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

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

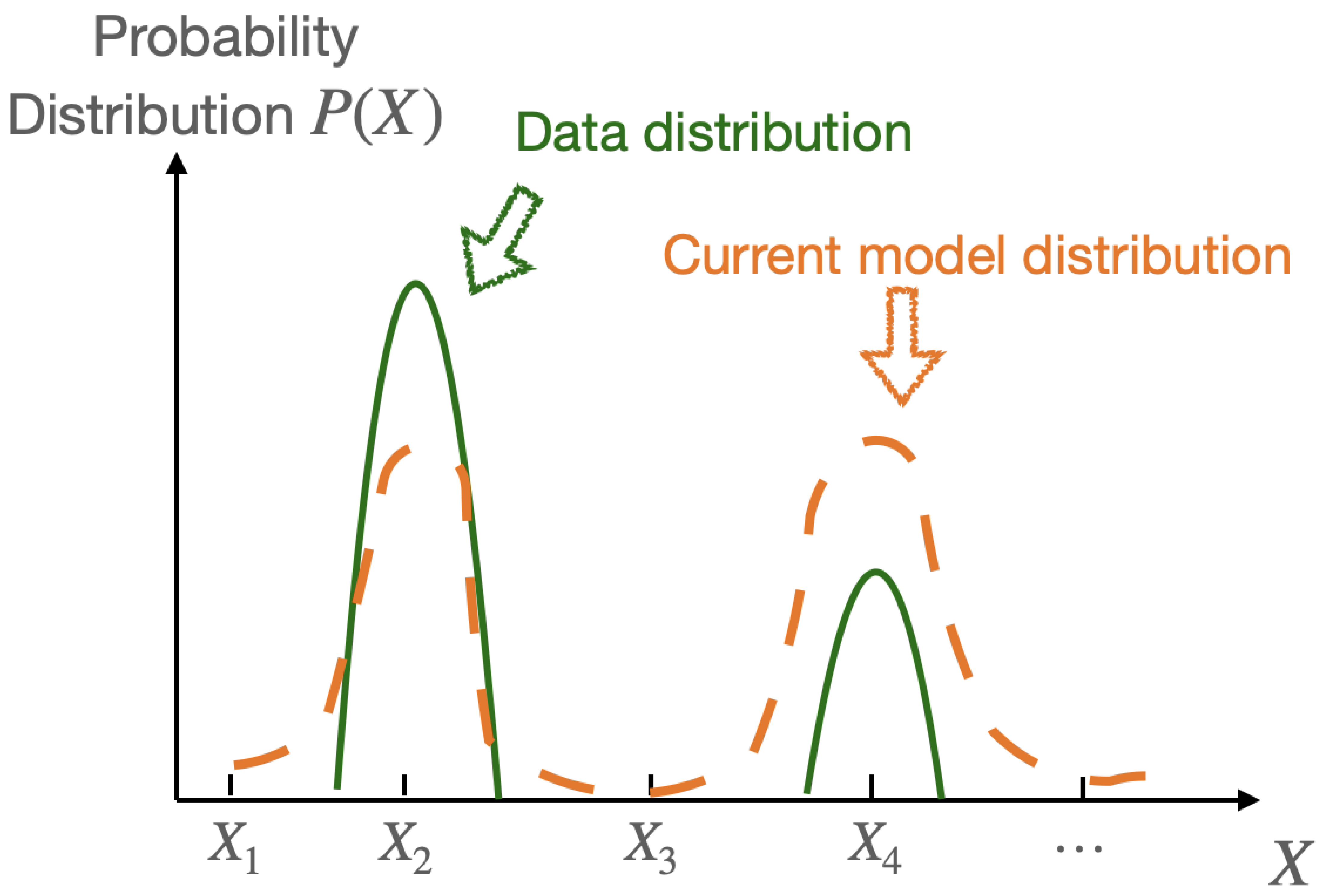
[1]Purdue University, [2]Northwestern University

{jiang631, yexiang}@purdue.edu, Yi.Gu@u.northwestern.edu

* The first two authors contribute equally.

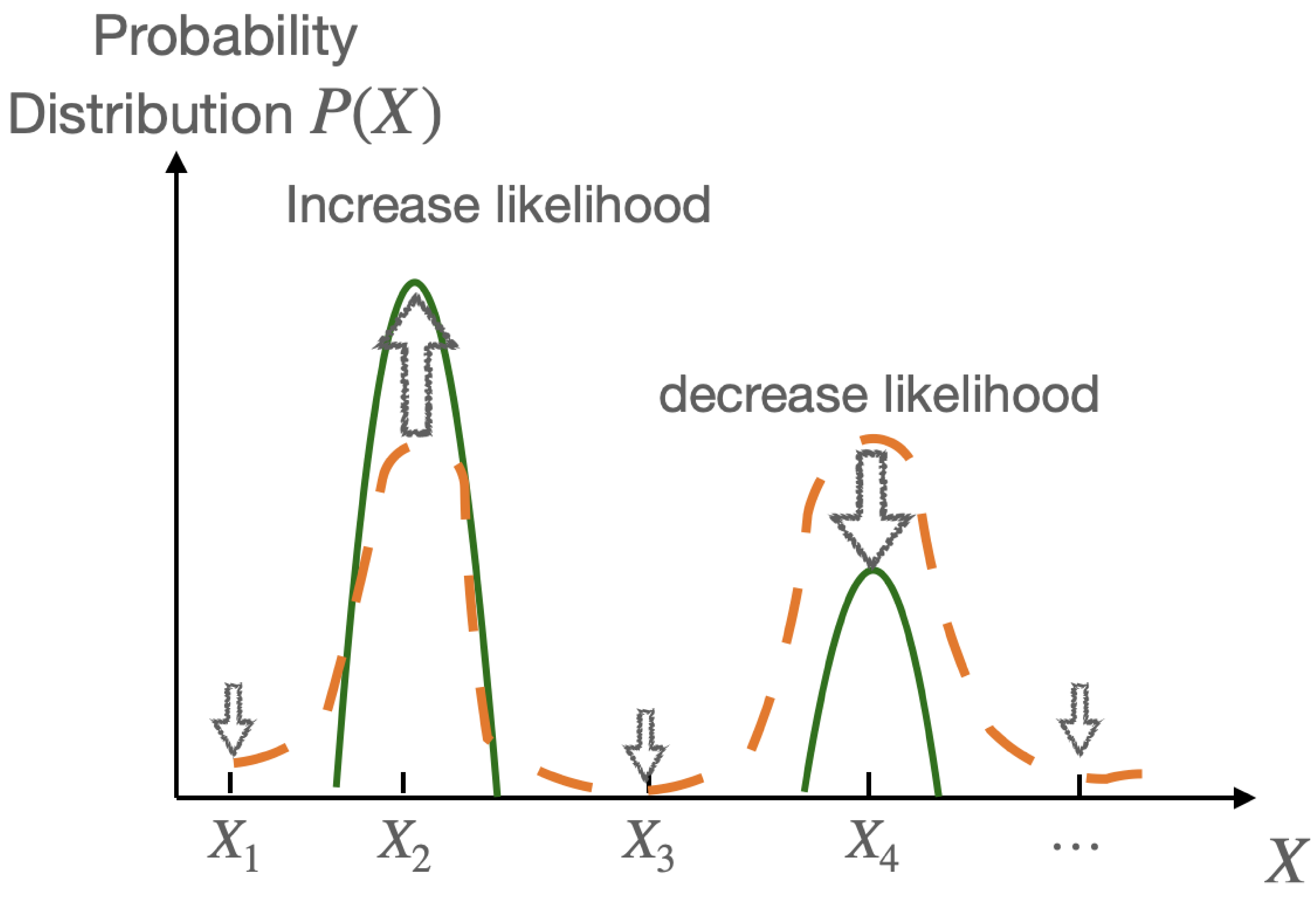
Learning to generate combinatorial structures

- Generative modeling received much success in AI
- Learning generative models over combinatorial structures involve matching the model distribution with the data distribution.



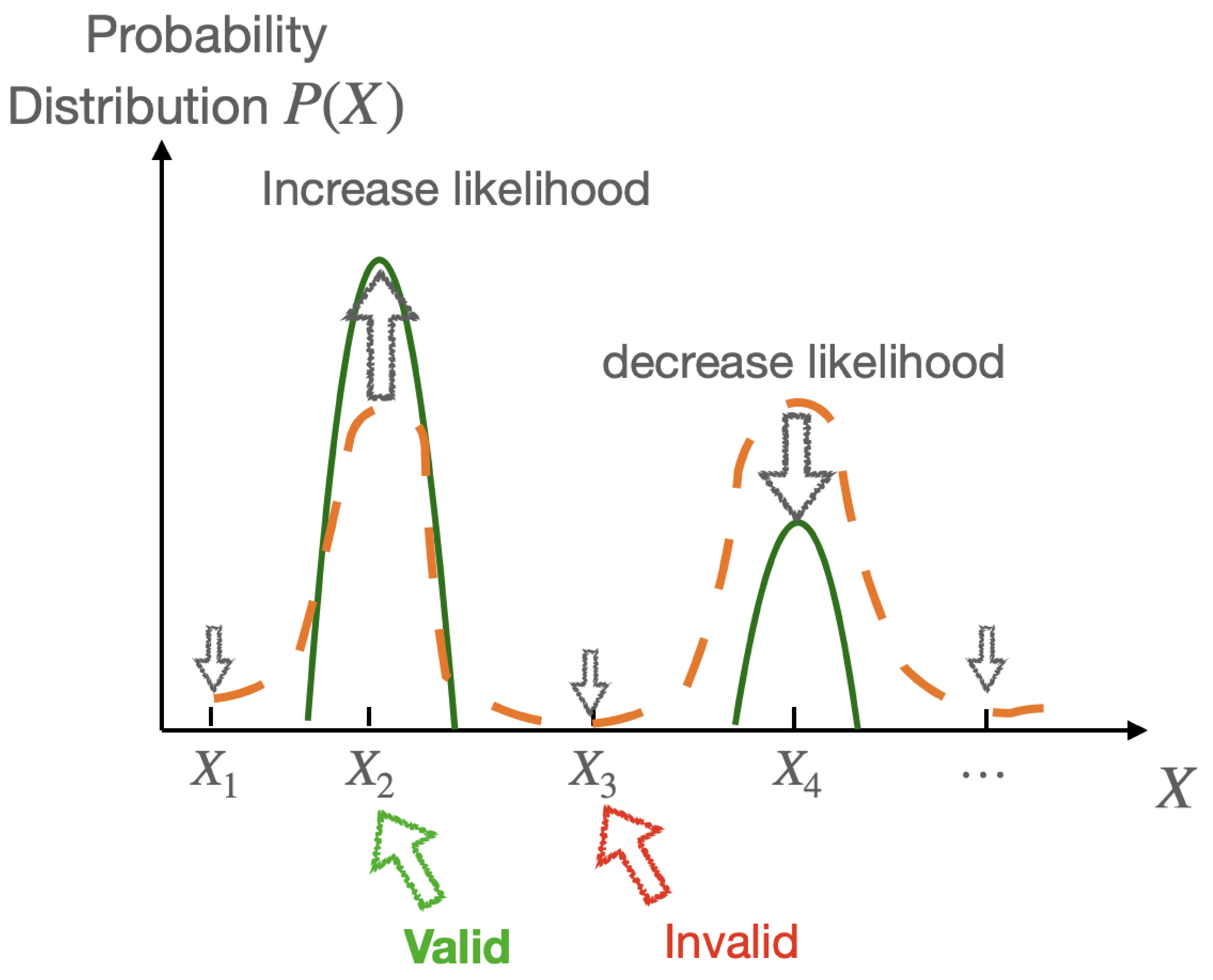
Learning to generate combinatorial structures

- Generative modeling received much success in AI
- Learning generative models over combinatorial structures involve matching the model distribution with the data distribution.
- **Increase** the likelihood of structures **in** the training data while **decreasing** those **not seen** (generated by reasoning algorithms)



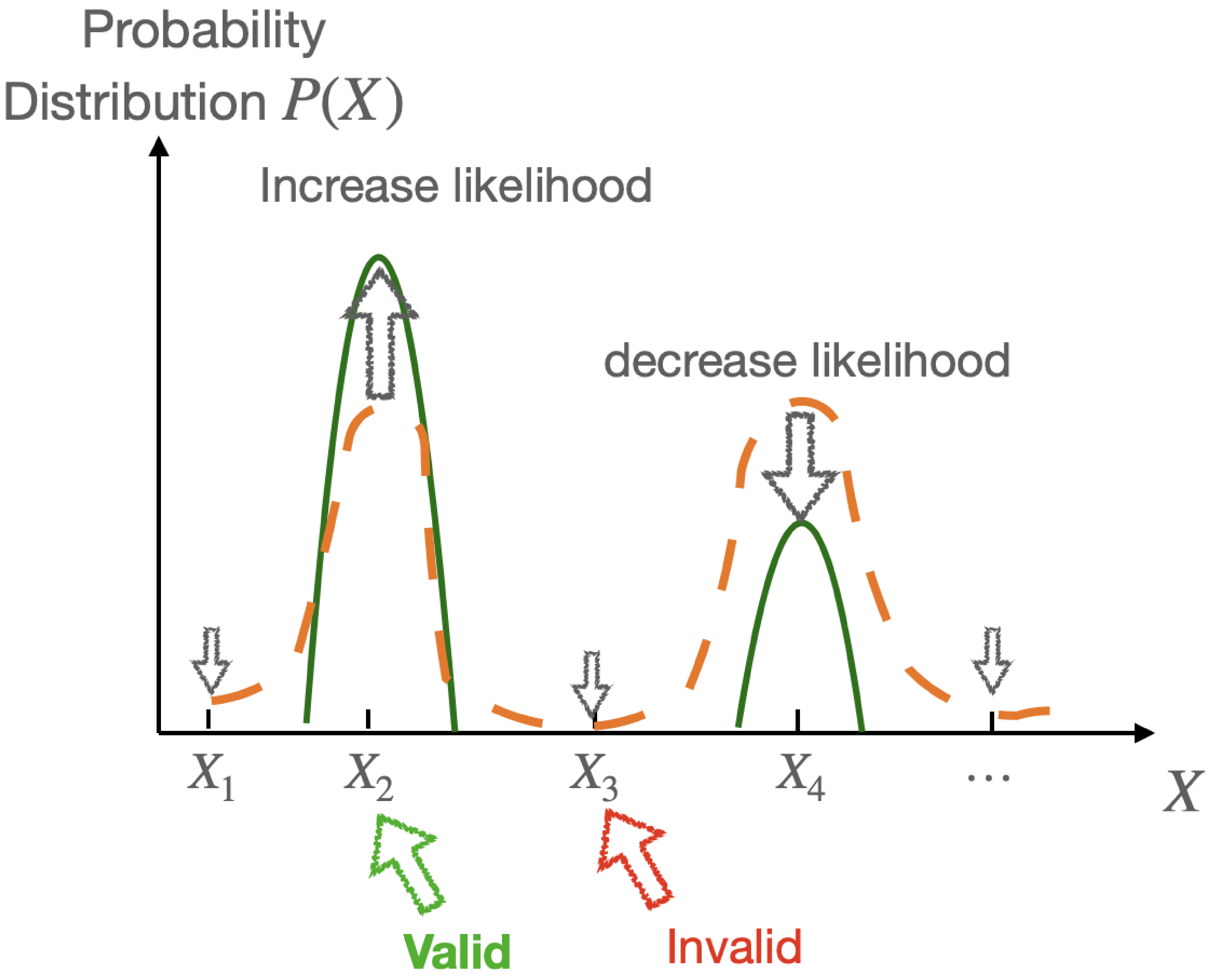
Learning to generate combinatorial structures

- Generative modeling received much success in AI
- Learning generative models over combinatorial structures involve matching the model distribution with the data distribution.
- **Increase** the likelihood of structures **in** the training data while **decreasing** those **not seen** (generated by reasoning algorithms)
- **GAP: Existing works generates invalid combinatorial structures; resulting in learning to separate valid and invalid structures, but NOT learning the structural difference between valid structures in the training dataset and those outside.**



Learning to generate combinatorial structures

- Generative modeling received much success in AI
- Learning generative models over combinatorial structures involve matching the model distribution with the data distribution.
- **Increase** the likelihood of structures **in** the training data while **decreasing** those **not seen** (generated by reasoning algorithms)
- **GAP: Existing works generates invalid combinatorial structures; resulting in learning to separate valid and invalid structures, but NOT learning the structural difference between valid structures in the training dataset and those outside.**
- **Our contribution: develop a fully-differentiable constraint reasoning layer based on Lovasz Local Lemma that samples valid structures for learning.**



Markov Random Fields for Combinatorial Structures

- Markov Random Fields (MRF)
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - Probability distribution:

$$P(X = x) = \frac{\exp(\phi_\theta\ (x))}{Z(\theta)}$$

Markov Random Fields for Combinatorial Structures

- Markov Random Fields (MRF)
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - Probability distribution:

$$P(X = x) = \frac{\exp(\phi_\theta\ (x))}{Z(\theta)}$$

  - Potential function: $\phi_\theta\ (x)$: $X \to R$.
  - Normalizing constant $Z(\theta) = \sum_{x' \in X} \exp(\phi_\theta\ (x'))$.

Markov Random Fields for Combinatorial Structures

- Markov Random Fields (MRF)
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - Probability distribution:

$$P(X = x) = \frac{\exp(\phi_\theta\ (x))}{Z(\theta)}$$

  - Potential function: $\phi_\theta\ (x)\colon X \to R$.
  - Normalizing constant $Z(\theta) = \sum_{x' \in X} \exp(\phi_\theta\ (x'))$.

- Constrained Markov Random Fields
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - **Constraints $\boldsymbol{C} = \{\boldsymbol{c_k}\}_{\boldsymbol{k=1}}^{\boldsymbol{K}}$.**

Markov Random Fields for Combinatorial Structures

- Markov Random Fields (MRF)
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - Probability distribution:

$$P(X = x) = \frac{\exp(\phi_\theta\ (x))}{Z(\theta)}$$

  - Potential function: $\phi_\theta\ (x): X \to R$.
  - Normalizing constant $Z(\theta) = \sum_{x' \in X} \exp(\phi_\theta\ (x'))$.

- Constrained Markov Random Fields
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - **Constraints $\boldsymbol{C} = \{\boldsymbol{c_k}\}_{\boldsymbol{k=1}}^{\boldsymbol{K}}$.**
  - Probability distribution:
  - $C(x)$: indicator function which evaluates to 1 if all constraints are satisfied.

Markov Random Fields for Combinatorial Structures

- Markov Random Fields (MRF)
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - Probability distribution:

$$P(X = x) = \frac{\exp(\phi_\theta\,(x))}{Z(\theta)}$$

  - Potential function: $\phi_\theta\,(x): X \rightarrow R$.
  - Normalizing constant $Z(\theta) = \sum_{x' \in X} \exp(\phi_\theta\,(x'))$.

- Constrained Markov Random Fields
  - Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X \in \{0,1\}^n$.
  - **Constraints $\boldsymbol{C = \{c_k\}_{k=1}^{K}}$.**
  - Probability distribution:

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

  - $C(x)$: indicator function which evaluates to 1 if all constraints are satisfied.
  - $Z_C\,(\theta) = \sum_{x' \in X} \exp\left(\phi_\theta\,(x')\right) C(x)$

Learning Constrained MRF

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

$$-\frac{1}{|D|}\sum_{x\sim D}\log P\big(X = x \mid \mathrm{C}\big)$$

Inference task: generates the structure which attains the highest likelihood under constraints.

$$x^* = \arg\max_{x'\in\{0,1\}^n} P\big(X = x' \mid \mathrm{C}\big)$$

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

$$-E_{x\sim D}\big(\nabla\phi_\theta(x)\big) + E_{\tilde{x}\sim P(X|C)}(\nabla\phi_\theta(\tilde{x}))$$

Sample from dataset D

Sample from constrained MRF

Learning Constrained MRF

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

$$-\frac{1}{|D|}\sum_{x\sim D}\log P\left(X=x\mid \mathrm{C}\right)$$

Inference task: generates the structure which attains the highest likelihood under constraints.

$$x^{*}=\arg\max_{x'\in\{0,1\}^n} P\left(X=x'\mid \mathrm{C}\right)$$

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

$$-E_{x\sim D}\left(\nabla\phi_\theta(x)\right)+E_{\tilde{x}\sim P(X|C)}(\nabla\phi_\theta(\tilde{x}))$$

Sample from dataset D

Our contribution: sample valid structures using Lovasz local lemma.

Sampling through Lovasz Local Lemma

Inputs:

- Discrete variables X_1, X_2, X_3, with $X_i \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$.
- Constraints (in Conjunctive Normal Form)

$$P(X_i) = \frac{\exp(\theta_i X_i)}{\sum_{X_i' \in \boldsymbol{X}_i} \exp(\theta_i X_i{}')}$$

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

Output:

Sample valid assignments.

Random initialize $X_i \sim P(X_i)$ →

X_1	X_2	X_3
1	0	0

Sampling through Lovasz Local Lemma

Inputs:

- Discrete variables X_1, X_2, X_3, with $X_i \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$.
- Constraints (in Conjunctive Normal Form)

$$P(X_i) = \frac{\exp(\theta_i X_i)}{\sum_{X_i' \in \boldsymbol{X}_i} \exp(\theta_i X_i{}')}$$

$$C = c_1 \land c_2,$$
$$c_1 = X_1 \lor X_2,$$
$$c_2 = \neg X_1 \lor X_3$$

Output:

Sample valid assignments.

c_2 is broken →

X_1	X_2	X_3
1	0	0

Sampling through Lovasz Local Lemma

Inputs:

- Discrete variables X_1, X_2, X_3, with $X_i \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$.
- Constraints (in Conjunctive Normal Form)

$$P(X_i) = \frac{\exp(\theta_i X_i)}{\sum_{X_i' \in \mathbf{X}_i} \exp(\theta_i X_i{}')}$$

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

Output:

Sample valid assignments.

Resample X_1, X_3 from $P(X_1), P(X_3)$ →

X_1	X_2	X_3
1	0	0

Sampling through Lovasz Local Lemma

Inputs:

- Discrete variables X_1, X_2, X_3, with $X_i \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$.
- Constraints (in Conjunctive Normal Form)

$$P(X_i) = \frac{\exp(\theta_i X_i)}{\sum_{X_i' \in \boldsymbol{X}_i} \exp(\theta_i X_i{}')}$$

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

Output:

Sample valid assignments.

	X_1	X_2	X_3
	1	0	0
c_1 is broken →	0	0	1

Sampling through Lovasz Local Lemma

Inputs:

- Discrete variables X_1, X_2, X_3, with $X_i \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$.
- Constraints (in Conjunctive Normal Form)

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

$$P(X_i) = \frac{\exp(\theta_i X_i)}{\sum_{X_i' \in \boldsymbol{X}_i} \exp(\theta_i X_i{}')}$$

Output:

Sample valid assignments.

Resample X_1, X_2 from $P(X_1), P(X_2)$ →

X_1	X_2	X_3
1	0	0
0	0	1

Sampling through Lovasz Local Lemma

Inputs:
- Discrete variables X_1, X_2, X_3, with $X_i \in \{0,1\}$.
- Marginal distributions $P(X_1), P(X_2), P(X_3)$.
- Constraints (in Conjunctive Normal Form)

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

$$P(X_i) = \frac{\exp(\theta_i X_i)}{\sum_{X_i' \in \boldsymbol{X}_i} \exp(\theta_i X_i{}')}$$

Output:
Sample valid assignments.

All constraints are satisfied! →

X_1	X_2	X_3
1	0	0
0	0	1
0	1	1

Implementing Sampling through Lovasz Local Lemma as several **Fully Differentiable** Neural Network Layers

- We convert constraints $C = \{c_k\}_{k=1}^{L}$ tensor W, matrix b.

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

X_1 is the 1st variable in c_1. **X_2 is the 2nd variable in c_1.**

$$W=\begin{bmatrix} [1 & 0 & 0] & [0 & 1 & 0] \\ [-1 & 0 & 0] & [0 & 0 & 1] \end{bmatrix}$$

$\neg X_1$ is the 1st variable in c_2. **X_3 is the 2nd variable in c_2.**

$$b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

the 1st variable in c_2 is negated.

- We also need a mapping matrix V

$$V=\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{array}{l} \rightarrow X_1 \text{ and } X_2 \text{ are in } c_1. \\ \rightarrow X_1 \text{ and } X_3 \text{ are in } c_2. \end{array}$$

Implementing Sampling through Lovasz Local Lemma as several **Fully Differentiable** Neural Network Layers

Input:
Discrete variables X_1, X_2, X_3,
with $X_1 \in \{0,1\}$.
Marginal distributions
$P(X_1), P(X_2), P(X_3)$.
Constraints

$C = c_1 \wedge c_2$,
$c_1 = X_1 \vee X_2$,
$c_2 = \neg X_1 \vee X_3$

$$\mathrm{W} = \begin{bmatrix} [1 & 0 & 0] & [0 & 1 & 0] \\ [-1 & 0 & 0] & [0 & 0 & 1] \end{bmatrix}$$

$$b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\mathrm{V} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

1. Initialize

$$x_i = 1[u_i > P(X_i)]$$

2. Extract violated constraints

$$Z = W \otimes x + b$$

$$S_j = 1 - \max_{1 \le k \le K} Z_{jk}$$

3. Variables to be resamples

$$A_i = 1\left[\sum_{j=1}^{L} S_j j V_{ji} \ge 1\right]$$

4. Resample variables

$$x = (1 - \mathrm{A}) * \mathrm{x} + \mathrm{A} * 1[u_i > P(X_i)]$$

Fully Differentiable

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

$$\mathrm{S} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow$$ c_1 **is violated**

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

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

Theoretical Guarantees

Theorem 1 (probability distribution) Given random variables $X = \{X_i\}_{i=1}^{n}$, constraints $C = \{c_k\}_{k=1}^{L}$ that satisfy the extreme condition (Condition 1) 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:

$$\mathrm{x} \sim P\big(X = x \mid C\big).$$

Lovasz local lemma guarantee we sample valid structures from the constrained MRF

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.

$$C = c_1 \wedge c_2,$$
$$c_1 = X_1 \vee X_2,$$
$$c_2 = \neg X_1 \vee X_3$$

Satisfy the extreme condition.

Theoretical Guarantees

Theorem 1 (probability distribution) Given random variables $X = \{X_i\}_{i=1}^{n}$, constraints $C = \{c_k\}_{k=1}^{L}$ that satisfy the extreme condition (Condition 1) 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:

$$\mathrm{x} \sim P\big(X = x \mid C\big).$$

Lovasz local lemma guarantee we sample valid structures from the constrained MRF

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 2 (time complexity) Let $\mathrm{q}_\emptyset$ 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 $\mathrm{q}_\emptyset \geq 0$, then the total number of re-sampling throughout the algorithm is $\frac{1}{\mathrm{q}_\emptyset}\sum_{j=1}^{L} \mathrm{q}_{c_j}$.

Time is proportional to the number of constraints.

Experimental Analysis: Learn Random K-SAT Solutions with Preference

Problem size	(a) Training time per iteration (Mins) (↓)								
	NELSON	XOR	WAPS	WeightGen	CMSGen	KUS	QuickSampler	Unigen	Gibbs
10	**0.13**	26.30	1.75	0.64	0.22	0.72	0.40	0.66	0.86
20	**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

Our method takes much less time

Experimental Analysis: Learn Random K-SAT Solutions with Preference

Problem size	(a) Training time per iteration (Mins) (↓)								
	NELSON	XOR	WAPS	WeightGen	CMSGen	KUS	QuickSampler	Unigen	Gibbs
10	**0.13**	26.30	1.75	0.64	0.22	0.72	0.40	0.66	0.86
20	**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

Our method generates 100% valid structures.

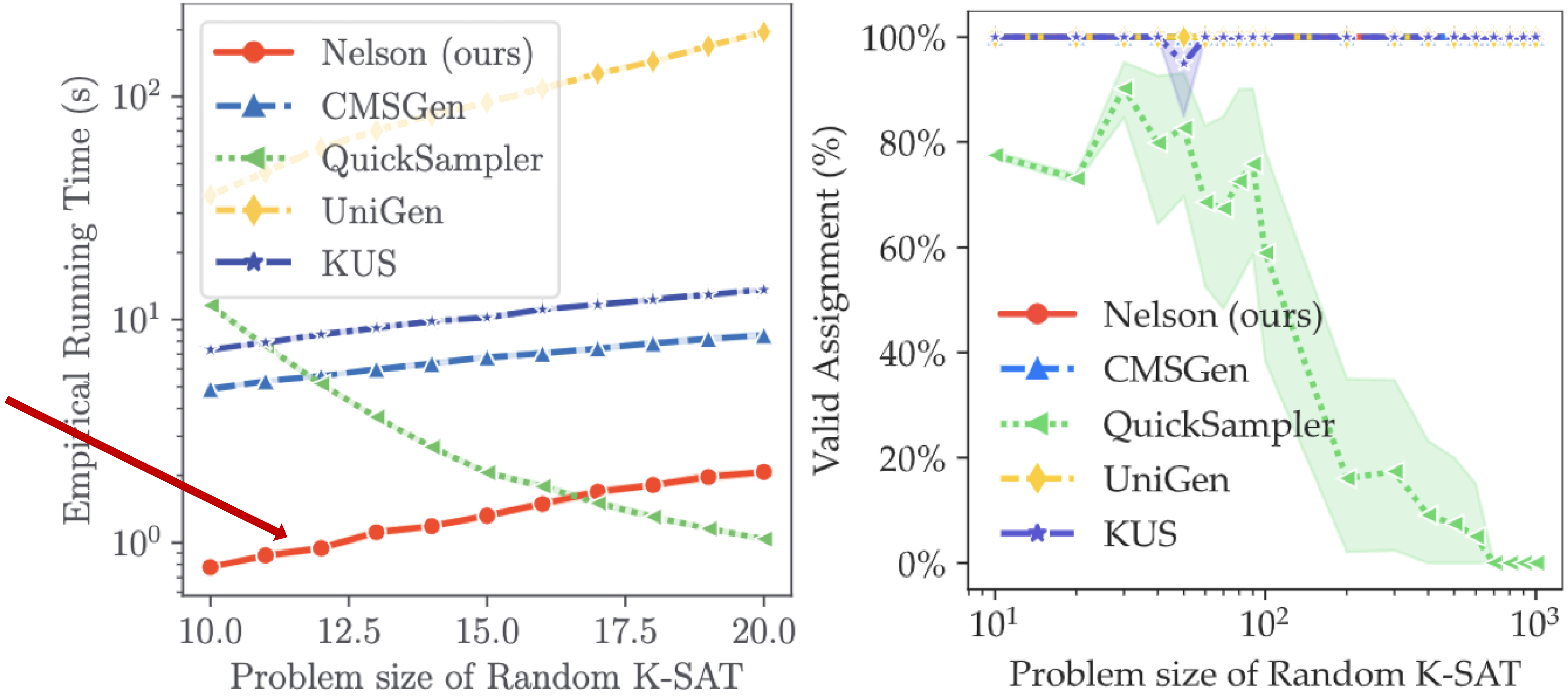
Experimental Analysis: Learn Random K-SAT Solutions with Preference

Problem size	(a) Training time per iteration (Mins) (↓)								
	NELSON	XOR	WAPS	WeightGen	CMSGen	KUS	QuickSampler	Unigen	Gibbs
10	**0.13**	26.30	1.75	0.64	0.22	0.72	0.40	0.66	0.86
20	**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 $\nabla \log Z_C(\theta)$ (↓)								
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
16	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 estimate gradient more accurately $-E_{x\sim D}\left(\nabla\phi_\theta(x)\right)+E_{\tilde{x}\sim P(X|C)}\left(\nabla\phi_\theta(\tilde{x})\right)$

Case studies

Our method scales better with respect to problem size.



Experimental Analysis: Learn Sink-Free Orientation in Undirected Graphs

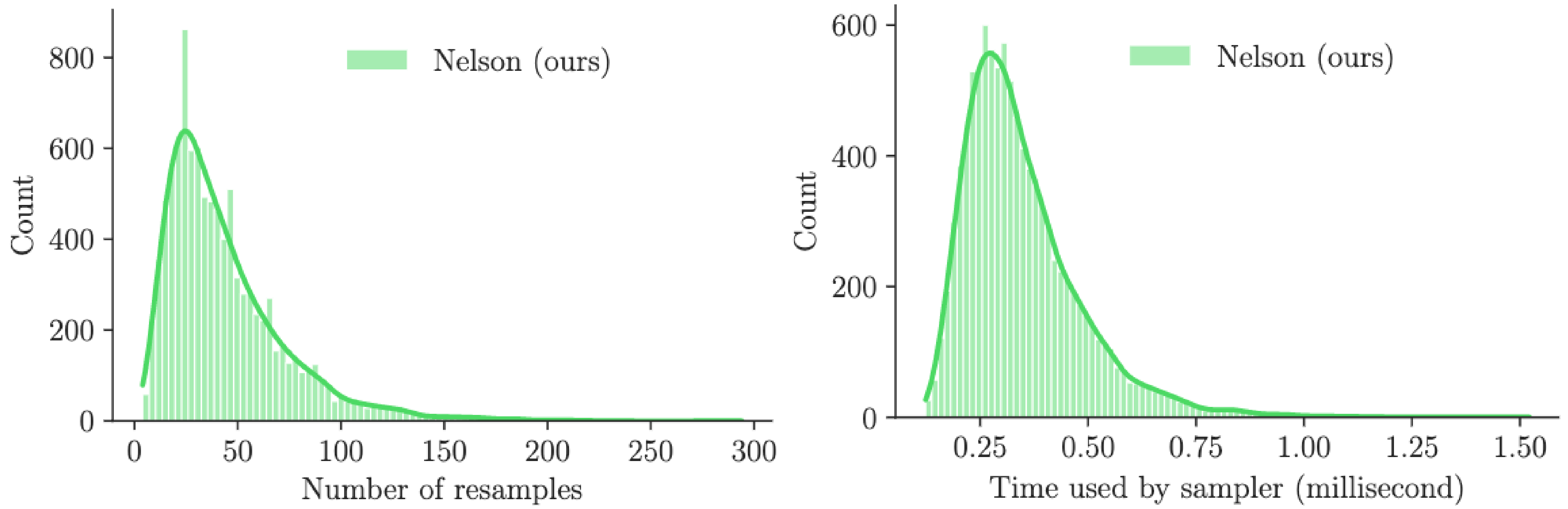
Problem size	(a) Training Time Per Epoch (Mins) (↓)		
	NELSON	Gibbs	CMSGen
10	**0.53**	9.85	0.69
20	**0.53**	80.12	1.93
30	**0.72**	256.38	3.65
40	**0.93**	777.01	5.99
50	**1.17**	T.O.	9.08
	(b) Validness of Orientations (%) (↑)		
7	**100**	50.16	**100**
8	**100**	64.63	**100**
9	**100**	47.20	**100**
10	**100**	62.60	**100**
11	**100**	84.95	**100**
	(c) Approximation Error of $\nabla \log Z_{\mathcal{C}}(\theta)$ (↓)		
5	**0.01**	0.09	0.21
7	**0.05**	0.08	2.37
9	**0.03**	0.11	2.37
11	**0.04**	0.17	8.62
13	**0.05**	0.28	11.27
	(d) MAP@10 (%) (↑)		
10	61.14	60.01	**64.56**
20	**55.26**	55.20	47.79
30	**100.00**	96.29	**100.00**
40	**40.01**	39.88	38.90
50	**46.12**	T.O.	42.11

Our method takes much less time

Our method generates 100% valid structures.

Our method estimate gradient more accurately

Experimental Analysis: Learn Vehicle Delivery Routes Problems



Our method is also efficient for other general NP-hard combinatorial problems

Conclusion

- Existing work focuses on separating invalid and valid instances rather than valid structures inside and outside of the training dataset.

- We propose a fully-differentiable constraint reasoning layer based on Lovasz Local Lemma that samples valid structures for learning.

Q & A?

https://github.com/jiangnanhugo/nelson-cd