$$\overline{Q}_{t,S}(D) = \mathbb{P}(X_{t+1} \in D \mid R_t = S) = \mathbb{E}[Q_{X_t}(D) \mid R_t = S]. \tag{5}$$

Here $Q_{X_t}(D)$ is the probability of D after one repair from the random current assignment. This average can change with t as the distribution of retained assignments changes, even though the model's resampling rule stays fixed. Although an individual repair may enlarge the set, the criterion below requires a decrease after averaging conditional on the current repair set.

To bound the next repair set, we must count new violations and neighbors that PRS selects even while they are satisfied . A satisfied neighbor j can be added after a violation at i when the values on variables shared by i and j agree with some assignment violating j. The joint event $A_i \cap U_{ij}$ captures this possibility. Constraint i is violated, while the shared values are compatible with a violation of j even if j is currently satisfied. More generally, every selected constraint can be traced to a violation through an *induced path* in the dependency graph, whose vertices are distinct and for which only consecutive vertices are adjacent. Each path is a possible route by which a violation enlarges the repair set. Bounding each path probability and adding the endpoint weights will bound the expected weight of the next set. A uniform decrease in that weight gives termination. Let $\mathcal{I}_i$ contain all such paths $P = (i_0, \dots, i_k)$ starting at i, including the length-zero path (i), and write $\mathrm{end}(P) = i_k$ for the endpoint. The corresponding event is

$$D_P = A_{i_0} \cap \bigcap\nolimits_{q=1}^{k} U_{i_{q-1} i_q}. \tag{6}$$

Only the first constraint must be violated. Subsequent constraints need only have overlap values compatible with a violation. After resampling a repair set S, every violation lies in $\Gamma^+(S)$ because constraints outside this neighborhood were satisfied and retain all their values. Thus we need only consider paths whose starting vertex, or *root*, lies in $\Gamma^+(S)$. For every round t and attainable nonempty repair set S, choose bounds

$$\overline{Q}_{t,S}(D_P) \le \beta_{t,S}(P) \quad \text{for } P \in \mathcal{I}_i, \quad i \in \Gamma^+(S). \tag{7}$$

The $\beta_{t,S}(P)$ are probability upper bounds to establish for the chosen sampler. We can calculate the averaged path probability directly or use an upper bound that holds at every assignment the sampler can reach. Below we explain how parallel resampling makes individual path probabilities explicit. Appendix C gives sufficient tests for choosing weights once valid bounds are available. Each bound covers all conditions in D_P jointly, including those involving retained values. The theorem requires the resulting sum of endpoint-weighted path bounds to be at most $\kappa W(S)$ for a fixed $\kappa < 1$. The following theorem turns this expected decrease into a guarantee of termination and a bound on repair rounds. Write $\tau = \inf\{t \mid R_t = \varnothing\}$ for the number of rounds, with $\tau = 0$ when the initial proposal is already feasible.

Theorem 3.1 (Termination of the masked diffusion sampler)**.** *Run Algorithm 1 from q_0 with fixed kernels $K_B(z, \cdot)$, using parallel or joint resampling. Suppose* (7) *holds. If fixed weights $w_i > 0$ and $0 < \kappa < 1$ satisfy, for every round t and attainable nonempty S,*

$$\sum\nolimits_{i \in \Gamma^+(S)} \sum\nolimits_{P \in \mathcal{I}_i} \beta_{t,S}(P) w_{\mathrm{end}(P)} \le \kappa W(S), \qquad W(S) = \sum\nolimits_{i \in S} w_i, \tag{8}$$

then the sampler stops almost surely at a feasible assignment. With $w_{\min} = \min_i w_i$, $L(0) = 0$, and $L(s) = 1 + \log(s/w_{\min})$ for $s \ge w_{\min}$,

$$\mathbb{E}\tau \le \frac{\mathbb{E}L(W(R_0))}{1 - \kappa}. \tag{9}$$

Every constraint in R_{t+1} can be traced along an induced path starting at a violation after the current repair. A union bound over these paths, followed by (8), gives $\mathbb{E}[W(R_{t+1}) \mid R_t] \le \kappa W(R_t)$. Thus the expected repair weight decreases geometrically. With unit weights, this gives

$$\mathbb{E}\tau \le \frac{1 + \log m}{1 - \kappa}, \qquad \mathbb{E}\sum\nolimits_{t<\tau} |R_t| \le \frac{m}{1 - \kappa}.$$

The first bound counts repair rounds, while the second counts each selected constraint once per round, including repeated selections in later rounds. We give the derivations in Appendix A. The weighted version of the second bound appears in Corollary A.6.

3.2 Recovery of classical PRS

We recover standard PRS as a special case by applying our certificate to a fixed product distribution. In this setting, we can verify the path condition in Theorem 3.1 using the unconditional violation and overlap bounds introduced in Section 2.

Corollary 3.2 (Recovery of classical PRS). *Let $\mu = \bigotimes_{v=1}^{n} \mu_v$ be a fixed product distribution. Initialize Algorithm 1 with $X_0 \sim \mu$ and use $K_B(z, u) = \prod_{v \in B} \mu_v(u_v)$ for every mask B and retained assignment z. If $\Delta \geq 2$ and the probabilities $p_\star = \max_i \mu(A_i)$ and $r_\star = \max_{i \sim j} \mu(U_{ij})$ satisfy* (3)*, then the certificate* (8) *holds with unit weights and $\kappa = 1/2$. The sampler terminates almost surely, outputs exactly $\mu(\cdot \mid \mathcal{F})$, and uses at most $2(1 + \log m)$ expected rounds and $2m$ expected constraint resamplings.*

The proof in Appendix C.4 converts the product-PRS bounds into our path certificate. Taking μ to be uniform recovers uniform sampling from feasible assignments.

4 Sampling the data distribution conditioned on feasibility

Section 3 addresses how quickly repair reaches feasibility. We now turn to the second challenge, ensuring that feasible assignments appear with the probabilities prescribed by the data distribution conditioned on feasibility. Our target is

$$p_{\mathcal{F}}(x) = p_{\mathrm{data}}(x \mid \mathcal{F}) = \frac{p_{\mathrm{data}}(x)\mathbf{1}\{x \in \mathcal{F}\}}{p_{\mathrm{data}}(\mathcal{F})}. \tag{10}$$

Write $p = p_{\mathrm{data}}$ and assume $p(\mathcal{F}) > 0$. Conditioning keeps the relative probabilities of feasible assignments. For uniform p, the target $p_{\mathcal{F}}$ is uniform on $\mathcal{F}$. We measure output error in total variation (TV), $\mathrm{d_{TV}}(q, p_{\mathcal{F}}) = \sup_E |q(E) - p_{\mathcal{F}}(E)|$. We first correct joint conditional repair by an acceptance test and restarts to sample $p_{\mathcal{F}}$ exactly. We then bound the error of learned repair without this correction, assuming its ideal counterpart returns $p_{\mathcal{F}}$.

4.1 Exact sampling with polynomial restart overhead

As discussed in Section 2.1, sequential exact conditionals recover the joint block conditional $p(X_B \mid X_{B^c})$ by the chain rule. Independent parallel draws recover it only when the variables in B are conditionally independent given the retained values. Even exact joint resampling can bias the final output because the repair set is selected from the current assignment (see Appendix D.1 for a two-bit example with biased feasible outputs). The corrected sampler starts with $X_0 \sim p$. Before each repair, it either accepts the repair or discards the entire current assignment and starts a new attempt from an independent draw from p. The new attempt computes its repair set from that fresh assignment.

For a selected repair set S, write $B = V(S)$, $u = X_B$ for the values about to be masked, and $z = X_{B^c}$ for the retained values. Choose $h_S(u) \geq 0$, independent of z, with $h_S(u) \leq p(u \mid z)$ for every retained assignment z with $p(X_{B^c} = z) > 0$. Accept the repair with probability

$$p_{\mathrm{accept}}(u, z; S) = \frac{h_S(u)}{p(u \mid z)}. \tag{11}$$

On acceptance, jointly resample B from $p(\cdot \mid z)$. On rejection, discard the current assignment and start afresh from p. The ratio cancels the old block's dependence on the retained values, yielding exact output with the PRS repair-set rule.

Theorem 4.1 (Exact sampling with polynomial restart overhead). *Assume $p(\mathcal{F}) > 0$. Suppose we use the corrected sampler described above, with $p_{\mathrm{accept}} > \alpha$ at every attainable repair for a fixed constant $\alpha \in (0, 1)$. Suppose its uncorrected version satisfies the certificate of Theorem 3.1, giving $\mathbb{E}\tau \leq L$. The corrected sampler terminates almost surely and outputs exactly $p_{\mathcal{F}}$. If N counts fresh starts, including initialization, and T counts all repair-set checks, including the final check, then*

$$\mathbb{E}N \leq \alpha^{-L}, \qquad \mathbb{E}T \leq (L + 1)\alpha^{-L}. \tag{12}$$

For $L = O(\log m)$, the restart bounds give polynomial expected overhead in m. Polynomial expected total time also requires polynomially bounded costs per operation. The proof can be found in Appendix D.5.

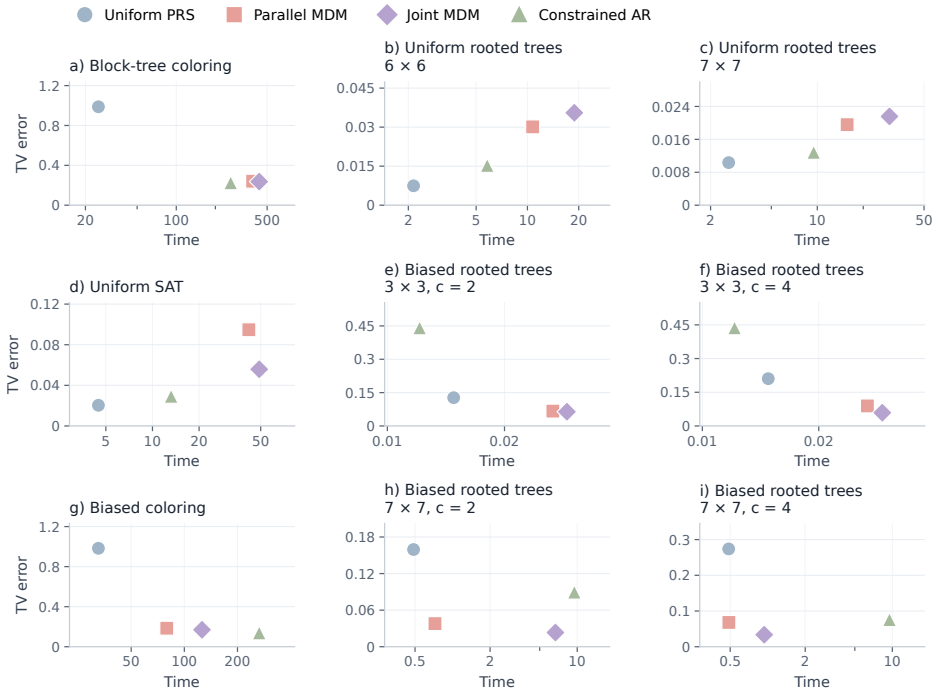

Figure 2: Distributional fidelity and sampling time across nine configurations. The horizontal axis shows time per 10,000 requested outputs on a logarithmic scale. Each panel uses the TV statistic and aggregation specified in Appendix G.9. Lower values on both axes indicate better performance.

Distributional accuracy of learned repair. In Appendix D.4, we provide a complementary analysis for learned repair without the acceptance correction, allowing imperfect initialization and resampling. If the learned sampler meets our termination condition and an ideal version using exact joint data conditionals returns $p_{\mathcal{F}}$, we bound the learned sampler's output error by its initialization error and the resampling errors accumulated during repair.

5 Experiments

5.1 Distributional fidelity across sampling tasks

We compare sampling methods on nine configurations spanning uniform SAT, biased hypergraph coloring, and uniform and biased rooted spanning trees. Each configuration is evaluated locally with Uniform PRS, parallel MDM, joint MDM, and constrained AR. Constrained AR is a standard autoregressive transformer that generates variables sequentially using feasibility masks while the MDM models are implemented with a PRS masking policy. Appendix G specifies the models, training, initialization, TV statistics, and timing for each dataset.

As expected, uniform PRS achieves the lowest measured TV and shortest time on the uniform targets. Both MDM methods reduce TV relative to Uniform PRS on every biased target, and joint MDM further reduces TV relative to parallel MDM. On the 32-step biased-coloring comparison, constrained AR has the lowest measured TV at 0.138, while parallel MDM takes 79.5 seconds per 10,000 requests compared with 265.5 seconds for constrained AR.

Acceptance on biased distributions. We also test the acceptance-and-restart rule in Section 4 on biased sink-free SAT and biased rooted spanning trees. Table 8 in the appendix reports empirical TV together with its matched finite-sample reference. See Appendix H for the detailed experimental setup.

5.2 RANDOM 3-SAT WITHOUT AN EVALUATED CERTIFICATE

PRS-based certificates focus on sparse regimes. We now ask whether the learning-enhanced PRS repair process remains useful far outside them, on near-threshold random 3-SAT. We keep Algorithm 1 and the full PRS repair mask fixed and change only the resampling kernel: fair bits (uniform PRS), or a uniform-noise (DiffUCO (Sanokowski et al., 2024)) or masked (MDNS (Zhu et al., 2025)) diffusion model, pretrained without solutions and fine-tuned on runs of the repair loop. Each round resamples the masked variables jointly with 16 denoising steps, conditioned on the retained values. We train on generated formulas with $n = 100$ and test on the 1000 SATLIB uf100 formulas (Appendix F.4).

Table 1: Solved SATLIB uf100 formulas (%), solved@1 / solved@10, after 1, 4, 16 and 64 rounds; round 1 is the initial draw. PRS-mask repair resamples the PRS repair mask with a fine-tuned uniform (DiffUCO) or masked (MDNS) diffusion kernel, using 16 denoising steps per round. Diffusion models without repair generate once. *COExpander supports at most 5 inference steps.

Method	1	4	16	64
Diffusion based				
PRS-mask repair (uniform)	43.4 / 76.5	63.7 / 87.3	80.5 / 94.2	88.9 / 96.9
PRS-mask repair (masked)	0.7 / 4.6	1.2 / 9.6	4.4 / 20.1	11.9 / 39.6
COExpander* (Ma et al., 2025)	50 / 89	–	–	–
DiffUCO (Sanokowski et al., 2024)	44.3 / 75.7	–	–	–
MDNS (Zhu et al., 2025)	0.1 / 2.4	–	–	–
Other baselines				
Uniform PRS (Guo et al., 2019)	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
WalkSAT (Selman et al., 1994)	2.8 / 23.2	19.8 / 72.2	55.7 / 94.9	85.5 / 99.7
probSAT (Balint & Schöning, 2012)	1.8 / 16.8	19.7 / 70.2	54.5 / 96.1	86.1 / 99.7

Results. A learned kernel turns PRS from a sampler that would not terminate in this regime into a competitive solver (Table 1). With fair bits, uniform PRS solves no formula even after 1024 rounds; with the same repair mask, the DiffUCO kernel solves 96.9% within 64 rounds, above every one-shot diffusion solver, including COExpander (89%). Repair, not the initial draw, drives the gain: from round 1 to 64, solved@10 rises from 76.5 to 96.9% for DiffUCO and from 4.6 to 39.6% for MDNS. In a single run, PRS-mask repair solves more formulas than WalkSAT and probSAT at every round budget, though rounds differ in cost. Trained at $n = 100$, the DiffUCO kernel still solves 86% of uf250 formulas versus 69% for COExpander (Table 7); Table 6 isolates the gain from fine-tuning.

Limitations. Our enumerable experiments do not show that exact certificate checks scale to large models, and the SATLIB study measures feasibility rather than the output distribution.

6 CONCLUSION

We view masked diffusion as a local repair process guided by the masking policy of partial rejection sampling. Through this connection, we give conditions for efficient repair and identify additional requirements for sampling feasible assignments from a target combinatorial distribution on the Boolean hypercube. For future work, it would be interesting to extend this analysis to soft constraints, building on existing PRS-style methods for graphical models (Jerrum, 2024; Feng et al., 2021), and to settings such as language generation, where the constraints may be implicit or not directly available.

AI USE STATEMENT

Generative AI tools (OpenAI Codex and Anthropic Claude Code) assisted with theoretical analysis, proof development and critique, literature search, manuscript revision, and implementation of the synthetic training experiments, finite-state verification, and document build tools. We executed all reported training runs and numerical checks programmatically. AI critiques are not human review or formal proof verification. The authors retain responsibility for the final submission.

REPRODUCIBILITY STATEMENT

We provide proofs, the full synthetic data distributions, and the training and verification procedures in the appendices. Appendix F reports three training seeds per small enumerable distribution. The supplementary code includes the frequency of each training example, saved model parameters, and exact calculations of the output probabilities for the samplers as implemented, including their probability rounding. Separate checks over all assignments in small examples complement the general proofs. For SATLIB, we specify the benchmark, training procedure, evaluation budgets, and baseline comparisons in Appendix F.4. Section 5.2 reports one training seed per kernel. The source archive includes commands to build the paper and run the checks.

ETHICS STATEMENT

The experiments use synthetic distributions and contain no human-subject data. Feasible outputs are not necessarily meaningful, fair, or safe. Conditioning preserves their relative probabilities in the data and may retain existing biases.

REFERENCES

Dimitris Achlioptas. A sampling Lovász Local Lemma, 2026. URL https://arxiv.org/abs/2609.18712. arXiv preprint.

Dimitris Achlioptas, Fotis Iliopoulos, and Vladimir Kolmogorov. A local lemma for focused stochastic algorithms. *SIAM Journal on Computing*, 48(5):1583–1602, 2019. doi: 10.1137/16M109332X. URL https://arxiv.org/abs/1809.01537.

Adrian Balint and Uwe Schöning. Choosing probability distributions for stochastic local search and the role of make versus break. In *Theory and Applications of Satisfiability Testing – SAT 2012*, volume 7317 of *Lecture Notes in Computer Science*, pp. 16–29, 2012. doi: 10.1007/978-3-642-31612-8_3. URL https://doi.org/10.1007/978-3-642-31612-8_3.

Vansh Bansal, Cho Cholyeon, Syamantak Kumar, Sujay Sanghavi, and Purnamrita Sarkar. Understanding parallel samplers in masked diffusion via random walks on graphs, 2026. URL https://arxiv.org/abs/2606.22976. arXiv preprint.

Alankrita Bhatt, Mukur Gupta, Germain Kolossov, and Andrea Montanari. Generating from discrete distributions using diffusions: Insights from random constraint satisfaction problems, 2026. URL https://arxiv.org/abs/2603.20589v2. Version 2.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete Gaussian for differential privacy. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html.

Michael Cardei, Jacob K. Christopher, Bhavya Kailkhura, Tom Hartvigsen, and Ferdinando Fioretto. Constrained discrete diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 12383–12416, 2025. doi: 10.52202/085713-0415. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/124c1cc9f7750f512ed81cdbf2fd5bf5-Abstract-Conference.html.

Hongrui Chen and Lexing Ying. Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *Journal of Machine Learning*, 4(2):108–127, 2025. doi: 10.4208/jml.240812. URL https://doi.org/10.4208/jml.240812.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: Theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.11215.

Sitan Chen, Kevin Cong, and Jerry Li. Optimal inference schedules for masked diffusion models. In *Proceedings of the Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 1279–1311, 2026. URL https://proceedings.mlr.press/v336/chen26e.html.

Meihua Dang and Stefano Ermon. Constrained decoding for diffusion language models via efficient inference over finite automata, 2026. URL https://arxiv.org/abs/2607.07026.

Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and Finite Sets, Vol. II*, volume 10 of *Colloquia Mathematica Societatis János Bolyai*, pp. 609–627. North-Holland, Amsterdam, 1975. URL https://www.renyi.hu/~p_erdos/1975-34.pdf.

Shengyu Feng and Yiming Yang. Regularized Langevin dynamics for combinatorial optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025. URL https://openreview.net/forum?id=bbJ0QCujU4.

Weiming Feng, Nisheeth K. Vishnoi, and Yitong Yin. Dynamic sampling from graphical models. *SIAM Journal on Computing*, 50(2):350–381, 2021. doi: 10.1137/20M1315099. URL https://doi.org/10.1137/20M1315099.

Weiming Feng, Heng Guo, and Yitong Yin. Perfect sampling from spatial mixing. *Random Structures & Algorithms*, 61(4):678–709, 2022. doi: 10.1002/rsa.21079. URL https://doi.org/10.1002/rsa.21079.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. Scalable verified training for provably robust image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4842–4851, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Gowal_Scalable_Verified_Training_for_Provably_Robust_Image_Classification_ICCV_2019_paper.html.

Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovász Local Lemma. *Journal of the ACM*, 66(3):18:1–18:31, 2019. doi: 10.1145/3310131. URL https://doi.org/10.1145/3310131.

Wei Guo, Jaemoo Choi, Yuchen Zhu, Molei Tao, and Yongxin Chen. Proximal diffusion neural sampler. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3a0e2de215bd17c39ad08ba1d16c1b12-Abstract-Conference.html.

Holger H. Hoos and Thomas Stützle. SATLIB: An online resource for research on SAT. In Ian P. Gent, Hans van Maaren, and Toby Walsh (eds.), *SAT 2000*, pp. 283–292. IOS Press, 2000. URL https://www.cs.ubc.ca/~hoos/Publ/sat2000-satlib.pdf.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4083–4093, 2019. doi: 10.18653/v1/D19-1419. URL https://aclanthology.org/D19-1419/.

Vishesh Jain, Clayton Mizgerd, and Huy Tuan Pham. A general counting and sampling Lovász Local Lemma, 2026. URL https://arxiv.org/abs/2609.23671. arXiv preprint.

Mark Jerrum. Fundamentals of partial rejection sampling. *Probability Surveys*, 21:171–199, 2024. doi: 10.1214/24-PS29. URL https://doi.org/10.1214/24-PS29.

Nan Jiang, Yi Gu, and Yexiang Xue. Learning Markov random fields for combinatorial structures via sampling through Lovász Local Lemma. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4016–4024, 2023. doi: 10.1609/aaai.v37i4.25516. URL https://ojs.aaai.org/index.php/AAAI/article/view/25516.

Taegu Kang, Jaesik Yoon, and Sungjin Ahn. Inference-time scaling in diffusion models through iterative partial refinement, 2026. URL https://arxiv.org/abs/2605.19317.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.01384.

Maximilian J. Kramer, Paul Boes, and Jens Eisert. Using deep learning to construct stochastic local search SAT solvers with performance bounds. *Machine Learning: Science and Technology*, 7:025057, 2026. doi: 10.1088/2632-2153/ae5b23. URL https://doi.org/10.1088/2632-2153/ae5b23.

Sanghyun Lee, Sunwoo Kim, Seungryong Kim, Jongho Park, and Dongmin Park. Effective test-time scaling of discrete diffusion through iterative refinement, 2025a. URL https://arxiv.org/abs/2511.05562.

Vint Lee, Minh Nguyen, Leena Elzeiny, Chun Deng, Pieter Abbeel, and John Wawrzynek. Chip placement with diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33499–33514. PMLR, 2025b. URL https://proceedings.mlr.press/v267/lee25y.html.

Yuchen Liang, Zhiheng Tan, Ness Shroff, and Yingbin Liang. Sharp convergence rates for masked diffusion models, 2026. URL https://arxiv.org/abs/2602.22505. arXiv preprint.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. Constrained graph variational autoencoders for molecule design. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/b8a03c5c15fcfa8dae0b03351eb1742f-Abstract.html.

Jiale Ma, Wenzheng Pan, Yang Li, and Junchi Yan. COExpander: Adaptive solution expansion for combinatorial optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42130–42164, 2025. URL https://proceedings.mlr.press/v267/ma25r.html.

Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM*, 57(2):11:1–11:15, 2010. doi: 10.1145/1667053.1667060. URL https://arxiv.org/abs/0903.0544.

Niels Mündler, Jasper Dekoninck, and Martin Vechev. Constrained decoding of diffusion LLMs with context-free grammars. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.10111.

Robert R. Nerem, Samantha Chen, Sanjoy Dasgupta, and Yusu Wang. Graph neural networks extrapolate out-of-distribution for shortest paths. In *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 5273–5331. PMLR, 2026. URL https://proceedings.mlr.press/v336/nerem26a.html.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2502.09992.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Paweł Parys, Sairam Vaidya, Taylor Berg-Kirkpatrick, and Loris D'Antoni. Constrained adaptive rejection sampling, 2025. URL https://arxiv.org/abs/2510.01902.

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, 2024. URL https://arxiv.org/abs/2406.07524.

Sebastian Sanokowski, Sepp Hochreiter, and Sebastian Lehner. A diffusion model framework for unsupervised neural combinatorial optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024. URL https://arxiv.org/abs/2406.01661.

Sebastian Sanokowski, Wilhelm Berghammer, Haoyu Wang, Martin Ennemoser, Sepp Hochreiter, and Sebastian Lehner. Scalable discrete diffusion samplers: Combinatorial optimization and statistical physics. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d87eadbe6114b768e76c5d5e8eb39388-Abstract-Conference.html.

Sakari Seitz, Mikko Alava, and Pekka Orponen. Focused local search for random 3-satisfiability. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(06):P06006, 2005. doi: 10.1088/1742-5468/2005/06/P06006. URL https://doi.org/10.1088/1742-5468/2005/06/P06006.

Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 337–343, 1994. URL https://cdn.aaai.org/AAAI/1994/AAAI94-051.pdf.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.04329.

Allan Sly and Nike Sun. Counting in two-spin models on d-regular graphs. *The Annals of Probability*, 42(6):2383–2416, 2014. doi: 10.1214/13-AOP888. URL https://doi.org/10.1214/13-AOP888.

Tarun Suresh, Debangshu Banerjee, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. DINGO: Constrained inference for diffusion LLMs. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23061.

Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, 2021. doi: 10.1016/j.patter.2021.100273. URL https://doi.org/10.1016/j.patter.2021.100273.

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1910.10593.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2503.00307.

Solveig Wittig, Antonis Vasileiou, Robert R. Nerem, Timo Stoll, Floris Geerts, Yusu Wang, and Christopher Morris. Which algorithms can graph neural networks learn? In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.13106.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJxbJeHFPS.

Morris Yau, Nikolaos Karalias, Eric Lu, Jessica Xu, and Stefanie Jegelka. Are graph neural networks optimal approximation algorithms? In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/85db52cc08c5e00cfb1d216b1c85ba35-Abstract-Conference.html.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. MDNS: Masked diffusion neural sampler via stochastic optimal control. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=xIH95kXNR2.

Algorithm 2 Selecting the repair set for the PRS repair mask

```
R ← Bad(x); N ← ∅
while Γ(R) \ N ≠ ∅ do
    U ← Γ(R) \ N                                   ▷ Freeze this batch
    for j ∈ U in the fixed order do
        if some y ∈ A_j agrees with x on V_j ∩ V(R) then
            R ← R ∪ {j}
        else
            N ← N ∪ {j}
        end if
    end for
end while
return R
```

A MEMORYLESS RESAMPLING WITH THE PRS REPAIR MASK

We first prove that resampling with the PRS repair mask terminates under the stated probability bounds. The model uses a fixed resampling rule that depends only on the current mask and retained values, with fresh randomness in each round. Its sampled coordinates may be dependent. We then apply the result to masked diffusion to obtain Theorem 3.1 and the separate bound on cumulative repair-set weight in Corollary A.6. The repair-mask construction follows the general partial-rejection sampling construction of Guo et al. (2019). The main certificate conditions only on the current repair set. It averages over the actual assignments that produce that set at the given round. The sampler still uses its actual visible coordinates in each execution.

A.1 CONSTRUCTING THE PRS REPAIR MASK

Let $n, m \geq 1$ and $\Omega = \prod_{v=1}^{n} \Sigma_v$, where every alphabet is finite and nonempty, and let $A_1, \ldots, A_m$ be bad events. Event A_i is determined by its scope $V_i \subseteq [n]$. Thus membership in A_i cannot change without changing a coordinate in V_i. Write

$$\mathrm{Bad}(x) = \{i : x \in A_i\}, \qquad \mathcal{F} = \bigcap_{i=1}^{m} A_i^c, \qquad V(S) = \bigcup_{i \in S} V_i.$$

The dependency graph joins distinct i, j when $V_i \cap V_j \neq \varnothing$. Its neighborhoods are

$$\Gamma(i) = \{j : j \sim i\}, \qquad \Gamma^+(i) = \Gamma(i) \cup \{i\}, \qquad \Gamma(S) = \left(\bigcup_{i \in S} \Gamma(i)\right) \setminus S, \qquad \Gamma^+(S) = S \cup \Gamma(S).$$

For a fixed assignment x, consider the coordinates that A_j shares with the events in S, namely $V_j \cap V(S)$. We call these coordinates its interface with S. The event A_j is compatible with this interface if some $y \in A_j$ agrees with x there. It is blocked otherwise: the shared values already rule it out. If an event is blocked at S, it remains blocked when S grows, because fixing more coordinates cannot make a previously impossible completion possible.

Fix an order in which to inspect event labels. The repair-set selection procedure works in batches and keeps two disjoint sets: included labels and blocked labels. Start with $R^{(0)} = \mathrm{Bad}(x)$ and $N^{(0)} = \varnothing$. At batch q, freeze

$$U^{(q)} = \Gamma(R^{(q)}) \setminus N^{(q)}.$$

This is the list to inspect in batch q, and it stays fixed until the batch ends. If the list is empty, stop. Otherwise initialize temporary sets $R = R^{(q)}$ and $N = N^{(q)}$, and inspect each $j \in U^{(q)}$ in the fixed order. Test compatibility against the *current* temporary R. If compatible, add j to R. Otherwise, add it to N. After the batch, set $R^{(q+1)} = R$ and $N^{(q+1)} = N$. The assignment x does not change during this construction. The final included set is the PRS repair set $\mathrm{Res}(x)$. Its variables $V(\mathrm{Res}(x))$ form the PRS repair mask. We record when each event was included using its addition depth $d_x(j)$. An initially bad event has depth zero, and an event first included in batch q has depth $q + 1$.

Lemma A.1 (Properties of the PRS repair set)**.** *Every label considered during repair-set selection is inspected at most once, so the procedure makes at most m inspections. Moreover,*

$$\mathrm{Bad}(x) \subseteq \mathrm{Res}(x), \qquad \mathrm{Res}(x) = \varnothing \iff x \in \mathcal{F},$$

and the shared coordinates block every event in $\Gamma(\mathrm{Res}(x))$.

Proof. An inspected label is permanently placed in R or N. It cannot be inspected again because included labels are outside $\Gamma(R)$ and blocked labels are excluded from subsequent candidate sets. At termination every label adjacent to R but outside it belongs to N. Since a blocked event stays blocked as R grows, all these events are still blocked at termination. Initially bad events are never removed. If there are no initially bad events, both $R^{(0)}$ and its boundary are empty, so the algorithm stops with an empty repair set. □

For an ordered adjacent pair define the overlap event

$$U_{ij} = \{x : \text{some } y \in A_j \text{ satisfies } y_{V_i \cap V_j} = x_{V_i \cap V_j}\}. \tag{13}$$

This event depends only on $V_i \cap V_j$. It says that the current shared values do not rule out A_j. Reversing i and j can give a different event. An induced path $P = (i_0, \ldots, i_\ell)$ has distinct vertices, adjacent consecutive vertices, and nonadjacent nonconsecutive vertices. Length-zero paths are included. Write $\mathrm{end}(P) = i_\ell$ and let $\mathcal{I}_i$ denote the finite set of induced paths starting at i. For this path, define

$$D_P = A_{i_0} \cap \bigcap_{q=1}^{\ell} U_{i_{q-1} i_q}. \tag{14}$$

The event D_P requires the first event to be bad and, at each step along the path, the shared values to allow the next bad event. We call it a witness event because the next lemma shows that every event included in the repair set can be accounted for by such a path.

Lemma A.2 (A path for each event in the repair set)**.** *If $j \in \mathrm{Res}(x)$ has addition depth ℓ, there is an induced path $P = (i_0, \ldots, i_\ell)$ ending at j, with $d_x(i_q) = q$ for every q, such that $x \in D_P$.*

Proof. For $\ell = 0$, take $P = (j)$, since j is initially bad. Suppose $\ell \geq 1$. The label j was in the frozen candidate set for batch $\ell - 1$. It has no neighbor in $R^{(\ell-2)}$, interpreting $R^{(-1)} = \varnothing$. Otherwise it would have been a boundary candidate in the preceding batch. Since it is eventually included, it cannot have been permanently blocked earlier. That preceding inspection would therefore contradict its addition depth.

Consequently j has a neighbor $i_{\ell-1}$ in $R^{(\ell-1)} \setminus R^{(\ell-2)}$. Apply the induction hypothesis to that neighbor. Every earlier vertex of the resulting path belongs to $R^{(\ell-2)}$ and hence is nonadjacent to j. Appending j preserves the property that no nonconsecutive vertices are adjacent, and gives the stated depths. At the inspection of j, the current included set contains $i_{\ell-1}$. Compatibility of A_j with that set implies $x \in U_{i_{\ell-1} j}$. Together with the inductive witness, this proves $x \in D_P$. □

The lemma says that at least one such path exists for each event in the repair set. The algorithm might not explicitly select that path. The first event on the path is the only one required to be bad under x. Later events need only satisfy the shared-coordinate conditions.

A.2 Memoryless resampling and the one-round bound

Let X_0 have a specified distribution q_0 on Ω. Set

$$R_t = \mathrm{Res}(X_t), \qquad B(x) = V(\mathrm{Res}(x)), \qquad B_t = B(X_t), \qquad \tau = \inf\{t \geq 0 : R_t = \varnothing\}.$$

For each mask B and retained assignment $z \in \Omega_{B^c}$, fix a probability distribution $K_B(z, \cdot)$ on Ω_B. Before termination, the sampler draws the entire block from $K_{B_t}(X_{t,B_t^c}, \cdot)$, using fresh randomness, and copies the coordinates outside B_t. The resulting transition distribution from a current assignment x is

$$Q_x(y) = \mathbf{1}\{y_{B(x)^c} = x_{B(x)^c}\} K_{B(x)}(x_{B(x)^c}, y_{B(x)}). \tag{15}$$

The kernel depends only on the current mask and visible coordinates. It does not use previous assignments, previous repair sets, or the round number. The newly sampled coordinates need not be independent. After termination, keep the assignment fixed, so Q_x is the point mass at x when $x \in \mathcal{F}$.

For every $t \geq 0$ and nonempty repair set S with $\mathbb{P}(R_t = S) > 0$, define

$$\overline{Q}_{t,S}(D) = \mathbb{P}(X_{t+1} \in D \mid R_t = S) = \mathbb{E}[Q_{X_t}(D) \mid R_t = S]. \tag{16}$$

This averages over the conditional distribution of X_t given its current repair set. That distribution can change with t even though the kernel is fixed.

These probabilities can be calculated from the distribution of the current assignment. Writing $q_t(x) = \mathbb{P}(X_t = x)$, finite-state propagation gives

$$q_{t+1}(y) = \sum_{x \in \Omega} q_t(x) Q_x(y), \tag{17}$$

$$\overline{Q}_{t,S}(D) = \frac{\displaystyle\sum_{x:\mathrm{Res}(x)=S} q_t(x) Q_x(D)}{\displaystyle\sum_{x:\mathrm{Res}(x)=S} q_t(x)}. \tag{18}$$

The denominator is positive for the sets under consideration. Exact propagation can be expensive when Ω is large. A proved invariant or valid pointwise bounds, developed below, can avoid this calculation.

Which repair sets must satisfy the bounds? At every round, impose the bounds for every nonempty repair set that has positive probability under the specified initialization and fixed kernel. Both the state space and the family of repair sets are finite. The stronger pointwise checks below need hold only at nonterminal assignments that have positive probability at some round. They may also be checked on a larger set of assignments. A certificate for one initialization need not hold for another, because the conditional distribution of retained values can change.

For each such pair (t, S), choose nonnegative bounds

$$\overline{Q}_{t,S}(D_P) \leq \beta_{t,S}(P) \quad \text{for paths rooted in } \Gamma^+(S). \tag{19}$$

Bounds larger than one may be replaced by one. For fixed positive weights $w = (w_i)_{i=1}^m$, define

$$W(S) = \sum_{i \in S} w_i, \qquad w_{\min} = \min_i w_i, \qquad G_{t,S,i}(w) = \sum_{P \in \mathcal{I}_i} \beta_{t,S}(P) w_{\mathrm{end}(P)}.$$

The certificate requires, for one fixed $0 < \kappa < 1$,

$$\sum_{i \in \Gamma^+(S)} G_{t,S,i}(w) \leq \kappa W(S) \quad \text{whenever } S \neq \varnothing \text{ and } \mathbb{P}(R_t = S) > 0. \tag{20}$$

Each root in the union $\Gamma^+(S)$ is counted once. The weights and contraction factor stay fixed, but the path bounds may depend on the round and current repair set.

Theorem A.3 (Memoryless resampling with the PRS repair mask)**.** *Consider the finite bad-event system and PRS repair-set rule defined above. From a specified initial distribution q_0 on Ω, set $R_t = \mathrm{Res}(X_t)$ and $\tau = \inf\{t : R_t = \varnothing\}$. At every nonterminal assignment x, use the transition distribution Q_x from* (15)*. The fixed block kernel may sample dependent coordinates.*

Suppose the current-repair-set path bounds in (19) *and the aggregate certificate* (20) *hold for fixed $w_i > 0$ and $0 < \kappa < 1$. Define $W(S) = \sum_{i \in S} w_i$, $w_{\min} = \min_i w_i$, and $L(0) = 0$, $L(s) = 1 + \log(s/w_{\min})$ for $s \geq w_{\min}$. Then $\tau < \infty$ and $X_\tau \in \mathcal{F}$ almost surely, and*

$$\mathbb{E}\left[\sum_{t<\tau} W(R_t)\right] \leq \frac{\mathbb{E}W(R_0)}{1-\kappa}, \qquad \mathbb{E}\tau \leq \frac{\mathbb{E}L(W(R_0))}{1-\kappa}. \tag{21}$$

The theorem concerns the resampling distributions actually used by the sampler. It applies beyond diffusion models: neither the initial coordinates nor the resampled coordinates need to be independent, and the resampling distribution may change with the current visible values and mask. The next lemma bounds the expected size of the next repair set. Repeating that bound proves termination.

Lemma A.4 (Where new violations can occur and how much repair they require)**.** *For every $t \geq 0$,*

$$\mathrm{Bad}(X_{t+1}) \subseteq \Gamma^+(R_t), \tag{22}$$

$$\mathbb{E}[W(R_{t+1}) \mid R_t] \leq \sum_{i \in \Gamma^+(R_t)} G_{t,R_t,i}(w). \tag{23}$$

Consequently, (20) *gives*

$$\mathbb{E}[W(R_{t+1}) \mid R_t] \leq \kappa W(R_t). \tag{24}$$

Proof. If $i \notin \Gamma^+(R_t)$, its scope is disjoint from the mask. It was not bad before the step, because $\mathrm{Bad}(X_t) \subseteq R_t$, and none of its coordinates changes. This proves (22) for every execution, without any probabilistic conditioning. Put $Y = X_{t+1}$. By Lemma A.2, each member of $\mathrm{Res}(Y)$ is accounted for by an induced path starting in $\mathrm{Bad}(Y) \subseteq \Gamma^+(R_t)$. Thus, on every execution,

$$W(\mathrm{Res}(Y)) \leq \sum_{i \in \Gamma^+(R_t)} \sum_{P \in \mathcal{I}_i} \mathbf{1}\{Y \in D_P\} w_{\mathrm{end}(P)}.$$

The root set is determined by R_t. Taking expectations conditional on R_t and using (19) proves (23). The aggregate certificate gives (24). After termination all repair sets are empty. Neither independence of coordinates under $\overline{Q}_{t,S}$ nor independence between rounds is required. In particular, some assignments with the same current repair set may have positive expected growth even though their average contracts. The repair-set process itself need not be Markov. □

A.3 Proof of the general repair theorem

Proof of Theorem A.3. Let $W_t = W(R_t)$. Taking expectations in (24) and iterating gives

$$\mathbb{E}W_t \leq \kappa^t \mathbb{E}W_0. \tag{25}$$

After termination $W_t = 0$. We may therefore sum over all t, and interchange the expectation and sum because the terms are nonnegative (Tonelli's theorem):

$$\mathbb{E}\left[\sum_{t<\tau} W_t\right] = \sum_{t \geq 0} \mathbb{E}W_t \leq \frac{\mathbb{E}W_0}{1-\kappa}. \tag{26}$$

Every round before termination has $W_t \geq w_{\min} > 0$. A run that never terminates would therefore have infinite cumulative repair-set weight. The finite expectation in (26) rules this out except on a set of probability zero. Thus $\tau < \infty$ almost surely. Lemma A.1 gives $X_\tau \in \mathcal{F}$.

To obtain the sharper bound on the number of rounds, apply the function L from the theorem to the remaining weight. For $s \geq w_{\min}$ and another attainable value r, one has

$$L(s) - L(r) \geq \frac{s-r}{s}. \tag{27}$$

For $r > 0$, this is $\log(s/r) \geq 1 - r/s$, which follows from $\log u \leq u - 1$. For $r = 0$, it follows from $L(s) \geq 1$. Conditioning on R_t and using (24) yields

$$\mathbb{E}[L(W_t) - L(W_{t+1}) \mid R_t] \geq (1-\kappa)\mathbf{1}\{t < \tau\}.$$

Sum this inequality for $t = 0, \ldots, T-1$, take expectations, and use $L(W_T) \geq 0$. Letting $T \to \infty$ gives

$$\mathbb{E}\tau \leq \frac{\mathbb{E}\left[\mathbf{1}\{W_0 > 0\}\big(1 + \log(W_0/w_{\min})\big)\right]}{1-\kappa}. \tag{28}$$

The integrand is defined to be zero when $W_0 = 0$. We never evaluate the logarithm of zero. The preceding sums are well defined for each finite T: W_t can take only the finitely many values obtained by adding weights from a subset of the events. All sums are over deterministic times, so this proof does not require a Markov property for R_t or an inequality conditioned on previous repair sets. Finally $1 + \log(s/w_{\min}) \leq s/w_{\min}$ for $s \geq w_{\min}$, so (28) also gives the simpler bound $\mathbb{E}\tau \leq \mathbb{E}W_0/[w_{\min}(1-\kappa)]$. □

A.4 MASKED DIFFUSION AS A COROLLARY

The masked diffusion instance of Algorithm 1 obtains a complete initial proposal and then repeatedly constructs the PRS repair mask to choose which coordinates to resample. Before each resampling step it computes $R_t = \mathrm{Res}(X_t)$ and masks exactly $B_t = V(R_t)$. It then uses the model to resample the masked coordinates while copying the visible coordinates. The resampling procedure may sample tokens independently in parallel, or may use several model calls to sample dependent tokens. The chosen repair mask remains fixed until the round is complete. The fixed block kernel $K_B(z, \cdot)$ is the distribution produced by the entire model procedure for a given mask B and visible assignment z. It includes all random choices within the round. Any diffusion time schedule or sampling settings are fixed parts of that procedure. The procedure uses fresh randomness each time it is invoked. The completed assignment has distribution Q_x from (15). The certificate averages Q_{X_t} conditional on R_t, as in (16).

Corollary A.5 (Termination of masked diffusion repair)**.** *Suppose complete masked diffusion rounds use fixed block kernels $K_B(z, \cdot)$ with fresh randomness and copy all coordinates outside the mask. Suppose their current-repair-set path probabilities satisfy* (19) *and some fixed positive weights and $0 < \kappa < 1$ satisfy* (20)*. Then the method terminates almost surely at a feasible assignment, and its number of resampling rounds satisfies*

$$\mathbb{E}\tau \le \frac{\mathbb{E}L(W(R_0))}{1-\kappa}$$

for the initial proposal distribution under which these averaged bounds hold. Both parallel and joint model resampling are covered.

Derivation of Theorem 3.1. At a current assignment x, the complete masked diffusion round has distribution Q_x. In a parallel resampling step, write $\nu_{x,v} = \nu_{B(x),x_{B(x)^c},v}$ for the distribution actually used to sample token v. This abbreviation depends only on the mask and visible assignment. Independence of these draws gives

$$Q_x(y) = \prod_{v \in B(x)} \nu_{x,v}(y_v) \prod_{v \notin B(x)} \mathbf{1}\{y_v = x_v\}. \tag{29}$$

For a joint resampling step, Q_x is instead the distribution produced by the entire procedure for filling the mask. Its coordinates need not be independent. In both cases, every coordinate outside $B(x)$ is copied, as required by Theorem A.3. The repair-set rule, path events, weights, and stopping rule are the same in the two statements. Averaging Q_{X_t} conditional on each attainable current repair set gives exactly the probabilities in (19). All assumptions of the general theorem therefore hold. Its conclusion that the sampler reaches a feasible assignment, together with the expected number of rounds in (21), gives Theorem 3.1. □

Corollary A.6 (Expected cumulative repair-set weight)**.** *Under Theorem 3.1,*

$$\mathbb{E}\sum_{t<\tau} W(R_t) \le \frac{\mathbb{E}W(R_0)}{1-\kappa}. \tag{30}$$

With unit weights, this bounds the expected total number of constraints selected for repair across rounds by $m/(1-\kappa)$.

Proof. Under the same assumptions, (24) bounds the expected remaining weight after each complete masked diffusion resampling step. Set $W_t = W(R_t)$ and keep $W_t = 0$ after termination. Iterating the contraction gives $\mathbb{E}W_t \le \kappa^t \mathbb{E}W_0$. Since all terms are nonnegative, Tonelli's theorem yields

$$\mathbb{E}\sum_{t<\tau} W(R_t) = \sum_{t=0}^{\infty} \mathbb{E}W_t \le \sum_{t=0}^{\infty} \kappa^t \mathbb{E}W_0 = \frac{\mathbb{E}W(R_0)}{1-\kappa}.$$

This is (30), which uses the same certificate as the termination theorem and no additional assumptions on the chosen initialization. □

The certificate applies to the probabilities used at inference, including any sampling controls. For independent parallel draws, factor each pointwise path probability as in Appendix C and then average the whole product, not its factors separately. The general theorem and its masked diffusion corollary also cover dependent draws using joint path probability bounds.

B BOUNDS ON THE NUMBER AND SIZE OF REPAIRS

The following consequences of Theorem A.3 require the same certificate for the chosen initialization and no further assumptions. They therefore apply to the masked diffusion method through Corollary A.5.

Corollary B.1 (Probability of a long run and expected number of rounds)**.** *Put $A = \mathbb{E}W(R_0)/w_{\min}$. For every integer $t \geq 0$,*

$$\mathbb{P}(\tau > t) \leq \min\{1, A\kappa^t\}. \tag{31}$$

If $A > 0$, then for $\delta \in (0,1)$ the choice

$$t_\delta = \max\left\{0, \left\lceil \frac{\log(A/\delta)}{\log(1/\kappa)} \right\rceil \right\}$$

ensures $\mathbb{P}(\tau > t_\delta) \leq \delta$. If $A = 0$, termination is immediate almost surely. Moreover,

$$\mathbb{E}\tau \leq \frac{1 + \log\left(\sum_i w_i / w_{\min}\right)}{1 - \kappa}. \tag{32}$$

For unit event weights this is $(1 + \log m)/(1 - \kappa)$.

Proof. The event $\{\tau > t\}$ is exactly $\{R_t \neq \varnothing\}$, on which $W_t \geq w_{\min}$. Apply Markov's inequality to (25). The round bound follows from (28) and $W_0 \leq \sum_i w_i$. □

Corollary B.2 (Repair costs and number of masked tokens)**.** *For nonnegative event costs c_i, put $c_w = \max_i c_i/w_i$. Then*

$$\mathbb{E}\left[\sum_{t<\tau} \sum_{i \in R_t} c_i\right] \leq c_w \frac{\mathbb{E}W(R_0)}{1 - \kappa}. \tag{33}$$

In particular, let $c_V = \max_i |V_i|/w_i$. The token masks satisfy

$$\mathbb{E}|B_t| \leq c_V \kappa^t \mathbb{E}W(R_0), \tag{34}$$

$$\mathbb{E}\left[\sum_{t<\tau} |B_t|\right] \leq c_V \frac{\mathbb{E}W(R_0)}{1 - \kappa}, \tag{35}$$

$$\mathbb{P}(|B_t| \geq b) \leq \min\{1, c_V \kappa^t \mathbb{E}W(R_0)/b\} \qquad (b > 0). \tag{36}$$

Proof. For every repair set, $\sum_{i \in R_t} c_i \leq c_w W_t$. Also $|B_t| \leq \sum_{i \in R_t} |V_i| \leq c_V W_t$, since overlaps can only decrease the union size. Apply (26), (25), and Markov's inequality. □

Remark B.3 (The mask can grow during a run). The certificate allows the mask to grow on an individual round. For example, take Boolean variables (x_1, x_2, x_3), bad events $A_1 = \{x_1 = 0, x_2 = 0\}$ and $A_2 = \{x_1 = 0, x_2 = 1, x_3 = 1\}$, and fair independent masked resampling. Neither bad event can occur together with the shared-coordinate condition for the other. Both paths of positive length therefore have probability zero. The bounds $\beta((1)) = \beta((2)) = 1/4$ satisfy the certificate with $w_1 = w_2 = 1$ and $\kappa = 1/2$. Starting at $(0, 0, 1)$ gives $R_0 = \{1\}$ and a two-token mask. With probability $1/4$ the resampling step sets $(x_1, x_2) = (0, 1)$, giving $R_1 = \{2\}$ and a three-token mask. Thus (34) gives an upper bound on the expected mask size that decreases with t. It does not say that $B_{t+1} \subseteq B_t$ or that $\mathbb{E}|B_{t+1}| \leq \kappa \mathbb{E}|B_t|$.

These repair counts give runtime bounds once we account for the costs of evaluating constraints, constructing the repair mask, computing model probabilities, and obtaining resampling distributions that satisfy the certificate.

When the certificate fails. The criterion fails when a repair is too likely to preserve or propagate a violation. In that case, the weighted path sum cannot be made uniformly smaller than the current repair weight. Consider one bit constrained to equal one, and suppose a repair from zero outputs one with probability $a \in [0, 1]$ and zero otherwise. The only path is the length-zero path, whose probability is $1 - a$, so (8) requires $\kappa \geq 1 - a$. From zero, the expected number of repair rounds is $1/a$ when $a > 0$. As a approaches zero, the required contraction factor approaches one and repair

becomes slow. At $a = 0$, no $\kappa < 1$ satisfies the criterion and the sampler remains infeasible forever. Conversely, a deterministic repair that always returns one terminates in a single round. The problem is persistent violations, not determinism itself. Failure of this sufficient test can also reflect loose path bounds rather than actual nontermination.

C Constructing and checking the certificate

C.1 Calculating path probabilities for independent resampling

Suppose that, at current assignment x, the sampler draws coordinates in $B(x)$ independently from distributions $\nu_{x,v} = \nu_{B(x),x_{B(x)^c},v}$ and copies all other coordinates. The distribution of the complete assignment after the resampling step is

$$Q_x(y) = \prod_{v \in B(x)} \nu_{x,v}(y_v) \prod_{v \notin B(x)} \mathbf{1}\{y_v = x_v\}. \tag{37}$$

For this current assignment, define

$$p_i(x) = Q_x(A_i), \qquad s_{ij}(x) = Q_x(A_i \cap U_{ij}), \qquad r_{ij}(x) = Q_x(U_{ij}).$$

Here $p_i(x)$ is the probability that event i is bad after the resampling step, $r_{ij}(x)$ is the probability that the shared values allow event j, and $s_{ij}(x)$ is the probability that both statements hold together.

Lemma C.1 (Path probabilities as products)**.** *For an induced path $P = (i_0, \dots, i_\ell)$,*

$$Q_x(D_P) = \begin{cases} p_{i_0}(x), & \ell = 0, \\ s_{i_0 i_1}(x) \displaystyle\prod_{q=2}^{\ell} r_{i_{q-1} i_q}(x), & \ell \geq 1. \end{cases} \tag{38}$$

The empty product for $\ell = 1$ is one.

Proof. The first event $A_{i_0} \cap U_{i_0 i_1}$ is determined by V_{i_0}. For $q \geq 2$, the subsequent overlap event is determined by $O_q = V_{i_{q-1}} \cap V_{i_q}$. These coordinate sets are pairwise disjoint. Indeed, an intersection of V_{i_0} with O_q would make i_0 adjacent to the nonconsecutive vertex i_q. If $2 \leq q < q'$, an intersection of O_q and $O_{q'}$ would make i_{q-1} adjacent to the nonconsecutive vertex $i_{q'}$. Both contradict the definition of an induced path. Independence under (37) now gives (38). The length-zero case is immediate. □

The first factor is generally s_{ij}, not $p_i r_{ij}$: the bad starting event and the first shared-coordinate condition use some of the same variables. The factorization uses independence of coordinates within one resampling step, conditional on the current assignment x.

To use the factorization in the main theorem, average its full right-hand side over the current assignments conditional on $R_t = S$:

$$\overline{Q}_{t,S}(D_P) = \begin{cases} \mathbb{E}[p_{i_0}(X_t) \mid R_t = S], & |P| = 0, \\ \mathbb{E}\Big[s_{i_0 i_1}(X_t) \prod_{q=2}^{|P|} r_{i_{q-1} i_q}(X_t) \,\Big|\, R_t = S \Big], & |P| \geq 1, \end{cases} \tag{39}$$

where $|P|$ is the number of edges. The mixture $\overline{Q}_{t,S}$ generally has dependent coordinates even when all Q_x are product distributions. Products of separate averaged p_i, s_{ij}, r_{ij} therefore need not bound a path probability.

As a stronger sufficient route, bounds holding at every attainable nonterminal assignment, $\bar{p}_i \geq \sup_x p_i(x)$, $\bar{s}_{ij} \geq \sup_x s_{ij}(x)$, and $\bar{r}_{ij} \geq \sup_x r_{ij}(x)$, give a fixed upper bound on the averaged path probabilities:

$$\beta_{\mathrm{fac}}(P) = \begin{cases} \bar{p}_{i_0}, & \ell = 0, \\ \bar{s}_{i_0 i_1} \displaystyle\prod_{q=2}^{\ell} \bar{r}_{i_{q-1} i_q}, & \ell \geq 1. \end{cases} \tag{40}$$

The suprema range over assignments that the process can encounter before termination. One may instead use a larger set of assignments if the bounds can be checked on that set, as described in Appendix A.2. Bounding a complete path's probability directly can give a smaller value than multiplying these separate worst-case bounds. If two bounds are available, use the smaller one for each path.

An optional pointwise symmetric bound. For parallel resampling with $p_i(x) \le \bar{p}$ and $r_{ij}(x) \le \bar{r}$ at every attainable nonterminal assignment, use $s_{ij}(x) \le p_i(x)$ in (40). There are at most Δ choices for the first edge of an induced path and $\Delta - 1$ for each later edge. Thus, for $\Delta \ge 1$, fixed path bounds give a contribution per closed neighborhood of at most

$$\kappa_{\mathrm{full}} = (\Delta + 1)\bar{p}\left(1 + \frac{\Delta}{1 - (\Delta - 1)\bar{r}}\right),$$

provided $(\Delta - 1)\bar{r} < 1$. Summing over $a \in R_t$ may count roots more than once but proves (20) whenever $\kappa_{\mathrm{full}} < 1$. The constants here and in Corollary C.2 reflect different assumptions, with pointwise bounds at every attainable assignment here and averages conditional on the current repair set there. For $\Delta = 1$ this pointwise route reduces to $4\bar{p} < 1$. For $\Delta = 0$, only length-zero paths occur and $\bar{p} < 1$ suffices.

C.2 A SYMMETRIC PATH CRITERION

A symmetric version of the certificate uses common bounds across constraints. It bounds each whole path event: averaging in (5) can create dependence, so separate averaged violation and overlap probabilities cannot simply be multiplied. The normalization below is chosen to recover the product PRS bounds.

Corollary C.2 (A symmetric criterion for the masked diffusion sampler)**.** *Let $\Delta \ge 2$ and $a_\Delta = (1 + 1/\Delta)^{\Delta+1}$. Suppose that for every round t, attainable nonempty S, and $P = (i_0, \dots, i_k)$ rooted in $\Gamma^+(S)$,*

$$\overline{Q}_{t,S}(D_P) \le \bar{p} \begin{cases} 1, & k = 0,\ i_0 \in S, \\ e, & k = 0,\ i_0 \in \Gamma(S), \\ a_\Delta^k \bar{r}^{\,k-1}, & k \ge 1. \end{cases} \tag{41}$$

If the fixed parameters $\bar{p}, \bar{r} \ge 0$ satisfy

$$6e\,\bar{p}\,\Delta^2 \le 1, \qquad 3e\,\bar{r}\,\Delta \le 1, \tag{42}$$

then Theorem 3.1 applies with unit weights and $\kappa = 1/2$: at most $2(1 + \log m)$ expected rounds and $2m$ expected constraint resamplings.

Here $\bar{p}$ controls new violations, with different allowances within the repair set and among its neighboring constraints. The parameter $\bar{r}$ controls how the whole-path bounds change with length. In general, $\bar{r}$ is not the averaged probability of an individual overlap.

Proof of Corollary C.2. Fix a round t and nonempty repair set S with $\mathbb{P}(R_t = S) > 0$, and put $a_\Delta = (1 + 1/\Delta)^{\Delta+1}$. The length-zero path assumptions bound their total contribution by

$$|S|\bar{p} + |\Gamma(S)|e\bar{p} \le \bar{p}(1 + e\Delta)|S|.$$

For length $k \ge 1$, there are at most $(\Delta + 1)|S|\Delta(\Delta - 1)^{k-1}$ induced paths rooted in $\Gamma^+(S)$. Their individual probabilities are at most $\bar{p}\bar{r}^{k-1}a_\Delta^k$ by the assumed averaged path bound. Consequently the complete unit-weight path sum is at most $k_\star|S|$, where

$$k_\star = \bar{p}(1 + e\Delta) + \frac{(\Delta + 1)\Delta\bar{p}a_\Delta}{1 - (\Delta - 1)\bar{r}a_\Delta}. \tag{43}$$

The denominator is positive under (42), as checked below. When $\bar{r} = 0$, only the length-one term remains in this series. As usual, $\bar{r}^0 = 1$.

The elementary bound $a_\Delta \le e(1 + 1/(2\Delta))$ follows by setting $x = 1/\Delta$ in

$$(1 + 1/x)\log(1 + x) \le 1 + \log(1 + x/2).$$

The difference between the right and left sides tends to zero as $x \downarrow 0$ and has derivative $[\log(1+x) - 2x/(2+x)]/x^2 \geq 0$. Therefore $6e\bar{p}\Delta^2 \leq 1$ and $3e\bar{r}\Delta \leq 1$ imply

$$(\Delta-1)\bar{r}a_\Delta \leq \frac{(\Delta-1)(2\Delta+1)}{6\Delta^2} < 1, \qquad k_\star \leq \frac{1}{6\Delta} + \frac{1}{6e\Delta^2} + \frac{(\Delta+1)(2\Delta+1)}{2(4\Delta^2+\Delta+1)}.$$

Each term on the right decreases for $\Delta \geq 2$, hence

$$k_\star \leq \frac{1}{12} + \frac{1}{24e} + \frac{15}{38} < \frac{1}{2}.$$

Theorem A.3 now applies with unit weights and $\kappa = 1/2$. Since $|R_0| \leq m$, it gives at most $2(1+\log m)$ expected resampling rounds and $2m$ expected constraint resamplings. The proof used only probabilities conditioned on the current repair set, with no factorization assumption on $\overline{Q}_{t,S}$. □

Equation (43) also gives a less conservative sufficient condition: convergence of the displayed geometric series together with $k_\star < 1$. It permits using the actual factor $k_\star$ in place of $1/2$ when $k_\star < 1/2$. The main text uses the simpler pair of inequalities to match the product PRS criterion.

C.3 Sufficient checks at current assignments

Corollary C.3 (Pointwise bounds imply the averaged certificate)**.** *Suppose $\beta_x(P) \geq Q_x(D_P)$ and there are fixed $w > 0$ and $0 < \kappa < 1$ such that, at every attainable nonterminal assignment,*

$$\sum_{i \in \Gamma^+(\mathrm{Res}(x))} \sum_{P \in \mathcal{I}_i} \beta_x(P) w_{\mathrm{end}(P)} \leq \kappa W(\mathrm{Res}(x)). \tag{44}$$

Then the current-repair-set certificate and all conclusions of Theorem A.3 hold.

Proof. For each t, S with $\mathbb{P}(R_t = S) > 0$, average the pointwise bounds to define $\beta_{t,S}(P) = \mathbb{E}[\beta_{X_t}(P) \mid R_t = S]$. Equation (16) gives (19). On $\{R_t = S\}$ the root set is $\Gamma^+(S)$, so averaging (44) gives (20) with the same w and κ. □

This condition checks each attainable current assignment and its model input. The main theorem allows rare assignments to fail this local check when their contribution averaged conditional on the current repair set is small.

C.4 Comparison with a product reference sampler

There is a second way to avoid requiring every conditional overlap probability to be small. Compare the model's resampling steps with those of a product reference distribution $\mu = \prod_v \mu_v$. Let $p_\star = \max_i \mu(A_i)$ and $r_\star = \max_{i \sim j} \mu(U_{ij})$, and suppose $\Delta \geq 2$ and

$$6e\, p_\star \Delta^2 \leq 1, \qquad 3e\, r_\star \Delta \leq 1. \tag{45}$$

These probabilities are unconditional. We first record a constant contraction factor for the reference sampler, using the estimates in the proof of Lemma 28 of Guo et al. (2019).

Lemma C.4 (Contraction of the product reference sampler)**.** *Run PRS from $X_0 \sim \mu$ with independent replacement variables drawn from their marginals under μ. Under* (45)*,*

$$\mathbb{E}_\mu[|R_{t+1}| \mid R_t] \leq \tfrac{1}{2}|R_t|, \qquad \mathbb{E}_\mu|R_t| \leq m\, 2^{-t}.$$

Proof. Conditional on the current repair set $R_t = S$, the assignment after resampling has distribution $\mu(\cdot \mid E_S)$, where $E_S = \bigcap_{j \notin \Gamma^+(S)} A_j^c$. This is the general PRS invariant of Guo et al. (2019). It uses both product initialization and independent reference resampling. Constraints in S use variables disjoint from those determining E_S, so

$$\mu(A_i \mid E_S) = \mu(A_i) \leq p_\star \quad (i \in S).$$

For $i \in \Gamma(S)$, the local lemma conditional-probability bound with parameters $x_j = 1/(\Delta+1)$ gives

$$\mu(A_i \mid E_S) \leq p_\star(1 + 1/\Delta)^\Delta \leq e p_\star.$$

These parameters satisfy the local lemma hypothesis under (45). For any induced path of length $k \geq 1$, its unconditioned probability is at most $p_\star r_\star^{k-1}$: the first bad event and overlap are bounded together by $p_\star$, and the subsequent overlaps have disjoint supports. The full path event is determined by variables in the first k path vertices, whose scopes meet at most $k(\Delta+1)$ constraint scopes. The same local lemma comparison therefore gives

$$\mu(D_P \mid E_S) \leq p_\star r_\star^{k-1}(1+1/\Delta)^{k(\Delta+1)}. \tag{46}$$

These are exactly the averaged path assumptions of Corollary C.2 with $\bar{p} = p_\star$, $\bar{r} = r_\star$. Its proof bounds the conditional path sum by

$$k_\star|S|, \qquad k_\star = p_\star(1+e\Delta) + \frac{(\Delta+1)\Delta p_\star a_\Delta}{1-(\Delta-1)r_\star a_\Delta} < \frac{1}{2}. \tag{47}$$

Lemma A.4 yields the claimed conditional contraction, and iteration gives $\mathbb{E}_\mu|R_t| \leq m2^{-t}$. After termination both sets are empty. □

Proof of Corollary 3.2. The initialization and kernels in the corollary give precisely the product reference sampler. The proof of Lemma C.4 therefore supplies bounds $\beta_{t,S}(P)$ on its actual averaged path probabilities, for every round t and attainable nonempty repair set S. Their unit-weight sum is at most $k_\star|S| < |S|/2$ by (47). Thus they satisfy (8) with unit weights and $\kappa = 1/2$. Theorem 3.1 gives almost-sure termination and at most $2(1+\log m)$ expected rounds, while Corollary A.6 gives at most $2m$ expected constraint resamplings.

The first inequality in (3), with $\Delta \geq 2$, also implies $ep_\star(\Delta+1) \leq 1$, so the local lemma ensures $\mu(\mathcal{F}) > 0$. The product PRS invariant identifies the distribution after resampling S as $\mu(\cdot \mid E_S)$ conditional on $R_t = S$. Since $\mathcal{F} \subseteq E_S$, conditioning a terminating check further on feasibility gives $\mu(\cdot \mid \mathcal{F})$. The same holds if the initial proposal is feasible. Almost-sure termination therefore yields this exact output distribution, as in the classical PRS correctness argument of Guo et al. (2019). □

In particular, if every masked token is drawn independently and uniformly, and the initial proposal is obtained by the same rule with all variables masked, the model process is uniform PRS. Corollary 3.2 then applies with the unconditional uniform probabilities $p_\star, r_\star$.

Proposition C.5 (Bounded changes to the reference resampling)**.** *Suppose* (45) *holds. Let the model's initial distribution q_0 and fixed kernels $K_B(z,\cdot)$ for the entire resampled block satisfy, for fixed $C_0 \geq 1$ and $1 \leq c < 2$,*

$$q_0(x) \leq C_0\mu(x), \qquad K_{B(x)}(x_{B(x)^c}, u) \leq c\,\mu_{B(x)}(u), \qquad \mu_B(u) = \prod_{v\in B}\mu_v(u_v). \tag{48}$$

The second inequality is required for every block assignment at each attainable nonterminal assignment x. Resampling steps may be joint and may depend on the current mask and visible values. With $\rho = c/2 < 1$, the sampler terminates almost surely at a feasible assignment, and

$$\mathbb{P}(\tau > t) \leq \min\{1, C_0 m\rho^t\}, \qquad t \geq 0, \tag{49}$$

$$\mathbb{E}\tau \leq \frac{1+\log(C_0 m)}{1-\rho}, \qquad \mathbb{E}\sum_{t<\tau}|R_t| \leq \frac{m(1+\log C_0)}{1-\rho}. \tag{50}$$

In particular, if $q_0 = \mu$ and $c \leq 3/2$, the expected rounds and constraint resamplings are at most $4(1+\log m)$ and $4m$, respectively.

Proof. Compare complete assignment trajectories with reference PRS using the same deterministic repair-set rule. The initial probability ratio is at most C_0, and each resampling step contributes a ratio at most c. All internal sampling randomness is included in the fixed block kernel. Keep both processes fixed after termination. Their subsequent transition ratio is one. The model's trajectory distribution through time t is therefore bounded by C_0c^t times the reference trajectory distribution. Lemma C.4 gives

$$\mathbb{E}|R_t| \leq C_0c^t\mathbb{E}_\mu|R_t| \leq C_0 m\rho^t.$$

Since $|R_t| \leq m$ and $|R_t| \geq 1$ whenever $\tau > t$, this proves (49) and $\mathbb{E}|R_t| \leq m\min\{1, C_0\rho^t\}$.

For $A \geq 1$, the scalar inequality

$$\sum_{t \geq 0} \min\{1, A\rho^t\} \leq \frac{1 + \log A}{1 - \rho}$$

follows by telescoping $F(A\rho^t)$, where $F(s) = s$ for $0 \leq s \leq 1$ and $F(s) = 1 + \log s$ for $s \geq 1$. Indeed, $F(s) - F(\rho s) \geq (1-\rho)\min\{1, s\}$. Apply this inequality with $A = C_0 m$ to the tail and with $A = C_0$ to the expected repair-set sizes. The tail tends to zero, proving almost-sure termination, and the stopping rule ensures feasibility. □

This proposition compares the distribution of the newly sampled block with the reference distribution. Its trajectory comparison proves unconditional bounds. It allows overlap events determined by coordinates left unchanged to have conditional probability one. Indeed, U_{ij} depends only on the shared variables $V_i \cap V_j$, so

$$(V_i \cap V_j) \cap B(x) = \varnothing \quad \Longrightarrow \quad Q_x(U_{ij}) = \mathbf{1}\{x \in U_{ij}\} \in \{0, 1\}.$$

For example, consider clauses $x \vee y$ and $y \vee z$ whose variables are all outside the current mask. At $x = z = 1$ and $y = 0$, both clauses are satisfied, but the shared value $y = 0$ can be extended to violate $y \vee z$ by setting $z = 0$. Thus U_{ij} holds and $Q_x(U_{ij}) = 1$. This probability concerns compatibility with a violation. The second clause itself remains satisfied because its actual value $z = 1$ is unchanged. This situation can occur while a separate violated constraint supplies the current repair mask.

For example, a score $f_{B,z}$ depending on the current mask and visible values defines a fixed resampling kernel

$$K_B(z, u) = \frac{\mu_B(u) \exp(f_{B,z}(u))}{\mathbb{E}_{\mu_B} \exp(f_{B,z})}.$$

If $\sup f_{B,z} - \inf f_{B,z} \leq \log c$ on the reference support, then $K_B(z, u)/\mu_B(u) \leq c$. Such bounded reweighting may introduce dependence among the masked variables.

The factor c controls the *whole block*. Even for parallel resampling, pointwise token probability ratios at most e^{ϵ} give only $e^{\epsilon|B|}$ for a block of $|B|$ tokens. A constant c can therefore be restrictive for large masks. Likewise, large C_0 increases the bounds on rounds and constraint selections through $\log C_0$. These bounds give $O(\log m)$ rounds and $O(m)$ constraint resamplings when C_0 is bounded and c stays below two by a fixed margin.

C.5 A MATRIX TEST AND BOUNDS FOR INDIVIDUAL EVENTS

As an optional stronger test, suppose fixed bounds $\beta(P)$ dominate $\overline{Q}_{t,S}(D_P)$ for every relevant round and current repair set. They may come from direct averaged estimates or from the pointwise envelope (40). Define

$$G_i(w) = \sum_{P \in \mathcal{I}_i} \beta(P) w_{\mathrm{end}(P)}.$$

A sufficient condition for the aggregate certificate is

$$\sum_{i \in \Gamma^+(a)} G_i(w) \leq \kappa w_a \qquad (a \in [m]). \tag{51}$$

Indeed, summing over $a \in R_t$ bounds the union of current root neighborhoods, possibly counting roots more than once. Let $\mathcal{I}(i, j)$ be the induced paths from i to j, and define nonnegative matrices

$$K_{ij} = \sum_{P \in \mathcal{I}(i,j)} \beta(P), \qquad C_{ai} = \mathbf{1}\{i \in \Gamma^+(a)\}, \qquad M = CK. \tag{52}$$

Then $(Kw)_i = G_i(w)$, so the sufficient condition (51) is $Mw \leq \kappa w$. The next proposition shows how to check whether suitable positive weights exist. Write $\rho(M)$ for the spectral radius of M, the largest absolute value of any of its eigenvalues.

Proposition C.6 (Matrix test for the chosen probability bounds)**.** *The following statements are equivalent:*

1. $\rho(M) < 1$;
2. *there exists* $w > 0$ *with* $Mw < w$ *componentwise;*
3. *there exist* $w > 0$ *and* $0 < \kappa < 1$ *with* $Mw \le \kappa w$.

Proof. The second and third statements are equivalent because there are finitely many rows. To show that the third implies the first, let $D = \mathrm{diag}(w)$. The row sums of $D^{-1}MD$ are at most κ, so $\rho(M) = \rho(D^{-1}MD) \le \|D^{-1}MD\|_\infty \le \kappa < 1$. Conversely, if $\rho(M) < 1$, choose $\kappa \in (\rho(M), 1)$ and set

$$w = (I - M/\kappa)^{-1}\mathbf{1} = \sum_{q \ge 0} (M/\kappa)^q \mathbf{1}.$$

The series converges, $w \ge \mathbf{1} > 0$, and $Mw = \kappa(w - \mathbf{1}) \le \kappa w$. □

For the chosen probability bounds in (52), $\rho(M) < 1$ is sufficient for termination.

Corollary C.7 (How often each event is selected for repair)**.** *Let* Z_t *be the column vector with* $(Z_t)_j = \mathbf{1}\{j \in R_t\}$*, and let* $N_j = \sum_{t<\tau} \mathbf{1}\{j \in R_t\}$*. For the fixed averaged path bounds defining* M*, if* $\rho(M) < 1$*, then*

$$\mathbb{E}[Z_t^\top] \le \mathbb{E}[Z_0^\top] M^t, \tag{53}$$

$$\mathbb{E}[N^\top] \le \mathbb{E}[Z_0^\top](I - M)^{-1}, \tag{54}$$

$$\mathbb{E}\left[\sum_{t<\tau} Z_t^\top c\right] \le \mathbb{E}[Z_0^\top](I - M)^{-1} c \qquad (c \ge 0). \tag{55}$$

The vector inequalities hold separately for every coordinate. In particular, choosing $c = \mathbf{1}$ *gives an upper bound on the expected number of rounds, and choosing* $c_j = |V_j|$ *gives an upper bound on the expected total number of sampled token values.*

Proof. For a fixed endpoint j, only events sharing coordinates with the current mask can become bad. Adding the probabilities of the paths from those events to j gives

$$\mathbb{P}(j \in R_{t+1} \mid R_t) \le \sum_{i \in \Gamma^+(R_t)} K_{ij} \le (Z_t^\top M)_j.$$

Take expectations and iterate, using nonnegativity of M. Summing over t, interchanging the nonnegative sums and expectations, and using $\sum_{t \ge 0} M^t = (I - M)^{-1}$ gives (54). Multiplication by $c \ge 0$ gives the cost bound. Each round before termination includes at least one event, and each mask is a union of its event scopes. □

C.6 A SIMPLER TEST USING ONE BOUND PER DIRECTED EDGE

Summing over all induced paths can be expensive. When $\beta \le \beta_{\mathrm{fac}}$, the following test provides a sufficient alternative. It introduces a nonnegative number ϕ_{ij} for each directed edge. This number will bound the contribution from continuing a path after traversing that edge.

Proposition C.8 (Edge certificate)**.** *Suppose* $w_i > 0$*,* $\phi_{ij} \ge 0$*, and* $0 < \kappa < 1$ *satisfy*

$$\phi_{ij} \ge w_j + \sum_{k \in \Gamma(j) \setminus \{i\}} \bar{r}_{jk} \phi_{jk}, \qquad (i \sim j), \tag{56}$$

$$\sum_{i \in \Gamma^+(a)} \widehat{G}_i \le \kappa w_a, \qquad (a \in [m]), \tag{57}$$

where

$$\widehat{G}_i = \bar{p}_i w_i + \sum_{j \in \Gamma(i)} \bar{s}_{ij} \phi_{ij}.$$

Then $G_i(w) \le \widehat{G}_i$*, so the certificate holds for every event.*

Proof. Consider walks that have just traversed the directed edge (i, j) and then continue for at most L further edges. A walk may revisit vertices, but it may not immediately retrace its last edge. Such a walk is called a nonbacktracking walk. Give each continuation a weight equal to the weight of its final vertex times the product of $\bar{r}_{jk}$ over the additional edges (j, k) it traverses. Let $F_{ij}^{(L)}$ be the sum of these weights, including w_j for stopping immediately at j. These finite sums satisfy

$$F_{ij}^{(0)} = w_j, \qquad F_{ij}^{(L+1)} = w_j + \sum_{k \in \Gamma(j) \setminus \{i\}} \bar{r}_{jk} F_{jk}^{(L)}.$$

Equation (56) gives $F_{ij}^{(L)} \le \phi_{ij}$ by induction. Every induced path is nonbacktracking. After taking out its first factor $\bar{s}_{ij}$, its remaining contribution is therefore included in one of these walk sums. The length-zero contribution is at most $\bar{p}_i w_i$. This proves $G_i(w) \le \widehat{G}_i$, and (57) gives the result. □

For fixed κ and valid bounds on the local probabilities, these are linear inequalities in m event weights and $2|E|$ edge bounds. We may impose $w_i \ge 1$: scaling all the event weights and edge bounds by the same positive factor preserves the inequalities, and any positive solution can be scaled to satisfy this condition. This sufficient test can be more restrictive than the induced-path certificate because it also counts walks that are not induced paths.

For illustration, if the maximum degree is $\Delta \ge 1$ and $\bar{p}_i, \bar{s}_{ij} \le p$, $\bar{r}_{ij} \le r$, then

$$(\Delta - 1)r < 1, \qquad (\Delta + 1)p \left(1 + \frac{\Delta}{1 - (\Delta - 1)r}\right) < 1$$

suffice: choose $w_i = 1$ and $\phi_{ij} = 1/[1-(\Delta-1)r]$ in the edge inequalities. For isolated events ($\Delta = 0$), there are only length-zero paths, and $\max_i \bar{p}_i < 1$ suffices. The PRS invariant in Appendix C.4 gives averaged path bounds for the product reference sampler.

C.7 Boolean clauses

Suppose each constraint is a Boolean clause: an OR of variables or their negations, with no variable repeated. Assume that the clause is not always true. Let A_i be the event that clause i is false, and let $f_{i,v} \in \{0, 1\}$ be the bit that makes its literal on coordinate $v \in V_i$ false. Then

$$A_i = \{x : x_v = f_{i,v} \text{ for all } v \in V_i\}, \qquad U_{ij} = \{x : x_v = f_{j,v} \text{ for all } v \in V_i \cap V_j\}.$$

For independent resampling, the probability of bit b at coordinate v after the resampling step is

$$q_{x,v}(b) = \begin{cases} \nu_{x,v}(b), & v \in B(x), \\ \mathbf{1}\{b = x_v\}, & v \notin B(x). \end{cases}$$

We can now calculate the three local probabilities directly:

$$p_i(x) = \prod_{v \in V_i} q_{x,v}(f_{i,v}), \qquad r_{ij}(x) = \prod_{v \in V_i \cap V_j} q_{x,v}(f_{j,v}), \qquad s_{ij}(x) = \chi_{ij} p_i(x), \tag{58}$$

where

$$\chi_{ij} = \mathbf{1}\{f_{i,v} = f_{j,v} \text{ for every } v \in V_i \cap V_j\}.$$

For the last identity, if the two clauses require different bits at a shared coordinate, $A_i \cap U_{ij}$ is empty. Otherwise, $A_i \subseteq U_{ij}$. Although $\chi_{ij} = \chi_{ji}$, the values $s_{ij}(x)$ and $s_{ji}(x)$ need not be equal.

We can also calculate the probability of a complete path by listing the bits that its witness event requires. Collect the requirements $y_v = f_{i_0,v}$ on V_{i_0} and $y_v = f_{i_q,v}$ on each $V_{i_{q-1}} \cap V_{i_q}$ for the next assignment y. If requirements conflict, $Q_x(D_P) = 0$. Otherwise let $T(P)$ be their union of coordinates and $b_P(v)$ their required bit. Then

$$Q_x(D_P) = \mathbf{1}\{x_v = b_P(v) \text{ for all } v \in T(P) \setminus B(x)\} \prod_{v \in T(P) \cap B(x)} \nu_{x,v}(b_P(v)). \tag{59}$$

Each coordinate is counted once. The indicator checks that the copied coordinates have the required bits, and the product gives the probability that the sampled coordinates also have them. Copied values enter each path probability through the indicator in (59). Average the full expression conditional on the current repair set, or use the pointwise checks in Corollary C.3.

C.8 A CERTIFICATE USING PATHS ONLY UP TO A CHOSEN LENGTH

The full path sum can contain exponentially many induced paths. While we cannot simply discard longer paths, we can trade the number of probabilities we bound against precision by bounding the endpoints of longer paths using the probability of the last examined prefix.

Fix a round t and a nonempty set S with $\mathbb{P}(R_t = S) > 0$. Write $\overline{Q}_{t,S}$ for the conditional distribution of the next assignment given $R_t = S$. For an induced path $P = (v_0, \dots, v_k)$, let $C(P)$ be the connected component of v_k in the graph induced by

$$\{v_k\} \cup \left([m] \setminus \bigcup_{q=0}^{k-1} \Gamma^+(v_q) \right), \qquad M_w(P) = \sum_{j \in C(P)} w_j. \tag{60}$$

For $k = 0$, the union is empty and $C((v_0))$ is the component of v_0 in the original graph. Every induced extension of P ends in $C(P)$: a new vertex cannot equal or neighbor any earlier vertex of P other than v_k. Conversely, a shortest path from v_k to any vertex of $C(P)$ gives an induced extension of P. Let $\mathcal{C}(P)$ contain the paths formed by adding one admissible vertex to P.

Choose a depth $d \in \{0, \dots, m-1\}$. Only for paths of length at most d starting in $\Gamma^+(S)$, choose bounds $\beta_{t,S}(P) \geq \overline{Q}_{t,S}(D_P)$. Define the following quantities backward from depth d:

$$T_{t,S,d}(P) = \begin{cases} M_w(P)\beta_{t,S}(P), & |P| = d, \\ \min\left\{ M_w(P)\beta_{t,S}(P),\ w_{\mathrm{end}(P)}\beta_{t,S}(P) + \displaystyle\sum_{P' \in \mathcal{C}(P)} T_{t,S,d}(P') \right\}, & |P| < d. \end{cases} \tag{61}$$

Here $|P|$ denotes the number of edges, and a sum over no extensions is zero. The first bound allows every remaining endpoint whenever the prefix event holds. The second accounts separately for the current endpoint and for each one-edge extension. Taking the smaller bound ensures that inspecting a longer prefix never makes the certificate worse.

Proposition C.9 (Finite-depth path certificate)**.** *Run the memoryless sampler with the PRS repair mask from its specified initial distribution. Suppose fixed positive weights w and $0 < \kappa < 1$ satisfy*

$$\sum_{i \in \Gamma^+(S)} T_{t,S,d}((i)) \leq \kappa W(S) \tag{62}$$

at every round t and attainable nonempty repair set S. Then the almost-sure termination and bounds on expected rounds and cumulative repair-set weight from Theorem 3.1 and Corollary A.6 hold with these same w, κ. The depth may be chosen separately for each pair (t, S). The sampler still uses its fixed kernel and full repair mask. For any fixed (t, S) and probability bounds on existing prefixes, increasing the depth can only decrease the left side of (62)*.*

Proof. For a completed assignment y, let $Z_P(y)$ contain the endpoints of all induced extensions Q of P, including P itself, for which $y \in D_Q$. Since $D_Q \subseteq D_P$ and every such endpoint belongs to $C(P)$,

$$W(Z_P(y)) \leq M_w(P)\mathbf{1}\{y \in D_P\}.$$

Separating the extension of length zero from those with a first new edge also gives

$$W(Z_P(y)) \leq w_{\mathrm{end}(P)}\mathbf{1}\{y \in D_P\} + \sum_{P' \in \mathcal{C}(P)} W(Z_{P'}(y)).$$

These are upper bounds on a union of endpoints. Repeated appearances can only increase the right sides. Conditional expectation under $\overline{Q}_{t,S}$, followed by backward induction, proves

$$\mathbb{E}[W(Z_P(X_{t+1})) \mid R_t = S] \leq T_{t,S,d}(P).$$

Locality and the induced-witness lemma (Lemmas A.4 and A.2) imply that every member of R_{t+1} is contained in some $Z_{(i)}(X_{t+1})$ with $i \in \Gamma^+(S)$. Hence (62) gives

$$\mathbb{E}[W(R_{t+1}) \mid R_t = S] \leq \kappa W(S).$$

The drift argument proving Theorem A.3 and Corollary A.6 now gives the claimed bounds. This proves their conclusions directly. A capped finite-depth sum need not upper-bound the uncapped sum of all path probabilities.

To compare depths d and $d+1$, consider an old boundary prefix of length d. Its new value is the minimum of its old cap $M_w(P)\beta_{t,S}(P)$ and an extension bound, so it cannot increase. The minimum and sum operations are monotone in all child values. Backward induction therefore gives the same conclusion at every root. □

Cost and strength. At depth zero, the test multiplies each violation probability by the total weight of the root's graph component. At depth $m-1$, it examines every induced extension and gives a bound no larger than the full path sum by limiting repeated counting of endpoints. If $r = |\Gamma^+(S)|$ and $\Delta \geq 2$, the number of queried prefixes is at most

$$N_d \leq r\left(1 + \Delta \sum_{k=0}^{d-1} (\Delta - 1)^k\right).$$

For fixed depth this is polynomial in the graph size. Constructing (60) by graph search takes $O(m + |E|)$ operations per prefix, in addition to checking extensions and obtaining the probability bounds. Here $|E|$ counts graph edges. Computing these probability bounds can be expensive because they average over assignments with $R_t = S$ at round t. Increasing depth reduces the conservative remainder while increasing the number of queries. Decreasing depth retains the same guarantee whenever the resulting inequality passes.

A two-variable example. Let X, Y be independent fair bits and let the three bad events be $A_1 = \{00\}$, $A_2 = \{01\}$, and $A_3 = \{10\}$, each with scope $\{X, Y\}$. The graph is a triangle. Every infeasible assignment has exactly one bad event, and its repair set is that singleton: all variables of either other event are already fixed to a different assignment, so neither is added. Repair resamples both bits uniformly. Thus at every round t and attainable nonempty repair set S,

$$\beta_{t,S}((i)) = \tfrac{1}{4}, \qquad \beta_{t,S}((i,j)) = 0 \quad (i \neq j),$$

because the first pair of bad events is disjoint and $U_{ij} = A_j$. With unit weights the current set has weight one. Depth zero gives three root bounds of $3/4$, totaling $9/4$, so it cannot certify contraction. Depth one gives three root bounds of $1/4$, totaling $3/4$. It therefore certifies $\kappa = 3/4$ using only length-zero and length-one probabilities. From uniform initialization, $\mathbb{E}L(W(R_0)) = \mathbb{E}W(R_0) = 3/4$, so the bounds on expected rounds and total constraint selections are both three, equal to their actual expectations.

C.9 Hardness of improving the degree exponent

Proposition C.10 concerns uniform sampling under a fixed product distribution and also excludes exact sampling with uniformly polynomial expected running time.

For completeness, a CNF formula is a conjunction of clauses, each an OR of variables or their negations. Total-variation error measures the largest difference between the probabilities that two distributions assign to the same set of outcomes.

Proposition C.10 (Degree-exponent barrier for uniform sampling)**.** *Fix $\varepsilon \in (0,2)$ and $C > 0$. Unless* NP = RP*, no randomized algorithm can, for every satisfiable CNF formula satisfying*

$$p_\star \Delta^{2-\varepsilon} \leq C,$$

sample a satisfying assignment within requested total-variation error $\eta \in (0,1)$ of the uniform distribution in time polynomial in the formula size and η^{-1}. Here $p_\star$ is measured under independent fair bits. The hardness holds even for monotone CNF formulas.

Proof of Proposition C.10. We use the reduction and hard-core sampling hardness result in Guo et al. (2019, Theorems 32–33). Choose a sufficiently large integer s, put $q = 2^s$, and set $d = 4q$. Given a simple d-regular graph $G = (V, E)$, replace each vertex v by s Boolean variables $x_{v,1}, \ldots, x_{v,s}$. For each edge uv, impose the positive clause

$$\left(\bigvee_{i=1}^{s} x_{u,i}\right) \vee \left(\bigvee_{i=1}^{s} x_{v,i}\right).$$

Let μ be uniform on these bits. A clause fails precisely when both endpoint blocks are zero. Two clauses share variables precisely when the corresponding edges share an endpoint. Thus, under μ, the dependency graph is the line graph of G and

$$p_\star = q^{-2}, \qquad \Delta = 2(d-1) = 8q-2. \tag{63}$$

Map a satisfying assignment to the set I of vertices whose blocks are entirely zero. The clauses ensure that I is an independent set. Send infeasible assignments to the empty independent set, defining the map on all possible approximate outputs. Exactly $(q-1)^{|V|-|I|}$ satisfying assignments map to a given I, so a uniform feasible assignment induces the hard-core distribution

$$\mathbb{P}(I) = \frac{\lambda^{|I|}}{Z_G(\lambda)}, \qquad \lambda = (q-1)^{-1}, \qquad Z_G(\lambda) = \sum_{J \text{ independent}} \lambda^{|J|}.$$

Sly & Sun (2014) show that approximate sampling from this distribution on d-regular graphs is NP-hard when

$$\lambda > \lambda_c(d) = \frac{(d-1)^{d-1}}{(d-2)^d}.$$

Our parameters satisfy this inequality for sufficiently large q: $d\lambda \to 4$ whereas $d\lambda_c(d) \to e < 4$. The deterministic map from assignments to independent sets cannot increase total-variation distance, so an approximate assignment sampler would give an approximate hard-core sampler.

Nevertheless, the same instances obey

$$p_\star \Delta^{2-\varepsilon} = q^{-2}(8q-2)^{2-\varepsilon} \le 8^{2-\varepsilon} q^{-\varepsilon} \longrightarrow 0.$$

To make the parameter choice explicit, let s_0 be an integer such that $\lambda > \lambda_c(d)$ whenever $q = 2^s$, $d = 4q$, and $s \ge s_0$. Such an s_0 exists by the strict limiting inequality above. For the fixed constants $\varepsilon \in (0,2)$ and $C > 0$, choose

$$s \ge \max\left\{1, s_0, \left\lceil \frac{\log_2(8^{2-\varepsilon}/C)}{\varepsilon} \right\rceil \right\}.$$

Then $8^{2-\varepsilon} 2^{-\varepsilon s} \le C$, so every formula in this construction satisfies the promised bound $p_\star \Delta^{2-\varepsilon} \le C$. The same choice also ensures $\lambda > \lambda_c(d)$. Crucially, choose s once for the proposed criterion: it may depend on ε, C, but not on the input graph size or the requested sampling accuracy. Hence q, d, and λ are fixed parameters throughout the reduction.

Write $N = |V|$. The constructed formula Φ_G has sN variables, $dN/2$ clauses, and sdN literal occurrences. It can be written by listing the $2s$ endpoint variables for each edge. With binary variable indices, its encoding length L is $O(sdN \log(sN+1)) = O(N \log(N+1))$, where the second equality uses the fixed values of s, d. Thus both constructing Φ_G and scanning its blocks to recover an independent set take polynomial time in the graph size. A large constant value of d does not invalidate the reduction. As N grows, we obtain a hard family of graphs of that fixed degree.

Now suppose the sampler ruled out by Proposition C.10 exists. Given G and an accuracy $\eta \in (0,1)$, construct Φ_G, run that sampler with the same accuracy η, and apply the zero-block map above to its output. Let U_G denote the uniform distribution on satisfying assignments and let ν_G be the sampler's output distribution, so $\mathrm{d_{TV}}(\nu_G, U_G) \le \eta$. Write f for the map to independent sets, $\pi_{G,\lambda}$ for the target hard-core distribution, and $\widehat{\pi}_G$ for the distribution of the mapped output. For every collection $\mathcal{B}$ of independent sets,

$$\left|\widehat{\pi}_G(\mathcal{B}) - \pi_{G,\lambda}(\mathcal{B})\right| = \left|\nu_G(f^{-1}(\mathcal{B})) - U_G(f^{-1}(\mathcal{B}))\right| \le \eta.$$

Taking the supremum over $\mathcal{B}$ gives $\mathrm{d_{TV}}(\widehat{\pi}_G, \pi_{G,\lambda}) \le \eta$. The construction, sampling, and projection together take time polynomial in N and η^{-1}. This would be a polynomial-time approximate sampler for the hard-core model at the fixed parameters d, λ above its hardness threshold, contradicting the cited result unless $\mathrm{NP} = \mathrm{RP}$.

Finally, suppose instead that an exact uniform sampler exists with expected running time at most $T(L)$ on every promised formula of encoding length L, for one fixed polynomial $T(L) \ge 1$. Its running time may vary with its random choices. To construct an approximate sampler on Φ_G, run it for at most

$$B = \left\lceil \frac{T(L)}{\eta} \right\rceil$$

computation steps. If it finishes, return its output. Otherwise, stop it and return the all-ones assignment, which satisfies every clause of Φ_G. If H denotes the exact sampler's random running time, Markov's inequality gives

$$\mathbb{P}(H > B) \le \frac{\mathbb{E}H}{B} \le \frac{T(L)}{B} \le \eta.$$

Couple the original and truncated samplers by using the same random choices until the cutoff. Let X be the original sampler's eventual output and $\widehat{X}$ the truncated output. On $\{H \le B\}$ they agree exactly, so the coupling inequality bounds the truncated output distribution ν_{cut} by

$$\mathrm{d}_{\mathrm{TV}}(\nu_{\mathrm{cut}}, U_G) \le \mathbb{P}(\widehat{X} \ne X) \le \mathbb{P}(H > B) \le \eta.$$

The truncated sampler always stops within $B + O(L)$ steps, including writing the fallback assignment. This is polynomial in L and η^{-1} and yields the approximate sampler just ruled out. Thus a uniform polynomial bound on the expected runtime of exact sampling would also imply $\mathrm{NP} = \mathrm{RP}$. □

Scope of the barrier. For neighboring clauses, the overlap event in (2) holds exactly when their shared block is zero. Therefore

$$r_\star = q^{-1}, \qquad p_\star \Delta^2 = \left(8 - \frac{2}{q}\right)^2 \longrightarrow 64, \qquad r_\star \Delta = 8 - \frac{2}{q} \longrightarrow 8.$$

Any fixed reduction of the exponent 2 admits these hard instances, regardless of how small the positive constant C is. This construction leaves the optimal constant at exponent 2 open. These instances violate the GJL condition $3er_\star\Delta \le 1$, and the optimal exponent could differ when that condition is retained. Here $p_\star = r_\star^2$, so imposing it gives $p_\star\Delta^2 \le 1/(9e^2) < 1/(6e)$ and restores the first GJL condition. The instances lie within the LLL existence regime, since $ep_\star(\Delta + 1) = e(8q-1)/q^2 \to 0$, and the all-ones assignment is feasible.

D THE DISTRIBUTION OF THE RETURNED SAMPLES

Throughout this appendix, $\Omega = \prod_{v=1}^n \Sigma_v$ is finite, p is a fixed probability distribution on Ω, and $a = p(\mathcal{F}) > 0$. Write $p_{\mathcal{F}} = p(\cdot \mid \mathcal{F})$. Some assignments may have probability zero under p. The variables used by each constraint and the deterministic order used to construct Res are fixed. Exactness assumes access to the conditional probabilities of p. We analyze learned prediction errors separately.

D.1 JOINT CONDITIONAL RESAMPLING

For a mask B, let z be an assignment to the unmasked coordinates with $p(X_{B^c} = z) > 0$. The ideal resampling step draws the masked coordinates from

$$K_B^p(z, u) = p(X_B = u \mid X_{B^c} = z). \tag{64}$$

This is a joint distribution: it specifies the probability of the whole masked block, including dependencies between its coordinates. We can sample from it one coordinate at a time if the model gives the exact conditional probability at each step. Under suitable training assumptions, ideal masked predictions give these data conditionals (Ou et al., 2025). How we use the predictions determines the distribution of the completed block. For an ordering $v_1, \dots, v_\ell$ of B, the chain rule gives

$$K_B^p(z, u) = \prod_{j=1}^{\ell} p\big(X_{v_j} = u_{v_j} \mid X_{B^c} = z,\, X_{v_i} = u_{v_i}\ (i < j)\big)\,. \tag{65}$$

The repair set remains fixed until this resampling step is complete. Every partial assignment reached with positive probability has a well-defined next conditional distribution. These probabilities condition on the visible values, but not on the event that selected B. We examine the resulting selection bias below.

Independent parallel draws give (64) only if the masked coordinates are conditionally independent given the visible values. Otherwise, even exact predictions for individual coordinates give a different resampling distribution from sequential draws. Writing path probabilities as products requires the relevant independence conditions. Exact joint resampling alone does not provide them. Averaging parallel draws over visible assignments can also introduce dependence, so bounds conditional on the repair set must cover entire path events.

Exact joint resampling can still give biased outputs. Consider binary (X, Y) with $p(0,0) = p(1,1) = 3/8$ and $p(1,0) = p(0,1) = 1/8$, and require $X = 0$. Start from p and, whenever $X = 1$, redraw X from its exact conditional given Y. This procedure stops, with certificate $\kappa = 3/4$: after any repair, its probability of remaining infeasible is at most $3/4$. But Y is never changed, so it remains equally likely to be zero or one. In the desired data distribution conditioned on $X = 0$, $Y = 0$ is three times as likely as $Y = 1$. Thus the output probabilities are $Q(0,0) = Q(0,1) = 1/2$, whereas $p_{\mathcal{F}}(0,0) = 3/4$ and $p_{\mathcal{F}}(0,1) = 1/4$. The output is therefore at TV distance $1/4$ from the target, even with exact initialization and conditional resampling. Choosing which coordinates to repair can bias the output.

D.2 How the repair set depends on the assignment

The following property of the repair-set rule is used in general PRS (Guo et al., 2019). It describes changes to an assignment that leave the selected repair set unchanged. We prove it for the rule used here, which inspects constraints in a fixed order within each batch.

Lemma D.1 (Changes outside the repair set)**.** *Let* $\mathrm{Res}(x) = S$. *Suppose* x' *agrees with* x *on* $B_S = V(S)$ *and* $\mathrm{Bad}(x') \subseteq S$. *Then* $\mathrm{Res}(x') = S$.

Proof. Every constraint in S uses only variables in B_S. It therefore has the same violation status under x and x'. Every violated constraint under x belongs to S, and by assumption none outside S is violated under x'. The two repair-set constructions thus start from the same violated constraints.

Now follow the two constructions batch by batch, and in the fixed inspection order within each batch. Suppose their selected and excluded sets agree so far. Then they have the same boundary for the current batch and inspect the same next constraint. The test of whether that constraint can be violated uses fixed values only from the currently selected variables, hence from B_S. The assignments agree on those values, so the test has the same outcome in both constructions. Their selected and excluded sets remain equal. By induction, they return the same final set. □

For any event set R, define

$$E_R = \bigcap_{j \notin \Gamma^+(R)} A_j^c. \tag{66}$$

The event E_R says that every constraint whose scope is disjoint from B_R is satisfied. It therefore depends only on coordinates outside B_R. In particular, $E_{[m]} = \Omega$, $E_\varnothing = \mathcal{F}$, and $\mathcal{F} \subseteq E_R$, so $p(E_R) > 0$. Before checking the first proposal $X_0 \sim p$, no previous repair exists: we use bookkeeping $R = [m]$, so $E_{[m]} = \Omega$ imposes no restriction. The first actual repair set is $S = \mathrm{Res}(X_0)$. Subsequently, R records the set just repaired. For an assignment u to B_S, define two functions taking values zero or one:

$$f_S(u) = \mathbf{1}\{\text{there exists an exterior } z \text{ with } \mathrm{Res}(u, z) = S\}, \tag{67}$$

$$f_{R,S}(u) = f_S(u) \prod_{j \in S \setminus \Gamma^+(R)} \mathbf{1}\{(u, z) \notin A_j\}. \tag{68}$$

The function $f_S(u)$ records whether some completion of u would select exactly S. The function $f_{R,S}(u)$ additionally requires the constraints in $S \setminus \Gamma^+(R)$ to be satisfied. These extra constraints use only variables in B_S, so the last expression does not depend on z.

Lemma D.2 (Separating conditions inside and outside the repair set)**.** *For every* $R, S \subseteq [m]$ *and* $x \in \Omega$,

$$\mathbf{1}\{x \in E_R\}\mathbf{1}\{\mathrm{Res}(x) = S\} = f_{R,S}(x_{B_S})\mathbf{1}\{x \in E_S\}. \tag{69}$$

Proof. First show $\mathbf{1}\{\mathrm{Res}(x) = S\} = f_S(x_{B_S})\mathbf{1}\{x \in E_S\}$. The forward implication follows because every violated constraint belongs to S. For the reverse implication, choose an assignment $\widetilde{x}$ that agrees with x on B_S and selects S. Every constraint in $\Gamma(S)$ must be satisfied whenever the coordinates in B_S have these values. Indeed, the construction excluded it because it could not be violated with the then-fixed values. Fixing more coordinates cannot change that conclusion. The same neighboring constraints are therefore satisfied under x. Constraints outside $\Gamma^+(S)$ are satisfied because $x \in E_S$. Thus $\mathrm{Bad}(x) \subseteq S$. Lemma D.1 applies to $x, \widetilde{x}$ and gives $\mathrm{Res}(x) = S$.

When the selected set is S, every violated constraint belongs to S. Requiring E_R therefore amounts to requiring that the constraints in $S \setminus \Gamma^+(R)$ are satisfied. This gives (69). □

D.3 When repair-set selection preserves the target distribution

Given a repair set S, the ideal sampler masks $B_S = V(S)$, the variables appearing in its constraints. It jointly resamples this block and retains the outside assignment $X_{B_S^c} = z$. These outside variables stay fixed during this step but may be resampled later.

Selecting S can change the distribution of retained values. Suppose the first proposal has two equally likely outside assignments and S is selected with probabilities $1/2$ and $1/4$, respectively. Among runs selecting S, their probabilities become $2/3$ and $1/3$, and resampling B_S leaves these proportions unchanged. The condition below excludes such reweighting beyond the constraints known to be satisfied. With exact initialization and joint resampling from p, it provides a distributional certificate: almost-sure termination and output exactly $p_{\mathcal{F}}$.

Let S be a nonempty set of constraints that could be selected. For a visible assignment z with $p(X_{B_S^c} = z) > 0$ that satisfies E_S, put

$$g_{R,S}(z) := \mathbb{P}_p\big(E_R,\ \mathrm{Res}(X) = S \mid X_{B_S^c} = z\big) \tag{70}$$

$$= \mathbb{E}_p\big[f_{R,S}(X_{B_S}) \mid X_{B_S^c} = z\big]\,. \tag{71}$$

The equality follows from Lemma D.2. The function $g_{R,S}(z)$ measures how likely a draw from p is to satisfy E_R and select S, given the visible values z. If p is a product distribution, this probability does not depend on z, because the function inside the expectation in (71) involves only the masked coordinates. This is the property used by general PRS with a fixed product distribution (Guo et al., 2019). For correlated data, constancy of these selection probabilities in z is enough. Formally, for each fixed pair R, S, we require $g_{R,S}(z)$ to be constant over all z that have positive probability under p and satisfy E_S. The event E_S restricts only unmasked variables. Constancy rules out additional weighting among these allowed assignments.

Lemma D.3 (Distribution after one resampling step)**.** *Suppose $X \sim p(\cdot \mid E_R)$. Condition on $\mathrm{Res}(X) = S$, an event of positive probability, and resample B_S from* (64)*, leaving the visible coordinates unchanged. The probability of the resulting assignment y is proportional to*

$$p(y)\mathbf{1}\{y \in E_S\}\, g_{R,S}(y_{B_S^c}). \tag{72}$$

The resulting distribution equals $p(\cdot \mid E_S)$ if and only if $g_{R,S}$ has the same value for every visible assignment satisfying E_S with positive probability under p.

Proof. Write $p_{B_S^c}$ for the distribution of the visible coordinates under p. Before conditioning on the selection of S, the probability of both selecting S and observing z is

$$\mathbb{P}(\mathrm{Res}(X) = S,\ X_{B_S^c} = z) = \frac{p_{B_S^c}(z)\mathbf{1}\{z \in E_S\}\, g_{R,S}(z)}{p(E_R)}. \tag{73}$$

Selecting S implies E_S, so visible assignments that fail E_S have probability zero. Multiply this expression by $p(y_{B_S} \mid z)$ and normalize. This gives (72). The event E_S depends only on the visible coordinates. Every visible assignment satisfying E_S that has positive probability under p also has positive probability under $p(\cdot \mid E_S)$. Thus the final factor in (72) leaves the desired distribution unchanged exactly when it is constant on those assignments. Since S is selected with positive probability, that constant must be positive. □

Proposition D.4 (Exact sampling when repair-set selection preserves retained probabilities)**.** *Assume $p(\mathcal{F}) > 0$, $X_0 \sim p$, and each repair draws from $p(X_{B_t} \mid X_{B_t^c})$. If the probability in* (70) *is constant across eligible retained assignments for every pair of repair sets that can be encountered, Algorithm 1 stops with probability one and outputs exactly $p_{\mathcal{F}}$.*

Proof of Proposition D.4. Set $R_{-1} = [m]$. At round t, let X_t be the current assignment, select $R_t = \mathrm{Res}(X_t)$, and obtain X_{t+1} by resampling its mask. After stopping, retain the returned assignment and the empty repair set. We claim that for every $t \geq 0$ and every R with positive probability,

$$\mathcal{L}(X_t \mid R_{t-1} = R) = p(\cdot \mid E_R). \tag{74}$$

This holds at $t = 0$ because initialization is from p and $E_{[m]} = \Omega$. Suppose it holds at round t. Condition further on $R_t = S \neq \varnothing$. The assumed selection property makes $g_{R,S}$ constant over the possible visible assignments, so Lemma D.3 gives $\mathcal{L}(X_{t+1} \mid R_{t-1} = R, R_t = S) = p(\cdot \mid E_S)$. This distribution is the same for every possible R. Averaging over R therefore proves (74) at round $t + 1$ for nonempty S. If $S = \varnothing$, conditioning the current distribution on feasibility gives $p_{\mathcal{F}}$, since $\mathcal{F} \subseteq E_R$. Retaining this assignment proves the same invariant for the empty set.

Before a reached check, conditional on the preceding repair set R, the probability of stopping is $p(\mathcal{F})/p(E_R) \geq a$. Averaging over R gives the same lower bound conditional only on reaching that check. Thus the probability of needing more than k checks is at most $(1 - a)^k$, and termination occurs with probability one. For every possible stopping round and preceding set, the returned assignment has distribution $p_{\mathcal{F}}$. Their mixture is also $p_{\mathcal{F}}$. □

In the main theorem's notation, this gives $\mathcal{L}(X_{t+1} \mid R_t = S) = p(\cdot \mid E_S)$. Here we average over current assignments selecting S without fixing their visible values.

When the selection property holds. Product distributions satisfy this property automatically, but correlated distributions need not. With one constraint on X and violation event A, it requires $p(A \mid Z = z)$ to be the same for every positive-probability assignment of the unmasked variables Z. Individual outcome probabilities can still depend on z. Sparse constraints, rare violations, and exact joint resampling alone do not ensure this property. Constancy of $g_{R,S}$ is necessary and sufficient to preserve the distribution in (74) after a selected transition from R to S. The final output can still be exact without this constancy, because biases from different sequences of repair sets can cancel (see Appendix D.7). Checking every pair R, S gives a finite sufficient test. It is enough to check pairs that occur with positive probability at some round. When this property holds, the correction can accept every repair in Appendix D.6, so no restart overhead is needed.

D.4 How resampling errors affect the output distribution

Suppose an ideal sampler satisfies Proposition D.4, or another exactness result proved for the same repair-set and stopping rules. Its initial distribution is p. The implemented sampler starts from q_0, uses a fixed memoryless resampling kernel, and terminates with probability one. For a current assignment x, put $B = V(\mathrm{Res}(x))$ and $z = x_{B^c}$. Compare the implemented masked-block distribution $K_B^{\mathrm{act}}(z, \cdot)$ with the ideal distribution $K_B^p(z, \cdot)$, and write

$$\delta_0 = \mathrm{d_{TV}}(q_0, p), \qquad e(x) = \mathrm{d_{TV}}\big(K_B^{\mathrm{act}}(z, \cdot), K_B^p(z, \cdot)\big)\,. \tag{75}$$

Both transitions copy the same visible coordinates, so this is also the distance between their next complete-assignment distributions. Set $e(x) = 0$ for feasible x. The implemented sampler may reach visible assignments that have probability zero under p. At those assignments, the ideal conditional distribution is undefined. At these states, choose any fixed resampling distribution that copies the same visible coordinates. This makes (75) well-defined at every state of the implemented sampler. Such a visible assignment cannot occur before the two coupled samplers first disagree.

Lemma D.5 (Accumulation of resampling errors)**.** *If Q is the distribution returned by the implemented sampler, then*

$$\mathrm{d_{TV}}(Q, p_{\mathcal{F}}) \leq \min\left\{1,\, \delta_0 + \mathbb{E}_{\mathrm{act}} \sum_{t<\tau} e(X_t)\right\}. \tag{76}$$

Proof. Construct the two samplers together so that each retains its prescribed distribution. Choose their initial assignments to disagree with probability exactly δ_0. Until their first disagreement, both have the same current assignment X_t. Couple their next assignments to disagree with probability exactly $e(X_t)$. Such choices are possible by the maximal coupling characterization of total variation distance. Let D_t denote agreement through round t, just before resampling. On D_t the two repair sets, masks, and stopping decisions agree. The probability of first disagreement during resampling step t is at most

$$\mathbb{E}[\mathbf{1}\{D_t\}\mathbf{1}\{t < \tau\}e(X_t)] \leq \mathbb{E}_{\mathrm{act}}[\mathbf{1}\{t < \tau\}e(X_t)]\,.$$

The expectation on the right uses the implemented sampler's own distribution. Sum over all possible rounds of first disagreement and add the initial disagreement probability δ_0. If the samplers never

disagree, they terminate with the same output. Their output distributions can therefore differ in total variation by at most the probability of a disagreement. This proves the claim, which also remains valid when the expected sum is infinite. □

For an attainable nonempty repair set S, define $\bar{e}_t(S) = \mathbb{E}_{\mathrm{act}}[e(X_t) \mid R_t = S]$, the average local resampling error at round t conditional on the implemented sampler selecting S.

Theorem D.6 (Output error of the learned sampler)**.** *Suppose the ideal version of Algorithm 1 starts from p, uses exact joint data resampling, and is known to terminate almost surely with output $p_{\mathcal{F}}$, for example by Proposition D.4. Suppose the learned sampler satisfies Theorem 3.1, and $\bar{e}_t(S) \le \varepsilon W(S)$ at every round t and attainable nonempty S under the learned sampler. Then its output distribution q satisfies*

$$\mathrm{d_{TV}}(q, p_{\mathcal{F}}) \le \min\left\{1, \delta_0 + \frac{\varepsilon\, \mathbb{E}_{\mathrm{actual}} W(R_0)}{1-\kappa}\right\}. \tag{77}$$

Proof of Theorem D.6. Retain $R_t = \varnothing$ and set $e(X_t) = 0$ after termination. Then $\{t < \tau\} = \{R_t \neq \varnothing\}$, and the assumed bound $\bar{e}_t(S) = \mathbb{E}_{\mathrm{act}}[e(X_t) \mid R_t = S] \le \varepsilon W(S)$ gives, by nonnegativity and the tower property,

$$\begin{aligned}\mathbb{E}_{\mathrm{act}} \sum_{t<\tau} e(X_t) &= \sum_{t\ge 0} \mathbb{E}_{\mathrm{act}}[\mathbf{1}\{R_t \neq \varnothing\} \mathbb{E}_{\mathrm{act}}[e(X_t) \mid R_t]] \\ &\le \varepsilon\, \mathbb{E}_{\mathrm{act}} \sum_{t<\tau} W(R_t) \le \frac{\varepsilon\, \mathbb{E}_{\mathrm{act}} W(R_0)}{1-\kappa}.\end{aligned}$$

The last inequality is Corollary A.6. Insert this bound in Lemma D.5. □

The assumption bounds the average of $e(X_t)$ over assignments that actually select S at round t. Averaging the two kernels first and then taking their TV distance can give a smaller value, so it does not provide the error bound needed by this coupling proof. The statewise inequality $e(x) \le \varepsilon W(\mathrm{Res}(x))$ is a stronger sufficient check, because it implies the required conditional expectation bound.

The resampling error measures every difference from the ideal joint distribution. These differences may come from approximate predictions, sampling coordinates independently in parallel, or adjusting probabilities to satisfy the termination certificate. For fixed finite weights and $1-\kappa$ bounded below by a positive constant, the output-error bound converges to zero as $\delta_0, \varepsilon \to 0$.

D.5 Exact acceptance without selection-probability calculations

The correction in Theorem 4.1 uses the values about to be replaced. At a nonterminal check, let $S = \mathrm{Res}(x)$, $B = B_S$, $u = x_B$, and $z = x_{B^c}$. For each nonempty S, fix a nonnegative function h_S, independent of the retained assignment and previous repair history, with

$$0 \le h_S(u) \le p(X_B = u \mid X_{B^c} = z) \quad \text{for all } u, z \text{ with } p_{B^c}(z) > 0. \tag{78}$$

The function need not sum to one. Before resampling, accept with probability

$$p_{\mathrm{accept}}(u, z; S) = \frac{h_S(u)}{p(X_B = u \mid X_{B^c} = z)}. \tag{79}$$

If accepted, draw new values of X_B jointly from $p(\cdot \mid z)$ and retain z. Otherwise, discard the assignment and restart independently from p. Feasible assignments are returned without an acceptance check. The ratio is well-defined at every state of positive probability. If $h_S(u) = 0$, reject without evaluating the denominator.

Theorem D.7 (Exact sampling by correcting the old masked values)**.** *Suppose $p(\mathcal{F}) > 0$, initialization and restarts draw exactly from p, accepted repairs use its exact joint conditionals, and* (78) *holds. Repair with the acceptance rule* (79) *terminates almost surely and outputs exactly $p_{\mathcal{F}}$. If T counts repair-set checks, including the final check, then $\mathbb{P}(T > k) \le (1 - p(\mathcal{F}))^k$ and $\mathbb{E}T \le 1/p(\mathcal{F})$.*

Proof. Store the previous repair set R, initially $[m]$ and reset to $[m]$ after every restart. At every reached check, conditional on its index and the stored set R, we maintain $X \sim p(\cdot \mid E_R)$. This holds initially since $E_{[m]} = \Omega$. By Lemma D.2, the mass of an old assignment (u, z) selecting S is

$$\frac{p_{B^c}(z)p(u \mid z)}{p(E_R)} f_{R,S}(u)\mathbf{1}\{z \in E_S\}.$$

Multiplication by the acceptance probability cancels $p(u \mid z)$. Summing over old u and then resampling the block gives

$$\mathbb{P}(S, \text{accept}, X' = y \mid R, \text{check reached}) = \frac{d_{R,S}}{p(E_R)} p(y)\mathbf{1}\{y \in E_S\}. \tag{80}$$

Here $d_{R,S} = \sum_u f_{R,S}(u)h_S(u)$ does not depend on z. Thus every accepted arrival with stored set S has distribution $p(\cdot \mid E_S)$. A restart has distribution p, which gives the same invariant for stored set $[m]$. Mixing these incoming possibilities proves the next-check invariant. At a reached check with stored set R, stopping has probability $p(\mathcal{F})/p(E_R) \geq p(\mathcal{F})$ and gives output distribution $p_{\mathcal{F}}$. The geometric tail proves termination and the check bound. Mixing over stopping checks proves exactness. □

A neural construction of the lower bound. Fix a sequential reveal order $v_1, \ldots, v_b$ for each block B. Write $q_i(a \mid z, u_{<i})$ for the denoiser's prediction at the next variable. Exactness requires compatible conditionals of the same p, so that $p(u \mid z) = \prod_{i=1}^{b} q_i(u_i \mid z, u_{<i})$.

At the old prefix $u_{<i}$, compute the actual logits $s_i(a, z)$ and reference logits $r_i(a)$ by masking every outside variable in a second network evaluation. The reference probabilities $\nu_i = \mathrm{softmax}(r_i)$ depend on the old inside prefix, but do not depend on z. With the prefix and the actual mask pattern fixed, compute bounds $L_{ia} \leq s_i(a, z') \leq U_{ia}$ for *every* outside assignment z'. These bounds must not fix the current outside values. Define

$$D_i(u_{<i}) = \max_j\{U_{ij} - r_i(j)\} - \min_a\{L_{ia} - r_i(a)\}, \qquad h_S(u) = \prod_{i=1}^{b} e^{-D_i(u_{<i})}\nu_i(u_i \mid u_{<i}). \tag{81}$$

For $d_i(a, z) = s_i(a, z) - r_i(a)$, the definition gives $d_i(j, z) - d_i(a, z) \leq D_i$. Hence

$$e^{-D_i} \leq \frac{q_i(a \mid z, u_{<i})}{\nu_i(a \mid u_{<i})} \leq e^{D_i},$$

which proves (78). This construction permits nonuniform probabilities and dependence among variables inside the block.

Generating the acceptance decision. For each old value $a = u_i$, independently draw $J \sim \nu_i$, then draw a coin with success probability $\exp\{d_i(J, z) - d_i(a, z) - D_i\}$. Its exponent is nonpositive and

$$\mathbb{E}_{J\sim\nu_i}\left[e^{d_i(J,z)-d_i(a,z)-D_i}\right] = e^{-D_i}\frac{\nu_i(a)}{q_i(a \mid z, u_{<i})}. \tag{82}$$

Reject on the first failure and accept the repair if all b coins succeed. Their product is exactly (79).

Polynomial cost of the decision. Use a specified finite-precision neural computation with rational logits of polynomial bit length. Sound interval propagation encloses the same computation, including its rounding. For example, enclose every unknown outside embedding coordinate by its vocabulary minimum and maximum. Propagate affine bounds using positive and negative weights and ReLU bounds by applying the activation to both endpoints. Products use their four endpoint products. Attention probabilities can conservatively be bounded in $[0, 1]$, and normalization uses its positive stabilizing constant. With polynomial-bit numerical operations and sound outward rounding, this gives a polynomial bound pass through the computation graph. Such interval bounds are standard for neural verification, including discrete symbol substitutions (Gowal et al., 2019; Huang et al., 2019). Looser bounds may reduce acceptance frequency, but remain valid and can still be evaluated in polynomial time.

The random decisions also have an explicit bit implementation. For rational $\gamma \geq 0$, Algorithm 2 and Proposition 14 of Canonne et al. (2020) generate an exact $\mathrm{Bernoulli}(e^{-\gamma})$ using rational coins,

with constant expected arithmetic-operation count. Their factorial and geometric loop tails give polynomial expected bit cost in the input length. An exact softmax draw from k rational logits follows by proposing a uniform label j and accepting it with probability $e^{r_j - \max_a r_a}$. At most k trials are needed in expectation. Thus one repair decision uses at most b actual network passes, b reference passes, b interval passes, and polynomial expected random-bit work. Exactness still requires the denoiser to give the conditional probabilities of the target distribution. An exact acceptance test alone does not ensure this.

Acceptance frequency and restart overhead. The same ratio bounds give

$$p_{\mathrm{accept}}(u, z; S) \geq \exp\left\{-2 \sum_{i=1}^{b} D_i(u_{<i})\right\}. \tag{83}$$

A uniform constant bound on the cumulative D_i therefore supplies a constant acceptance lower bound. Corollary D.10 below combines any lower bound $p_{\mathrm{accept}} \geq \alpha > 0$ with the uncorrected repair bound $\mathbb{E}\tau \leq L$ to give at most $(L+1)\alpha^{-L}$ expected checks. Together with the explicit decision cost above, this proves Theorem 4.1. This lower bound must hold for the acceptance probability in (79). The alternative correction in Appendix D.6 uses a different probability and needs its own bound.

Symmetric bounds under the data distribution.

Corollary D.8 (Symmetric PRS for the data distribution)**.** *Assume $p_{\mathrm{data}}(\mathcal{F}) > 0$ and use the corrected sampler defined above with $p = p_{\mathrm{data}}$. Let $\Delta \geq 2$. Choose common worst-case bounds $\bar{p}, \bar{r}$ for the path events in* (41) *(Appendix C.2) under $p_{\mathrm{data}}(\cdot \mid E_S)$, over all attainable nonempty repair sets S and the paths rooted in $\Gamma^+(S)$. For product p_{data}, it suffices to take the worst-case violation and overlap probabilities $\bar{p} = \max_i p_{\mathrm{data}}(A_i)$ and $\bar{r} = \max_{i \sim j} p_{\mathrm{data}}(U_{ij})$. If*

$$6e\,\bar{p}\,\Delta^2 \leq 1, \qquad 3e\,\bar{r}\,\Delta \leq 1, \tag{84}$$

and $p_{\mathrm{accept}} \geq \alpha$ at every attainable repair for a fixed constant $\alpha \in (0, 1]$, then the sampler terminates almost surely and outputs exactly $p_{\mathrm{data}}(\cdot \mid \mathcal{F})$ with polynomial expected restart overhead.

Proof of Corollary D.8. Write $p = p_{\mathrm{data}}$. An attempt begins with an independent draw from p and ends at the first rejected repair or feasible output. The assignment at its first check has distribution $p = p(\cdot \mid E_{[m]})$. Suppose that, conditional on a particular accepted repair-set history ending at R, the next assignment has distribution $p(\cdot \mid E_R)$. The cancellation in (80) then applies conditional on this same history: selecting and accepting S, followed by exact joint resampling, gives a multiple of $p(y)\mathbf{1}\{y \in E_S\}$. Normalizing shows that the new assignment has distribution $p(\cdot \mid E_S)$. Induction therefore proves this invariant conditional on every accepted repair-set history of positive probability. At any terminating check, conditioning further on $\mathcal{F} \subseteq E_R$ gives output distribution $p(\cdot \mid \mathcal{F})$.

Define the initial repair-set probabilities and conditional-data transitions by

$$\pi(S) = p(\mathrm{Res}(X) = S), \qquad P_R(S) = p(\mathrm{Res}(X) = S \mid E_R) \quad (R \neq \varnothing).$$

These conditionals exist because $p(E_R) \geq p(\mathcal{F}) > 0$. For nonempty S, abbreviate its acceptance probability at x by $a_S(x) = p_{\mathrm{accept}}(x_{B_S}, x_{B_S^c}; S)$, and define

$$K_{\mathrm{init}}(S) = \mathbb{E}_{X \sim p}[\mathbf{1}\{\mathrm{Res}(X) = S\} a_S(X)],$$
$$K_R(S) = \mathbb{E}_{X \sim p(\cdot \mid E_R)}[\mathbf{1}\{\mathrm{Res}(X) = S\} a_S(X)].$$

Each expectation only uses a_S on assignments selecting S. By the invariant, these are the probabilities of selecting and accepting S initially and after an accepted repair of R, respectively. Feasible termination has probability $\pi(\varnothing)$ initially and $P_R(\varnothing)$ subsequently. The remaining mass is rejection. At every attainable check the acceptance assumption gives

$$\alpha\pi(S) \leq K_{\mathrm{init}}(S) \leq \pi(S), \qquad \alpha P_R(S) \leq K_R(S) \leq P_R(S).$$

For the analysis, let J_0 have distribution π and define an auxiliary chain with transitions P_R and absorbing state $\varnothing$. Every nonempty set with $\pi(S) > 0$ is selected initially and, because $\alpha > 0$, can be accepted. If R can be reached through accepted repairs and $P_R(S) > 0$, the invariant makes

S attainable at the next check, and its acceptance again has positive probability. Induction shows that all nonempty sets reachable in the auxiliary chain are attainable through accepted repairs. The assumed data bounds and the displayed comparisons therefore hold at every auxiliary state that is used below.

If $X \in E_R$, then $\mathrm{Bad}(X) \subseteq \Gamma^+(R)$. Lemma A.2 bounds $|\mathrm{Res}(X)|$ by the sum of indicators of induced path events rooted in $\Gamma^+(R)$. Taking expectations under $p(\cdot \mid E_R)$ and summing the symmetric path bounds as in Appendix C.2 gives

$$\mathbb{E}[|J_{t+1}| \mid J_t = R] = \sum_S P_R(S)|S| \le \tfrac{1}{2}|R|.$$

Let $\tau_* = \inf\{t \ge 0 : J_t = \varnothing\}$, the number of nonempty sets on an auxiliary trajectory. Since $|J_0| \le m$, the logarithmic-potential argument in (28) gives

$$\mathbb{E}\tau_* \le L := 2(1 + \log m).$$

In particular, the auxiliary chain terminates almost surely. Its analysis starts from J_0, the repair set of a fresh proposal. The implemented sampler continues to use the joint conditional repairs and restarts specified in Section 4.

For $k \ge 1$, a terminating auxiliary path with nonempty sets $s_0, \dots, s_{k-1}$ has probability

$$w = \pi(s_0) \left(\prod_{i=0}^{k-2} P_{s_i}(s_{i+1}) \right) P_{s_{k-1}}(\varnothing).$$

By the history-conditional invariant, the probability that one corrected attempt accepts exactly this sequence and then terminates is

$$\widetilde{w} = K_{\mathrm{init}}(s_0) \left(\prod_{i=0}^{k-2} K_{s_i}(s_{i+1}) \right) P_{s_{k-1}}(\varnothing) \ge \alpha^k w.$$

Empty products are one. For $k = 0$, both probabilities equal $\pi(\varnothing)$. Summing over these disjoint terminating paths and using convexity of $t \mapsto \alpha^t$ gives the probability q that an attempt succeeds:

$$q \ge \mathbb{E}[\alpha^{\tau_*}] \ge \alpha^{\mathbb{E}\tau_*} \ge \alpha^L > 0.$$

For $\alpha = 1$ the same inequalities hold with equality in the powers.

Let H count checks in one attempt, including its final rejection or feasibility check. Reaching check $h + 1$ requires h accepted nonempty repair sets. The corresponding prefix probabilities use K_{init} and K_R, each bounded above by its auxiliary counterpart. Summing over prefixes gives

$$\mathbb{P}(H \ge h + 1) \le \mathbb{P}(\tau_* \ge h) \quad (h \ge 0), \qquad \mathbb{E}H \le \mathbb{E}(\tau_* + 1) \le L + 1.$$

As in Theorem 4.1, let N count fresh starts, including initialization, and T all repair-set checks, including the final successful check. Fresh initialization and subsequent randomness are independent across attempts, so N is geometric with success probability q. If H_j denotes the length of attempt j, the event $\{N \ge j\}$ depends only on earlier attempts and is independent of H_j. Thus, even though an attempt's length and success can be dependent,

$$\begin{aligned} \mathbb{E}N &= q^{-1} \le \alpha^{-L}, \\ \mathbb{E}T &= \sum_{j \ge 1} \mathbb{E}[\mathbf{1}\{N \ge j\} H_j] \\ &= \sum_{j \ge 1} (1 - q)^{j-1} \mathbb{E}H = \frac{\mathbb{E}H}{q} \le (L + 1)\alpha^{-L}. \end{aligned}$$

Finite expected checks imply almost-sure termination, and the invariant already proved gives exact output. For fixed $\alpha > 0$, $\alpha^{-L} = \alpha^{-2} m^{2\log(1/\alpha)}$ is polynomial in m. The neural procedure above gives polynomial expected acceptance cost. With polynomial conditional expected costs for the original operations, the total expected running time is polynomial as well.

Finally, for product p, Lemma C.4 supplies the conditional-data path bounds from the unconditional violation and overlap bounds in the corollary. Independence gives $p(X_{B_S} = u \mid X_{B_S^c} = z) = p(X_{B_S} = u)$, so choosing $h_S(u) = p(X_{B_S} = u)$ makes every attainable repair accepted with probability one. The corrected sampler then reduces to product PRS. □

Algorithm 3 Repair with an acceptance test and restarts

Draw $X \sim p$; set $R \leftarrow [m]$.
loop
  $S \leftarrow \mathrm{Res}(X)$.
  if $S = \varnothing$ **then**
    return X.
  end if
  $z \leftarrow X_{B_S^c}$.
  if $c_{R,S} > 0$ **then**
    Draw an acceptance coin with probability $c_{R,S}/g_{R,S}(z)$.
  else
    Set the coin outcome to rejection.
  end if
  if accepted **then**
    Draw $X_{B_S} \sim p(X_{B_S} \mid z)$; retain z; set $R \leftarrow S$.
  else
    Draw a fresh independent $X \sim p$; set $R \leftarrow [m]$.
  end if
end loop

D.6 When exactness requires restarts

An alternative acceptance check corrects the selection probability directly when selection depends on retained values. Unlike the construction in Appendix D.5, it integrates out the old masked values before constructing the correction. Before a local resampling step, the sampler either accepts that repair or discards the whole assignment and starts again with a fresh draw from p. Repairs selected more often for particular retained values are accepted less often. The correction requires exact conditional probability calculations and stores the preceding repair set as part of its current state. Related exact local-resampling and perfect-sampling methods are developed by Feng et al. (2021; 2022). The elementary correction here is a sufficient way to obtain exact samples.

For each pair R, S with $S \neq \varnothing$, choose a fixed number

$$0 \le c_{R,S} \le \min_{\substack{p_{B_S^c}(z)>0 \\ z \in E_S}} g_{R,S}(z). \tag{85}$$

The set over which the minimum is taken is nonempty because $p(\mathcal{F}) > 0$, and it is finite. The number $c_{R,S}$ must be fixed for each pair R, S. It must not depend on the currently observed visible values. It may be zero.

Whenever S and z are observed with positive probability, $g_{R,S}(z) > 0$. The ratio used by Algorithm 3 is therefore well-defined with probability one. If $c_{R,S} = 0$, the algorithm restarts without dividing by $g_{R,S}(z)$ or evaluating that conditional probability.

Theorem D.9 (Exact sampling with acceptance and restarts)**.** *With exact joint draws and valid bounds* (85)*, Algorithm 3 terminates almost surely and outputs $p_{\mathcal{F}}$. If T counts repair-set checks including the final check, then*

$$\mathbb{P}(T > k) \le (1-a)^k \quad (k \ge 0), \qquad \mathbb{E}T \le \frac{1}{a}. \tag{86}$$

Proof. Let J_t denote the stored set R before check t, and let $X^{(t)}$ be the assignment at that check, for $t \ge 1$. The current state is the pair $(J_t, X^{(t)})$. All transition probabilities are fixed functions of this pair. We maintain the invariant

$$\mathcal{L}(X^{(t)} \mid T \ge t, J_t = R) = p(\cdot \mid E_R),$$

whenever the conditioning event has positive probability. It holds at the first check, where $J_1 = [m]$ and $X^{(1)} \sim p$. For a nonempty selected set S, multiply (73) by the acceptance probability and then the conditional resampling probability. This gives

$$\mathbb{P}(S, \text{accept}, X^{(t+1)} = y \mid T \ge t, J_t = R) = \frac{c_{R,S}}{p(E_R)} p(y) \mathbf{1}\{y \in E_S\}. \tag{87}$$

For $c_{R,S} > 0$, conditioning on this accepted transition gives $p(\cdot \mid E_S)$ and sets $J_{t+1} = S$. If $c_{R,S} = 0$, that transition is never accepted. A rejection instead produces an independent draw from p and sets $J_{t+1} = [m]$. For each fixed value of J_{t+1}, all ways of reaching it therefore yield the same conditional distribution. Averaging over those transitions proves the invariant at check $t + 1$.

Conditional on reaching check t and on $J_t = R$, its stopping probability is $p(\mathcal{F})/p(E_R) \geq a$ and its output, upon stopping, has distribution $p_{\mathcal{F}}$. Averaging over R gives the same statements conditional only on reaching the check. Iterating the survival bound proves (86). Termination occurs with probability one, and mixing the conditional output distributions over all stopping checks gives $p_{\mathcal{F}}$. □

Scope and computational cost. The acceptance probability cancels the factor $g_{R,S}$ introduced by selection. The repair-set rule provides the inside–outside identity in Lemma D.2, which also explains why no correction is needed for product distributions. Computing $g_{R,S}$ may require summing over all masked assignments. Finding a positive lower bound may also require checking every visible assignment that satisfies E_S and has positive probability under p. Exact conditional predictions for arbitrary masks allow these calculations by summation and the chain rule, but both searches can be very large. One sufficient case can be checked by enumeration. Let $N \geq 2$ be the input length, assume a fixed finite alphabet and full support of p, and suppose exact conditional probabilities and $f_{R,S}$ can be evaluated in polynomial time. For each encountered pair R, S, suppose $|B_S| = O(\log N)$ and we can identify in polynomial time a set $D_S \subseteq B_S^c$ of size $O(\log N)$ such that $p(X_{B_S} \mid X_{B_S^c} = z) = p(X_{B_S} \mid X_{D_S} = z_{D_S})$. Enumerate masked assignments u and retained patterns d to compute

$$h_{R,S}(d) = \sum_u f_{R,S}(u)\, p(X_{B_S} = u \mid X_{D_S} = d), \qquad c_{R,S} = \min_d h_{R,S}(d).$$

There are polynomially many terms. By (71), $g_{R,S}(z) = h_{R,S}(z_{D_S})$ for allowed z, so this minimum is a valid, possibly conservative lower bound: it also includes patterns that may not extend to an assignment satisfying E_S. A small mask reduces the sum over masked values, but finding a lower bound valid for every retained assignment can still be costly.

If p gives every assignment positive probability and $f_{R,S}(u) = 1$ for some u, this minimum is positive. A constant acceptance lower bound still requires control of $c_{R,S}/g_{R,S}(z)$. Positivity and efficient evaluation alone do not provide one. Setting every $c_{R,S}$ to zero is always valid. It gives ordinary rejection sampling, which draws independently from p until it obtains a feasible assignment.

Equation (86) bounds the number of repair-set checks. Acceptance and resampling costs enter the runtime separately. A restart changes variables outside the selected mask and can create new violations. The expected decrease in Theorem 3.1 may therefore fail at a restart, so Corollary A.6 does not bound the corrected algorithm's cumulative repair-set weight. The output-error theorem in the main text compares the original sampler with an ideal sampler that has the correct output distribution. Errors in the acceptance calculation require a separate analysis. When the selection probability is constant over the relevant visible assignments, $g_{R,S}$ takes the same value for each of them. Choosing $c_{R,S}$ equal to that constant makes every local resampling step that occurs accepted, so the algorithm reduces to the original repair procedure.

Acceptance probabilities when selection depends on retained values. The correction accounts for variation in $g_{R,S}(z)$. Without a correction, this function must be constant over retained assignments that satisfy E_S and have positive probability under p. Write $g_{\min}$ and $g_{\max}$ for its minimum and maximum over those assignments. When $g_{\min} > 0$, the largest valid choice is $c_{R,S} = g_{\min}$, and the smallest acceptance probability for this pair is

$$\inf_z \frac{c_{R,S}}{g_{R,S}(z)} = \frac{g_{\min}}{g_{\max}}.$$

For example, selection probabilities 0.4 and 0.5 for two possible retained assignments vary, but give acceptance probabilities 1 and 0.8. Thus a constant acceptance lower bound permits selection probabilities that differ by at most a constant factor. We need to control the ratio of selection probabilities. A small difference between them can still give a small ratio when both are small. If $g_{\min} = 0 < g_{\max}$, every valid $c_{R,S}$ is zero and this correction rejects every occurrence of the pair.

The following restart bound applies to either acceptance rule. Its coupling argument separates acceptance frequency from the computational cost of generating an acceptance decision.

Corollary D.10 (Overhead from a lower bound on acceptance)**.** *Assume the hypotheses of Theorem D.7 or Theorem D.9, using its corresponding acceptance rule. Starting from a fresh draw from p, suppose the uncorrected repair sampler has expected number of resampling steps $\mathbb{E}\tau \le L < \infty$. Suppose also that for every attainable nonterminal state (R, X) of that uncorrected sampler, with $S = \mathrm{Res}(X)$, $u = X_{B_S}$, and $z = X_{B_S^c}$, its acceptance probability $p_{\mathrm{accept}}(R, X)$ satisfies*

$$p_{\mathrm{accept}}(R, X) \ge \alpha, \qquad 0 < \alpha \le 1. \tag{88}$$

Here $p_{\mathrm{accept}}(R, X) = p_{\mathrm{accept}}(u, z; S)$ for (79)*, or $p_{\mathrm{accept}}(R, X) = c_{R,S}/g_{R,S}(z)$ for Algorithm 3. Let N count fresh starts, including initialization, and let T count all repair-set checks, including the final check. Then the corrected sampler outputs exactly $p_{\mathcal{F}}$ and*

$$\mathbb{E}N \le \alpha^{-L}, \qquad \mathbb{E}T \le (L+1)\alpha^{-L}. \tag{89}$$

If initialization and preprocessing have expected cost at most C_0, and each reached iteration has conditional expected cost at most C given its current state (R, X), then

$$\mathbb{E}[\mathsf{Time}] \le C_0 + C(L+1)\alpha^{-L}. \tag{90}$$

The cost C includes constructing the repair set, all computations for the acceptance decision, and any ensuing exact resampling or restart. Consequently, if $m \ge 2$, $L \le c_0 + c_1 \log m$ for fixed nonnegative constants c_0, c_1, and α is bounded below by a positive constant independent of m, then

$$\mathbb{E}T = O\Big((1 + \log m)\, m^{c_1 \log(1/\alpha)}\Big).$$

Polynomial bounds on C_0 and C therefore give polynomial expected running time.

Proof. An attempt starts with an independent draw from p and ends at the first rejection or at feasible termination. Couple it to a complete run of the uncorrected sampler from the same initial assignment, using the same exact resampling draws until rejection. Conditional on this complete run, an attempt reaches feasible termination without a restart with probability at least α^{τ}. If q denotes the unconditional probability of such success, convexity and $\mathbb{E}\tau \le L$ give

$$q \ge \mathbb{E}[\alpha^{\tau}] \ge \alpha^{\mathbb{E}\tau} \ge \alpha^{L} > 0.$$

This also holds when $\alpha = 1$. An attempt uses at most $\tau + 1$ checks in the coupling: rejection can only truncate the uncorrected run, and successful termination requires one final check. Its number H of checks therefore satisfies $\mathbb{E}H \le L + 1$.

Every restart draws independently from p, resets the stored repair set to $[m]$, and uses fresh subsequent randomness. The attempts are therefore independent and identically distributed. The number N of attempts has mean $1/q$, and summing the expected costs of reached attempts gives

$$\mathbb{E}T = \sum_{j \ge 1} (1-q)^{j-1} \mathbb{E}H = \frac{\mathbb{E}H}{q} \le (L+1)\alpha^{-L}.$$

There is no need to assume independence between the length and success of an individual attempt. Exactness follows from the corresponding exactness theorem. For the runtime bound, let D_t be the cost of check t and its ensuing update, set to zero when that check is not reached. The conditional cost assumption gives $\mathbb{E}D_t \le C\mathbb{P}(T \ge t)$. Summing nonnegative costs yields $\mathbb{E}[\mathsf{Time}] \le C_0 + C\sum_{t \ge 1} \mathbb{P}(T \ge t) = C_0 + C\mathbb{E}T$, which proves (90). Finally, $\alpha^{-L} \le \alpha^{-c_0} m^{c_1 \log(1/\alpha)}$ gives the polynomial bound. □

The bound $\mathbb{E}\tau \le L$ for the uncorrected sampler needs a proof that does not assume this selection property, such as checking the averaged path bounds in Theorem 3.1 for the sampler's actual distributions. Without that property, using $p(\cdot \mid E_S)$ in place of these distributions requires a separate justification. The acceptance bound must also hold at every reachable state before termination, not just on average at the first repair. Under both bounds, the expected number of checks is at most $\min\{1/p(\mathcal{F}), (L+1)\alpha^{-L}\}$. For fixed $\alpha < 1$, the factor α^{-L} grows exponentially when L is proportional to m. Taking $L = O(\log m)$ makes this factor polynomial.

Corollary D.11 (Running time from a lower bound on feasibility)**.** *Under the assumptions of Theorem D.9, suppose*

$$p(\mathcal{F}) \geq \rho, \qquad 0 < \rho \leq 1.$$

Assume initialization and any preprocessing have expected cost at most C_0. For each reached iteration, assume its conditional expected cost, given its current state (R, X), is at most C. This includes constructing the repair set and its mask, all computations needed for the acceptance decision, and any ensuing exact resampling or restart. If Time *is the total computational cost and T counts repair-set checks including the final check, then*

$$\mathbb{P}(T > k) \leq (1-\rho)^k \quad (k \geq 0), \qquad \mathbb{E}T \leq \frac{1}{\rho}, \qquad \mathbb{E}[\mathsf{Time}] \leq C_0 + \frac{C}{\rho}. \tag{91}$$

Proof. The tail and check-count bounds follow from (86) and $p(\mathcal{F}) \geq \rho$. Let D_t be the cost of iteration t, set to zero if that iteration is not reached. The conditional cost assumption gives $\mathbb{E}D_t \leq C\mathbb{P}(T \geq t)$. Summing nonnegative costs yields

$$\mathbb{E}[\mathsf{Time}] \leq C_0 + \sum_{t \geq 1} C\mathbb{P}(T \geq t) = C_0 + C\mathbb{E}T \leq C_0 + C/\rho.$$

□

For an input of encoding length $N \geq 2$, a sufficient assumption is $p(\mathcal{F}) \geq \rho(N) \geq N^{-c}$ for some fixed $c > 0$, together with polynomial bounds on C_0 and C. The corrected sampler then has polynomial expected running time. A constant lower bound $\rho > 0$ gives a constant expected number of repair-set checks. The runtime guarantee also requires polynomial costs for initialization, acceptance calculations, and resampling.

One sufficient condition for the probability assumption is

$$\sum_{i=1}^{m} p(A_i) \leq 1 - \rho,$$

since the union bound gives $p(\mathcal{F}) \geq 1 - \sum_i p(A_i)$ without requiring independence. Even independent constraints can make feasibility exponentially rare. For example, m constraints with disjoint scopes ($\Delta = 0$) and independent violations of fixed probability $q > 0$ give $p(\mathcal{F}) = (1-q)^m$. Ordinary rejection sampling satisfies the same bound in Corollary D.11, so this bound alone cannot show that local repair is faster.

Using one learned distribution throughout. Suppose initialization, conditional resampling, acceptance probabilities, and valid lower bounds all come from the same learned joint distribution q, with $b = q(\mathcal{F}) > 0$. Theorem D.9 then gives output exactly $q_{\mathcal{F}} = q(\cdot \mid \mathcal{F})$. For $\delta = \mathrm{d_{TV}}(q, p)$,

$$\mathrm{d_{TV}}(q_{\mathcal{F}}, p_{\mathcal{F}}) \leq \min\left\{1, \frac{\delta}{\max\{a, b\}}\right\} \leq \min\left\{1, \frac{\delta}{a}\right\}. \tag{92}$$

To prove the first bound when $a \geq b$, use

$$\mathrm{d_{TV}}(p_{\mathcal{F}}, q_{\mathcal{F}}) = \frac{1}{a}\sum_{x \in \mathcal{F}} \left[p(x) - \frac{a}{b}q(x)\right]_+ \leq \frac{1}{a}\sum_{x \in \Omega}[p(x) - q(x)]_+ = \frac{\delta}{a}.$$

Swap p, q when $b \geq a$. For fixed $a > 0$, the output error converges to zero as the learned distribution approaches p in total variation. The factor $1/a$ also shows why conditioning on a rare feasible event can amplify model error. Predictions for all masks must be conditional probabilities of the same q, which training them separately does not ensure. Exactness also requires exact selection probabilities and proved lower bounds.

D.7 Examples with coordinates that are never repaired

Theorem D.12 (Exactness when a visible context is retained)**.** *Split the data coordinates into a visible context Z and coordinates X that may be repaired. Suppose*

$$p(z, x) = \rho(z)\mu^z(x), \qquad \mu^z(x) = \prod_v \mu_v^z(x_v). \tag{93}$$

Thus, conditional on $Z = z$, the coordinates of X are independent. Suppose Z is always visible and is never masked. Initialize from p and, conditional on $Z = z$, run the original repair algorithm, drawing each masked coordinate from its exact distribution μ_v^z. Write $\mathcal{F}_z$ for the feasible assignments to X when $Z = z$, and suppose $a_z = \mu^z(\mathcal{F}_z) > 0$ for every z with $\rho(z) > 0$. Then the output distribution is

$$Q(z, x) = \rho(z)\mu^z(x \mid \mathcal{F}_z), \qquad \mathrm{d_{TV}}(Q, p_{\mathcal{F}}) = \frac{1}{2}\mathbb{E}_\rho \left| 1 - \frac{a_Z}{a} \right|. \tag{94}$$

In particular, $Q = p_{\mathcal{F}}$ if and only if $a_Z = a$ almost surely.

Proof. Conditional on $Z = z$, the initial coordinates of X are independent, each coordinate has a fixed resampling distribution, and the repair-set rule is the usual deterministic PRS rule. Proposition D.4, or fixed-product general PRS (Guo et al., 2019), gives the exact conditional output distribution $\mu^z(\cdot \mid \mathcal{F}_z)$ and almost-sure termination. The algorithm leaves Z unchanged, so its output has the same distribution ρ over contexts as its initial draw. This proves the first identity. The target $p_{\mathcal{F}}$ gives the same distribution of X within each context, but assigns probability $\rho(z)a_z/a$ to context z. Summing the absolute difference over x for each z gives (94) and the claimed equivalence. □

If Z is part of the sampled data, no constraint may use it when applying this theorem to our algorithm. Repairing such a constraint would mask Z too. If instead we fix an external context z before sampling, the target is $p(\cdot \mid Z = z, \mathcal{F}_z)$, so there is no distribution over contexts to match. A latent Z must be explicitly sampled and retained for Theorem D.12 to apply.

Rare feasibility with efficient exact local repair. Let Z be a binary coordinate with positive probability of each value. Given Z, let $X_1, \ldots, X_{2m}$ be independent and satisfy $\mathbb{P}(X_i = Z \mid Z) = \theta$, where $0 < \theta < 1$ and $\theta \neq 1/2$. For each j, the bad event is $A_j = \{X_{2j-1} \neq X_{2j}\}$. Each constraint uses its corresponding pair, and the pairs are disjoint. The repair set therefore consists exactly of the pairs whose bits are currently unequal. Put

$$b_\theta = 2\theta(1-\theta), \qquad s_\theta = \theta^2 + (1-\theta)^2 = 1 - b_\theta.$$

The feasibility probability is $a_z = s_\theta^m$, independent of z. Theorem D.12 therefore proves exact sampling of $p_{\mathcal{F}}$ using independent parallel resampling. The data coordinates are correlated through their shared dependence on Z, even though they are independent when Z is given. Each pair has probability at most b_θ of being unequal after the next round, so the path certificate holds with unit weights and $\kappa = b_\theta < 1$. Moreover,

$$\mathbb{E} \sum_{t<\tau} |R_t| = m \frac{b_\theta}{s_\theta}. \tag{95}$$

Indeed, each pair is initially unequal with probability b_θ. Such a pair needs a geometric number of resampling steps with mean $1/s_\theta$, and a pair is never changed again once its bits agree. At $\theta = 0.8$, the probability of a feasible initial draw is 0.68^m, while the expected total number of pair resampling steps is $m(0.32/0.68)$. For $m = 50$, the feasible probability is approximately 4.22×10^{-9} and the expected number of pair resampling steps is $400/17 \approx 23.53$. Rejection sampling with independent full assignments instead requires about 2.37×10^8 proposals. These counts compare different operations and do not establish which method runs faster.

The selection condition can fail even when the output is exact. Let Z be equally likely to be zero or one, and let X_1, X_2 be independent conditional on Z. Their probabilities of being one are $(1/5, 2/5)$ when $Z = 0$ and $(2/5, 1/5)$ when $Z = 1$. Use bad events $A_i = \{X_i = 1\}$. Both contexts have feasibility probability $(4/5)(3/5) = 12/25$, so Theorem D.12 gives exact output. However, for $R = \{1, 2\}$ and $S = \{1\}$, visible assignments satisfying E_S have $X_2 = 0$, and $g_{R,S}(X_2 = 0, Z)$ equals $1/5$ or $2/5$ according to Z. Thus selecting a repair set can change the relative probabilities of the two contexts, and (74) fails. Nevertheless, after averaging over all sequences of repair sets, the final output distribution is correct.

Proposition D.13 (Fast repair and exact predictions can still give biased samples)**.** *For every $k, \eta \in (0, 1)$, there is a finite data distribution that gives positive probability to every assignment, with exact initial sampling and exact joint conditional resampling for which the path certificate holds with $\kappa \leq k$, but $\mathrm{d_{TV}}(Q, p_{\mathcal{F}}) > 1 - \eta$.*

Proof. Choose $0 < \gamma < \eta$, set $\mathbb{P}(Z = 0) = \gamma$, and choose $0 < r_0 < r_1 \leq k$. Conditional on $Z = z$, let $X_1, \ldots, X_n$ be independent Bernoulli variables with parameter r_z. Let $A_i = \{X_i = 1\}$, with disjoint singleton scopes, and retain Z. Every assignment has positive probability under this data distribution. Exact conditional resampling redraws each bit independently if it is currently one, with probability r_Z of returning another one. Each length-zero path therefore has probability bound at most r_1. There are no positive-length paths, because the scopes are disjoint. Unit weights give $\kappa = r_1$. The algorithm terminates with all $X_i = 0$, retaining $Q(Z = 0) = \gamma$. In contrast,

$$p_{\mathcal{F}}(Z = 0) = \frac{\gamma(1 - r_0)^n}{\gamma(1 - r_0)^n + (1 - \gamma)(1 - r_1)^n}.$$

Both output distributions put all their probability on the two assignments with all $X_i = 0$, one for each value of Z. Their total variation distance is therefore $p_{\mathcal{F}}(Z = 0) - \gamma$. Since $r_1 > r_0$, this tends to $1 - \gamma > 1 - \eta$ as n grows. Choose a sufficiently large finite n. □

Exact individual predictions can give an incorrect joint resampling distribution. Let $p(00) = p(11) = 2/5$ and $p(01) = p(10) = 1/10$, with sole bad event $A = \{11\}$ and full mask $\{1, 2\}$. The selection probability cannot depend on visible values because no coordinates remain visible. The selection probability therefore cannot vary with retained values. Joint resampling yields $p_{\mathcal{F}} = (2/3, 1/6, 1/6)$ on $(00, 01, 10)$. Each exact token marginal is Bernoulli$(1/2)$. If the two coordinates are sampled independently in parallel at every repair, then, starting from the infeasible assignment 11, the algorithm returns each of the three feasible assignments with equal probability. Even with exact initialization from p, the resulting output is $(8/15, 7/30, 7/30)$ and has TV distance $2/15$ from $p_{\mathcal{F}}$. Even when selection cannot bias retained values, accurate predictions for individual coordinates are not enough. Resampling must reproduce their joint distribution.

E Exhaustive verification on small examples

The checks in this appendix use specified probability distributions, with every calculation performed using exact fractions from Python's standard library. The trained-model experiments are described separately in Appendix F. We enumerate all assignments, including those assigned probability zero, and construct the PRS repair set using the prescribed batches and inspection order. For every (R, S, x), we check the identity in Lemma D.2. This identity describes the selection of S in terms of values inside its mask and satisfied constraints outside the mask. We obtain selection probabilities by summing over assignments. We then check that the acceptance correction gives p conditioned on the required event, calculating that conditional distribution separately from its definition.

To compute final output probabilities, we solve $(I - T)H = B$ by Gauss–Jordan elimination with exact fractions. Here T contains the transition probabilities between states where the algorithm has not stopped, B contains the probabilities of transitions to feasible outputs, and each row of H gives the eventual output distribution from one state. Replacing B by a column of ones gives the expected number of transitions before stopping. We substitute each solution back into its original equations to check it. For the algorithm with the acceptance correction, the state records both the assignment and the previous repair set. In the example where some assignments have probability zero, the uncorrected algorithm can remain trapped at an infeasible assignment.

E.1 Distributional checks

We check the following seven distributions. All variables are bits unless another set of values is specified.

1. **Overlapping constraints with independent data.** Four independent bits are true with probabilities $(1/3, 2/5, 1/2, 3/4)$. The three bad events are adjacent pairs equal to 00. This tests a repair set that can include satisfied constraints and checks exact sampling from independent data conditioned on feasibility. The probability of a feasible assignment is $9/20$.
2. **Correlated data with selection independent of retained values.** Y is fair, $X \in \{0, 1, 2\}$, and the conditional probabilities of X are $(1/5, 7/10, 1/10)$ at $Y = 0$ and $(1/5, 1/10, 7/10)$ at $Y = 1$. The only bad event is $X = 0$. Its probability is $1/5$ for

Table 2: Exact calculations of output distributions. Every terminating method returns feasible assignments, but their relative probabilities can still differ from $p_{\mathcal{F}}$.

Model and resampling rule	$p(\mathcal{F})$	$\mathrm{d_{TV}}(q, p_{\mathcal{F}})$
Biased two-bit distribution, exact joint local repair	$1/2$	$1/4$
Same distribution, selection-corrected repair	$1/2$	0
Full mask forbidding 11, exact joint resampling	$3/5$	0
Same distribution, independent exact marginals	$3/5$	$2/15$
Two correlated pairs with visible Z	$289/625$	0

either value of Y, so selecting the repair set does not bias the retained value of Y, even though X and Y are correlated. The probability of feasibility is $4/5$.

3. **Two-bit data with biased repair.** The distribution in Appendix D.1 has bad event $X = 1$. Using the exact data conditionals for local resampling gives output TV $1/4$ and an expected number of resampling steps of $5/3$. Adding the acceptance correction gives TV zero.
4. **Joint versus independent resampling.** The distribution on $(00, 01, 10, 11)$ is $(2/5, 1/10, 1/10, 2/5)$ and the only bad event is 11, with both bits in its scope. With the starting sample drawn exactly from p, joint resampling gives output probabilities $(2/3, 1/6, 1/6)$ on $(00, 01, 10)$. Independent resampling using the correct probabilities for each individual bit gives $(8/15, 7/30, 7/30)$, at TV distance $2/15$ from the target.
5. **Correlated pairs.** Use two pairs in the family of Appendix D.7, with $\theta = 4/5$ and fair visible Z. The probability of feasibility is $289/625$. Both joint and parallel conditional resampling steps are exact.
6. **Failure probabilities exchanged across contexts.** Z is fair. Conditional on Z, X_1, X_2 are independent with failure probabilities $(1/5, 2/5)$ or $(2/5, 1/5)$. The bad events are $X_1 = 1$ and $X_2 = 1$. Feasibility has probability $12/25$ in both contexts. The ordinary algorithm is exact even though some repair-set selection probabilities depend on the variables kept fixed. The corrected version is also exact. Its verification covers multiple possible previous repair sets.
7. **Data with zero-probability assignments.** Assign probability $1/2$ each to 00 and 11, and use bad event $X = 1$. Exact local repair cannot escape from 11. A zero lower bound in the acceptance correction forces a restart, and the corrected sampler returns 00 with probability one.

The total number of checks of Lemma D.2 is $16{\cdot}8^2{+}6{\cdot}2^2{+}4{\cdot}2^2{+}4{\cdot}2^2{+}32{\cdot}4^2{+}8{\cdot}4^2{+}4{\cdot}2^2 = 1736$. For the large family, where a draw is rarely feasible, the code evaluates the proved formulas at $m = 50$ rather than enumerating all assignments.

E.2 Termination and asymmetry checks

The checks below evaluate statewise path bounds and drift inequalities. For the memoryless resampling rules in cases 1, 3, and 4, the stronger criterion (51) implies the main criterion by averaging over assignments with the same repair set. Case 2 checks the path bounds and expected decrease for a resampling rule outside the main theorem. The asymmetric example uses $X, Y \in \{0, 1, 2\}$ and bad events $A_1 = \{X = 0\}$, $A_2 = \{X = Y = 1\}$, and $A_3 = \{Y = 0\}$. Masked coordinates are resampled independently with probabilities $(0.49, 0.10, 0.41)$. The dependency graph is 1–2–3. No path containing an edge can have a witness: its first two events require different values for a shared variable. For paths containing a single vertex, the witness probabilities are bounded by $(0.49, 0.10, 0.49)$ across all rounds. The middle bound includes the case where one variable is kept fixed at one. With unit weights, the largest left-hand side of the certificate is 1.08, so this choice does not certify termination. Weights $(1, 2, 1)$ instead give ratios $(0.69, 0.59, 0.69)$ after dividing by the weight of the corresponding constraint. Thus $\kappa = 0.69$ certifies termination for the same sampler.

The termination code enumerates all induced paths. It checks that each event selected for the next repair set has a path witness and that each witness depends only on its stated variables. For each path, the largest witness probability over all states before a resampling step gives β. The code also

computes the probability that each constraint belongs to the next repair set and checks it against the path bounds. From the transition matrix, it calculates the expected total number of repaired constraints, drawn tokens, and resampling rounds. We use the following four cases:

1. The three-bit expanding-mask example of Remark B.3, with initial state $(0, 0, 1)$.
2. Four bits with bad events $A_i = \{X_i = X_{i+1} = 0\}$ for $i = 1, 2, 3$. With phase $b = t \bmod 2$, a masked bit v has probability $(1 + b + x_{v+1})/40$ of zero, with cyclic coordinate indexing and all x values taken before the resampling step. Masked draws are independent. The initial assignment is 0000 at phase zero. This rule uses the round's phase and old values of masked variables, so it cannot be written as $K_B(z, \cdot)$. Including the phase in the state defines its transition matrix. These checks test the underlying calculations, not the main theorem.
3. The ternary asymmetric example above, started at $(0, 0)$. Using the probability bounds above, the matrix defined in Appendix C is
$$M = \begin{pmatrix} 49/100 & 1/10 & 0 \\ 49/100 & 1/10 & 49/100 \\ 0 & 1/10 & 49/100 \end{pmatrix}.$$
The largest ratio in this sufficient criterion with unit weights is $27/25$, whereas $w = (1, 2, 1)$ gives ratios $(69/100, 59/100, 69/100)$. The exact expected cumulative repair-set weight from $(0, 0)$ is $10200/2501$, below the bound $200/31$.
4. The four-bit chain of case 2, now using a joint resampling step that sets every masked bit to zero with probability $1/100$ and every masked bit to one with probability $99/100$. Start at 0000. Calculating path probabilities directly gives a valid certificate. For the length-two path at the initial state, the witness probability is $1/100$. Multiplying the probabilities of its separate requirements would incorrectly give $1/10000$, because the drawn bits are dependent. This example checks why joint resampling steps require bounds on the joint path probability unless the needed independence has been established.

Across the four models, the code passes 50 witness checks, 210 checks that joint probabilities equal the claimed products, and 96 bounds on the probability of selecting a particular constraint. It also compares the exact expected cumulative repair-set weight and expected rounds against their bounds from all 33 states where the algorithm has not stopped, including phase information where needed. To check the logarithmic bound using exact fractions, the code uses the first 24 terms of $\log r = 2\sum_{k\geq 0}[(r-1)/(r+1)]^{2k+1}/(2k+1)$. For $r \geq 1$, this partial sum is a lower bound, so the resulting comparison is conservative. The code also checks the probability of still running and the expected number of masked tokens through round 12. Finally, it checks the matrix-inverse bounds on the expected number of repairs of each constraint.

F Experiment specifications, proofs, and additional results

The experiments ask three questions: which distribution a sampler returns when its resampling probabilities are known exactly, how closely a trained model reproduces that distribution, and how many repair rounds each sampler requires. When we call a sampler exact below, we mean that its output has the stated target distribution. The trained networks have small but nonzero errors. We compute these errors for the probabilities actually used by the implementation.

F.1 Reference distributions and exactness proofs

Repairing the entire block. The eight-bit examples each use one constraint involving all eight bits. Whenever it fails, the algorithm replaces the entire block. This is rejection sampling, a special case of PRS. The following calculation gives the output distribution and expected number of repair rounds for any distribution of the starting sample.

Proposition F.1 (Full-block repair)**.** *Let q_0 be the distribution of the starting sample, and let every repair draw independently from a fixed distribution q, with $a = q(\mathcal{F}) > 0$. The output distribution $\widehat{p}$ and the expected number of repair rounds are*
$$\widehat{p}(y) = q_0(y) + q_0(\mathcal{F}^c)\frac{q(y)}{a} \quad (y \in \mathcal{F}), \qquad \mathbb{E}\tau = \frac{q_0(\mathcal{F}^c)}{a}. \tag{96}$$

The unit-weight path certificate is $\kappa = 1 - a$ when $a < 1$. If $a = 1$, any $\kappa \in (0, 1)$ satisfies the certificate. In particular, using q for both the starting sample and subsequent resampling returns $q(\cdot \mid \mathcal{F})$ exactly.

Proof. A feasible assignment y is returned either as the starting sample, with probability $q_0(y)$, or after that sample fails and one or more resampling steps are performed. In the second case, every resampling step before the final draw of y must fail. Summing these probabilities gives $q_0(\mathcal{F}^c)\sum_{j\geq 0}(1-a)^j q(y) = q_0(\mathcal{F}^c)q(y)/a$. If the starting sample fails, the number of resampling steps is geometric with success probability a and mean $1/a$. The dependency graph has one vertex and no edges, so $m = 1$ and $\Delta = 0$. Its only path witness is the bad event $\mathcal{F}^c$. Its resampling probability is $1 - a$ at every round. □

Eight-bit data. For $x \in \{0,1\}^8$, write $S(x) = \sum_{i=1}^{8} x_i$ and define

$$\pi_{12} = \mathrm{Uniform}\{x : S(x) \in \{1, 2\}\}, \qquad \pi_1 = \mathrm{Uniform}\{x : S(x) = 1\}. \tag{97}$$

There are 36 strings with one or two true bits and 8 with exactly one. For the OR and one-or-two experiments, the data distribution is $p = \frac{1}{4}\delta_{0^8} + \frac{3}{4}\pi_{12}$. For the exactly-one experiment, it is $p = \frac{1}{4}\delta_{0^8} + \frac{3}{4}\pi_1$. Thus $p(\mathcal{F}) = 3/4$ in all three experiments, although their feasible sets differ. This probability is part of the data construction, rather than a conclusion about training. All eight coordinates are masked on failure. An ideal joint data resampling step therefore draws from p and succeeds with probability $3/4$. Drawing bits one at a time from the exact data conditionals gives this same joint distribution by (65).

Let u denote the uniform distribution on all 256 bit strings. Table 3 gives its acceptance probabilities and those of the ideal joint data resampling step. Both methods in the table draw their starting sample from p, so their differences come entirely from the resampling distribution. Their expected numbers of repairs follow from Proposition F.1.

Table 3: Exact calculations when every repair replaces all eight bits. Both methods start from $q_0 = p$. The ideal joint data resampling method has $q = p$, success probability $3/4$, certificate $\kappa = 1/4$, and expected number of repairs $1/3$ in every row.

Constraint	$u(\mathcal{F})$	Uniform κ	Uniform $\mathbb{E}\tau$	Uniform output TV to $p_{\mathcal{F}}$
$S \geq 1$	$255/256$	$1/256$	$64/255$	$73/340$
$1 \leq S \leq 2$	$9/64$	$55/64$	$16/9$	0
$S = 1$	$1/32$	$31/32$	8	0

For OR, a uniform starting sample and uniform resampling return the uniform distribution u_F on all 255 nonzero strings. Its distance from the desired target is $\mathrm{d_{TV}}(u_F, \pi_{12}) = 1 - 36/255 = 73/85$. When the starting sample is drawn from the data, the output instead has distribution $\frac{3}{4}\pi_{12} + \frac{1}{4}u_F$, with distance $73/340$. The preference for one or two true bits is a property of the data in this experiment, not a requirement imposed by the constraint. Uniform PRS returns exactly the uniform distribution conditioned on OR. That target differs from the data distribution conditioned on OR.

For $1 \leq S \leq 2$ and $S = 1$, the uniform distribution conditioned on feasibility equals $p_{\mathcal{F}}$. A feasible starting sample drawn from p also has distribution $p_{\mathcal{F}}$. Uniform resampling therefore returns $p_{\mathcal{F}}$ with this starting distribution as well. Here joint data resampling reduces the expected number of repair rounds while preserving the target: the ideal reductions are factors of $16/3$ and 24, respectively. If independent copies of these blocks are used with independent data and resampling, each block has the same calculation. Only failing blocks need repair, and the expected total number of block repairs is the sum for the individual blocks. The training experiments use one block. The same calculation applies to several blocks because their data and resampling are independent.

Independent resampling with correct individual predictions. Under the sparse data distribution, a given bit is true with probability $1/6$. When all bits are masked, a perfect prediction for each individual bit is therefore Bernoulli$(1/6)$. Drawing independently from these predictions gives the parallel resampling distribution $r = \mathrm{Bernoulli}(1/6)^{\otimes 8}$, which differs from p. Conditioned on

$1 \le S \le 2$, each string with one true bit has probability $5/68$ and each string with two true bits has probability $1/68$. The target assigns $1/36$ to every string. Consequently,

$$\mathrm{d_{TV}}(r(\cdot \mid 1 \le S \le 2), \pi_{12}) = \frac{56}{153}, \qquad \mathrm{d_{TV}}(\widehat{p}, \pi_{12}) = \frac{14}{153} \quad \text{when } q_0 = p. \tag{98}$$

The second equality follows from the mixture formula in (96). Thus correct probabilities for each individual bit and a valid termination certificate do not imply the correct output distribution. For exactly-one, each bit is true with probability $3/32$ under the data. If all bits are drawn independently with this same probability, then conditioning on exactly one true bit gives a uniform distribution over the eight possible strings. Parallel resampling therefore gives the correct target in this particular ideal limit, although it requires more repair rounds than joint data resampling.

Overlapping backup-clause constraints. Let $x = (a, b, c, d) \in \{0, 1\}^4$ and require

$$(a \vee b \vee c) \ \wedge \ (\neg a \vee \neg b \vee d). \tag{99}$$

Under the data distribution, the pair (a, b) has probabilities $(1/20, 9/20, 9/20, 1/20)$ on $(00, 01, 10, 11)$, respectively. The bits c, d are independent Bernoulli$(1/10)$ variables and are independent of the pair. Each clause fails with probability $9/200$. The two clauses cannot fail at the same time, so $p(\mathcal{F}) = 91/100$. Among feasible data, the probability of having one or two true bits is $64/65$.

Proposition F.2 (Exact partial repair for the backup-clause example)**.** *For* (99)*, a uniform starting sample and uniform resampling return the uniform distribution on the* 12 *feasible assignments. A starting sample from* p *and exact joint data resampling return* $p_{\mathcal{F}}$*. The unit-weight path certificates are* $1/2$ *and* $1/10$*, respectively. Every round before stopping replaces three coordinates and keeps the fourth fixed.*

Proof. Write $A_1 = \{a = b = c = 0\}$ and $A_2 = \{a = b = 1, d = 0\}$. The dependency graph consists of two vertices joined by one edge, since the scopes share a, b. Thus $m = 2$ and $\Delta = 1$. If A_1 occurs, then the shared pair is $(a, b) = 00$. These values rule out A_2, which requires $(a, b) = 11$. The repair set is therefore $\{1\}$ and the PRS repair mask is $\{a, b, c\}$. The other case gives repair set $\{2\}$ and mask $\{a, b, d\}$. Neither case selects a satisfied clause for repair.

Under the data distribution, regard (a, b) as one four-valued variable. This variable, c, and d are independent, and both masks contain the whole pair. The two bad events share this variable but cannot occur together. This is the setting called extremal PRS: bad events that share variables are mutually exclusive. The PRS exactness result therefore applies to these three independent variables, so repair returns their data distribution conditioned on feasibility (Guo et al., 2019). The same argument applies to uniform bits. Termination follows from the certificates below.

Neither path containing an edge can have a witness: each would require the shared pair to be both 00 and 11. Only paths containing a single vertex remain. With uniform resampling, each such witness has maximum probability $1/4$ over all infeasible starting assignments. For a given clause, this maximum occurs when its backup bit is kept fixed at zero while the other clause is repaired. With joint data resampling, these maxima are $1/20$ each. Adding the two bounds in each closed neighborhood gives $\kappa = 1/2$ and $\kappa = 1/10$ in (51). □

The symmetric criterion (42) assumes $\Delta \ge 2$ and does not apply directly to this two-constraint graph. After grouping (a, b), ideal data resampling is product resampling over the grouped variables, so the $\Delta = 1$ specialization in Appendix C.1 does apply: $\bar{p} = 1/20$ gives $4\bar{p} = 1/5 < 1$. The path certificate $1/10$ is sharper because it uses the impossible edge witnesses. Uniform resampling has $\bar{p} = 1/4$ and fails the strict symmetric condition $4\bar{p} < 1$, but still satisfies the path certificate with $\kappa = 1/2$. Learned joint resampling distributions need not factor even over the grouped variables. We check their certificates directly using the whole-path sufficient criterion (51), which implies Theorem 3.1 without that factorization.

This example has overlapping constraints and replaces only part of the assignment, but its repair set never includes a satisfied constraint. We also check that, for every pair of selected sets, the probability of selecting the next set does not change with the variables kept fixed. When both methods start from p, ideal data resampling requires $9/91$ repairs on average and uniform resampling

requires approximately 0.133714. The output of uniform resampling has TV distance approximately 0.045099 from $p_{\mathcal{F}}$. Parallel draws from the correct probabilities for individual bits also give the wrong output distribution, because they discard the dependence between a and b. These comparisons use the feasible data as the target. Starting uniformly and resampling uniformly still returns the uniform distribution over feasible assignments exactly.

F.2 Training and evaluation protocol

Data and objective. We choose the datasets so that the fraction of rows equal to each assignment matches its specified data probability exactly. The sparse dataset has 144 rows: 36 copies of 0^8 and three copies of each of the 36 sparse strings. The exactly-one dataset has 32 rows: eight copies of 0^8 and three copies of each string with exactly one true bit. The backup-clause dataset contains 2000 rows, including 180 infeasible rows. The repeated rows let us study how well the network learns the specified distribution without adding error from randomly sampled training data. These experiments therefore do not measure generalization from an independent sample. The OR and one-or-two experiments share the same trained models because they have the same training data.

For each data row, training masks every nonempty subset of coordinates with equal weight. The loss sums token cross-entropies over the masked coordinates, normalized by the total number of masked tokens. If $M_B x$ replaces coordinates in B by MASK, the objective is

$$\mathcal{L}(\theta) = \frac{\mathbb{E}_{x\sim p}\mathbb{E}_B \sum_{v\in B} -\log \nu_{\theta,v}(x_v \mid M_B x)}{\mathbb{E}_B|B|}, \qquad B \sim \mathrm{Uniform}(2^{[n]} \setminus \{\varnothing\}). \tag{100}$$

For repeated masked inputs, we count how often each target bit occurs and compute the loss from those counts. This gives the same objective as summing over the original rows. All rows, including infeasible ones, are used in training. The loss contains no term for constraint violations or for the termination certificate.

Model and optimization. The denoiser is a small feedforward neural network with one hidden layer of 32 tanh units. The input records each position separately, using a one-hot encoding for its token $(0, 1, \mathrm{MASK})$, and includes the fraction of masked coordinates. An affine output layer followed by a separate softmax at each coordinate predicts the probabilities of zero and one. There are 1360 parameters for eight bits and 712 for four bits. We initialize weights independently with centered normal entries scaled by the inverse square root of each layer's input width. Biases start at zero. We train on each dataset using random seeds $0, 1, 2$ for 6000 full-batch Adam updates, with learning rates 0.01 for updates 1–1000, 0.003 for 1001–3000, and 0.0005 thereafter. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and denominator offset 10^{-8}. The final checkpoint is evaluated for every seed.

Generation and repair. To generate the starting sample, the network begins with all tokens masked and reveals coordinates in descending index order. Joint neural resampling uses the same ordering within the selected mask. The set of variables to repair stays fixed until all of them have been drawn. For comparison, a parallel resampling step predicts once from the masked input and draws the masked bits independently. Both resampling methods use the same trained network. We first evaluate the sampler with the network generating the starting sample and performing the resampling steps. We then repeat the evaluation with the starting sample drawn exactly from the data distribution. The second comparison isolates the effect of the resampling distribution.

Probability rounding and exhaustive verification. To check the sampler using exact fractions, we round its predicted probabilities to multiples of $1/65536$. Specifically, each predicted categorical distribution is represented by integer counts with sum $D = 65536$: first take the floors of D times its probabilities, then allocate the remaining counts to the largest fractional remainders, using a fixed tie order. Sampling uses a uniform integer in $\{0, \ldots, D-1\}$ and these counts. All final certificates and output probabilities refer to the sampler that uses these rounded probabilities. The verification therefore covers exactly the probabilities used to generate the reported samples.

For each eight-bit model, we enumerate all 256 assignments. We evaluate the network on all 255 partial assignments in its fixed generation order, then multiply the successive conditional probabilities

Table 4: The network generates the starting sample and performs the resampling steps. The starting sample is generated one bit at a time. TV is measured against $p_{\mathcal{F}}$. Ranges cover seeds $0, 1, 2$. Each endpoint is rounded from an exact calculation using the probabilities in the implemented sampler.

Constraint	Resampling	Certificate κ	Output TV	Expected repairs
OR	Joint	0.23956–0.24342	0.01299–0.02463	0.31503–0.32174
	Parallel	0.21656–0.22399	0.11295–0.11809	0.30578–0.31368
One or two	Joint	0.23984–0.24359	0.01286–0.02442	0.31552–0.32204
	Parallel	0.36558–0.36624	0.09422–0.09862	0.37805–0.38437
Exactly one	Joint	0.24970–0.25250	0.00327–0.00420	0.33281–0.33780
	Parallel	0.62306–0.62346	0.00288–0.00330	0.66316–0.66988
Backups	Joint	0.09998–0.10000	0.000082–0.000152	0.098881–0.098899
	Parallel	0.49994–0.50002	0.007197–0.007223	0.16360–0.16363

to obtain the probability of each complete assignment. Proposition F.1 gives the output distribution and expected number of repair rounds exactly. For the four-bit example, we enumerate all 16 assignments, including all four infeasible ones, and calculate the probability of every possible transition. Let T contain the transition probabilities between infeasible assignments and B those from infeasible to feasible assignments. We solve $(I - T)H = B$ using exact fractions. Each row of H gives the eventual output distribution from one infeasible starting assignment. Combining these rows with the starting distribution gives the probability of each final feasible assignment. We also compute the expected number of repair rounds exactly from these transition probabilities. The distances in the tables use the resulting output distributions, rather than estimates from sample histograms.

To check termination, we construct the PRS repair mask and enumerate every possible resampling outcome for each infeasible assignment. For each path, we calculate the probability of its witness event and take the largest value over all these assignments. We then check (51) with unit weights. These stronger bounds imply the main criterion by averaging over current assignments with the same repair set at each round. The enumeration is exhaustive for these memoryless implementations: the trained network, token ordering, and mask construction are fixed, and only the current mask and visible values determine the resampling distribution. These calculations do not assume that an average of product distributions remains a product distribution. For the overlapping example, we additionally check the expected decrease in repair-set size at each step, the bounds on expected rounds and total constraint selections, and the sampling-fidelity inequality in Theorem D.6. Certificate checking takes place after training. Cross-entropy alone does not establish these conditions.

Separate checks loaded and evaluated the six eight-bit and three four-bit trained models. We also ran the sampler to generate 1000 feasible outputs for each of the three eight-bit constraints and for the overlap instance. These runs check that the implementation executes and returns feasible assignments. The claims about output probabilities follow from the exact calculations described above.

F.3 Additional numerical results

Tables 4 and 5 report the range over all three training seeds. Joint and parallel rows use the same trained networks. All reported certificates are below one. This includes the parallel resampling methods whose output probabilities differ substantially from the target.

When the backup-clause sampler uses the network for both the starting sample and joint resampling, the bound in Theorem D.6 ranges from 0.0001544 to 0.0002497 and exceeds the computed TV for each seed. For exactly-one, symmetry makes ideal parallel resampling exact, and the trained parallel resampling method has small error. This property does not hold for the one-or-two constraint or for the correlated variables a, b.

Expected repairs count the completed resampling rounds, not the individual bit predictions within a round. Because each repair set contains one constraint, the expected number of rounds also equals the expected total number of constraints selected for repair. Each eight-bit repair draws eight bits. Each backup-clause repair draws three. A sequential joint neural resampling step uses one network prediction per drawn bit, whereas a parallel resampling step uses one prediction for the entire mask.

Table 5: Resampling comparison when the starting sample is drawn exactly from the data distribution, $q_0 = p$. The neural network is used only for repairs.

Constraint	Resampling	Output TV to $p_{\mathcal{F}}$	Expected repairs
OR	Joint	0.00325–0.00616	0.32876–0.33043
	Parallel	0.10851–0.10899	0.31911–0.32216
One or two	Joint	0.00322–0.00611	0.32888–0.33051
	Parallel	0.08832–0.08981	0.39406–0.39447
Exactly one	Joint	0.000819–0.001049	0.33320–0.33445
	Parallel	0.000359–0.000631	0.66324–0.66395
Backups	Joint	0.0000055–0.0000157	0.098899–0.098901
	Parallel	0.007191–0.007195	0.163633–0.163640

Uniform resampling requires no network predictions. Fewer repairs therefore do not by themselves show that the sampler runs faster.

Reproduction. Training and evaluation are implemented in `experiments/sparse_blocks.py` and `experiments/overlap_neural.py`. Exact checks are in `experiments/verify_sparse_results.py` and `experiments/overlap_candidate_check.py`. Datasets, checkpoints, exact probability tables, and numerical summaries are included in `experiments/results_sparse/` and `experiments/results_overlap/`.

F.4 Random 3-SAT: protocol, kernels, and full results

Protocol. We train on 5000 generated satisfiable formulas with $n = 100$, $m = 426$, and evaluate on the 1000 formulas of SATLIB uf100 (Hoos & Stützle, 2000) with $K = 10$ runs per formula. The PRS repair set is computed with Algorithm 2.

Kernels. Both learned kernels use DiffUCO's graph network for SAT (Sanokowski et al., 2024). We add verifier inputs computed from the current state: make and break counts, violated-clause indicators, and the fraction of violated clauses.

- *DiffUCO* follows the original method and code with $T = 4$ Bernoulli denoising steps. To resample a mask, we draw x_B from fair bits and run the reverse chain on B only, giving retained variables the final time index. Larger step counts repeat each time index.
- *MDNS* (Zhu et al., 2025) is trained with PDNS (Guo et al., 2026) and its code at $\beta = 10$ with 128 samples per formula. Variables are unmasked in random order. The network has no time input, so one model serves every step count. The pretrained kernel treats retained values as if generated earlier in the same resampling step.

Fine-tuning. Each training run has a geometric number of rounds with mean 4, capped at 16; for DiffUCO this count includes the initial draw. Because the horizon is geometric, the optimal kernel does not depend on the round index. The reward is the number of violated clauses after the final step. DiffUCO adds a REINFORCE term with this reward to its original objective. MDNS applies PDNS's proximal stages to whole runs with $\eta = 0.5$. Both fine-tuned kernels receive an input marking retained variables. We checked that retained variables never change and that the kernel ignores the previous values of masked variables.

Full results. Table 6 isolates fine-tuning. After 64 rounds it raises solved@10 from 94.1 to 96.9% for DiffUCO and from 32.5 to 39.6% for MDNS. For MDNS, part of this gain is already present in the first round, which the fine-tuned model also generates.

Instance sizes. Table 7 compares PRS-mask repair with the baselines of Table 1 on the SATLIB families uf100, uf150, uf200, and uf250, which have 430, 645, 860, and 1065 clauses. The learned kernels are trained at $n = 100$ and are not retrained for larger sizes.

Table 6: Effect of fine-tuning the kernel for PRS-mask repair on SATLIB uf100: solved@1 / solved@10 (%) after 1, 4, 16 and 64 rounds of 16 denoising steps. A pretrained kernel unrolls the pretrained denoiser on the PRS repair mask. A fine-tuned kernel is further trained with reinforcement learning (REINFORCE for DiffUCO, PDNS stages for MDNS) to minimize the number of violated clauses.

Kernel	Training	1	4	16	64
DiffUCO	pretrained	44.3 / 75.7	60.5 / 85.3	75.5 / 91.0	86.0 / 94.1
	fine-tuned	43.4 / 76.5	63.7 / 87.3	80.5 / 94.2	88.9 / 96.9
MDNS	pretrained	0.1 / 2.4	0.6 / 6.3	2.1 / 14.4	7.6 / 32.5
	fine-tuned	0.7 / 4.6	1.2 / 9.6	4.4 / 20.1	11.9 / 39.6

Table 7: Solved SATLIB formulas (%), solved@1 / solved@10, after 64 rounds on uf100 (1000 formulas) and uf150–uf250 (100 each). Diffusion models without repair generate once. *COExpander supports at most 5 inference steps. All learned methods are trained at $n = 100$.

Method	uf100	uf150	uf200	uf250
Diffusion based				
PRS-mask repair (uniform)	88.9 / 96.9	81.0 / 95.0	67.0 / 85.0	67.0 / 86.0
PRS-mask repair (masked)	11.9 / 39.6	1.0 / 6.0	0.0 / 0.0	0.0 / 0.0
COExpander* (Ma et al., 2025)	50 / 89	38 / 82	32 / 65	29 / 69
DiffUCO (Sanokowski et al., 2024)	44.3 / 75.7	37.0 / 72.0	11.0 / 33.0	14.0 / 37.0
MDNS (Zhu et al., 2025)	0.1 / 2.4	0.0 / 1.0	0.0 / 0.0	0.0 / 0.0
Other baselines				
Uniform PRS (Guo et al., 2019)	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Moser–Tardos (Moser & Tardos, 2010)	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
WalkSAT (Selman et al., 1994)	85.5 / 99.7	81.0 / 96.0	56.0 / 92.0	52.0 / 95.0
probSAT (Balint & Schöning, 2012)	86.1 / 99.7	82.0 / 100.0	57.0 / 90.0	67.0 / 94.0
FMS (Seitz et al., 2005)	65.4 / 97.7	53.0 / 91.0	37.0 / 82.0	27.0 / 84.0
RLSA (Feng & Yang, 2025)	0.1 / 1.7	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0

G Models and protocols for the distributional sampling comparison

This appendix specifies the nine locally evaluated configurations in Figure 2. The methods are Uniform PRS, parallel masked diffusion, joint masked diffusion, and constrained autoregressive generation. Each configuration uses a fixed constraint instance and target distribution. The following subsections give the training data, objective, parameter initialization, optimization, selected checkpoint, initial sampling distribution, and resampling procedure for each model.

G.1 Transformer models and training objectives

For the locally trained SAT and coloring models, the masked diffusion model (MDM) is a bidirectional Transformer with four layers, hidden dimension 128, four attention heads, and feedforward dimension 512. Each layer uses GELU activations, dropout 0.1, and pre-layer normalization, followed by a final layer normalization. The input vocabulary contains the variable values and a dedicated mask token. Learned token and position embeddings encode each partially observed assignment. A two-layer MLP with SiLU activation embeds the scalar masking probability and adds its output at every position. The visible variable values, positions, mask tokens, and scalar time therefore specify the denoiser's conditioning context. Each model is trained for its fixed constraint instance.

Let $x \in \{0, \dots, q-1\}^n$ be a training assignment and let M_i indicate whether coordinate i is masked. Training draws t uniformly from $[10^{-3}, 1]$ and samples the indicators independently with probability t. Writing $\widetilde{x}$ for the resulting corrupted assignment, the absorbing-mask denoising objective is

$$\mathcal{L}_{\mathrm{MDM}}(\theta) = \mathbb{E}_{x,t,M}\left[-\frac{1}{nt}\sum_{i=1}^{n} M_i \log p_{\theta,i}(x_i \mid \widetilde{x}, t)\right]. \tag{101}$$

The implementation averages the weighted token losses over coordinates and examples in the mini-batch.

The autoregressive (AR) Transformer has the same layer count, hidden dimension, attention heads, feedforward dimension, activation, dropout, and normalization scheme. Causal attention exposes the preceding variables in increasing stored variable order. Training shifts the target sequence by one position and prepends a beginning-of-sequence token. The teacher-forced objective is

$$\mathcal{L}_{\mathrm{AR}}(\phi) = -\mathbb{E}_x \left[\frac{1}{n} \sum_{i=1}^{n} \log p_\phi(x_i \mid x_{<i}) \right]. \tag{102}$$

Fresh training uses seed 0. The initialization routine samples embedding and linear-layer weights from $\mathcal{N}(0, 0.02^2)$ and sets linear biases to zero. The remaining Transformer parameters use their PyTorch initialization. Both models use AdamW with learning rate 3×10^{-4}, batch size 128, weight decay 0.01, momentum parameters $(0.9, 0.999)$, numerical stabilizer 10^{-8}, and gradient norm clipping at 1. The learning rate remains constant. Each dataset contains 50,000 assignments, split into 45,000 training and 5,000 validation examples using seed 0. Examples are shuffled each epoch. Checkpoint selection minimizes validation loss. For the MDM, the validation corruption generator is reset to seed 1 at each evaluation so that checkpoint comparisons use fixed validation masks.

G.2 Uniform SAT

The SAT instance is a read-twice 3-SAT formula with 60 Boolean variables and 40 clauses. A connected cubic graph defines the formula, with graph vertices representing clauses and graph edges representing variables. Each variable occurs in its two endpoint clauses. Forty-five variables have opposite signs across their two occurrences, giving an opposite-sign fraction of 0.75. The graph seed is 0 and the polarity seed is 10,000. The target distribution is uniform over satisfying assignments. Training examples are obtained by rejection sampling independent Bernoulli$(1/2)$ assignments with seed 40,000. An independent reference dataset contains 50,000 satisfying assignments generated with seed 63,000.

The parallel MDM is trained for 50 epochs using Equation (101), and validation selects epoch 45. Sampling begins with every coordinate masked and performs 60 ancestral reveal steps. At step $s \in \{0, \dots, 59\}$, the denoiser receives $t_s = 1 - s/60$. Each remaining masked coordinate is revealed independently with probability $1 - t_{s+1}/t_s$, and its value is drawn from the corresponding predicted categorical distribution at temperature 1.

The joint MDM is initialized from this epoch-45 checkpoint and receives 20 further training epochs on the same split. Continued training uses independent Bernoulli masks, learning rate 10^{-4}, batch size 128, weight decay 0.01, gradient clipping at 1, and seed 80,000. Validation selects continuation epoch 17 by equally averaging the independent-mask validation loss and a repair-mask validation loss. The repair-mask bank contains 9,867 masks collected from 5,000 baseline sampling trajectories using seed 70,000. A validation mask is drawn independently of the clean validation assignment. With probability one half, a uniformly chosen nonempty size and a random reveal order select a subset of the bank mask. Otherwise, the full bank mask is used. The time input for these masks is their fraction of the 60 coordinates. The joint sampler initializes an assignment in 60 single-coordinate denoising calls following an independent random permutation for each sample.

Both MDM methods compute a PRS repair set S of clauses from the current assignment after initialization and after each resampling step. The set S initially contains all violated clauses, and the variable mask is $B = V(S)$, the union of their variables. Boundary clauses intersecting B are examined in canonical order and added to S when their variables inside B agree with the corresponding forbidden values. Each addition expands B by the variables of the added clause. Newly exposed clauses enter the next boundary batch, and incompatible clauses remain excluded during that construction. The final mask B specifies the variables supplied as mask tokens to the MDM.

The parallel kernel replaces the variables in B by mask tokens, calls the denoiser once with $t = |B|/60$, and independently draws all variables in B from the predicted categorical probabilities. The joint kernel samples a fresh random order on B and resamples one coordinate per denoiser call. The values outside B remain fixed, and every call conditions on the coordinates already generated inside B. Its time input is the remaining-mask fraction. Both kernels use temperature 1.

The AR model is trained for 50 epochs, with epoch 48 selected by validation. It generates the 60 variables in increasing index order. Before each draw, a value receives zero probability if assigning it would complete a violated clause, and the remaining probabilities are renormalized. A trajectory reaching an empty allowed set contributes to the failure count. Uniform PRS initializes every bit independently from Bernoulli$(1/2)$ and draws fresh independent Bernoulli$(1/2)$ values on each selected variable mask B. All SAT repair methods use a cap of 100 resampling steps and batches of 256. Every plotted method requests 10,000 assignments for each of sampling seeds 4001, 4002, and 4003.

G.3 Block-tree coloring with 256-step initialization

Panel (a) compares four locally evaluated methods on the same block-tree coloring instance. The graph has 128 blocks of two variables, arranged in a binary tree. Each tree edge joins its endpoint blocks into a four-variable hyperedge, giving 256 variables and 127 hyperedges. The topology seed is 0. The target uses three colors and the equal mixture of conditioned product distributions specified in Section G.4. Each hyperedge must be nonmonochromatic.

Parallel MDM and joint MDM use the same epoch-98 checkpoint from the training protocol in Section G.4. Both generate an initial assignment with 256 ancestral reveal steps at temperature 1. Parallel repair replaces the PRS repair mask by mask tokens, evaluates the denoiser once, and draws all replacement colors independently from its predicted marginals. Joint repair draws a fresh random permutation of each mask and reveals one variable per model call, conditioning on the retained exterior and all preceding draws. The time input is the fraction of variables still masked. Model parameters remain fixed during sampling.

All three PRS methods use the original implementation's ordered hyperedge scan to construct each repair set. Starting from all monochromatic hyperedges, a scan adds any intersecting hyperedge whose current values inside the repair mask contain at most one distinct color, expands the mask, and repeats until the set stops growing. Uniform PRS initializes independent uniform colors and makes fresh independent uniform draws on each repair mask. The three methods use batches of 256 and a cap of 100 resampling steps. Sampling seeds are 0 for parallel MDM, 391001 for Uniform PRS, and 392001 for joint MDM.

Constrained AR uses the epoch-44 checkpoint selected under the protocol in Section G.4. It generates the 256 colors in increasing variable order, setting a color's probability to zero whenever that choice would complete a monochromatic hyperedge. The remaining probabilities are renormalized before sampling. It uses seed 0, temperature 1, and batches of 256. Its retained run completes every trajectory with zero deadlocks.

Each method contributes $10{,}000$ feasible outputs. Their color-count statistic (n_0, n_1, n_2) is compared with the same independent empirical reference of $50{,}000$ exact-DP samples generated with seed 1. Uniform PRS has count TV 0.98766, mean resampling count 2.0038, and runtime 25.18 seconds. Parallel MDM has count TV 0.24072, mean resampling count 1.7308, and runtime 386.71 seconds. Joint MDM has count TV 0.23682, mean resampling count 1.2551, and runtime 434.57 seconds, comprising 358.50 seconds of initialization and 76.07 seconds of resampling. Constrained AR has count TV 0.22474 and runtime 262.36 seconds. The three neural methods run on the local NVIDIA GeForce RTX 2080 Ti. The results retain each run's seed and measured timer boundaries, specified in Section G.9.

Panel (g) retains the 32-step coloring comparison described below. That panel pools three runs per method and evaluates against the deterministic exact count target.

G.4 Biased coloring

The coloring instance has 256 variables and three colors. A breadth-first binary tree has 128 blocks, each containing two variables. Each of its 127 edges defines a four-variable hyperedge by joining the endpoint blocks. Variables are relabeled using seed 0. A feasible assignment gives at least two distinct colors to every hyperedge.

The target selects one of three components with equal probability. Their color probabilities are $(0.48, 0.32, 0.20)$, $(0.20, 0.48, 0.32)$, and $(0.32, 0.20, 0.48)$. Within the selected component, its

product distribution is conditioned on all hyperedges being nonmonochromatic. Exact sampling on the block tree produces the 50,000-example training dataset with seed 0 and an independent 50,000-example reference dataset with seed 1.

The MDM is trained for 100 epochs using the architecture and objective in Section G.1. Validation selects epoch 98, which supplies both plotted MDM methods. Each method starts from the same saved assignments generated by 32 ancestral reveal steps from the fully masked state at temperature 1. These initial steps follow the reveal schedule described for SAT, with 32 in place of 60.

For each resampling step, the PRS repair set S initially contains the monochromatic hyperedges, and $B = V(S)$ is the union of their variables. A boundary hyperedge intersecting B is added to S when the current values of its variables inside B contain at most one distinct color. Each addition expands B by the variables of the added hyperedge. Boundary batches and event ordering follow the SAT construction. The parallel MDM resamples the variables in B in one denoiser call. The joint MDM resamples these variables one coordinate per call in a fresh random order, preserving the values outside B and conditioning on preceding draws. Each denoiser call receives the fraction of coordinates still masked. The shared epoch-98 parameters remain fixed throughout sampling.

The AR model is trained for 50 epochs and uses its epoch-44 checkpoint. It generates variables in increasing stored index order. A color receives zero probability when assigning it would complete a monochromatic hyperedge. Uniform PRS initializes all variables independently from the uniform three-color distribution and draws fresh independent uniform colors on every selected variable mask B.

Sampling uses batch size 256 and a cap of 100 resampling steps. The MDM and Uniform PRS rows each contain 10,000 requested assignments for seeds 93001, 93002, and 93003. Parallel and joint MDM repair randomness uses seeds 94001, 94002, and 94003 after the shared initialization. Uniform initialization uses seeds 293001, 293002, and 293003, followed by the same three repair seeds. The AR row pools 10,000 requested assignments for each of seeds 91001, 91002, and 91003.

G.5 Rooted spanning-tree representation and targets

The rooted-tree experiments use the toroidal grid $C_L \square C_L$, with vertices numbered in row-major order and vertex 0 designated as the root. Each nonroot vertex has one categorical variable recording its parent direction. The four labels represent north, east, south, and west, in that order. The resulting assignment has $n = L^2 - 1$ variables. An assignment is feasible when following parent pointers from every vertex eventually reaches the root. Equivalently, the parent pointers contain no directed cycle among the nonroot vertices. Directed cycles are the bad events. At every PRS resampling step, the repair set S contains the currently present directed cycles, and its repair mask $B = V(S)$ is the union of their vertices. Values outside B remain fixed during the resampling step.

The uniform experiments use $L \in \{5, 6, 7\}$, giving $n \in \{24, 35, 48\}$, and target the uniform distribution over rooted spanning trees. Figure 2 displays the 6×6 and 7×7 cases. The biased experiments use $L \in \{3, 7\}$ and bias factors $c \in \{2, 4\}$. Writing $H(x)$ for the number of east or west parent directions, their training distribution is

$$p_c(x) = \frac{4^{-n}}{Z_c}\, c^{\mathbf{1}\{H(x) \geq \lceil n/2 \rceil\}}, \qquad x \in \{0, 1, 2, 3\}^n. \tag{103}$$

The desired sampling distribution is p_c conditioned on feasibility. Thus the bias favors assignments with at least half of their parent directions horizontal. The Uniform PRS baseline initializes every parent direction independently and uniformly and applies the same independent uniform distribution to each variable in the repair mask. Its output distribution supplies the uniform-tree reference in all rooted-tree panels.

G.6 Uniform rooted trees

We train separate models for each of the three fixed toroidal grids. For a grid of side length L, Wilson's loop-erased random-walk algorithm generates $50{,}000$ independent uniform spanning trees using seed $1000 + L$. Orienting their edges toward the root gives the parent-direction training examples. A fixed Torch permutation with seed 0 assigns $45{,}000$ examples to training and $5{,}000$ to

validation. The MDM and AR models use this common split and the shared Transformer architecture described above, with a sequence position for each nonroot vertex and a four-direction output vocabulary.

Both models are initialized with seed 0 and trained for 50 epochs using AdamW with learning rate 3×10^{-4}, weight decay 0.01, batch size 128, and gradient-norm clipping at 1. The MDM uses the inverse-time-weighted absorbing-mask objective with independent Bernoulli masks. The AR model uses teacher-forced next-direction cross entropy in ascending vertex order. Validation is evaluated with dropout disabled. The MDM validation corruption generator is reset to seed 1 each epoch, keeping its validation noise fixed. Selecting the minimum held-out objective gives base MDM checkpoints at epochs 49, 49, and 50 for $L = 5$, 6, and 7, respectively. The selected AR checkpoints are at epochs 47, 47, and 50.

The MDM then undergoes a 20-epoch continuation initialized from its selected base checkpoint. This stage uses a fresh AdamW optimizer with learning rate 10^{-4}, weight decay 0.01, batch size 128, gradient-norm clipping at 1, and seed 80000. Training continues on the same examples with independent Bernoulli corruption and the same denoising loss. Checkpoint selection uses the mean of two held-out losses, one evaluated with fixed IID masks and the other with fixed masks derived from cycle-resampling trajectories. For the second loss, a bank stores the first 4096 nonempty repair masks encountered under uniform cycle popping with seed $3000 + L$. Each stored mask is applied to a held-out tree. With probability $1/2$, a uniformly chosen nonempty subset replaces the full stored mask, with its cardinality selected uniformly from 1 through the original mask size. The model receives the resulting mask fraction as its time input. The selected continuation epochs are 20, 20, and 19 for the three grids. Parallel MDM and Joint MDM use the same selected continued checkpoint on each grid.

Both MDM samplers initialize from an entirely masked sequence and perform 32 ancestral denoising steps at temperature 1. At step $k \in \{0, \dots, 31\}$, the model receives time $t_k = 1 - k/32$. Each remaining masked coordinate is revealed independently with probability $(t_k - t_{k+1})/t_k$, and its value is drawn from the predicted categorical distribution. Revealed values are retained through the remaining initialization steps. The two repair methods share the resulting saved initial assignments exactly.

Parallel MDM replaces every direction in the repair mask by the MASK token, evaluates the model once at time $|B|/n$, and draws all replacement directions independently from those predictions. Joint MDM draws a fresh random ordering of the repair mask and reveals one variable per model evaluation. Each prediction sees the fixed exterior and all directions already revealed during that resampling step. Its time input is the current number of masked variables divided by n. Both use temperature 1, sampling batches of 256, and a cap of 1000 resampling steps per requested output.

Constrained AR starts with the BOS token and generates parent directions in ascending nonroot-vertex order. For each candidate direction, the completion test fixes the outgoing edges already assigned and retains all possible outgoing edges at unassigned vertices. The candidate is permitted precisely when every vertex can reach the root in this allowed directed graph. The AR probabilities are renormalized over the permitted candidates before drawing the next direction. Inference uses the Transformer KV cache, exact GELU evaluation, CPU completion tests, and batches of 256.

For each graph, the new MDM and constrained AR measurements use three sampling seeds and $10{,}000$ requested outputs per seed. For evaluation index $i \in \{0, 1, 2\}$, the initialization seed is $260000+1000L+i$, and the parallel, joint, and AR seeds are $360000+1000L+i$, $460000+1000L+i$, and $560000 + 1000L + i$. The Uniform PRS reference uses the retained outputs from seeds 20000, 20001, and 20002.

G.7 Biased rooted trees on the 3×3 torus

For each bias factor, we draw $50{,}000$ examples from the unconditioned distribution p_c. Data generation proposes independent uniform parent directions and retains each proposal with probability $c^{\mathbf{1}\{H \geq 4\}}/c$, using seed $97100 + c$. A NumPy permutation with seed 0 determines a common $45{,}000$-example training set and $5{,}000$-example validation set. Each MDM and AR configuration is trained with seeds 0, 1, and 2 on these fixed data.

The models use the shared Transformer architecture on eight parent variables. Both are trained from their seeded initializations for 50 epochs using AdamW with learning rate 3×10^{-4}, weight decay 0.01, batch size 256, and gradient-norm clipping at 1. The shuffle generator uses seed $1000 + s$ for training seed s. The MDM corruption generator uses seed $46000 + s$, and its fixed validation corruption uses seed $47000 + s$. The denoising objective is the shared inverse-time-weighted cross entropy. Validation is evaluated at initialization and after every epoch, and the checkpoint with minimum validation loss is retained. For $c = 2$, the selected MDM epochs are $(48, 50, 48)$ across the three seeds. For $c = 4$, they are $(40, 49, 42)$. These checkpoints supply both plotted MDM methods.

MDM initialization reveals the eight variables one at a time in ascending order from the fully masked sequence. Before each reveal, the time input is the number of remaining masks divided by eight. Joint MDM applies the same ascending reveal rule inside each cycle repair mask while retaining its exterior. Parallel MDM makes independent categorical draws for the entire repair mask from one unchanged masked context. Both samplers use temperature 1 and share the saved ascending-generation initial assignments for each checkpoint and evaluation seed. The resampling cap is $100{,}000$ nonempty repair decisions.

For this finite instance, inference precomputes predictions for all 5^8 contexts formed from four directions and MASK. Network logits are computed in float32, converted to float64, and normalized into categorical tables. Sampling then performs CPU table lookups in batches of 1024. The AR model is trained with the shared teacher-forced loss and selected by minimum held-out negative log likelihood, including the initialization checkpoint. Its selected epochs are $(49, 29, 40)$ for $c = 2$ and $(35, 34, 39)$ for $c = 4$. Ascending AR prefix probabilities and their exact rooted-tree completion masks are also precomputed. Constrained generation starts at BOS and samples from the renormalized permitted directions at each prefix. Sampling measurements use evaluation seeds 98301, 98302, and 98303, with $100{,}000$ requested outputs per seed and checkpoint.

G.8 Biased rooted trees on the 7×7 torus

The larger biased-tree MDM uses an exchangeable count-energy distribution

$$q_\theta(x) \propto \exp f_\theta\big(H(x)/48\big). \tag{104}$$

The network f_θ has scalar input, two hidden layers of width 64 with SiLU activations, and a scalar output, totaling 4353 parameters. Its masked predictions are conditional marginals of this distribution. The number of masked variables and the horizontal count in the visible exterior determine those marginals.

For each bias factor, integer-weight rejection generates $50{,}000$ independent samples from p_c with data seed $97100 + c$. A seed-0 NumPy permutation selects $45{,}000$ training and 5000 validation examples. The empirical horizontal-count histograms summarize these samples for training. With model seeds 0, 1, and 2, the network uses PyTorch's seeded linear-layer initialization and is optimized in float64 on CPU using Adam at learning rate 0.003, gradient-norm clipping at 1, and 2000 full-objective updates.

The training objective averages the absorbing-mask loss exactly over corruption times $t \sim \mathrm{Uniform}(0, 1)$ and mask subsets. Let $\mathcal{U}_m$ denote the uniform distribution over the m-element subsets of $[n]$. For an empirical data distribution $\widehat{p}$ on $n = 48$ variables, the objective is

$$\mathcal{L}_{\mathrm{count}}(\theta) = \frac{1}{n} \sum_{m=1}^{n} \mathbb{E}_{\substack{x \sim \widehat{p} \\ B \sim \mathcal{U}_m}} \left[\frac{1}{m} \sum_{i \in B} - \log q_\theta(x_i \mid x_{\overline{B}}) \right]. \tag{105}$$

The implementation evaluates this expectation from the empirical count histogram and hypergeometric probabilities for the horizontal count in the visible subset. Validation uses the corresponding exact expectation on the held-out histogram every 20 updates, including update 0. Minimum validation loss selects updates $(1760, 1780, 1860)$ for $c = 2$ and $(1900, 1920, 1960)$ for $c = 4$.

For deployment, the selected network defines positive integer weights

$$\begin{aligned} a_h &= f_\theta(h/n) - \max_{0 \le k \le n} f_\theta(k/n), \\ w_h &= \max\{1, \mathrm{round}(65536 \exp a_h)\}, \qquad h = 0, \dots, n. \end{aligned} \tag{106}$$

Initialization samples the joint distribution proportional to $w_{H(x)}$ by proposing independent uniform directions and accepting with probability $w_{H(x)}/\max_h w_h$. Joint repair uses this same proposal-and-accept procedure on the masked variables while holding the exterior fixed. The accepted block follows the deployed model's conditional distribution.

Parallel repair draws each masked direction independently from its conditional marginal. If the repair mask B has size $b = |B|$ and the exterior contains a horizontal directions, the horizontal probability is

$$p_{\mathrm{H}}(b, a) = \frac{\sum_{j=1}^{b} \binom{b-1}{j-1} w_{a+j}}{\sum_{j=0}^{b} \binom{b}{j} w_{a+j}}. \tag{107}$$

The four directional probabilities are $((1-p_{\mathrm{H}})/2, p_{\mathrm{H}}/2, (1-p_{\mathrm{H}})/2, p_{\mathrm{H}}/2)$. Integer sums are formed before the final float64 division. Parallel and joint sampling share saved initial assignments, use CPU batches of 1024, and allow $100{,}000$ nonempty repair decisions per output.

The AR comparator uses the shared causal Transformer architecture with 48 parent positions. It receives the same training examples and split as the count-energy model for each bias factor and is initialized separately with seeds 0, 1, and 2. Training runs for 50 epochs with AdamW at learning rate 3×10^{-4}, weight decay 0.01, batch size 256, gradient-norm clipping at 1, and teacher-forced cross entropy. Minimum held-out negative log likelihood selects epochs $(12, 6, 2)$ for $c = 2$ and $(22, 20, 35)$ for $c = 4$. Generation uses the online Transformer KV cache and the exact prefix completion rule described for uniform trees, at temperature 1 with batches of 256. Each trained checkpoint is evaluated with seeds 98401, 98402, and 98403, requesting $10{,}000$ outputs per seed.

G.9 Evaluation and aggregation

Distributional fidelity. For a distribution q over reported outcomes and target p, total variation is $\frac{1}{2}\sum_z |q(z) - p(z)|$. In the empirical evaluations, the frequency of each outcome is divided by the number of requested outputs. Invalid assignments and unfinished requests enter a failure outcome whose target probability is zero. SAT uses the clause-profile vector (C_1, C_2, C_3), where C_j counts clauses containing exactly j true literals. Coloring uses its three-color count vector. The uniform and large biased-tree panels use the horizontal-parent count H. These statistics define the distributions compared in the corresponding panels. For the 3×3 biased trees, finite-state propagation over all 4^8 parent assignments gives the complete terminal distribution and its full-joint TV.

The SAT target profile is estimated from the independent 50,000-assignment reference sample. The 32-step coloring and tree count targets are obtained from exact counting. The 256-step coloring panel compares each method's single 10,000-output run with the same 50,000-sample empirical exact-DP reference, as described in Section G.3. Among the eight retained configurations, SAT, coloring, and uniform-tree empirical distributions each pool three sampling runs of 10,000 requests for every method. For the large biased trees, this pooling is performed separately for each trained checkpoint, and the three resulting TV values are averaged. Small biased-tree panels average the numerically propagated TV values over the three trained checkpoints. Their timing measurements use three runs of 100,000 requests per checkpoint. Finite sampling contributes variation to the empirical TV measurements, including those of the uniform controls.

Uniform PRS supplies the uniform feasible distribution in each local comparison. For a biased target, its plotted TV is evaluated against that panel's specified target. The same retained uniform-tree sample is evaluated at both bias factors. The uniform 5×5, 6×6, and 7×7 controls use 30,000 saved outputs from the three stated seeds, with the count histograms recomputed for this comparison. The nine additional neural evaluations on these uniform trees reuse their trained checkpoints and add three sampling runs per method and graph.

The SAT validity fractions are 98.7067% for parallel MDM, 99.8067% for joint MDM, and 99.0200% for constrained AR. The other displayed methods complete all requests with feasible outputs. The failure outcome is included in every reported TV calculation.

Elapsed time. The horizontal axis reports seconds per 10,000 requested outputs on a logarithmic scale. Uniform-tree MDM timing includes generation of the shared initial assignments and the subsequent resampling loop, with the initialization cost charged to each method. The loop includes its constraint checks and ends before the separate CPU final-validity cross-check. Uniform-tree AR

Distribution	Bias	MDM	Baseline TV	Acceptance + MDM
Sink-free SAT	2×	0.07373	0.07054	0.07188
Sink-free SAT	4×	0.07739	0.06845	0.06819
Rooted trees 3×3	2×	0.09924	0.07789	0.09926
Rooted trees 3×3	4×	0.09743	0.07598	0.08990
Rooted trees $7\times7^{\dagger}$	2×	0.02330	0.00793	0.02291
Rooted trees $7\times7^{\dagger}$	4×	0.03367	0.00756	0.01382

Table 8: The effect of the acceptance algorithm on the distributional fidelity achieved by MDM. The acceptance mechanism described in section 4 is able to improve consistently the model's distributional fidelity, especially in more biased distributions.

timing includes generation, prefix-completion masks, categorical draws, transfers, bookkeeping, and the final validity check. These neural evaluations use an NVIDIA GeForce RTX 2080 Ti, PyTorch 2.7.1 with CUDA 11.8, two CPU threads, and batches of 256.

The retained uniform-tree Uniform PRS measurements cover resampling and constraint checks from saved uniform initial assignments. SAT and 32-step coloring measurements cover their recorded generation and resampling procedures. The 32-step coloring measurements include the first model call.

For the 256-step coloring panel, the retained parallel MDM timer covers its complete initialization and repair loop and ends before the final conversion of the output tensors to NumPy arrays. The constrained AR timer includes sequential generation and that final conversion. Both retained runs include the first sampling call without a dedicated warmup. Uniform PRS times CPU initialization and the complete resampling loop. Joint MDM synchronizes CUDA at timer boundaries and records initialization and resampling separately, including progress bookkeeping. For the neural methods, model loading and generator setup precede these timers. The sampling seeds and the original timing wrapper for each run are retained with its results.

Small biased-tree measurements begin after the probability and feasibility tables have been compiled and measure CPU table sampling. Large biased-tree MDM measurements use the structured CPU samplers, while their AR measurements use online GPU inference. Each panel therefore reports the runtime of its specified model and implementation. Model loading, archival, and distributional calculations are performed separately from the sampling timers.

H Acceptance on biased SAT and rooted-tree distributions

We compare joint MDM resampling with the acceptance-and-restart rule in Section 4 on biased sink-free SAT and biased rooted spanning trees. The SAT instance encodes sink-free orientations of the cube graph using 12 binary variables and eight clauses. Its target increases the weight of assignments with at least six forward edges by a factor of 2 or 4, then conditions on feasibility. The tree targets apply the same factors to assignments with at least half of their parent directions horizontal. Table 8 reports empirical TV together with its matched finite-sample reference.

Table 8 compares joint MDM repair with the old-value acceptance rule in Equation (79). This appendix specifies the biased SAT experiment and the evaluation shared with the biased rooted-tree rows. Each method pair uses the same trained checkpoint and saved initial assignments. The count-energy models define all resampling distributions from a common learned joint, allowing the acceptance theorem to apply directly to that joint.

H.1 An extremal SAT instance with a global count bias

The graph is the three-dimensional cube on vertices $0,\dots,7$, where two vertices are adjacent when their binary labels differ in one coordinate. Its 12 edges are ordered lexicographically. For an edge $\{u,v\}$ with $u<v$, its Boolean variable equals one when the edge points from u to v. Each vertex contributes a three-literal clause requiring at least one outgoing edge. Thus the formula has 12 variables, eight clauses, and 962 satisfying assignments among its 4096 complete assignments. Write $\mathcal{F}$ for this feasible set.

A bad event is a sink vertex. Adjacent sink events are mutually exclusive, since their shared edge can point toward only one endpoint. Nonadjacent sink events depend on disjoint edge variables. This is an extremal event system, for which product PRS samples the uniform distribution conditioned on feasibility. At each check, the PRS repair set S consists of the current sinks and its repair mask $B = V(S)$ contains their incident edges. Exhaustive enumeration verifies that this mask agrees with the general PRS construction on every assignment.

Let $H(x) = \sum_{i=1}^{12} x_i$. For $c \in \{2, 4\}$, the unconditioned data distribution and desired feasible target are

$$p_c(x) = \frac{2^{-12}}{Z_c}\, c^{\mathbf{1}\{H(x)\geq 6\}}, \qquad p_{c,\mathcal{F}}(x) = p_c(x \mid \mathcal{F}). \tag{108}$$

The tilt increases the relative probability of assignments with at least six forward edges. The experiment uses this one fixed graph at both bias strengths, with three independently initialized models per strength.

H.2 Training and conditioning of the count-energy MDM

For each c, exact sampling from p_c produces $50{,}000$ unconditioned assignments using data seed $381000 + c$. A NumPy permutation with seed 0 assigns $45{,}000$ examples to training and 5000 to validation. The three model seeds are 0, 1, and 2, with the data and split shared across them.

The model defines

$$q_\theta(x) \propto \exp f_\theta\big(H(x)/n\big), \qquad n = 12. \tag{109}$$

The scalar network f_θ has two hidden layers of width 64 with SiLU activations and a scalar output, giving 4353 parameters. Linear layers use PyTorch's default initialization after setting the model seed. Computation uses float64 and two CPU threads.

The denoiser's context consists of the mask size $b = |B|$ and the number a of ones retained outside B. Exchangeability makes these sufficient for every conditional prediction. With $v_h = \exp f_\theta(h/n)$, define

$$D_b(a) = \sum_{j=0}^{b} \binom{b}{j} v_{a+j}, \qquad \pi_\theta(b, a) = \frac{\sum_{j=1}^{b} \binom{b-1}{j-1} v_{a+j}}{D_b(a)}. \tag{110}$$

Every masked coordinate has conditional probability $\pi_\theta(b, a)$ of being one. Joint block probabilities additionally depend on the number of ones in the entire proposed block. The model's fixed predictions use the current repair mask and retained values, with fresh randomness for each resampling step.

Training minimizes the absorbing-mask denoising objective

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{x\sim\widehat{p} \\ \tau\sim\mathrm{Uniform}(0,1) \\ B\sim\mathrm{Bernoulli}(\tau)^n}} \left[-\frac{1}{n\tau} \sum_{i\in B} \log q_\theta(x_i \mid x_{B^c}) \right]. \tag{111}$$

Here $\widehat{p}$ is the empirical training distribution and each coordinate is masked independently with probability τ. Integrating over τ gives weight $1/n$ to each nonempty mask size when the token losses are averaged within the mask. Conditional on a complete count H, the retained count is hypergeometric. We therefore evaluate the full expectation exactly from the empirical count histogram.

Adam uses learning rate 0.003, gradient-norm clipping at 1, and 2000 full-objective updates. The held-out version of Equation (111) is evaluated every 20 updates, including update 0. Its minimum selects the checkpoint. The selected updates are $(520, 760, 400)$ for $c = 2$ and $(960, 620, 600)$ for $c = 4$, in model-seed order.

Deployment tabulates positive integer weights

$$w_h = \max\left\{1, \mathrm{round}\left(65536 \exp\left[f_\theta(h/n) - \max_{0\leq k\leq n} f_\theta(k/n)\right]\right)\right\}, \qquad q_w(x) \propto w_{H(x)}. \tag{112}$$

This same distribution supplies initialization, conditional resampling, and restarts. Across the six models, quantization changes the feasible joint by at most 3.31×10^{-6} in TV and any block conditional by at most 1.41×10^{-5}.

Bias	Model seed	Selected update	MDM TV	Acceptance + MDM TV
$2\times$	0	520	0.019236	0.007842
$2\times$	1	760	0.019408	0.008122
$2\times$	2	400	0.020782	0.013248
$4\times$	0	960	0.033791	0.004439
$4\times$	1	620	0.034245	0.004841
$4\times$	2	600	0.034177	0.005045

Table 9: Numerically propagated full-joint terminal TV to $p_{c,\mathcal{F}}$ on the cube SAT instance. Checkpoints are selected by held-out denoising loss.

H.3 Joint repair and the acceptance calculation

Joint MDM initializes from q_w. Each nonempty PRS repair mask is resampled jointly conditional on its retained exterior. To draw a block, the implementation proposes independent fair bits in the mask and accepts the proposal with probability equal to its complete-count weight divided by $\max_{0\le h\le n} w_h$. Repeated proposals give an exact conditional draw. Applying the same procedure to the full mask generates unconditional samples. Integer weights permit exact random integer comparisons.

For a repair mask of size b, let u be its old values, let $k = \sum_{i\in B} u_i$, and let a be the exterior count. Using $D_b(a) = \sum_{j=0}^{b} \binom{b}{j} w_{a+j}$ for the deployed weights, its conditional block probability and tight envelope are

$$q_w(u \mid z) = \frac{w_{a+k}}{D_b(a)}, \qquad h_S(u) = \min_{0\le a'\le n-b} \frac{w_{a'+k}}{D_b(a')}. \tag{113}$$

The minimum includes every exterior assignment through its count and depends only on the old block and mask. Integer cross products compare its candidate fractions exactly. Before resampling, the corrected sampler accepts with probability $h_S(u)/q_w(u \mid z)$. Acceptance triggers the same joint conditional draw used by the uncorrected sampler. Rejection discards the full assignment and independently restarts from q_w. A feasible assignment is returned immediately.

The positive weights, compatible joint conditionals, and exterior-independent lower envelope satisfy Theorem D.7 with $p = q_w$. Thus the uncapped corrected sampler has output distribution $q_w(\cdot \mid \mathcal{F})$. Its distance from the intended target $p_{c,\mathcal{F}}$ measures the learned distribution's approximation error.

H.4 Numerical distribution and operational measurements

The finite state space permits direct propagation of each sampler's probability distribution. Feasible assignments absorb probability mass. Each infeasible assignment sends mass to its possible conditional resampling outcomes and, for the corrected sampler, to unconditional restarts. Propagation stops when the remaining active mass is below 10^{-13}. The saved residual bounds the unpropagated contribution, and the largest recorded mass-conservation discrepancy is below 8×10^{-15}.

Table 9 gives the resulting full-joint TV to the intended feasible target. Acceptance reduces this TV for all six checkpoints. The mean decreases from 0.01981 to 0.00974 at $c = 2$ and from 0.03407 to 0.00478 at $c = 4$. For every corrected checkpoint, the numerical terminal distribution agrees with $q_w(\cdot \mid \mathcal{F})$ to within 10^{-13} in TV. Uniform PRS has terminal TV 0.14404 and 0.24416 to these two biased targets.

Operational evaluation requests $10{,}000$ outputs for each sampling seed 382001, 382002, and 382003, separately for every checkpoint and method. Each pair shares its saved unconditional initial assignments. Batches contain 1024 requests, with a cap of $10{,}000$ nonempty repair decisions per output. The two learned methods produce $360{,}000$ outputs, and a shared Uniform PRS control supplies another $30{,}000$. All requests finish with feasible outputs. Evaluation seeds are pooled within each checkpoint before averaging the three checkpoint results.

Table 10 separates completed resampling steps, full restarts, and all nonempty repair decisions. Acceptance rates count accepted old-value checks divided by all such checks. Wall time includes initial sampling, table setup, repair checks, block proposals, acceptance decisions, restarts, and event

Bias	Method	Resamplings	Restarts	Decisions	Accept	Time (s)
2×	MDM	1.8786	0	1.8786	1	0.0773
2×	Acceptance + MDM	1.6532	0.4374	2.0906	0.7908	0.1579
4×	MDM	1.8683	0	1.8683	1	0.1045
4×	Acceptance + MDM	1.3856	0.9223	2.3079	0.6004	0.2033

Table 10: CPU sampling measurements for biased SAT, averaged over three trained checkpoints. Work counts are per requested output and time is seconds per $10{,}000$ outputs. A full restart contributes its generation cost in addition to the accepted local resampling steps.

collection. Training, finite-state analysis, file compression, and final diagnostic aggregation are timed separately. Model weights are tabulated before sampling and table setup is charged to each method.

H.5 The rooted-tree comparison and finite-sample reference

The rooted-tree representations and biased targets are specified in Section G.5. The 3×3 rows of Table 8 use the native Transformer MDM trained in Section G.7. For each mask, its ascending sequential denoising predictions define a block distribution $K_B(\cdot \mid z)$. The corresponding acceptance envelope minimizes $K_B(u \mid z)$ over every exterior, using the precomputed float64 probability tables for all nonempty masks. These rows evaluate the correction empirically using the native predictions. The largest conditional TV discrepancy from the model's ascending initialization joint ranges from 0.05502 to 0.09903 at $c = 2$ and from 0.17346 to 0.22669 at $c = 4$ across the three seeds. The theorem's common-joint compatibility assumption holds by construction for the count-energy SAT and 7×7 tree models.

The 7×7 rows use the trained count-energy MDM in Section G.8. Its acceptance rule also minimizes the old-block conditional over all exterior horizontal counts, with exact integer comparisons. Both tree comparisons preserve the checkpoint, initialization arrays, and evaluation seeds used by their uncorrected counterparts. The 3×3 comparison pools $300{,}000$ outputs per checkpoint and measures full-joint empirical TV. The 7×7 comparison pools $30{,}000$ outputs per checkpoint and measures TV of the horizontal-parent count. The SAT rows pool $30{,}000$ outputs per checkpoint and measure full-joint empirical TV.

For a target distribution p on the reported outcomes and N independent target samples, the finite-sample reference in the main table is

$$\beta_N(p) = \frac{1}{2} \sum_y \mathbb{E} \left| \frac{C_y}{N} - p(y) \right|, \qquad C_y \sim \mathrm{Binomial}(N, p(y)). \tag{114}$$

It is evaluated from the exact target probabilities with the same outcome statistic and sample count as the corresponding model row. Equivalently, each outcome contributes $p(y)(1 - p(y)) \Pr\{\mathrm{Binomial}(N-1, p(y)) = \lfloor Np(y) \rfloor\}$. The reference is an expectation and individual empirical measurements can lie below it. All model entries first pool evaluation seeds within a checkpoint, then average the three checkpoint TVs. The finite-sample measurements in the main table and the numerical terminal distributions in Table 9 therefore quantify the same target comparison at different levels of sampling variation.

The SAT protocol and source snapshot were frozen before training. The archived records retain actual datasets, selected parameters, training histories, integer deployment weights, shared initial arrays, raw outputs, per-output work, acceptance events, and numerical terminal distributions. SHA256 receipts bind these artifacts to the protocol and source snapshot. The independent artifact audit passed $17{,}378$ checks, including all saved acceptance fractions, output accounting, target probabilities, residual mass, and corrected-distribution comparisons. The combined main table is regenerated from the audited SAT and rooted-tree records with their sample sizes and target statistics preserved.

I FURTHER RELATED WORK

Constructive resampling and exact sampling. Moser–Tardos finds feasible assignments under local lemma conditions (Moser & Tardos, 2010), while broader analyses allow randomized local moves that differ from a reference distribution (Achlioptas et al., 2019). General PRS supplies our repair-set rule and path argument, together with exact sampling from conditioned product distributions (Guo et al., 2019). Recent preprints give approximate sampling under degree-square LLL conditions (Achlioptas, 2026; Jain et al., 2026). The latter handles general product distributions without a separate overlap condition. Dynamic and perfect sampling also correct local updates to preserve target distributions (Feng et al., 2021; 2022). We extend the PRS repair analysis to model conditionals that depend on the current mask and retained values.

Learning with local resampling. NELSON learns constrained Markov random fields through PRS, with an exactness theorem for extremal constraints and fixed product resampling (Jiang et al., 2023). Learned product sampling oracles also guide stochastic local search, with LLL-based work bounds for Moser–Tardos (Kramer et al., 2026). We instead study context-dependent masked predictions and bound the number of constraints selected across repair rounds separately from correctness of the conditioned output distribution.

Discrete diffusion guarantees. Masked diffusion training (Sahoo et al., 2024; Shi et al., 2024) yields ideal predictions corresponding to clean-data conditionals (Ou et al., 2025). Recent analyses bound sampling error and parallel-unmasking complexity under assumptions on prediction accuracy and dependence (Liang et al., 2026; Chen et al., 2026). For fixed-order Markov walks, exact conditional predictions and separator-based unmasking give exact samples in logarithmic parallel depth (Bansal et al., 2026). These guarantees exploit properties of the modeled distribution, while we study repair under additional overlapping constraints.

Constrained diffusion and correction. Constrained decoders provide block optimality under regular-language constraints (Suresh et al., 2025), grammar-compatible completions (Mündler et al., 2026), or feasibility bounds under projection assumptions (Cardei et al., 2025). Dang & Ermon (2026) exactly sample each step's constrained product of token predictions for deterministic finite automata. These guarantees do not establish the final distribution obtained by conditioning the full diffusion process. Diffusion trajectory probabilities also support importance sampling and Metropolis–Hastings correction (Sanokowski et al., 2025), while adaptive rejection samples constrained language-model distributions (Parys et al., 2025). We give separate conditions for exact output from local repair, including an acceptance-and-restart correction with a bound on its overhead.

Remasking and local refinement. MDMs can also revisit generated values: ReMDM remasks visible tokens at a scheduled rate (Wang et al., 2025), and PRISM remasks tokens flagged by learned quality scores (Kim et al., 2026). IterRef uses remasking to propose moves in Multiple-Try Metropolis, targeting reward-weighted distributions (Lee et al., 2025a). Closest to our algorithm, Iterative Partial Refinement (IPR) resamples randomly selected regions of a completed sample, improving constraint satisfaction on tasks such as Sudoku (Kang et al., 2026). In probabilistic combinatorics, local repair algorithms such as partial rejection sampling (PRS) (Guo et al., 2019) also resample regions of a completed sample until it is feasible. Under suitable local-lemma-type conditions, general PRS reaches feasibility efficiently and samples exactly from a fixed product distribution conditioned on feasibility. Its degree dependence nearly matches a known hardness barrier (Proposition C.10).

J PARALLEL AND JOINT RESAMPLING: WHAT CHANGES IN THE GUARANTEES?

Parallel resampling is simple to perform and gives especially convenient conditions for proving termination. Joint resampling can preserve dependencies between the missing tokens, which matters when the goal is to reproduce the data distribution conditioned on feasibility. These advantages concern different questions. This section compares their guarantees and computational costs, explains what each choice changes, and discusses what our experiments establish.

J.1 Masked-diffusion training and mask selection

Masked diffusion models (MDMs) learn to reverse a corruption process that replaces discrete variables with a mask symbol (Sahoo et al., 2024; Shi et al., 2024). Let p_{data} be a distribution on $\Omega = \prod_{v=1}^{n} \Sigma_v$ with each Σ_v finite. For $B \subseteq [n]$, $M_B x$ masks the variables in B and keeps the *visible values* (or *visible assignment*) $z = x_{B^c}$, where $B^c = [n] \setminus B$. For each $v \in B$, a denoiser with parameters θ predicts a distribution $\nu_{B,z,v}$ on Σ_v. Standard variational objectives reduce to a weighted masked cross-entropy

$$\mathcal{L}(\theta) = \mathbb{E}_{X \sim p_{\mathrm{data}},\, B \sim \rho} \left[w(B) \sum_{v \in B} - \log \nu_{B, X_{B^c}, v}(X_v) \right], \tag{115}$$

where the masking schedule determines a data-independent mask distribution ρ and positive weights $w(B)$. If the model is expressive enough and exactly minimizes (115), its predictions equal $p_{\mathrm{data}}(X_v \mid X_{B^c} = z)$ for every visible assignment z of positive probability (Ou et al., 2025). Generation starts from a fully masked assignment. The denoiser predicts a distribution over values at each masked position, selected positions are sampled and revealed, and this process repeats until the assignment is complete. From any $M_B x$, unmasking one variable at a time with exact predictions samples X_B from $p_{\mathrm{data}}(X_B \mid X_{B^c} = z)$ by the chain rule. The masking policy determines which positions to reveal and, when remasking is allowed, which generated values to reconsider. These choices can substantially affect generation quality at a fixed sampling budget: selecting positions based on prediction confidence improves task performance (Nie et al., 2025), while allowing previously generated tokens to be remasked improves generation quality (Wang et al., 2025).

J.2 The difference is in the distribution of the completed mask

Fix a repair mask B and visible values $z = x_{B^c}$ with $p(X_{B^c} = z) > 0$. A parallel resampling step evaluates the predictor once and samples all missing tokens independently from their predictions. Even if each prediction is exactly correct for the data, its resampling distribution is

$$K_B^{\mathrm{par}}(z, u) = \prod_{v \in B} p(X_v = u_v \mid X_{B^c} = z). \tag{116}$$

The ideal joint data resampling step instead draws from

$$K_B^{p}(z, u) = p(X_B = u \mid X_{B^c} = z). \tag{117}$$

These distributions agree if and only if the missing tokens are conditionally independent given these visible values.

For example, suppose two missing bits must agree under the data distribution, with 00 and 11 each having probability $1/2$. Each bit individually has probability $1/2$ of being one. Independent draws give 01 or 10 half the time, although neither occurs in the data. A joint resampling step can assign probability $1/2$ to each of 00 and 11 and zero to the other two outcomes.

"Joint" describes the distribution of the completed mask, not the timing of the draws. Our joint implementation draws tokens one at a time, revealing each before predicting the next. Exact data conditionals then give (117) by the chain rule in Appendix D. Our comparison uses the one-pass independent resampling step of Section 3. Multiple prediction stages can produce dependent samples even when each stage draws some tokens in parallel.

J.3 Joint resampling retains the main termination theorem

The repair-set rule and the variables left unchanged are the same for both choices. Consequently, the argument that a new violation must overlap the previous repair mask still holds. The path argument also holds: it describes how the rule adds constraints, rather than how the model draws tokens. Theorem 3.1 and Corollary A.6 therefore apply to joint resampling whenever their stated probability bounds and inequality are satisfied.

For the memoryless model, Q_x is the next-assignment distribution obtained from the current mask and its visible values. The theorem conditions only on the current repair set $R_t = S$, without revealing those values:

$$\overline{Q}_{t,S}(D_P) = \mathbb{E}[Q_{X_t}(D_P) \mid R_t = S].$$

This expectation uses the actual distribution of current assignments selecting S at round t. Although the resampling kernel is fixed, this conditional distribution can change with t.

For a fixed current assignment, parallel resampling gives independent coordinates. After combining the starting violation with the first overlap, the remaining factors along an induced path depend on disjoint coordinates, giving (38). One can average these complete products conditional on $R_t = S$, or use a uniform upper bound on each complete product. Averaging each factor separately can lose the dependence caused by shared visible values.

Under joint resampling, dependence can already occur at a fixed visible assignment. Two conditions with individual probabilities 0.1 may occur together with probability 0.1, rather than 0.01. The constraint graph records shared variables, not all statistical dependencies. Joint resampling therefore requires direct bounds on whole path probabilities or additional assumptions that justify a bound on them.

Corollary C.2 also bounds whole-path probabilities conditional on the current repair set, so it allows both joint and parallel resampling. Correct individual token probabilities alone do not ensure these path bounds. The stronger parallel sufficient condition in Appendix C.1 remains available when its statewise bounds hold.

J.4 Parallel resampling needs more assumptions for exact data sampling

The output-error theorem, Theorem D.6, compares each complete resampling step with an ideal sampler whose final output is known to be $p_{\mathcal{F}}$. It can be used for either a parallel or a joint learned resampling step. Its assumption bounds the expected local total variation error given the current repair set at each round.

For parallel resampling, even perfect individual predictions can leave

$$\mathrm{d}_{\mathrm{TV}}(K_B^{\mathrm{par}}(z,\cdot), K_B^{p}(z,\cdot)) > 0. \tag{118}$$

Training the individual predictions more accurately does not remove this difference: it comes from drawing the tokens independently. The output-error bound can consequently remain positive even in the limit of perfect predictions. Exact sequential data conditionals remove this particular difference for joint resampling.

A mismatch in one resampling step need not cause final output error, because later repairs can still yield the correct distribution.

Selecting a mask from violated constraints can bias the output even with exact initialization and exact joint resampling. Proposition D.4 gives a condition ensuring exact output, and Appendix D.6 gives a correction using restarts.

J.5 Computational cost and evidence from our examples

The parallel implementation uses one model evaluation per resampling step. Our sequential joint implementation uses $|B|$ evaluations for a mask of $|B|$ tokens. The training objective can remain masked cross-entropy. Only the use of predictions during generation changes. To translate the theorem's resampling-step bounds into model evaluations or running time, we must also account for the cost of each step.

The sparse budget example makes the distributional difference explicit. With the first proposal drawn exactly from p, perfect independent token predictions give output TV error $14/153$, while exact joint data resampling gives $p_{\mathcal{F}}$. For the exactly-one constraint, however, identical independent bit probabilities give a uniform distribution among the accepted strings with one true bit. In that case ideal parallel resampling also gives the correct final target. The proof and the trained-model results are in Appendix F.

We check the joint samplers in these experiments by enumerating their possible assignments and resampling outcomes. This establishes the required probability bounds for these small implementations. It does not show that joint path probabilities can be checked efficiently for a large neural model.

The tradeoff is therefore concrete: joint resampling can represent more of the data's dependencies, while one-pass parallel resampling offers cheaper generation and a product formula at each fixed visible assignment.