BELIEF PROPAGATION ENABLES COMPOSITIONAL DIFFUSION FOR CONSTRAINT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have shown promise as iterative solvers for constraint reasoning problems. Most approaches train a single model on complete problem instances, inhibiting systematic generalization to novel combinations of the same constraints. *Compositional diffusion* instead seeks to learn reusable local constraint "experts" that are assembled at inference time, enabling generalization to new problem structures. Existing methods in this setting inherit two challenges of compositional diffusion without resolving them. First, noising and composition do not commute: composing independently noised experts does not recover the distribution obtained by first composing the experts in the clean space and then applying the forward diffusion process. Second, to produce a consistent solution, variables that participate in multiple constraints must resolve conflicts in the experts' value preferences. We introduce *CompDiff-BP*, a compositional diffusion method that addresses both challenges by performing inference over clean assignments using loopy belief propagation. At each diffusion step, we construct a constraint-based factor graph for the posterior over clean assignments conditioned on the current noisy state, then use belief propagation to approximate variable marginals. Their means then parameterize the next reverse step. In this way, without retraining the local experts, we directly approximate the denoiser of the forward-noised clean composition while reconciling overlapping experts on their shared variables. We evaluate *CompDiff-BP* on 3-SAT, Graph Coloring, N-Queens, and Crosswords, where it consistently outperforms existing compositional approaches, with particularly strong gains on larger problem instances.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Austin et al., 2021) have shown promise in learning heuristics for (discrete) constraint satisfaction (Sun & Yang, 2023; Sanokowski et al., 2024). They learn to iteratively refine a noisy variable assignment towards a feasible solution. Existing approaches are typically trained end-to-end on complete problem instances and can struggle to generalize systematically to configurations that differ from those seen in training (Du & Kaelbling, 2024). For instance, a model trained to solve Graph Coloring on Erdos-Renyi graphs with n nodes and m edges may not work well on smaller/larger graphs or ones with a different structure. This is despite all Graph Coloring instances consisting of a conjunction of not-equal constraints on edges.

We leverage compositional diffusion to explore an alternate path: learn an expert for a single local component, then compose copies of these experts at inference time (Liu et al., 2022; Du & Kaelbling, 2024; Zhang et al., 2023b). A problem instance can be tackled by composing its own local experts without training a separate global denoiser. This modularity provides a natural mechanism for generalization across problem instances and structures. In constraint reasoning, the local components correspond to individual constraints. As visualized in Figure 1, the same learned experts can be reused as the number and arrangement of constraints change. Existing compositional diffusion methods exploit this modularity by combining local scores or energies, sometimes with enhanced sampling techniques (Yang et al., 2023; Kong et al., 2025; Oarga & Du, 2025).

However, existing approaches do not explicitly address two known challenges of compositional diffusion (Figure 2). First, forward noising and composition do not commute. Composing the clean distributions and then noising their joint distribution is generally not equivalent to independently

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

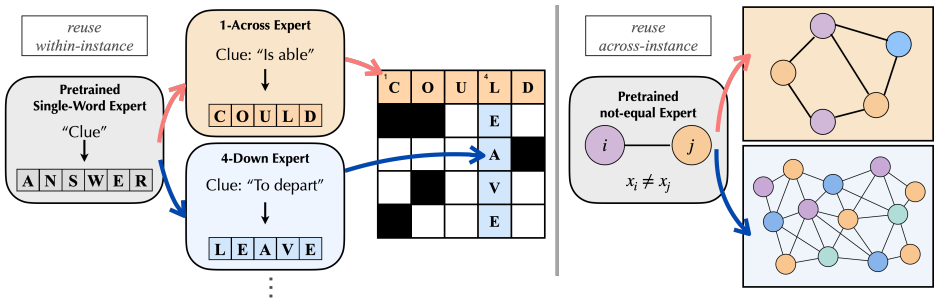

Figure 1: Compositional reasoning with reusable local experts. A pretrained expert for a local constraint can be reused to solve a larger problem at inference time. In crosswords (left), the same single-word expert is conditioned on different clues and applied to different answer slots within the puzzle. In graph coloring (right), copies of the same not-equal expert are applied across the edges of graphs with different sizes and structures. Assembling local experts allows one to construct solutions to a new instance family without training a separate model.

noising each factor and multiplying the results. Consequently, the noisy marginal of the desired clean distribution may not preserve the original factorization, and adding factor-wise scores at intermediate timesteps may not recover the score of the forward-noised composition. Second, local experts may disagree on shared variables. Simply combining their scores or energies does not explicitly resolve which preferences are mutually compatible. This is especially important in constraint reasoning, where overlapping experts correspond to constraints that must agree on shared variables.

We propose *CompDiff-BP*, a compositional diffusion method that addresses both challenges by reconciling learned experts through loopy belief propagation (LBP) in the predicted clean solution space. Each expert defines an energy-based potential over the variable assignments for one constraint. At each reverse step, a factor graph is built using the learned energies and the current noisy state. To reconcile expert disagreements, we run LBP on this factor graph and estimate the marginal distribution for each variable. Furthermore, since the experts are composed over predicted clean assignments rather than in noisy space, *CompDiff-BP* avoids the mismatch between noising and composition. The reconciled predictions are then used to perform the next reverse-diffusion step. *CompDiff-BP* achieves state-of-the-art performance among compositional approaches on 3-SAT, Graph Coloring, N-Queens, and Crosswords, with particularly strong gains on larger instances.

2 Background

2.1 Constraint satisfaction problems

A constraint satisfaction problem consists of variables $x = (x^{(1)}, \ldots, x^{(N)})$, each with a finite discrete domain $\mathcal{D}_i$, and a collection of constraints. Constraint $k \in [K]$ acts on a subset of variables $s_k \subseteq [N]$, its scope. A solution is an assignment to all variables that satisfies all constraints.

2.2 Factor graphs and loopy belief propagation

A factor graph represents a distribution as a product of local non-negative factor potentials. Given K factors with scopes $\{s_k\}_{k=1}^K$, the corresponding distribution over assignments factorizes as $p(x) = \frac{1}{Z}\prod_{k=1}^K \psi_k(x^{s_k})$, where Z is a normalizing constant, x^{s_k} denotes the variables in the scope of factor k and ψ_k is a non-negative function of those variables.

This factorization allows inference to be performed locally through message passing. Sum-product belief propagation estimates the marginal distribution of each variable by exchanging messages between variable and factor nodes (Kschischang et al., 2001; Wainwright & Jordan, 2008; Koller & Friedman, 2009). Let $\mathcal{N}(i)$ denote the factors neighboring variable i. A variable-to-factor message aggregates incoming messages from the variable's other neighboring factors, while a factor-to-

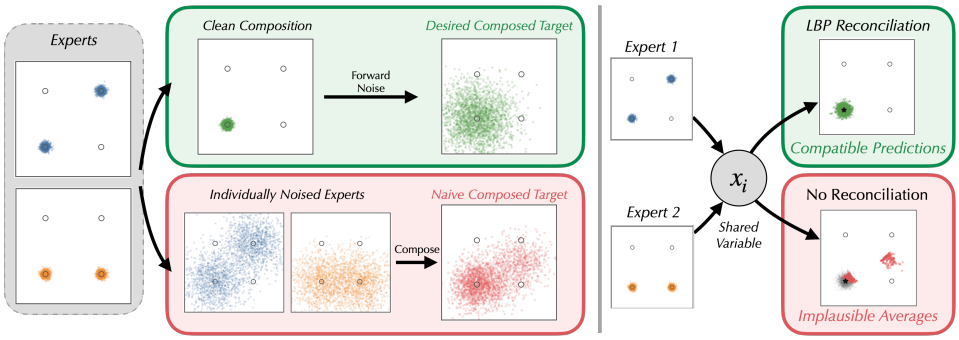

Figure 2: Two challenges in compositional diffusion. Left: Forward noising and composition do not commute. Composing the clean experts before applying the forward process yields the desired composed target, whereas composing the individually noised experts produces a different, naive target. *CompDiff-BP* targets the former. Right: Overlapping experts may favor incompatible values for shared variables. Without reconciliation, combining their predictions can produce implausible averages, whereas *CompDiff-BP* reconciles the experts toward compatible predictions.

variable message marginalizes over the remaining variables in that factor's scope:

$$m_{i\to k}(x^{(i)}) \propto \prod_{h\in\mathcal{N}(i)\setminus\{k\}} m_{h\to i}(x^{(i)}), \qquad m_{k\to i}(x^{(i)}) \propto \sum_{x^{s_k\setminus\{i\}}} \psi_k(x^{s_k}) \prod_{j\in s_k\setminus\{i\}} m_{j\to k}(x^{(j)}).$$

On trees, these updates recover the exact variable marginals. On graphs with loops, loopy belief propagation repeatedly applies the updates to obtain approximate marginals (Murphy et al., 1999).

Constraint satisfaction problems can be formulated with this representation by associating each problem variable with a variable node and each constraint with a factor over its scope. The factor potential assigns a weight to each local assignment according to how well it satisfies the corresponding constraint: hard constraints assign zero weight to violations, while relaxed or learned constraints may assign continuous values. Belief propagation has consequently also been studied as an inference procedure for constraint reasoning (Pesant, 2019).

2.3 Diffusion

The goal of a diffusion model is to learn a data distribution $q_0(x)$ so that new samples can be generated from it. It consists of two processes: starting from a clean sample $x_0 \sim q_0$, the fixed forward process progressively adds Gaussian noise,

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t I\right), \qquad q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right), \tag{1}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. The noise schedule is chosen such that q_T is close to a standard Gaussian. The learned reverse process then starts from a Gaussian noise sample $x_T \sim \mathcal{N}(0, I)$ and repeatedly applies the reverse transition $p_\theta(x_{t-1} \mid x_t)$ to recover an approximate sample from q_0 (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021).

Instead of directly learning the reverse transition, the reverse process is predominantly parameterized through one of three equivalent predictions. A model may predict the clean posterior mean $\widehat{x}_{0,\theta}(x_t, t) \approx \mathbb{E}[x_0 \mid x_t]$, the forward noise $\epsilon_\theta(x_t, t)$, or the score $s_\theta(x_t, t) \approx \nabla_{x_t} \log q_t(x_t)$. The predicted value is then used to obtain the reverse transition.

The reverse transition can also be parameterized through an energy function, which defines a probability density through the Boltzmann distribution (Du & Mordatch, 2019; Song & Kingma, 2021): $p_{\theta,t}(x_t) \propto \exp\left(-E_\theta(x_t, t)\right)$. The score can then be derived from the energy: $s_t(x_t) = \nabla_{x_t} \log p_{\theta,t}(x_t) = -\nabla_{x_t} E_\theta(x_t, t)$. An energy function assigns a scalar energy to each x, which we use to define factor potentials in Section 3.

2.4 Diffusion denoisers for known distributions

Diffusion can also be used when the target distribution $q(x_0)$ is known. Learning-free inference methods can be implemented as the denoiser and new samples can be generated (El Alaoui et al.,

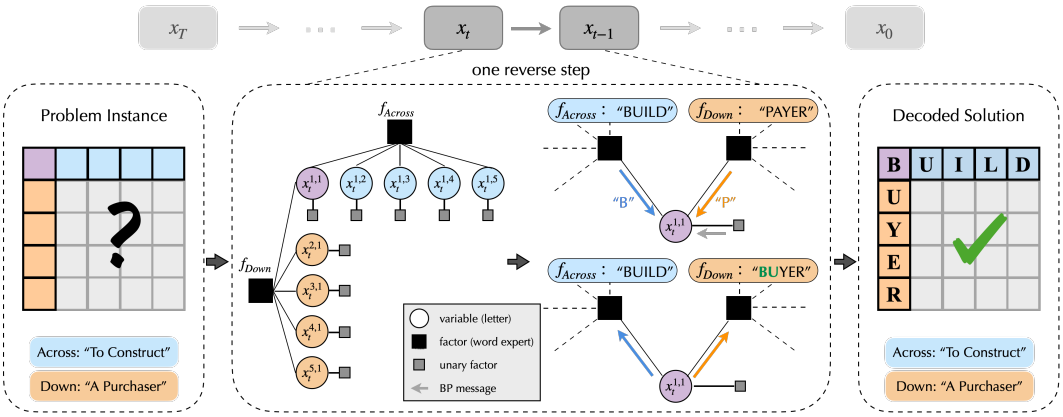

Figure 3: Overview of *CompDiff-BP* on a crossword example. The crossword instance defines a factor graph whose variable nodes correspond to cells and whose factor nodes correspond to clue experts. At timestep t in the reverse diffusion process, the across and down experts initially disagree on the shared letter $x_t^{1,1}$: the across clue *"to construct"* favors **BUILD**, while the down clue *"a purchaser"* favors **PAYER**. Loopy belief propagation passes messages through the shared variable, reconciling the experts so that the down expert shifts toward the compatible answer **BUYER**. The resulting variable marginals define the clean posterior mean used to parameterize the transition to x_{t-1}. Repeating this procedure across reverse-diffusion steps and decoding the final state produces the crossword solution shown on the right.

2022; Montanari & Wu, 2023; Ghio et al., 2024; Mei & Wu, 2025; Bhatt et al., 2026). At timestep t, the posterior over the clean sample can be defined as:

$$q(x_0 \mid x_t) = \frac{q_0(x_0)q(x_t|x_0)}{q_t(x_t)} \propto q_0(x_0)q(x_t \mid x_0),$$

where $q_0(x_0)$ is known and $q(x_t \mid x_0)$ is defined in Equation (1). The mean of this distribution, $\widehat{x}_{0|t}(x_t) = \mathbb{E}[x_0 \mid x_t]$, provides the clean prediction used in the reverse update. When the known target distribution factorizes, belief propagation can be used to infer this posterior mean. Bhatt et al. (2026) use this approach to construct the denoiser without learning. We build on this use of belief propagation within the reverse process for the compositional diffusion setting.

2.5 Compositional Diffusion

Compositional diffusion combines pretrained models at inference time so that a generated sample satisfies several factors jointly (Liu et al., 2022; Du & Kaelbling, 2024). Pretrained models with scopes s_k naturally define factors in a product-of-experts distribution (Hinton, 2002): $p_0^{\text{composed}}(x_0) \propto \prod_{k=1}^K p_{k,0}(x_0^{s_k})$. A common approach applies the same composition independently at each noise level t. Since products of distributions correspond to sums of energies and scores, this gives $E_t^{\text{composed}}(x_t) = \sum_{k=1}^K E_{k,t}(x_t^{s_k})$, and $s_t^{\text{composed}}(x_t) = \sum_{k=1}^K \nabla_{x_t} \log p_{k,t}(x_t^{s_k})$. This construction composes the independently noised factors. However, noising and composition do not generally commute:

$$\mathcal{T}_t\left[\prod_{k=1}^K p_{k,0}\right] \neq \prod_{k=1}^K \mathcal{T}_t[p_{k,0}], \tag{2}$$

where $\mathcal{T}_t[p_0](x_t) = \sum_{x_0} q(x_t \mid x_0)\, p_0(x_0)$ denotes the forward noising operation. Thus, multiplying the noisy factor distributions generally does not follow the forward diffusion of the desired clean composition (Du et al., 2023).

A second problem arises when several factors overlap in scope. Different factors may favor incompatible values for a shared variable. Adding their score directions can produce an intermediate value that is unlikely under either factor rather than selecting a compatible combination of modes.

Figure 2 visualizes these two issues, which motivate performing inference over the clean composition while allowing overlapping factors to reconcile their predictions.

3 COMPOSITIONAL DIFFUSION VIA LOOPY BELIEF PROPAGATION

We propose *CompDiff-BP*: a compositional diffusion approach for constraint satisfaction problems. Given pretrained constraint experts, *CompDiff-BP* composes them at inference time without retraining. It simultaneously addresses the two challenges identified in Section 2.5 through two corresponding design choices. Figure 3 and Algorithm 1 summarize the overall procedure.

Targeting the noised clean composition. Instead of composing the individually noised experts at timestep t, we directly target the distribution obtained by first composing the experts in clean space and then applying the forward process (left-hand side of Equation (2)). Let Z_0 denote the normalizing constant of the clean product. Our target at timestep t is

$$p_t^{\text{composed}}(x_t) = \mathcal{T}_t[p_0^{\text{composed}}](x_t) = \tfrac{1}{Z_0}\mathcal{T}_t\left[\textstyle\prod_{k=1}^K p_{k,0}(x_0^{s_k})\right](x_t).$$

The reverse process for this target can be parameterized by its clean posterior mean, $\widehat{x}_{0|t}^{\text{composed}}(x_t) = \mathbb{E}_{p^{\text{composed}}(x_0|x_t)}[x_0]$. Thus, each reverse step reduces to estimating the posterior marginals over clean assignments given the current noisy state.

Reconciling overlapping experts. Rather than obtaining the composed denoising prediction by summing the individually noised expert energies, we use the learned clean energies to construct a posterior over clean assignments and run loopy belief propagation over the resulting factor graph. This allows overlapping experts to exchange information and reconcile their preferences on shared variables. The resulting marginal means estimate $\widehat{x}_{0|t}^{\text{composed}}(x_t)$ and parameterize the next reverse-diffusion step. Thus, *CompDiff-BP* approximates the denoiser for the forward-noised clean composition while reconciling overlapping experts through a shared set of variable marginals.

Algorithm 1 *CompDiff-BP* overview

Require: $\{E_{\theta_k}, s_k\}_{k=1}^K$
1: $\psi_{\theta_k} \leftarrow \exp(-E_{\theta_k}/\tau)$ for all k
2: $x_T \sim \mathcal{N}(0, I)$
3: **for** $t = T, \dots, 1$ **do**
4: $\quad \mathcal{G}_t \leftarrow \textsc{PosteriorGraph}(x_t, \{\psi_{\theta_k}\})$
5: $\quad \{b_{i,t}\}_{i=1}^N \leftarrow \textsc{LoopyBP}(\mathcal{G}_t)$
6: $\quad \widehat{x}_{0|t} \leftarrow \{\mathbb{E}_{b_{i,t}}[x_0^{(i)}]\}_{i=1}^N$
7: $\quad x_{t-1} \sim p(x_{t-1} \mid x_t, \widehat{x}_{0|t})$
8: **end for**
9: **return** x_0

3.1 CONSTRUCTING THE POSTERIOR FACTOR GRAPH

We first specify how the pretrained local experts define the factors in the clean posterior. Let x_0 denote the clean variable assignment. Each discrete variable $x_0^{(i)}$ takes values in $\mathcal{D}_i$. Binary variables are represented directly, while variables with larger categorical domains are represented as one-hot vectors in $\mathbb{R}^{|\mathcal{D}_i|}$ for diffusion, following Chen et al. (2023). For each constraint k, we assume a pretrained local energy expert E_{θ_k}. Given an assignment $x_0^{s_k}$ to the variables in its scope, the expert returns a scalar energy $E_{\theta_k}(x_0^{s_k})$, with lower energy indicating greater preference under the learned constraint. We convert this energy into the categorical factor potential ψ_{θ_k} corresponding to the clean distribution: $p_{k,0}(x_0^{s_k}) \propto \psi_{\theta_k}(x_0^{s_k}) = \exp\left(-\frac{1}{\tau}E_{\theta_k}(x_0^{s_k})\right)$, where $\tau > 0$ is a temperature.

At reverse timestep t, the noisy state x_t is known while the clean assignment x_0 is unknown. Applying Bayes' rule to the composed clean distribution gives

$$p^{\text{composed}}(x_0 \mid x_t) \propto q(x_t \mid x_0) \textstyle\prod_{k=1}^K \psi_{\theta_k}(x_0^{s_k}).$$

Under the standard forward diffusion process, the forward transition has a diagonal covariance. It therefore factorizes across variables conditioned on x_0: $q(x_t \mid x_0) = \prod_{i=1}^N q(x_t^{(i)} \mid x_0^{(i)})$. The posterior therefore factorizes as

$$p^{\text{composed}}(x_0 \mid x_t) \propto \textstyle\prod_{i=1}^N q(x_t^{(i)} \mid x_0^{(i)}) \prod_{k=1}^K \psi_{\theta_k}(x_0^{s_k}). \tag{3}$$

Thus, the current noisy state contributes a unary potential for each variable, while the learned experts provide the potentials per constraint. This factor graph is the distribution on which we run loopy belief propagation.

3.2 RECONCILING OVERLAPPING EXPERTS WITH LOOPY BELIEF PROPAGATION

We approximate the variable marginals of Equation (3) using loopy sum-product belief propagation. Let $\mathcal{N}(i)$ denote the set of factor nodes neighboring variable i. At each round, we update the variable-to-factor and factor-to-variable messages as

$$\begin{aligned} m_{i\to k,t}(x_0^{(i)}) &\propto q\left(x_t^{(i)} \mid x_0^{(i)}\right) \prod_{h\in\mathcal{N}(i)\setminus\{k\}} m_{h\to i,t}(x_0^{(i)}). \\ m_{k\to i,t}(x_0^{(i)}) &\propto \sum_{x_0^{s_k\setminus\{i\}}} \psi_{\theta_k}(x_0^{s_k}) \prod_{j\in s_k\setminus\{i\}} m_{j\to k,t}(x_0^{(j)}). \end{aligned} \tag{4}$$

The first update combines the likelihood of the current noisy state with the incoming messages from all other neighboring factors, while the second marginalizes over the remaining variables in factor scope. At each reverse step, we initialize the factor messages uniformly and iterate Equation (4) for a fixed number of rounds. The resulting belief for variable i is

$$b_{i,t}(x_0^{(i)}) \propto q\left(x_t^{(i)} \mid x_0^{(i)}\right) \prod_{k\in\mathcal{N}(i)} m_{k\to i,t}(x_0^{(i)}), \tag{5}$$

Each expert influences this belief through its factor-to-variable message, which depends both on the assignments favored by that expert and on the current beliefs of the other variables in its scope. Repeated message passing therefore allows overlapping experts to reconcile their preferences through their shared variables. Because the full factor graph may contain loops, these beliefs are approximate posterior marginals. Nevertheless, they provide the variable-wise estimates needed to construct the clean posterior mean for the next reverse-diffusion step.

3.3 FROM BP MARGINALS TO THE REVERSE-DIFFUSION UPDATE

The BP beliefs from Equation (5) provide an approximation to the marginal posterior of each clean variable. We use these marginals to estimate the clean posterior mean: $\widehat{x}_{0|t}^{(i)} = \mathbb{E}_{b_{i,t}}[x_0^{(i)}]$. Collecting the variable-wise means gives the clean prediction $\widehat{x}_{0|t}$, which we use to parameterize the standard reverse-diffusion transition. The procedure is repeated at every reverse timestep: given x_t, we construct the posterior factor graph, run loopy belief propagation to obtain the variable marginals, form $\widehat{x}_{0|t}$, and sample x_{t-1}. Therefore, the reverse-diffusion process remains unchanged: *CompDiff-BP* replaces the denoising prediction at each timestep with one obtained by posterior inference over the composed constraint experts.

4 EXPERIMENTS

Our experiments aim to address four research questions: **(RQ1)** How effective is *CompDiff-BP* across constraint reasoning tasks? **(RQ2)** How does *CompDiff-BP* compare against existing compositional approaches? **(RQ3)** Is iterative reconciliation between experts important? **(RQ4)** Is diffusion a useful paradigm compared to directly optimizing or performing BP on the learned clean energies? We answer these questions empirically across four constraint reasoning benchmarks.

Benchmarks. Following Oarga & Du (2025), we evaluate on four constraint reasoning domains. In 3-SAT, the goal is to assign boolean values to variables so that every clause is satisfied. Each clause defines a local constraint over the variables it contains. In graph coloring, vertices must be assigned colors so that adjacent vertices have different colors. Each edge defines a local not-equal constraint. In 8-Queens, eight queens must be placed on an 8×8 board so that no two share a row, column, or diagonal. The problem decomposes into constraints over rows, columns, and diagonals. In Mini Crosswords, letters are assigned to grid cells so that the across and down words answer their clues and agree at intersecting cells. Each clue defines a constraint over the letters of its answer. For 3-SAT, we evaluate on UF20–91 and the larger UF50–218 instances. Graph coloring includes multiple graph families and scales, and crossword evaluation uses the 5×5 Crosswords benchmark from Yao et al. (2023). Full benchmark and instance details are provided in Appendix C.

Baselines. For compositional comparisons, we use the same pretrained local energy models across methods, so that comparisons reflect differences in compositional inference rather than in the learned experts. We also use the same candidate budget P, where each candidate corresponds to an independently sampled trajectory or optimization run. We compare *CompDiff-BP* with two compositional

Table 1: Results on 8-Queens (Left) and Mini Crosswords (Right). Compositional approaches share the same learned experts, 8-Queens results uses a budget of $P = 1024$ candidate trajectories. Mini Crosswords use a single candidate, except for the $P = 1024$ PEM setting used by Oarga & Du (2025).

Model	Correct (%)	Size ↑
Non-compositional approaches		
LWD	22	7.1000 ± 0.5744
GFlowNets	14	6.9293 ± 0.5904
DIFUSCO	17	6.9400 ± 0.6452
Fast T2T	41	7.3800 ± 0.5436
Compositional approaches		
LBP *(No Diffusion)*	0	5.2300 ± 0.4208
GD *(No Diffusion)*	**100**	$\mathbf{8.0000 \pm 0.0000}$
Bethe Correction	19	7.1900 ± 0.3923
PEM	97	7.9700 ± 0.1714
CompDiff-BP	**100**	$\mathbf{8.0000 \pm 0.0000}$

Model	Letter (%)	Word (%)	Game (%)
Non-compositional approaches			
IO	38.7	14.0	0
CoT	40.6	15.6	5
ToT	78.0	60.0	20
Compositional approaches			
LBP *(No Diffusion)*	73.6	51.5	20
GD *(No Diffusion)*	60.8	27.5	0
Bethe Correction	71.6	31.0	0
PEM	70.8	35.5	0
PEM ($P = 1024$)	81.6	50.0	0
CompDiff-BP	**85.8**	**65.5**	**25**

diffusion baselines. Parallel Energy Minimization (PEM) maintains a population of candidate solutions and optimizes the composed learned energy (Oarga & Du, 2025). The Bethe-style baseline composes the noisy experts using a static correction for repeated marginal information, without iteratively exchanging messages between factors (Mishra et al., 2023; Zhang et al., 2023b). We additionally evaluate two no-diffusion compositional baselines using the same energies: running loopy belief propagation directly on the constructed factor graph with uniform unary terms, as well as running gradient descent directly on the assignments to minimize the sum of the learned energies. For non-compositional baselines, we report the results from Oarga & Du (2025) directly. Additional baseline and implementation details are provided in Appendix D.

4.1 Results

RQ1: Overall performance. As seen in Tables 1 to 3, *CompDiff-BP* performs strongly across all four domains. On 3-SAT, it solves 92% of UF20–91 instances and 71% of the larger UF50–218 instances, improving substantially over the strongest reported baselines on the latter. On 8-Queens, it reaches 100% correct instances, tied with gradient descent and above the remaining approaches. On graph coloring, *CompDiff-BP* achieves the lowest number of conflicts in seven of the eight graph families. Finally, on Crosswords it achieves 85.8% letter accuracy, 65.5% word accuracy, and 25% exact puzzle accuracy. Together, these results show that *CompDiff-BP* is effective across constraint problems with substantially different structures and problem scales.

RQ2: Comparison with existing compositional approaches. Tables 1 to 3 also compare against existing compositional diffusion approaches PEM and the Bethe correction. *CompDiff-BP* provides the largest gains on settings that require stronger coordination across factors. On UF50–218, it solves 71% of instances compared with 43% for PEM and 33% for Bethe Correction. Similar gains appear on Mini Crosswords, where word accuracy increases to 65.5% from 35.5% for PEM and 31.0% for Bethe. These results establish *CompDiff-BP* as a strong compositional approach and provide evidence that both the noising–composition mismatch and the need to reconcile overlapping experts are important considerations in compositional diffusion.

RQ3: Importance of iterative reconciliation. To isolate the effect of iterative communication between experts with shared variables, we compare a single LBP round against multiple rounds (Table 4). With $R = 1$, factor messages are computed from the local expert and unary evidence before any messages from neighboring experts can propagate through shared variables. The resulting beliefs therefore combine the experts, but the experts cannot yet revise their messages in response to one another. For $R > 1$, messages from neighboring factors are fed back through the shared variables, enabling iterative reconciliation. This distinction has a substantial effect. Increasing R from 1 to 3 raises 8-Queens correctness from 12% to 100% and UF50–218 success from 2% to 47%, while graph-coloring conflicts decrease from 790 to 510. Further rounds improve UF50–218 and graph coloring to 71% and 386 conflicts, respectively. These results further showcase the importance of expert reconciliation and that static aggregation is often insufficient.

Table 2: 3-SAT results on 100 instances from each SATLIB distribution. All compositional approaches use the same refined SAT checkpoint and a budget of $P = 1024$ candidate trajectories. PEM (reported) denotes the values reported in Oarga & Du (2025).

	UF20–91		UF50–218	
Model	Correct (%)	Satisfied clauses ↑	Correct (%)	Satisfied clauses ↑
Non-compositional approaches				
GCN	5	0.9617 ± 0.0264	0	0.9569 ± 0.0203
DGL	10	0.9520 ± 0.0330	0	0.8705 ± 0.0405
DIFUSCO	6	0.9734 ± 0.0156	0	0.9738 ± 0.0156
Fast T2T	22	0.9760 ± 0.0273	20	0.9734 ± 0.0159
NeuroSAT	8	0.9742 ± 0.0154	0	0.9697 ± 0.0111
NSNet	58	0.9856 ± 0.0266	37	0.9846 ± 0.0205
Compositional approaches				
LBP *(No Diffusion)*	53	0.9896 ± 0.0135	21	0.9861 ± 0.0125
GD *(No Diffusion)*	3	0.9626 ± 0.0156	0	0.9383 ± 0.0114
Bethe Correction	89	0.9988 ± 0.0034	33	0.9954 ± 0.0043
PEM (reported)	91	0.9985 ± 0.0048	43	0.9963 ± 0.0046
PEM	81	0.9978 ± 0.0047	21	0.9940 ± 0.0045
CompDiff-BP	**92**	$\mathbf{0.9991 \pm 0.0030}$	**71**	$\mathbf{0.9986 \pm 0.0022}$

Table 3: Graph-coloring performance measured by conflicting edges (↓), reported as population mean and standard deviation over five graphs in each category. Compositional approaches use a budget of $P = 128$ candidate trajectories. Additional results are in Appendix E

	Erdős–Rényi		Holme–Kim		Regular Expander		Paley	Complete
Graph statistics								
$\mathcal{V}$	[20, 39]	[81, 99]	[22, 34]	[86, 100]	[21, 40]	[86, 100]	[19, 37]	[8, 12]
$\mathcal{E}$	[29, 76]	[193, 225]	[56, 92]	[398, 469]	[63, 120]	[184, 200]	[171, 465]	[36, 66]
d	0.12	0.05	0.26	0.10	0.22	0.23	0.80	1.00
Non-compositional approaches								
GCN	46.80 ± 20.47	151.60 ± 12.09	74.00 ± 14.74	408.00 ± 26.40	87.60 ± 22.58	144.80 ± 6.90	285.00 ± 117.70	46.00 ± 15.04
GAT	34.00 ± 11.55	130.20 ± 11.47	51.20 ± 10.03	253.20 ± 50.71	58.60 ± 12.44	118.80 ± 12.59	239.20 ± 159.43	46.00 ± 15.04
XLVIN	25.00 ± 7.81	93.80 ± 31.12	29.00 ± 7.75	182.60 ± 24.73	29.00 ± 7.75	112.60 ± 10.97	151.80 ± 92.13	34.80 ± 16.42
GNN-GCP	15.20 ± 4.32	53.80 ± 8.34	13.20 ± 7.46	55.20 ± 12.63	15.40 ± 6.65	141.60 ± 69.47	91.20 ± 63.14	30.00 ± 2.54
Compositional approaches								
LBP *(No Diffusion)*	0.40 ± 0.49	1.60 ± 0.80	0.80 ± 0.75	$\mathbf{7.60 \pm 3.44}$	79.20 ± 17.66	1.60 ± 1.02	194.20 ± 126.66	29.40 ± 10.98
GD *(No Diffusion)*	9.20 ± 4.58	40.00 ± 1.41	9.20 ± 2.32	62.80 ± 9.87	11.80 ± 4.35	44.80 ± 3.82	32.00 ± 16.24	2.40 ± 1.50
Bethe Correction	10.00 ± 5.06	42.80 ± 6.43	11.80 ± 3.19	69.00 ± 5.40	12.00 ± 5.10	47.00 ± 3.69	33.20 ± 12.69	2.40 ± 1.50
PEM	6.40 ± 3.61	23.40 ± 4.03	8.00 ± 2.90	49.80 ± 10.65	11.60 ± 5.95	24.80 ± 3.76	32.00 ± 13.51	2.60 ± 1.36
CompDiff-BP	$\mathbf{0.20 \pm 0.40}$	$\mathbf{0.60 \pm 0.80}$	$\mathbf{0.40 \pm 0.49}$	7.80 ± 3.66	$\mathbf{0.80 \pm 0.75}$	$\mathbf{0.40 \pm 0.49}$	$\mathbf{28.60 \pm 21.46}$	$\mathbf{2.20 \pm 1.17}$

RQ4: Role of diffusion. With the learned energy function available, a natural question is whether diffusion is needed in the first place. Can't we simply utilize the energy function directly for desired solutions? To answer this question, we implement loopy belief propagation and gradient descent using the same learned energies. We observe that neither provides a consistently strong alternative across tasks. LBP performs well on several graph-coloring families and reaches 51.5% word accuracy on Mini Crosswords, but solves only 21% of UF50–218 instances and produces no valid decoded 8-Queens boards. Gradient descent reaches 100% correct instances on 8-Queens, matching *CompDiff-BP*, but performs poorly on SAT and Mini Crosswords. In contrast, *CompDiff-BP* performs strongly across all four domains, suggesting that the compositional diffusion paradigm provides a useful mechanism for compositional reasoning.

4.2 Ablations

LBP rounds and sampling steps. We study sensitivity to the number of reverse-diffusion steps S and LBP rounds R in Table 4. To isolate each parameter, we vary S with $R = 2$ fixed and vary R with $S = 100$ fixed. Increasing S generally improves performance at this relatively shallow message-passing depth, most clearly on N-Queens and graph coloring, although the gains largely saturate by $S = 100$–200. The preferred LBP depth is more task dependent: SAT and graph coloring continue to benefit from additional rounds, N-Queens saturates by $R = 3$, and Mini Crosswords perform best at $R = 2$. Thus, neither deeper diffusion trajectories nor more message passing are uniformly beneficial, and the appropriate balance depends on the task.

Table 4: Ablation of reverse-diffusion steps S and BP rounds R. For the sampling-step ablation, we fix $R = 2$; for the BP-depth ablation, we fix $S = 100$. SAT and N-Queens report solved instances out of 100 (↑), coloring reports total conflicts (↓), and crosswords report word accuracy (↑).

	Reverse-diffusion steps S ($R = 2$)				BP rounds R ($S = 100$)					
Task	20	50	100	200	1	2	3	5	10	25
SAT UF20-91 ↑	89	**92**	**92**	**92**	87	**92**	**92**	**92**	**92**	**92**
SAT UF50-218 ↑	7	23	20	**27**	2	20	47	60	69	**71**
N-Queens ↑	28	72	84	**95**	12	84	**100**	**100**	**100**	**100**
Coloring ↓	1080	910	850	**830**	790	850	510	428	**386**	388
Crosswords ↑	52.5	60.0	**65.5**	58.5	44.5	**65.5**	60.0	57.0	53.5	54.0

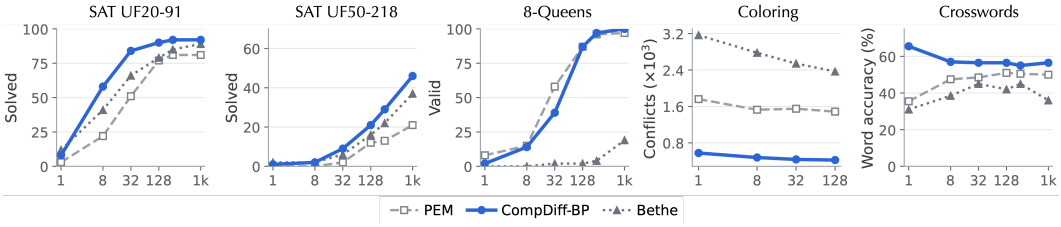

Figure 4: Candidate-efficiency curves for compositional diffusion approaches. Each panel reports one primary metric. PEM coloring results timed out on large candidate counts.

Candidate count. We next study sensitivity to the number of candidate trajectories P in Figure 4. Across several tasks, *CompDiff-BP* reaches strong performance with substantially fewer candidate assignments than PEM and the Bethe correction. Performance does not always improve with increasing P because candidate selection uses the learned composed energy rather than an oracle measure of solution quality. This is particularly noticeable for Crosswords, whose expert is trained without the contrastive loss used to align low-energy regions with ground-truth solutions (Appendix B), making energy-based candidate selection less reliable as P increases.

5 Related Work

Existing compositional diffusion methods improve composed generation along two complementary axes: defining the composed target and sampling from it. Methods such as DiffCollage (Zhang et al., 2023b) and Generative Skill Chaining (Mishra et al., 2023) modify the composition rule to account for overlap between local models, using Bethe-style corrections for repeated information. Compositional Visual Planning introduces additional boundary constraints to facilitate expert reconciliation (Zhang et al., 2026). Other approaches focus on improving the sampling procedure for a specified composed target. Reduce, Reuse, Recycle uses MCMC-based sampling (Du et al., 2023), Feynman–Kac Correctors use sequential Monte Carlo (Skreta et al., 2025), and Parallel Energy Minimization maintains a population of candidate solutions (Oarga & Du, 2025). *CompDiff-BP* belongs to the first group: by performing LBP on the forward-noised clean composition, it directly targets the desired composed distribution. An extended related work section is provided in Appendix A.

6 Limitations and Conclusion

We introduced *CompDiff-BP*, a compositional diffusion method for constraint reasoning. At each reverse step, *CompDiff-BP* performs inference over the clean composition using loopy belief propagation, allowing overlapping experts to reconcile their predictions. Across 3-SAT, graph coloring, 8-Queens, and Crosswords, *CompDiff-BP* achieves state-of-the-art results among existing compositional approaches. One limitation of our work is that factor-to-variable updates enumerate local assignments, making the current implementation best suited to discrete problems with tractable factor scopes. Approximate message representations, such as Gaussian belief propagation (Ortiz et al., 2021) or particle belief propagation (Ihler & McAllester, 2009), could extend the framework to larger or continuous domains. Our work focuses on the composed denoising target and uses the standard reverse-diffusion sampler. Future work could combine *CompDiff-BP* with more advanced compositional samplers discussed in Section 5.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing or refining hypotheses, designing research methodology or experiments, or interpreting results, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for creating and editing software code, drafting and editing parts of the paper, searching for and identifying relevant literature, and summarizing existing literature. We have reviewed all AI-assisted work. AI-assisted code was reviewed and tested by the authors, references and literature summaries were checked against the original sources, and AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sungsoo Ahn, Younggyo Seo, and Jinwoo Shin. Learning what to defer for maximum independent sets. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 134–144. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/ahn20a.html.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Alankrita Bhatt, Mukur Gupta, Germain Kolossov, and Andrea Montanari. Generating from discrete distributions using diffusions: Insights from random constraint satisfaction problems. *arXiv preprint arXiv:2603.20589*, 2026.

Maximilian Böther, Otto Kißig, Martin Taraz, Sarel Cohen, Karen Seidel, and Tobias Friedrich. What's wrong with deep learning in tree search for combinatorial optimization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=mk0HzdqY7i1.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3itjR9QxFw.

Andreea Deac, Petar Veličković, Ognjen Milinković, Pierre-Luc Bacon, Jian Tang, and Mladen Nikolić. XLVIN: executed latent value iteration nets. In *Learning Meets Combinatorial Algorithms at NeurIPS2020*, 2020. URL https://openreview.net/forum?id=yDKZFz_bT6p.

Yilun Du and Leslie Pack Kaelbling. Position: Compositional generative modeling: A single model is not all you need. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11721–11732. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/du24d.html.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023.

Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Sampling from the sherrington-kirkpatrick gibbs measure via algorithmic stochastic localization. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 323–334. IEEE, 2022.

Davide Ghio, Yatin Dandi, Florent Krzakala, and Lenka Zdeborová. Sampling with flows, diffusion, and autoregressive neural networks from a spin-glass perspective. *Proceedings of the National Academy of Sciences*, 121(27):e2311810121, 2024.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Alexander Ihler and David McAllester. Particle belief propagation. In David van Dyk and Max Welling (eds.), *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 256–263, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/ihler09a.html.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Dongxia Wu, Haorui Wang, Aaron M Ferber, Yian Ma, Carla P Gomes, and Chao Zhang. Diffusion models as constrained samplers for optimization with unknown constraints. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL https://openreview.net/forum?id=dEwvHEHxPc.

F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. doi: 10.1109/18.910572.

Henrique Lemos, Marcelo Prates, Pedro Avelar, and Luis Lamb. Graph colouring meets deep learning: Effective graph neural network models for combinatorial problems. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 879–885, 2019. doi: 10.1109/ICTAI.2019.00125.

Yang Li, Jinpei Guo, Runzhong Wang, Hongyuan Zha, and Junchi Yan. Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xDrKZOZEOc.

Zhaoyu Li and Xujie Si. Nsnet: a general neural probabilistic framework for satisfiability problems. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Song Mei and Yuchen Wu. Deep networks as denoising algorithms: Sample-efficient learning of diffusion models in high-dimensional graphical models. *IEEE Transactions on Information Theory*, 71(4):2930–2954, 2025.

Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pp. 2905–2925. PMLR, 2023.

Utkarsh Aashu Mishra, Yongxin Chen, and Danfei Xu. Generative factor chaining: Coordinated manipulation with diffusion-based factor graph. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=p6Wq6TjjHH.

Utkarsh Aashu Mishra, David He, Yongxin Chen, and Danfei Xu. Compositional diffusion with guided search for long-horizon planning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=b8avf4F2hn.

Andrea Montanari and Yuchen Wu. Posterior sampling in high dimension via diffusion processes. *arXiv preprint arXiv:2304.11449*, 2023.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pp. 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606149.

Alexandru Oarga and Yilun Du. Generalizable reasoning through compositional energy minimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=5k0AHYc4MJ.

Joseph Ortiz, Talfan Evans, and Andrew J Davison. A visual introduction to gaussian belief propagation. *arXiv preprint arXiv:2107.02308*, 2021.

Gilles Pesant. From support propagation to belief propagation in constraint programming. *Journal of Artificial Intelligence Research*, 66:123–150, 2019.

Sebastian Sanokowski, Sepp Hochreiter, and Sebastian Lehner. A diffusion model framework for unsupervised neural combinatorial optimization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=AFfXlKFHXJ.

Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. Learning a SAT solver from single-bit supervision. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJMC_iA5tm.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. In *International Conference on Machine Learning*, pp. 55906–55949. PMLR, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in neural information processing systems*, 36:3706–3731, 2023.

Michael A. Trick. Color dataset, 2002. URL https://mat.tepper.cmu.edu/COLOR02/.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Eric Wallace, Nicholas Tomlin, Albert Xu, Kevin Yang, Eshaan Pathak, Matthew Ginsberg, and Dan Klein. Automated crossword solving. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3073–3085, 2022.

Josh Wardle. Wordle. The New York Times, 2021. URL https://www.nytimes.com/games/wordle/index.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Zhutian Yang, Jiayuan Mao, Yilun Du, Jiajun Wu, Joshua B Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Compositional diffusion-based continuous constraint solvers. In *Conference on Robot Learning*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7): 2282–2312, 2005.

Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron C Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial problems with gflownets. *Advances in neural information processing systems*, 36:11952–11969, 2023a.

Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10188–10198, 2023b. doi: 10.1109/CVPR52729.2023.00982.

Yixin Zhang, Yunhao Luo, Utkarsh Aashu Mishra, Woo Chul Shin, Yongxin Chen, and Danfei Xu. Compositional visual planning via inference-time diffusion scaling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EEONns7ae4.

A Extended related work

Existing compositional diffusion methods differ in both how they define the composition and how they sample from it. Direct structured methods instantiate local modules on a factor graph and combine their scores or energies (Yang et al., 2023; Oarga & Du, 2025). When modules overlap, direct multiplication can double-count information associated with shared variables. DiffCollage addresses this using a Bethe-style correction to combine overlapping image factors, while Generative Factor Chaining applies a related overlap-aware composition to spatial and temporal factors for long-horizon planning (Zhang et al., 2023b; Mishra et al., 2024). These methods use a fixed aggregation rule at each denoising step. The Bethe approximation is closely connected to loopy belief propagation: fixed points of loopy BP correspond to stationary points of the Bethe free energy (Yedidia et al., 2005). Rather than using the resulting Bethe form as a fixed aggregation rule, *CompDiff-BP* explicitly runs sum-product belief propagation over the learned factor potentials. This iteratively updates the variable beliefs, allowing neighboring experts to influence one another through their shared variables.

Compositional Visual Planning also performs dynamic structured composition in clean-prediction space (Zhang et al., 2026). It represents long-horizon plans as a chain of overlapping diffusion factors, introduces explicit agreement potentials between neighboring Tweedie estimates, and derives specialized message-passing updates to enforce these constraints. In contrast, *CompDiff-BP* formulates reconciliation directly as posterior inference in a general constraint factor graph: each problem variable is represented once, the learned experts define factors over these shared variables, and standard sum-product loopy belief propagation is used to approximate the resulting marginals. This avoids introducing separate agreement variables or a task-specific message-passing procedure.

A complementary body of work improves inference for a chosen composition. Reduce, Reuse, Recycle applies annealed Langevin and Hamiltonian Monte Carlo to energy-based compositions (Du et al., 2023). Feynman–Kac Correctors use sequential Monte Carlo to propagate, reweight, and resample a population of candidate trajectories (Skreta et al., 2025). Parallel Energy Minimization similarly maintains a population to explore difficult composed energy landscapes (Oarga & Du, 2025). and Compositional Diffusion with Guided Search uses population-based search, iterative resampling, and likelihood-based pruning to avoid incompatible local modes (Mishra et al., 2026).

B Implementation Details

We retrain the local energy experts using the architectures, training procedure, and released implementation of Oarga & Du (2025). The resulting checkpoints are fixed during composition and shared across all compositional methods. Our contribution begins at inference time, where these learned energies are converted into categorical factor potentials and composed using ***CompDiff-BP***. Our code will be made available upon publication.

Table 5: Architectures of the local energy experts used in our experiments.

Expert	$\dim(u_t)$	Condition c	Blocks	h	w	$\dim(g_\theta)$
8-Queens	8	None	3	128	2	1
3-SAT	3	Clause signs, 3-D	3	128	2	1
Coloring	2×14	None	3	128	2	1
Crosswords	5×26	Clue embedding, 1536-D	3	1024	1	1

B.1 Local expert training

Following the analog representation of Chen et al. (2023), categorical local variables are represented by concatenated one-hot or binary vectors and corrupted by the Gaussian forward process. Let u_t denote a noised local assignment to variable x_t and c an optional condition, such as SAT literal signs or a crossword clue embedding. Each expert parameterizes a scalar energy as

$$E_\theta(u_t, t; c) = \|g_\theta([u_t; c], t)\|_2 , \tag{6}$$

where the c is omitted for unconditional experts.

All experts use the residual MLP architecture of Oarga & Du (2025). We retrain each expert using the model dimensions reported in Table 5. The experts are trained with a denoising diffusion objective that supervises the gradient of the energy function at each timestep. Given a clean local assignment u_0 and its Gaussian-corrupted counterpart u_t, the objective is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{u_0,t,\epsilon}\left[\|\epsilon - \sigma_t \nabla_{u_t} E_\theta(u_t, t; c)\|_2^2\right], \tag{7}$$

where σ_t is determined by the noise schedule. This directly supervises the energy gradient at each noise level without backpropagating through the full sequence of denoising steps.

For 8-Queens, 3-SAT, and graph coloring, training additionally uses a contrastive objective. Given a positive assignment u^+ and task-specific negative assignments $\{u_r^-\}$, all assignments are corrupted at the same timestep and ranked by energy:

$$\mathcal{L}_{\text{con}} = -\mathbb{E} \log \frac{\exp(-E_\theta(u_t^+, t; c))}{\exp(-E_\theta(u_t^+, t; c)) + \sum_r \exp(-E_\theta(u_{t,r}^-, t; c))}. \tag{8}$$

The total training loss is $\mathcal{L}_{\text{diff}} + 0.5\mathcal{L}_{\text{con}}$ for these three tasks, while the crossword expert uses only $\mathcal{L}_{\text{diff}}$. The SAT expert is subsequently refined for 10,000 epochs on complete formulas using the composed denoising objective of Oarga & Du (2025).

3-SAT. The SAT expert is trained on individual three-literal clauses. Training formulas contain between 10 and 20 variables and $4.258n$ clauses for n variables. For each clause, the satisfying

assignment of the corresponding formula provides a positive local example, while the unique assignment that violates the clause provides a negative example. The three literal signs are supplied as a conditioning vector. After local training, the SAT expert is refined for 10,000 epochs on complete formulas using the composed denoising objective of Oarga & Du (2025).

Graph coloring. The coloring expert is trained on pairs of colors from a vocabulary of 14 colors. Positive examples assign different colors to the two endpoints. Negative examples assign the same color to both endpoints or perturb an otherwise valid pair.

8-Queens. The 8-Queens expert is trained on individual rows represented as binary vectors of length 8. Positive examples are obtained from a valid 8×8 placement. Negative examples are constructed by removing a queen, adding an additional queen, or otherwise perturbing the vector.

Crosswords. The crossword expert is trained on individual clue–answer pairs. A five-letter answer is represented by five concatenated one-hot vectors of length 26. Gaussian noise is applied to this word representation. The clue is represented by a fixed 1536-dimensional embedding which is then concatenated with the word representation to form the input. Following Oarga & Du (2025), the embeddings are produced using `text-embedding-3-small`, with approximately 32,700 training pairs and 6,800 validation pairs from Crosswords QA (Wallace et al., 2022).

All experts are trained with AdamW at learning rate 10^{-4} using a linear noise schedule with $T = 100$. Training lengths and batch sizes are summarized in Table 6. Checkpoints are selected by validation loss.

Table 6: Training configurations for the local energy experts.

Expert	Training examples	Loss	Epochs	Batch	Noise steps
8-Queens	Lines from one 8×8 solution	Diffusion + contrastive	20,000	2,048	100
3-SAT	Clauses from 4,000 formulas	Diffusion + contrastive	20,000	1,024	100
3-SAT refinement	4,000 complete formulas	Diffusion	10,000	1,024	100
Coloring	Valid pairs from 14 colors	Diffusion + contrastive	50,000	1,024	100
Crosswords	32.7k clue–answer pairs	Diffusion	20,000	2,048	100

B.2 Task-specific factor construction

At inference time, the trained local experts are evaluated on clean local assignments to construct the factor potentials used by *CompDiff-BP*.

3-SAT. Each clause defines a factor over its three boolean variables, conditioned on three indicators specifying which literals are negated. We evaluate the refined SAT expert on all $2^3 = 8$ assignments to construct the factor table.

Graph coloring. Each graph edge defines a factor over the colors of its two endpoints. We evaluate the learned edge expert on all 14^2 ordered color pairs to build the factor table.

8-Queens. The same expert is applied to row, column, and diagonal factors. Shorter diagonals are zero-padded to length eight. For each factor, we enumerate the assignments on its unpadded scope, apply the same padding, and evaluate the learned energy.

Crosswords. Each across or down answer defines a clue-conditioned factor over its cells. At evaluation time, we use the stored clue embeddings rather than querying an embedding model. Because enumerating all 26^5 possible five-letter assignments is inefficient, we use the restricted construction described next.

B.2.1 Candidate-restricted crossword factors

For crosswords, each grid cell is a categorical variable over the alphabet and each across or down answer defines a factor. Directly enumerating all 26^5 assignments of a five-letter factor is inefficient,

so we restrict each factor to a fixed lexicon $\mathcal{A}_k$ of five-letter words defined by the set of valid guesses accepted by the game Wordle (Wardle, 2021). The lexicon determines the support of the factor, while the learned clue-conditioned energy determines the relative weight of each candidate. The lexicon is fixed before evaluation and is not constructed from the test answers.

For each clue c_k, we evaluate the learned expert on every word in $\mathcal{A}_k$ to construct the factor potential. Belief propagation then uses the same message updates as in Equation (4), with the factor support restricted to words in $\mathcal{A}_k$. Thus, factor-to-variable messages marginalize over candidate words consistent with each possible letter at the corresponding cell. The update is exact over the restricted lexicon without enumerating the full 26^5 state space.

B.3 Inference details

Table 7 reports the hyperparameters used for the main results. Here, S is the number of reverse-diffusion steps, R is the number of LBP rounds at each step, τ is the factor temperature, and P is the number of independently sampled trajectories.

We implement belief propagation in log space and damp the normalized factor-to-variable messages. Writing $\mu_{k\to i} = \log m_{k\to i}$, let $\widetilde{\mu}^{(r)}_{k\to i}$ denote the message produced by the standard sum-product update at round r. We update

$$\mu^{(r)}_{k\to i} = (1-\lambda)\mu^{(r-1)}_{k\to i} + \lambda\widetilde{\mu}^{(r)}_{k\to i}. \tag{9}$$

We use $\lambda = 0.5$ in all experiments.

Table 7: Inference settings used for the main results.

Task	S	R	τ	Damping	P
3-SAT UF20-91	100	3	1.0	0.5	1024
3-SAT UF50-218	100	25	1.0	0.5	1024
8-Queens	100	3	1.0	0.5	1024
Graph coloring	100	5	1.0	0.5	128
Mini Crosswords	100	2	0.1	0.5	1

B.3.1 Candidate selection

When $P > 1$, we run P independent reverse trajectories and select the final candidate with the lowest learned composed energy. Selection uses no oracle information, such as constraint violations, reference solutions, or the evaluation metric. For $P = 1$, including the main crossword experiment, the single endpoint is decoded directly.

C Benchmark and Evaluation Details

We follow the benchmark instances and evaluation protocols of Oarga & Du (2025).

The 3-SAT evaluation contains 100 SATLIB UF20–91 formulas with 20 variables and 91 clauses and 100 UF50–218 formulas with 50 variables and 218 clauses. The clause expert is trained on formulas with at most 20 variables, so UF50–218 evaluates composition of the same local expert on larger problem instances.

For 8-Queens, we evaluate 100 generated boards for the fixed 8×8 problem. Graph coloring uses five graphs from each of eight categories: small and large Erdős–Rényi, Holme–Kim, and regular-expander graphs, together with Paley and complete graphs. We additionally evaluate on 18 instances from the COLOR benchmark (Trick, 2002), with results reported in Section E. The local edge expert is trained over a fixed 14-color vocabulary. For graphs requiring fewer colors, evaluation restricts the expert to the corresponding color subset. The crossword evaluation uses 20 5×5 Mini Crosswords, each with five across and five down clues. The clue text is provided to the learned word expert, while the answer grid is used only for evaluation.

C.1 METRICS AND DECODING

For 3-SAT, a decoded boolean assignment is correct only if it satisfies every clause. We additionally report the fraction of satisfied clauses. For graph coloring, each node is decoded to its highest-valued color and we count conflicting edges without post-processing. Results are reported as the mean and standard deviation over the five graphs in each category.

For 8-Queens, we follow the greedy decoding procedure described by Oarga & Du (2025) for all compositional methods. We rank cells by their predicted value and consider the eight highest-scoring cells in descending order, retaining a queen only if it does not conflict with any previously retained queen. A decoded board is correct if all eight queens are retained, and *size* denotes the number of queens retained by the decoder.

For Mini Crosswords, each cell is decoded by argmax over the alphabet. Letter accuracy measures the fraction of correct cells, word accuracy requires all five letters of an answer to match, and game accuracy requires the complete puzzle to match.

D BASELINE DETAILS

D.1 NON-COMPOSITIONAL BASELINES

For the non-compositional baselines, we report the results from Oarga & Du (2025) and do not retrain or re-evaluate the methods.

For 3-SAT, we include GCN, a graph convolutional network baseline (Kipf & Welling, 2017). DGL is a graph-based solver with tree-search inference (Böther et al., 2022). DIFUSCO is a supervised diffusion solver for combinatorial optimization (Sun & Yang, 2023). Fast T2T is a diffusion-based solver that combines sampling with optional gradient search (Li et al., 2024). NeuroSAT is a recurrent neural SAT solver that iteratively refines variable embeddings (Selsam et al., 2019). NSNet is a neural SAT solver based on belief-propagation-style inference (Li & Si, 2022).

For 8-Queens, the problem is encoded as maximum independent set. LWD is a reinforcement-learning method that learns which nodes to defer (Ahn et al., 2020). GFlowNets use an unsupervised generative formulation for graph combinatorial problems (Zhang et al., 2023a). We also include DIFUSCO (Sun & Yang, 2023) and Fast T2T (Li et al., 2024).

For graph coloring, we include GCN (Kipf & Welling, 2017) and GAT (Veličković et al., 2018). GNN-GCP is a graph neural network designed specifically for graph coloring (Lemos et al., 2019). XLVIN performs reinforcement learning with a learned neural algorithmic reasoner (Deac et al., 2020).

For Mini Crosswords, we include the LLM prompting baselines reported by Yao et al. (2023). These are standard input–output prompting (IO), chain-of-thought prompting (CoT) (Wei et al., 2022), and tree-of-thought prompting (ToT) (Yao et al., 2023).

D.2 PARALLEL ENERGY MINIMIZATION

Parallel Energy Minimization (PEM) is the compositional inference method of Oarga & Du (2025). PEM maintains P candidate assignments throughout optimization. At timestep t, candidates are weighted according to

$$w_t^{(p)} = \frac{\exp\Big(-E_t^{\mathrm{PEM}}(x_t^{(p)})\Big)}{\sum_{p'=1}^{P} \exp\Big(-E_t^{\mathrm{PEM}}(x_t^{(p')})\Big)}, \tag{10}$$

where E_t^{PEM} denotes the composed noisy energy. Candidate assignments are resampled according to these weights, perturbed with scheduled Gaussian noise, and then updated using the gradient of the composed energy. Repeating this procedure across timesteps allows low-energy candidates to be retained while maintaining exploration of the composed energy landscape.

We rerun PEM using the released implementation with our retrained local checkpoints, the same evaluation instances, task decoders, and candidate budgets as ***CompDiff-BP***. For 3-SAT, our rerun

does not recover the performance reported by Oarga & Du (2025). We therefore report both results: *PEM (reported)* denotes the values from the original paper, while *PEM* denotes our run using the same retrained SAT checkpoint and evaluation setup as *CompDiff-BP*. For Mini Crosswords, we additionally evaluate PEM with $P = 1$ to provide a candidate-matched comparison with *CompDiff-BP*.

D.3 BETHE CORRECTION

We implement a noisy-space Bethe composition following DiffCollage (Zhang et al., 2023b). Let $\mathbf{u} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$ denote the full state, let $f^{(k)}$ denote the variables in factor k, and let d_i be the degree of variable $\mathbf{x}^{(i)}$. At each diffusion timestep, the Bethe approximation is

$$p_t^{\mathrm{Bethe}}(\mathbf{u}_t) \propto \frac{\prod_{k=1}^{K} p_{k,t}(f_t^{(k)})}{\prod_{i=1}^{N} p_{i,t}(\mathbf{x}_t^{(i)})^{d_i-1}}. \tag{11}$$

Accordingly, its score is

$$s_t^{\mathrm{Bethe}}(\mathbf{u}_t) = \sum_{k=1}^{K} R_k^\top s_{k,t}(f_t^{(k)}) + \sum_{i=1}^{N} (1-d_i) R_i^\top s_{i,t}(\mathbf{x}_t^{(i)}), \tag{12}$$

where R_k and R_i place the local predictions into the corresponding coordinates of the global state. This is the noisy-space composition used for the Bethe baseline.

Unlike DiffCollage, we do not train separate diffusion models for the singleton marginals. Instead, for each discrete variable we specify a categorical prior $\pi_i(c)$ and compute its noisy marginal analytically:

$$p_{i,t}(\mathbf{x}_t^{(i)}) = \sum_{c \in \mathcal{D}_i} \pi_i(c)\, q\Big(\mathbf{x}_t^{(i)} \mid \mathbf{x}_0^{(i)} = c\Big). \tag{13}$$

Its score, and equivalently its noise prediction, can therefore be evaluated in closed form. Since the learned experts predict noise rather than scores, we implement Equation (12) using the corresponding noise predictions with the same coefficients.

We use a uniform boolean prior for 3-SAT, a Bernoulli prior with probability $1/8$ for a queen, a uniform prior over the 14-color vocabulary, and empirical letter frequencies from the crossword training set. These priors are fixed before evaluation and do not use evaluation labels. The correction is recomputed at every reverse step, but no messages are exchanged between factors. We use the same local checkpoints, reverse-diffusion schedule, task decoder, and candidate budget as *CompDiff-BP*.

D.4 NO-DIFFUSION BASELINES

We implement two baselines that use the learned factors without a reverse-diffusion trajectory.

Loopy Belief Propagation removes the Gaussian unary terms from Equation (3) and performs inference on

$$p^{\mathrm{factor}}(x_0) \propto \prod_{k=1}^{K} \psi_{\theta_k}(x_0^{s_k}). \tag{14}$$

Equivalently, the unary log-potentials are set to zero. The variable-to-factor update and final belief become:

$$\begin{aligned} m_{i\to k}^{(r)}(x_0^{(i)}) &\propto \prod_{h \in \mathcal{N}(i)\setminus\{k\}} m_{h\to i}^{(r-1)}(x_0^{(i)}), \\ b_i(x_0^{(i)}) &\propto \prod_{k \in \mathcal{N}(i)} m_{k\to i}^{(R)}(x_0^{(i)}), \end{aligned} \tag{15}$$

while the factor-to-variable update is the second update in Equation (4). We run 50 BP rounds using the damping rule in Equation (9). For $P > 1$, we independently sample each variable from b_i to obtain candidate assignments and rank them using the learned composed factor energy. For the $P = 1$ Mini Crossword result, the marginal means are passed directly to the task decoder.

Gradient Descent minimizes the summed clean-time energy

$$E_0^{\text{sum}}(y) = \sum_{k=1}^{K} E_{\theta_k}(y^{s_k}, 0) \tag{16}$$

over a continuous assignment $y \in [0,1]^D$. Each restart is initialized as $y^{(p,0)} \sim \text{Unif}([0,1]^D)$. We perform 100 gradient-descent updates on the summed clean-time energy:

$$y^{(p,r+1)} = \text{clip}_{[0,1]}\left(y^{(p,r)} - \eta \nabla_y E_0^{\text{sum}}(y^{(p,r)})\right), \tag{17}$$

where $\text{clip}_{[0,1]}$ clips each coordinate to the interval $[0,1]$. We use the step size 0.001 for UF20–91 and 0.01 for UF50–218, 8-Queens, graph coloring, and Mini Crosswords. Final states are passed through the same task-specific decoders as the diffusion methods. When $P > 1$, candidates are selected using the learned composed energy.

E Additional graph coloring results

Table 8: Graph Coloring instances solved by compositional approaches. We report the percentage of instances with zero conflicting edges.

Method	Solved instances	Solved (%) ↑
LBP *(No Diffusion)*	$6/58$	10.3
GD *(No Diffusion)*	$1/58$	1.7
Bethe Correction	$1/58$	1.7
PEM	$0/58$	0.0
CompDiff-BP	$\mathbf{19/58}$	**32.8**

Table 9: Results on 18 COLOR benchmark instances (Trick, 2002). Entries are the number of conflicting edges before repair (↓). GCN, GAT, GNN-GCP results are as reported in Oarga & Du (2025). Compositional results are evaluated with $P = 128$ using the same learned edge expert. Best results are in bold.

					Non-compositional			Compositional				
Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	d	χ	GCN	GAT	GNN-GCP	LBP	GD	Bethe	PEM	*CompDiff-BP*
myciel3	11	20	0.36	4	17	10	6	1	**0**	**0**	1	**0**
myciel4	23	71	0.28	5	71	31	10	1	6	8	8	**0**
queen5_5	25	160	0.53	5	160	160	96	160	26	22	26	**2**
queen6_6	36	290	0.46	7	290	290	116	150	36	33	35	**8**
myciel5	47	236	0.22	6	171	97	42	13	27	21	17	**1**
queen7_7	49	476	0.40	7	476	476	216	476	59	75	59	**26**
queen8_8	64	728	0.36	9	728	728	272	705	75	95	79	**26**
1-Insertions_4	67	232	0.10	5	220	100	42	1	25	35	27	**0**
huck	74	301	0.11	11	216	234	172	149	16	27	18	**10**
jean	77	254	0.09	10	208	146	206	29	20	31	19	**6**
david	87	406	0.11	11	333	352	156	142	35	59	27	**10**
mug88_1	88	146	0.04	4	117	127	98	6	25	22	18	**0**
myciel6	95	755	0.17	7	755	755	100	8	105	109	71	**1**
queen8_12	96	1368	0.30	12	1368	1368	408	1337	130	242	93	**75**
games120	120	638	0.09	9	574	438	418	71	57	61	50	**12**
anna	138	493	0.05	11	260	392	110	11	34	53	27	**8**
2-Insertions_4	149	541	0.05	5	270	304	198	**1**	86	94	49	**1**
homer	556	1629	0.01	13	1625	1454	667	97	123	239	69	**31**

F RUNTIME ANALYSIS

Table 10 reports end-to-end wall-clock time per instance, including sampling, decoding, and candidate selection. Under the evaluated configurations, *CompDiff-BP* is between $2.1\times$ and $206.7\times$ faster than the released PEM implementation. SAT, 8-Queens, and graph coloring use the same candidate budgets across methods, while all Mini Crossword timings use $P = 1$. The *CompDiff-BP* timings use the configurations reported in the main results, as reported in Table 7. All timings were measured on an NVIDIA H100 GPU using a 10GB MIG partition.

Table 10: End-to-end wall-clock time in seconds per instance. SAT, 8-Queens, and graph coloring share candidate budgets across methods. All Mini Crossword configurations use $P = 1$. Values above one in the last column indicate that *CompDiff-BP* is faster than PEM.

Task	PEM	Bethe	*CompDiff-BP*	PEM / *CompDiff-BP*
SAT UF20	303.28	6.00	1.47	$206.7\times$
SAT UF50	724.93	13.71	14.43	$50.2\times$
8-Queens	41.52	10.13	19.79	$2.1\times$
Coloring	174.69	14.79	9.78	$17.9\times$
Crosswords	18.36	1.01	3.76	$4.9\times$

G FULL DIFFUSION-STEP AND BP-DEPTH ABLATION

Table 11 reports the complete sweep underlying the one-dimensional comparisons in Table 4. Within each task, all settings other than the number of reverse-diffusion steps S and BP rounds R are held fixed. Performance is not monotonic in either quantity, and the preferred combination of diffusion and message-passing depth varies across tasks.

Table 11: Ablation over the number of reverse-diffusion steps S and BP rounds R. For SAT and N-Queens, we report the number of solved instances out of 100 (↑). For graph coloring, we report total conflicts (↓). For Mini Crosswords, we report word accuracy in percent (↑).

Task	S	$R = 1$	$R = 2$	$R = 3$	$R = 5$	$R = 10$	$R = 25$
SAT UF20-91 ↑	20	74	89	92	92	92	92
	50	84	92	92	92	92	92
	100	87	92	92	92	92	92
	200	83	92	92	92	92	92
SAT UF50-218 ↑	20	0	7	33	56	69	72
	50	0	23	40	67	71	70
	100	2	20	47	60	69	71
	200	1	27	46	67	69	71
N-Queens ↑	20	1	28	100	100	100	100
	50	3	72	100	100	100	100
	100	12	84	100	100	100	100
	200	20	95	100	100	100	100
Coloring ↓	20	1044	1080	584	578	461	387
	50	836	910	518	450	434	398
	100	790	850	510	428	386	388
	200	801	830	466	434	390	399
Crosswords ↑	20	39.5	52.5	58.5	51.0	52.5	53.5
	50	44.0	60.0	59.0	56.5	53.5	55.5
	100	44.5	65.5	60.0	57.0	53.5	54.0
	200	46.0	58.5	57.0	56.5	53.5	55.5