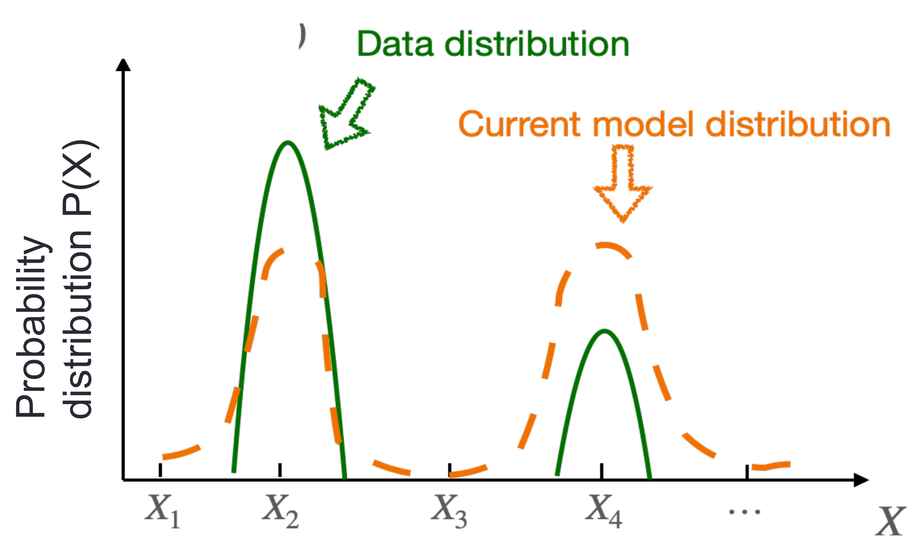
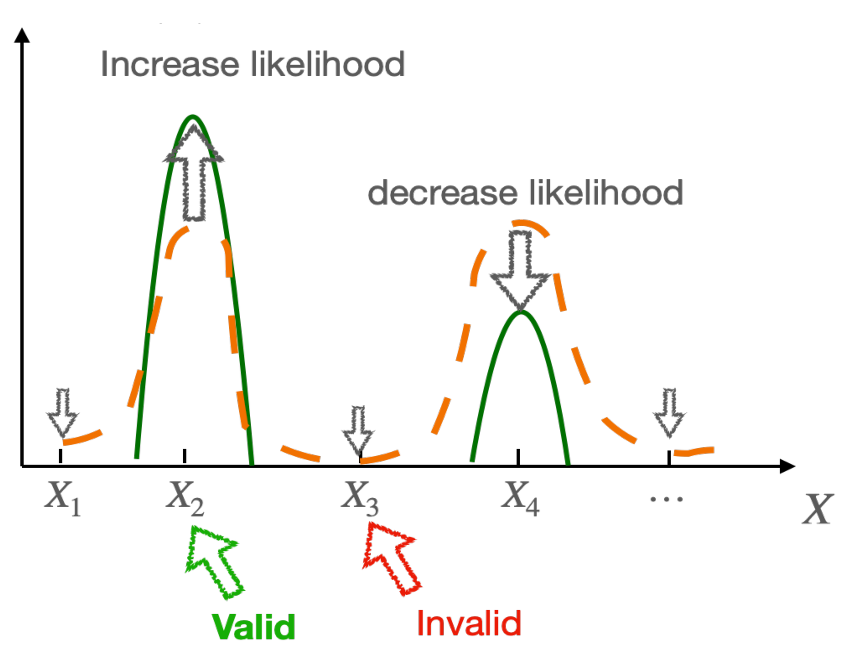
Learning Markov Random Fields for Combinatorial Structures via Sampling through Lovász Local Lemma

Nan Jiang[1*], Yi Gu[2*], Yexiang Xue[1]   [1]Purdue University, [2]Northwestern University

Problem: Generative Modeling For Combinatorial Structures

- Learning generative models over combinatorial structures involve matching the model distribution with the data distribution.





- **GAP**: Existing works generates **invalid** structures, resulting in learning to separate **valid** and **invalid** structures, but **NOT** learning the structural difference between **valid** structures **inside** and **outside** the dataset.
- **Our contribution**: A **fully-differentiable constraint reasoning** layer based on **Lovász Local Lemma** that samples **valid** structures for learning.

Background: Constrained Markov random fields (MRF)

- Discrete variables $X = \{X_i\}_{i=1}^n$, with $X \in \{0,1\}^n$
- Constraints $C = \{c_k\}_{k=1}^L$.

The probability distribution for constrained Markov random fields is:

$$P_\theta(X = x \mid C) = \frac{\exp\left(\phi_\theta(x)\right) C(x)}{Z_C(\theta)}$$

- $C(x)$ is the indicator function that evaluates to 1 if all constraints are satisfied.
- $\phi_\theta(x) : X \rightarrow \mathbb{R}$, is the potential function.
- $Z_C(\theta) = \sum_{x' \in X} \exp\left(\phi_\theta(x)\right) C(x)$, is the normalizing constant.

Learning task: minimize the negative log-likelihood over a dataset D:

$$-\frac{1}{|D|} \sum_{x^k \sim D}^{N} \log P_\theta(X = x^k \mid C)$$

The **gradient** of the negative log-likelihood is:

$$-\mathbb{E}_{x \sim D}\left(\nabla \phi_\theta(x)\right) + \mathbb{E}_{\tilde{x} \sim P_\theta(x|C)}\left(\nabla \phi_\theta(\tilde{x})\right).$$

Sample from dataset D Sample from constrained MRF

Our Contribution: **sample valid structures based on Lovász local lemma.**

Method: Sampling through Lovász Local Lemma

- Discrete variables $X_1, X_2, X_3 \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$, $P(X_i = x_i) = \frac{\exp(\theta_i x_i)}{\sum_{x_i' \in \{0,1\}} \exp(\theta_i x_i')}$.
- Constraints $C = (X_1 \vee X_2) \wedge (\neg X_1 \vee X_3)$.
- normalizing constant $Z_C(\theta) = \sum_{x' \in \{0,1\}^n} \exp(\phi_\theta(x'))C(x)$.

GOAL: Sample valid assignments from $P(X_1)P(X_2)P(X_3)$ subject to constraints C.

	X_1	X_2	X_3
initialize $X_i \sim P(X_i)$ →	1	0	0
Resample X_1, X_3 from $P(X_1), P(X_3)$ →	0	0	1
Resample X_1, X_2 from $P(X_1), P(X_2)$ →	0	1	1
All constraints are satisfied! →			

Fully Differentiable Neural Network-based Implementation

Given constraints $C = (X_1 \vee X_2) \wedge (\neg X_1 \vee X_3)$, construct

$$w = \begin{bmatrix} [1,0,0] & [0,1,0] \\ [-1,0,0] & [0,0,1] \end{bmatrix}, \quad b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix},$$

1. Initialize. $x_i = \mathbf{1}[u_i \geq P(X_i)]$
2. Extract violated constraints.
$$Z = W \otimes x + b$$
$$S_j = 1 - \max_{1 \leq k \leq K} Z_{jk}$$
3. Variables to be resamples,
$$A_i = \mathbf{1}\left[\sum_{j=1}^{L} S_j V_{ji} \geq 1\right]$$
4. Resample variables
$$x = (1 - A) \times x + A \times \mathbf{1}[u_i \geq P(X_i)]$$

Fully differentiable implementation

$x = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ → $X_1 = 0$, $X_2 = 0$, $X_3 = 1$

$S = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ → c_1 is violated

$A = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ → X_1 and X_2 will be resampled.

$x = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ → $X_1 = 0$, $X_2 = 1$, $X_3 = 1$

Theoretical Guarantees

Condition 1 "Extreme Condition": Constraints C is called ``Extreme" if for constraints $c_i, c_j \in C$, 1) Either their domain variables do not intersect. 2) Or no variable assignment violates c_i, c_j sharing variables.

Theorem (Probability distribution) Given random variables $X = \{X_i\}_{i=1}^n$, constraints $C = \{c_k\}_{k=1}^L$ that satisfy the extreme condition and the parameters of the constrained MRF in the single variable form θ. Upon termination, Algorithm outputs an assignment x randomly drawn from the constrained MRF distribution: $x \sim P_\theta(X = x \mid C)$

Theorem (Time complexity) Let $q_\varnothing$ be a non-zero probability of all the constraints are satisfied. Let q_{c_j} denote the probability that only constraint c_j is broken and the rest all hold. If $q_\varnothing \geq 0$, the total number of re-samples throughout is $\frac{1}{q_\varnothing} \sum_{k=1}^{L} q_{c_k}$.

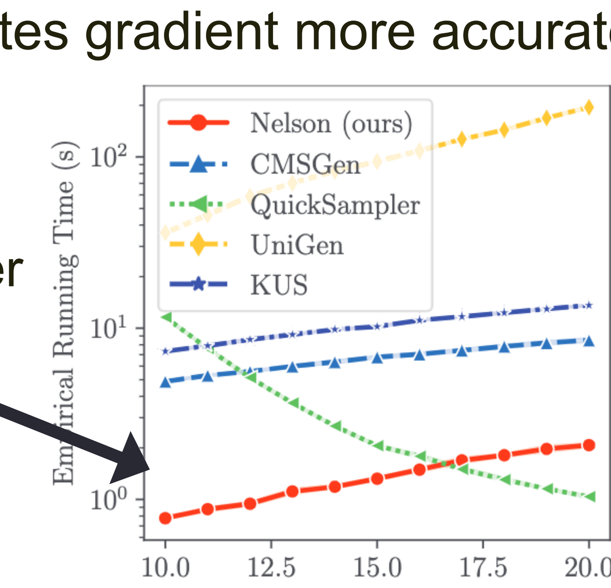
Experiment: Learn random K-SAT Solution with preference

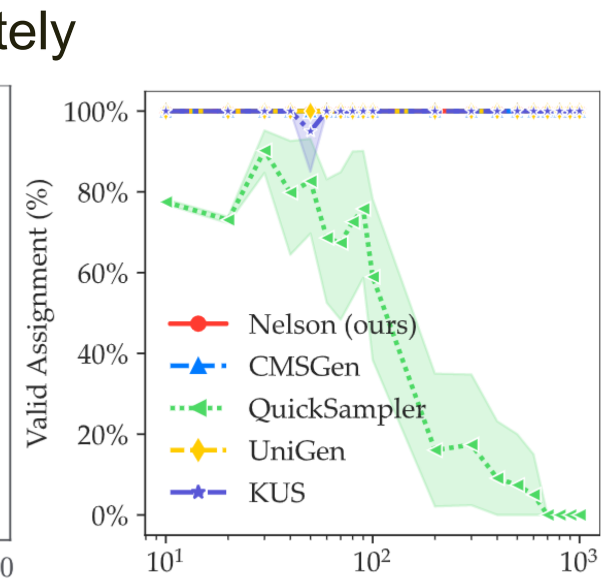
Our method is time efficient   Our method generates 100% valid structures

Problem size	Nelson	XOR	WAPS	WeightGen	CMSGen	KUS	QuickSampler	Unigen	Gibbs
	(a) Training time per iteration (Mins) (↓)								
10	**0.13**	26.30	1.75	0.64	0.22	0.72	0.40	0.66	0.86
[illegible]	**0.15**	134.50	3.04	T.O.	0.26	0.90	0.30	2.12	1.72
30	**0.19**	1102.95	6.62	T.O.	0.28	2.24	0.32	4.72	2.77
40	**0.23**	T.O.	33.70	T.O.	0.31	19.77	0.39	9.38	3.93
50	**0.24**	T.O.	909.18	T.O.	0.33	1532.22	0.37	13.29	5.27
500	**5.99**	T.O.	T.O.	T.O.	34.17	T.O.	T.O.	T.O.	221.83
1000	**34.01**	T.O.	T.O.	T.O.	177.39	T.O.	T.O.	T.O.	854.59
	(b) Validness of generated solutions (%) (↑)								
10 – 50	**100**	**100**	**100**	**100**	**100**	**100**	82.65	**100**	90.58
500	**100**	T.O.	T.O.	T.O.	**100**	T.O.	7.42	**100**	54.27
1000	**100**	T.O.	T.O.	T.O.	**100**	T.O.	0.00	**100**	33.91
	(c) Approximation error of $\nabla \log Z_{\mathcal{C}}(\theta)$ (↓)								
10	**0.10**	0.21	0.12	3.58	3.96	4.08	3.93	4.16	0.69
12	**0.14**	0.19	0.16	5.58	5.50	5.49	5.55	5.48	0.75
14	**0.15**	0.25	0.19	T.O.	6.55	6.24	7.79	6.34	1.30
[illegible]	0.16	0.25	**0.15**	T.O.	9.08	9.05	9.35	9.03	1.67
18	**0.18**	0.30	0.23	T.O.	10.44	10.30	11.73	10.20	1.90

Our method estimates gradient more accurately

Our method scales better with respect to problem size.





Code & Acknowledgement

This research was supported by NSF grants IIS-1850243, CCF-1918327. Our code implementation →