



The 39th Annual AAAI Conference on Artificial Intelligence

FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, PENNSYLVANIA, USA

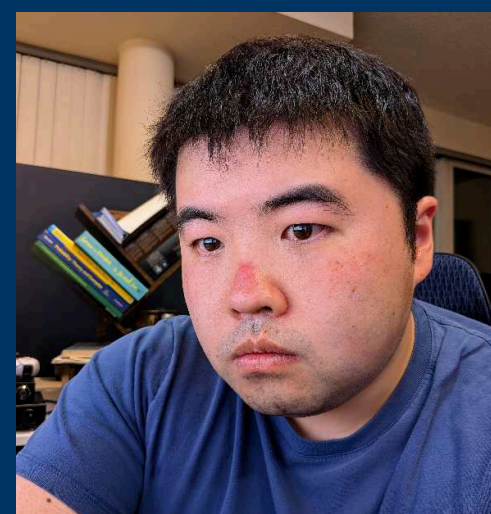


Mini Tutorial at CPMML: Learning Markov Random Fields for Combinatorial Structures via Sampling through Lovasz Local Lemma

Nan Jiang,



Yi Gu,

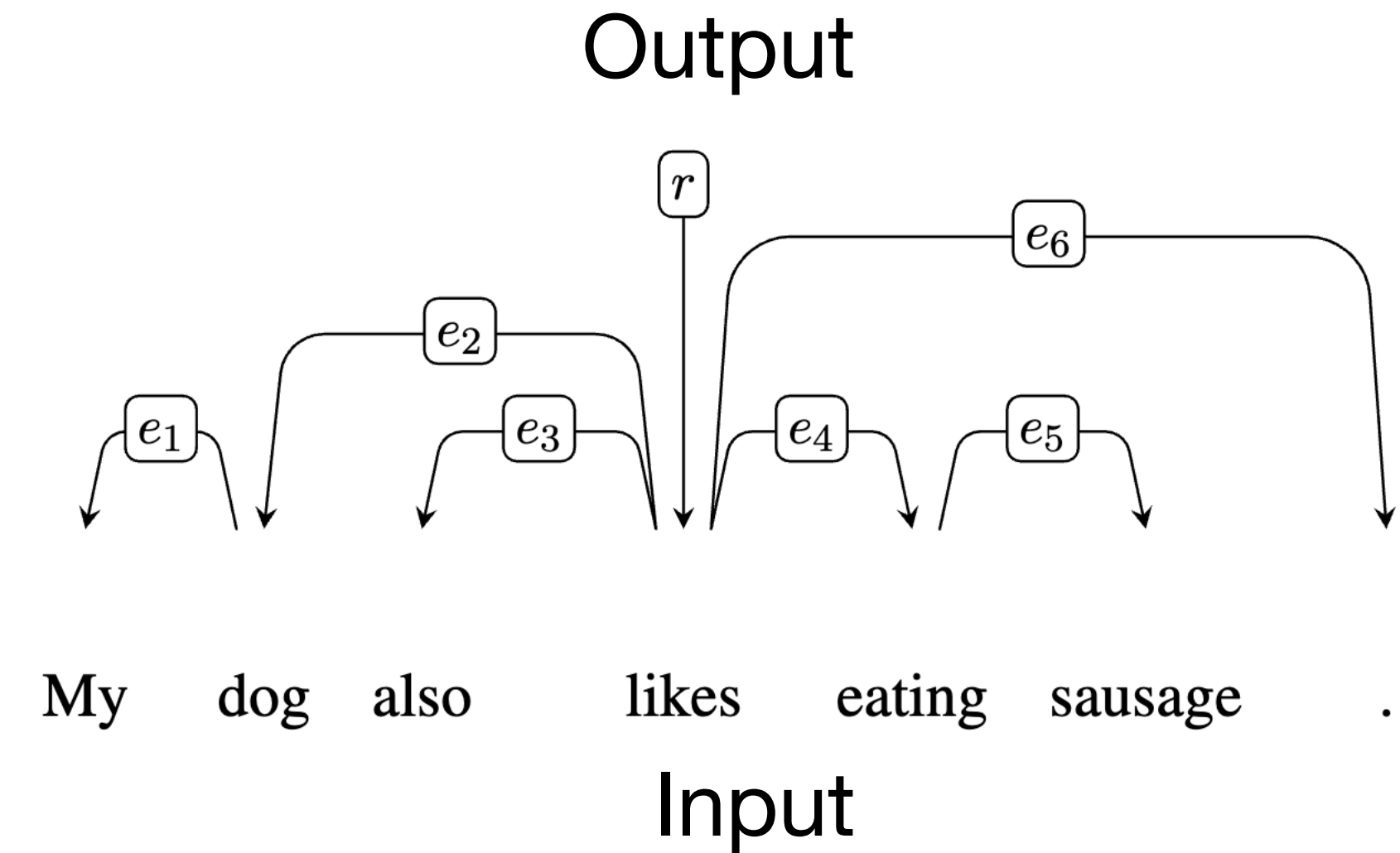


Yexiang Xue

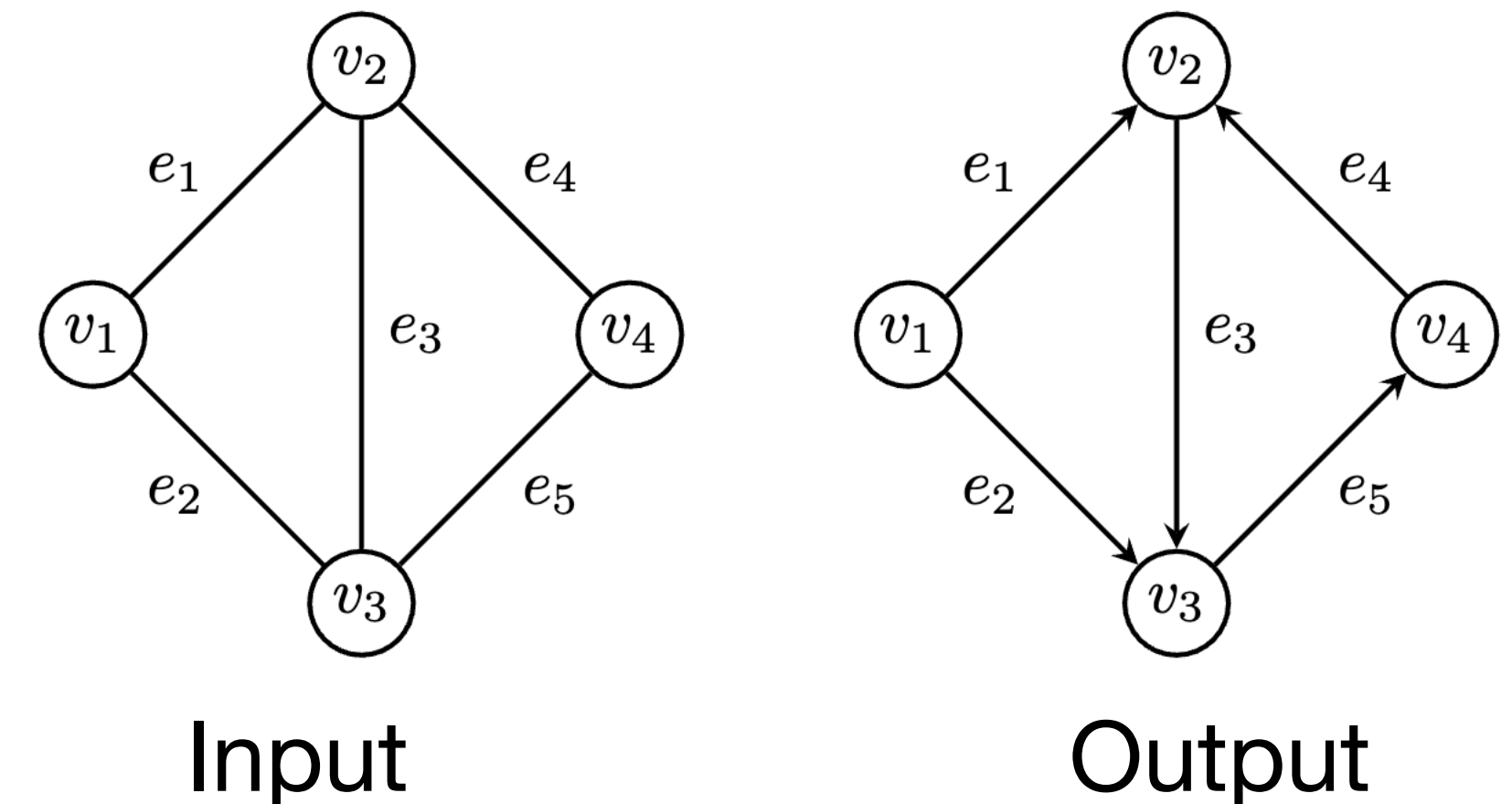


What are combinatorial structured problems?

- Dependency parsing in NLP
 - **Input:** sequence of words
 - **Output:** the grammar relationship between words.
 - **Constraint:** must be a tree.

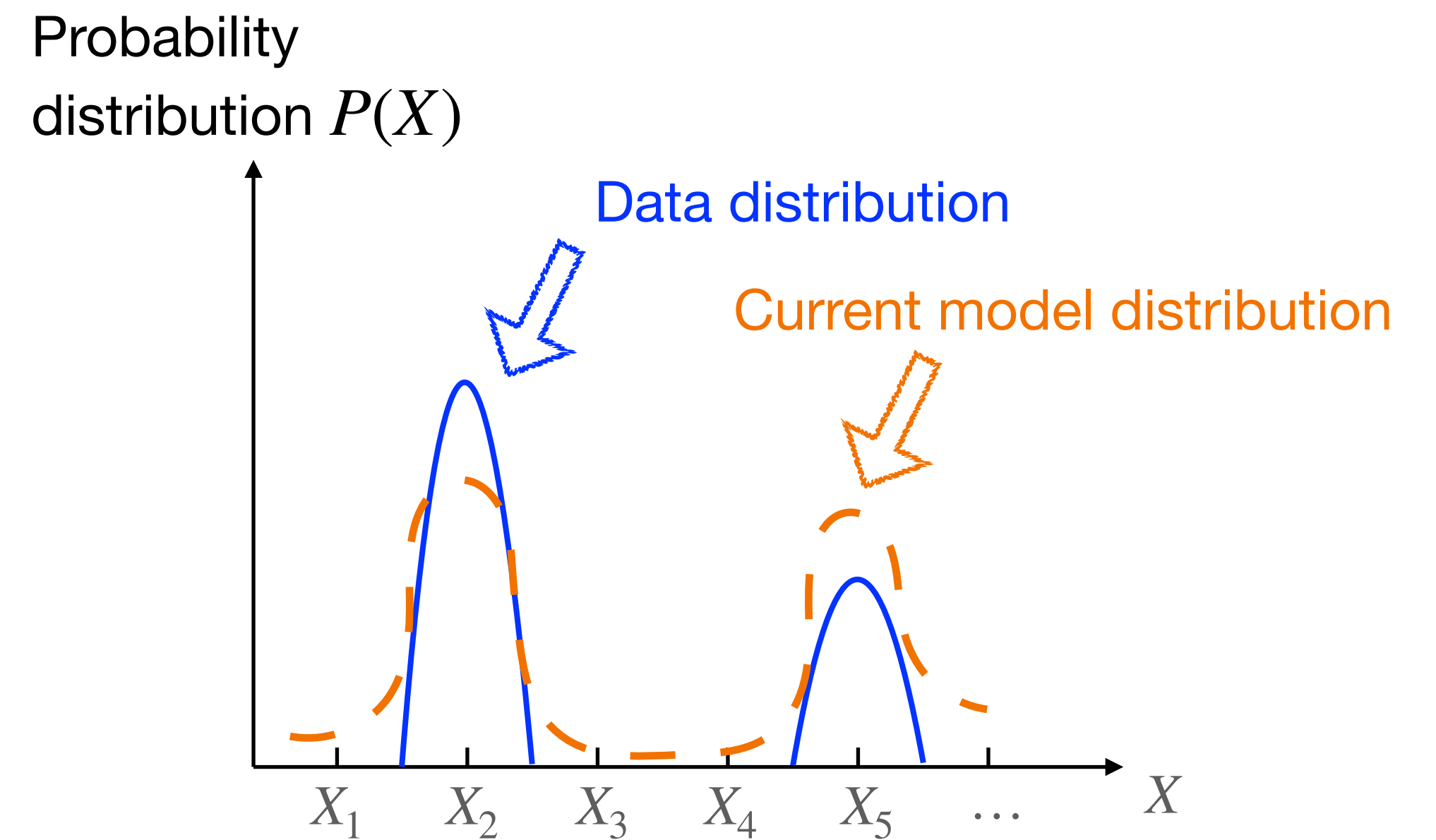


- Sink-free orientation of graph
 - **Input:** a connected undirected graph
 - **Output:** edge orientation.
 - **Constraint:** no sink



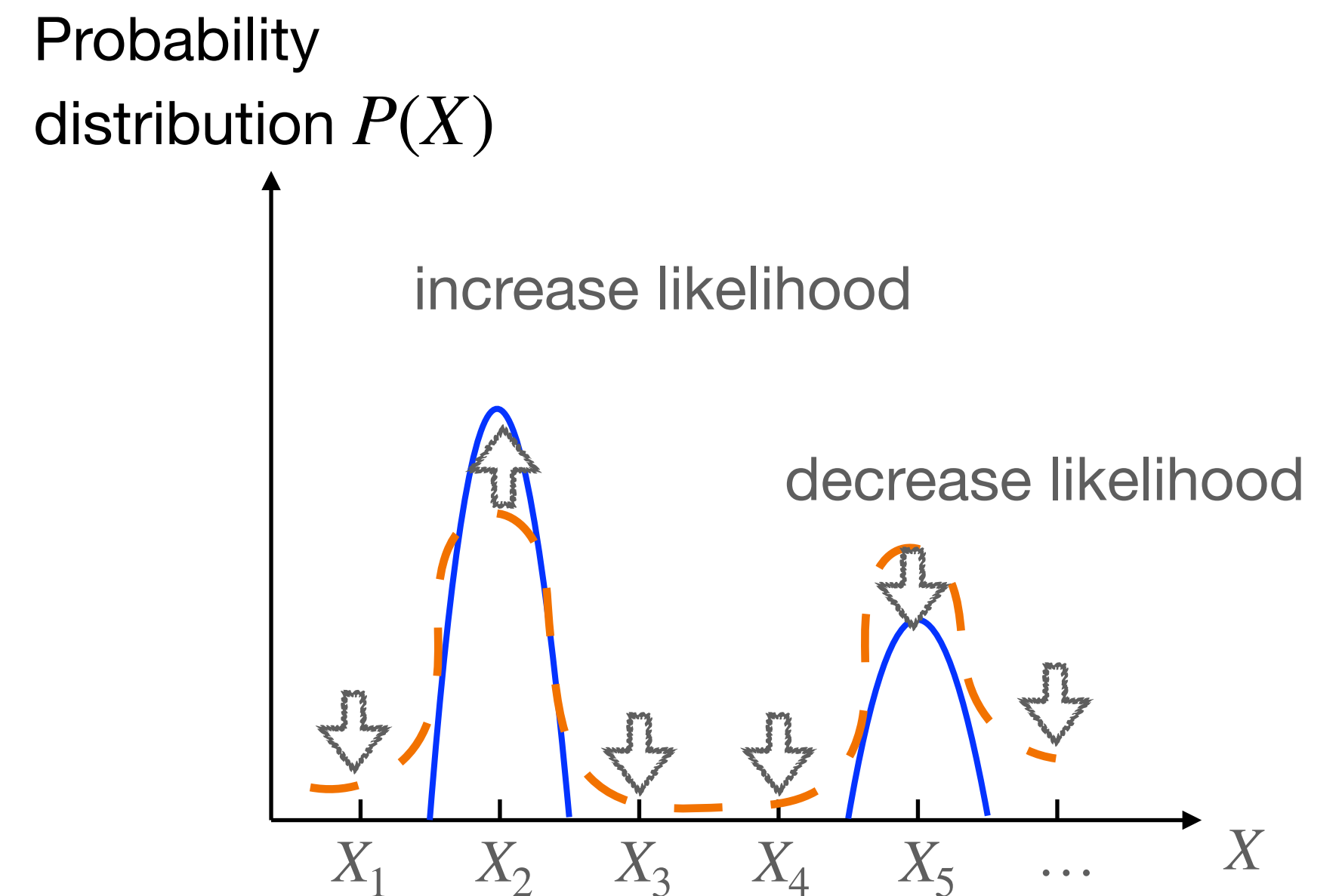
Learning to generate combinatorial structures

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



Learning to generate combinatorial structures

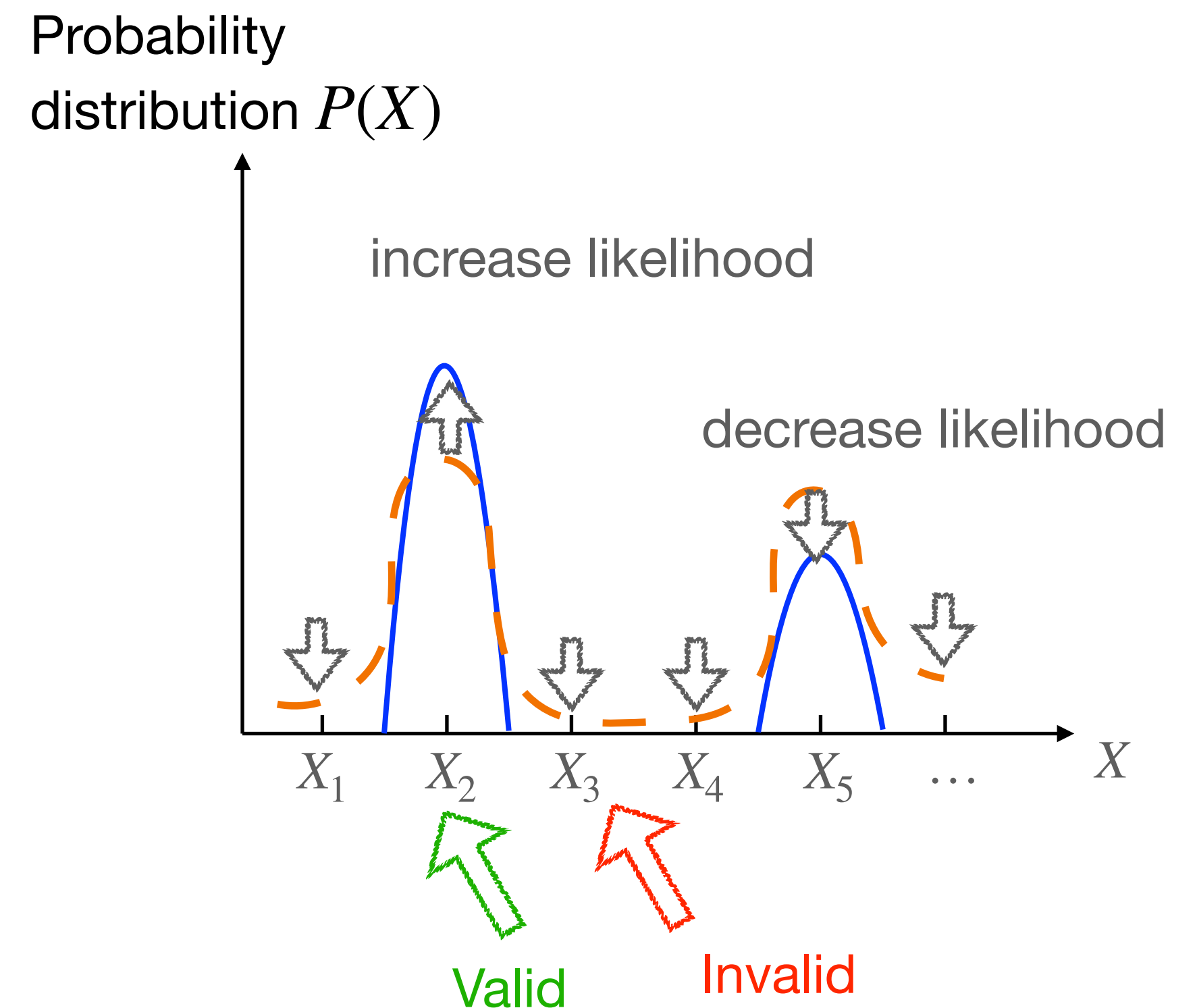
Increase the likelihood of structures **in** the training data while **decreasing** those **not seen** (generated by reasoning algorithms)



Learning to generate combinatorial structures

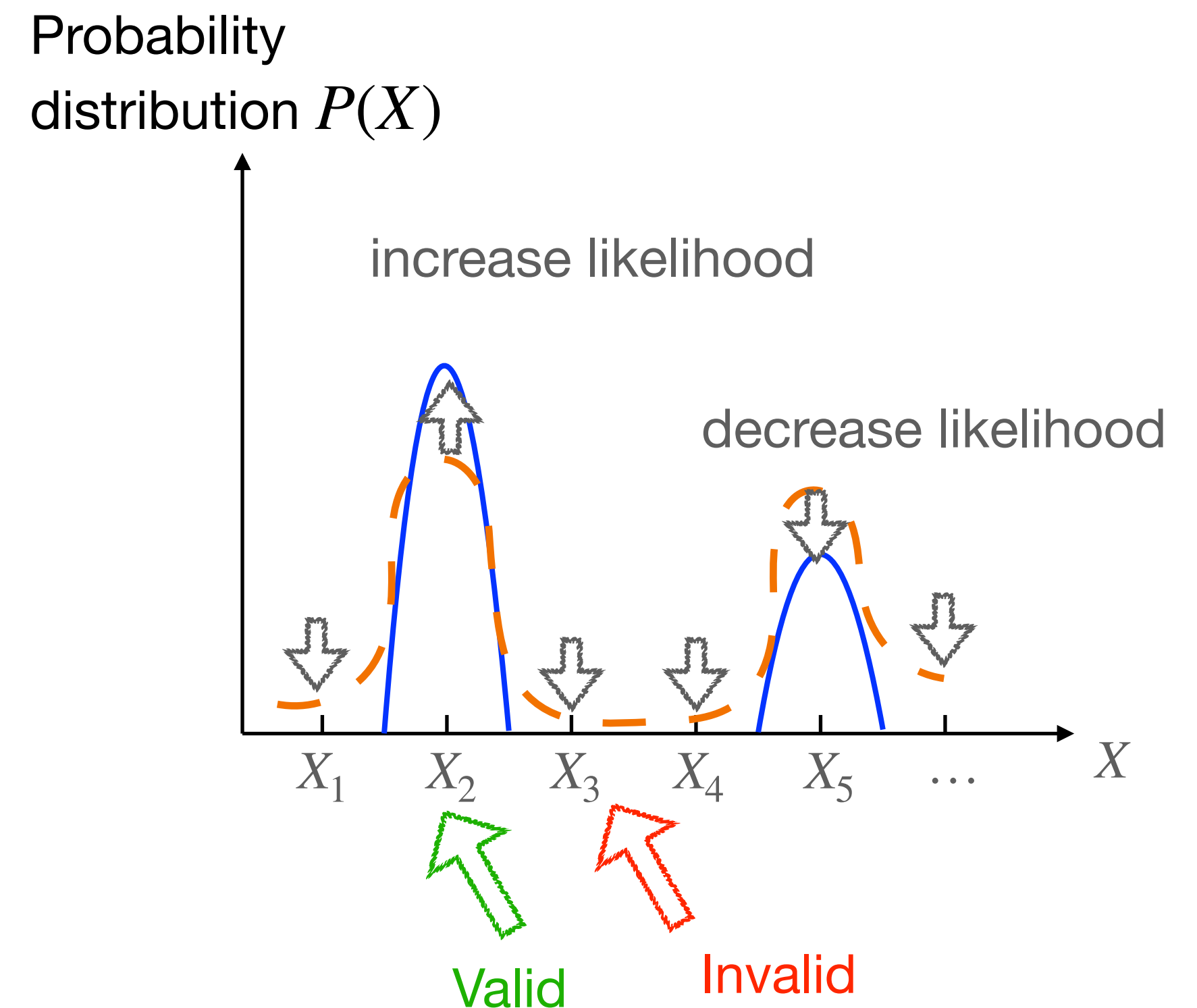
Gap: Existing works generates **invalid** combinatorial structures; resulting in learning to separate valid and invalid structures.

But **NOT** learning the structural difference between **valid** structures **in** the training dataset and those **outside**.



Learning to generate combinatorial structures

Our **contribution**: develop a **fully-differentiable** constraint reasoning layer based on Lovasz Local Lemma that samples **valid structures** for learning.



Constrained Markov Random Field (MRF)

Given:

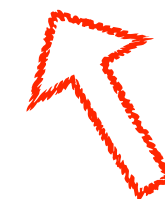
- Discrete variables $X = \{X_i\}_{i=1}^n$, with $X \in \{0,1\}^n$.
- Constraints $C = \{c_1, c_2, \dots\}$.

Probability distribution:

$$P_{\theta}(X = x | C) = \frac{\exp(f_{\theta}(x))}{Z_C(\theta)}$$

Indicator function $C(x) = 1$ when all constraints are satisfied.

Normalizing constant $Z_C(\theta)$ ensures the probability sums up to 1.



Main difficulty: Hard to compute

Learning Constrained MRF

Learning Objective: Minimize the negative log-likelihood over a training set D :

$$-\frac{1}{|D|} \sum_{x \in D} \log P_{\theta}(X = x | C)$$

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

$$-\mathbb{E}_{x \sim D} [\nabla_{\theta} \phi_{\theta}(x)] + \mathbb{E}_{x' \sim P(X|C)} [\nabla_{\theta} \phi_{\theta}(x')].$$

Sample from the dataset D

Sample from the constrained MRF

Learning Constrained MRF

Learning Objective: Minimize the negative log-likelihood over a training set:

$$-\frac{1}{|D|} \sum_{x \in D} \log P_{\theta}(X = x | C)$$

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

$$-\mathbb{E}_{x \sim D} [\nabla \phi_{\theta}(x)] + \mathbb{E}_{x' \sim P(X|C)} [\nabla \phi_{\theta}(x')].$$

Sample from the dataset D



Our contribution:
sample valid structures via the Lovasz local lemma.



Background on Lovasz Local Lemma

- In 1973, Erdos and Lovasz give a existence proof:

there exists a positive probability that none of a series of **bad** events occur, as long as these events are **mostly independent** from one another and are **not too likely individually**.

- In 2010, Moser and Tardos proposed Algorithmic-LLL:

which is an randomized algorithm to find solutions without breaching any bad events

The Gödel Prize 2020 - Laudation

The 2020 [Gödel Prize](#) is awarded to **Robin A. Moser** and **Gábor Tardos** for their algorithmic version of the Lovász Local Lemma in the paper:

"A constructive proof of the general Lovász Local Lemma," Journal of the ACM 57(2): 11:1-11:15 (2010).

- In 2019 (JACM), Heng Guo, Mark Jerrum et al. confirmed:

The probability distribution of Algorithmic-LLL and the required condition.

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$, where $P(X_i = x_i) = \frac{\exp(\theta_i x_i)}{\sum_{x'_i \in \{0,1\}} \exp(\theta_i x'_i)}$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C .

c_2 is violated 

X_1	X_2	X_3
1	0	0

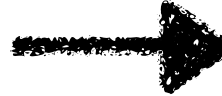
Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$, where $P(X_i = x_i) = \frac{\exp(\theta_i x_i)}{\sum_{x'_i \in \{0,1\}} \exp(\theta_i x'_i)}$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C .

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

X_1	X_2	X_3
1	0	0
0	0	1

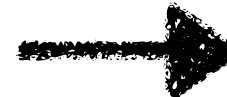
Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$, where $P(X_i = x_i) = \frac{\exp(\theta_i x_i)}{\sum_{x'_i \in \{0,1\}} \exp(\theta_i x'_i)}$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C .

c_1 is violated 

X_1	X_2	X_3
1	0	0
0	0	1

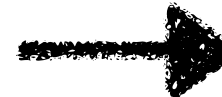
Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$, where $P(X_i = x_i) = \frac{\exp(\theta_i x_i)}{\sum_{x'_i \in \{0,1\}} \exp(\theta_i x'_i)}$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C .

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

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

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$, where $P(X_i = x_i) = \frac{\exp(\theta_i x_i)}{\sum_{x'_i \in \{0,1\}} \exp(\theta_i x'_i)}$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C .

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

All constraints are satisfied! 

Our contribution: we formulate a **fully-differentiable and efficient** neural network modules that simulates sampling through Lovasz Local Lemma.

Guarantee on constraint satisfaction

[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: $x \sim P_\theta(X = x \mid C)$.



The sampler exactly draw valid structures from the constrained MRF model.

[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.

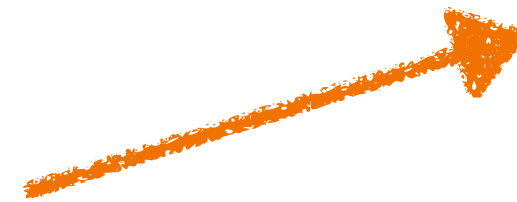
$$\text{Constraints: } C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$$

satisfy the extreme the condition

Guarantee on time efficiency

Theorem (Time complexity) Let $q_{\emptyset}$ 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

$q_{\emptyset} \geq 0$, the total number of re-samples throughout is $\frac{1}{q_{\emptyset}} \sum_{k=1}^L q_{c_k}$.



Time is proportional to the number of constraints.

[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.

Experimental Analysis: constraint satisfaction guarantee

Problem size	(a) Training Time Per Epoch (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 CNF Assignments (%) (↑)									
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% of valid structures

Experimental Analysis: more accurate gradient estimates

Problem size	(a) Training Time Per Epoch (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 CNF Assignments (%) (↑)									
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 Gradient (↓)									
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 estimates gradient more accurately

Experiments: converge better

	(a) log-likelihood ($\uparrow$)			
Problem size	NELSON	Gibbs	CMSGen	Quicksampler WeightGen,KUS XOR, WAPS
100	-49.16	- 36.36	-60.12	T.O.
300	- 52.61	-53.11	-128.39	
500	- 196.47	-197.21	-272.49	
700	- 238.60	-238.75	-389.44	
1000	- 294.22	-296.33	-532.85	
	(b) MAP@10 (%) ($\uparrow$)			
100	82.13	83.32	86.34	T.O.
300	66.37	64.42	64.50	
500	90.03	73.14	70.67	
700	69.74	69.74	48.10	
1000	91.70	77.56	78.72	

Our method converges to better model with smaller log-likelihood values

Experiments: 2. Learn Sink-Free Orientation in Undirected Graph

Time efficient	→	(a) Training Time Per Epoch (Mins) (↓)		
		Problem size	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
generates 100% of valid structures	→	(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
estimates gradient more accurately	→	(c) Approximation Error of $\nabla \log Z_c(\theta)$ (↓)		
		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
Our method converges to better model with higher learning-based metrics.	→	(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

Reference

1. Nan Jiang, Yi Gu, Yexiang Xue. Learning Markov Random Fields for Combinatorial Structures via Sampling through Lovasz Local Lemma. In AAI, 2023.
2. Fan Ding, Jianzhu Ma, Jinbo Xu, Yexiang Xue. XOR-CD: Linearly Convergent Constrained Structure Generation. In ICML, 2021.
3. Miguel A. Carreira-Perpinan and Geoffrey E. Hinton. On Contrastive Divergence Learning. AISTATS, 2005.
4. Heng Guo and Mark Jerrum and Jingcheng Liu. Uniform Sampling Through the Lovasz Local Lemma. In Journal of ACM, 2019.
- 6.