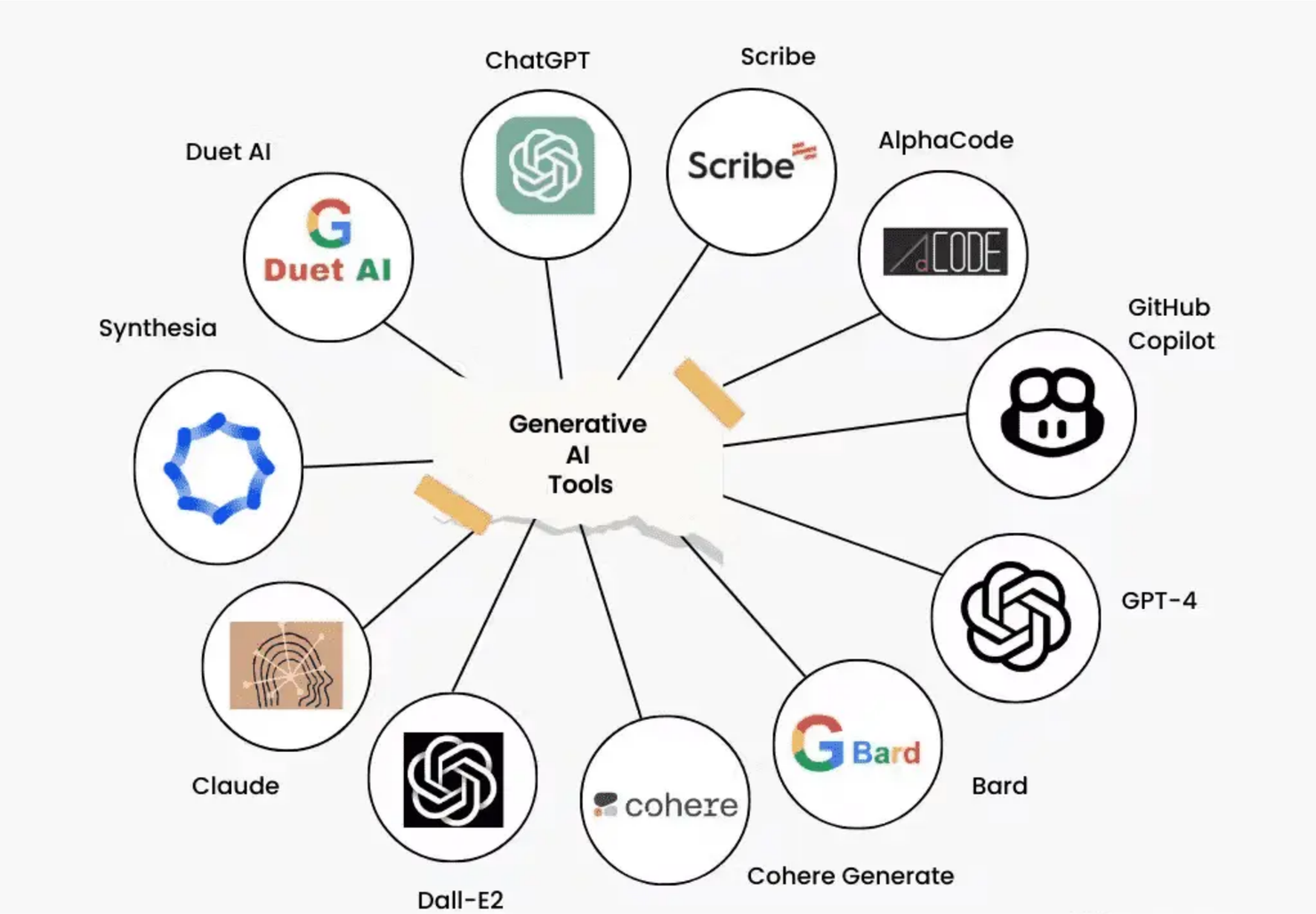
Constraining Deep Generative Models with Neurosymbolic Approach

Speaker: **Zhe Zeng**

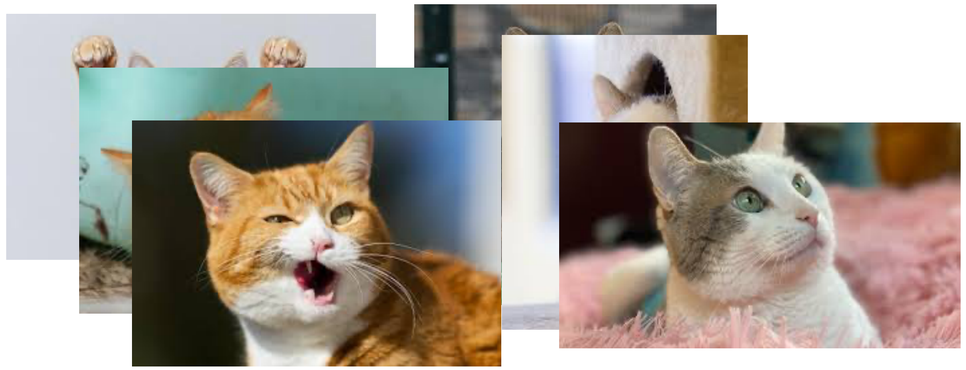
UVA AI/ML Seminar — *Oct 29th*

Progress of deep generative models

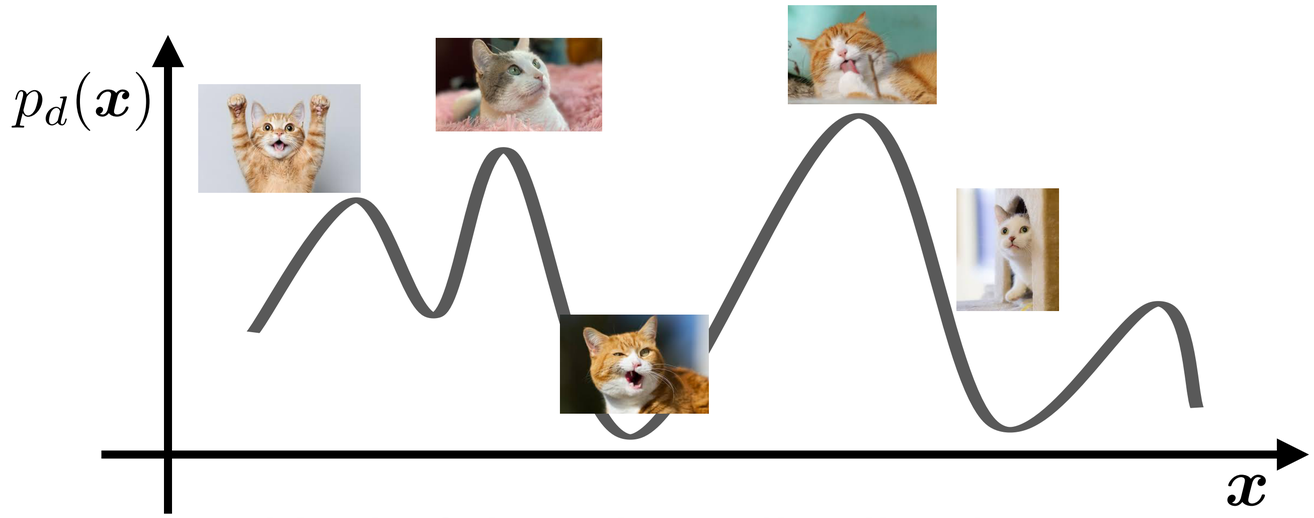


Estimating the probability distribution of data

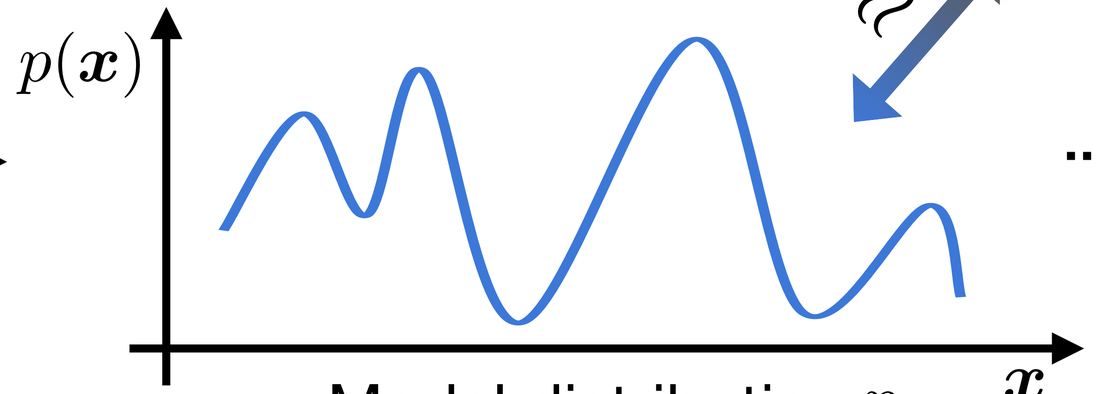
How they work?



Data samples



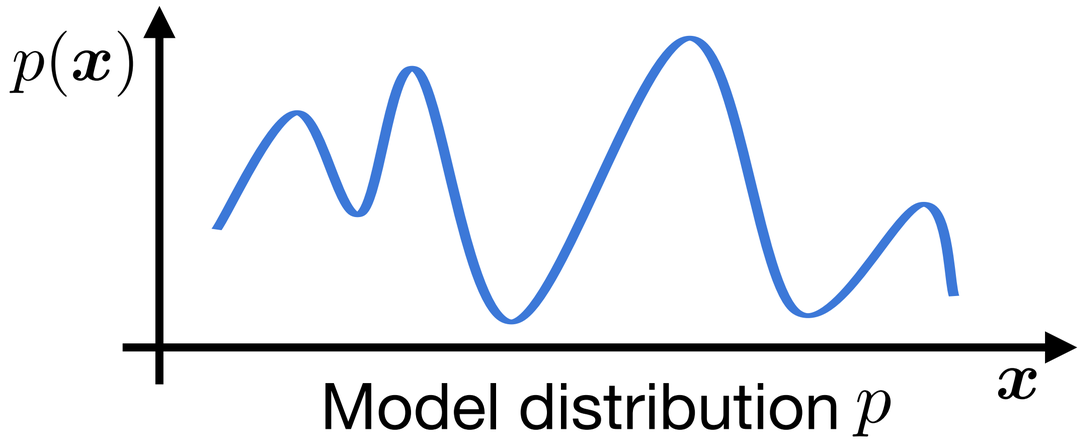
Underlying data distribution p_d
(unknown)



Model distribution p

Sample

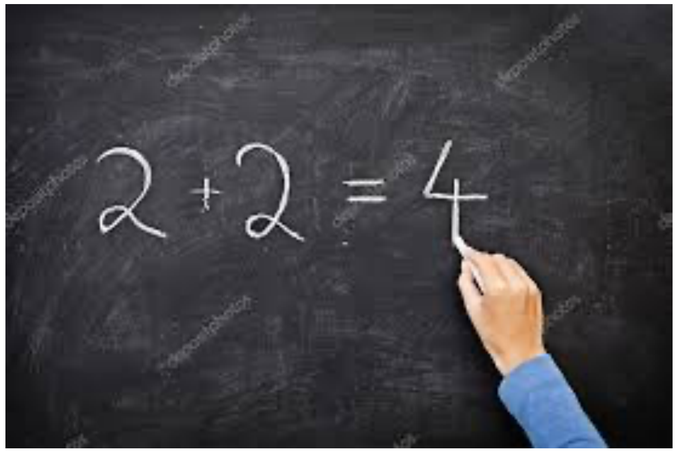
Realistic data points



- They are ***unconstrained***
- … but some events are certain
- and *constraint is a must!*

Why do we need constraints?

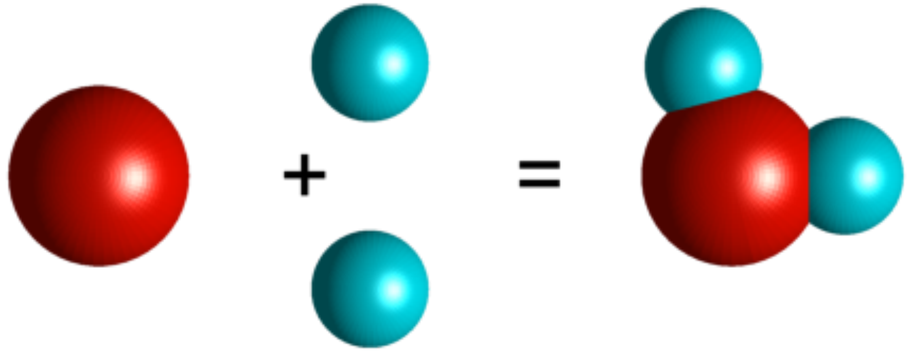
Math reasoning and logical deduction



Constraints: carrying out arithmetic tasks, but also proving theorems

Why do we need constraints?

Physics law

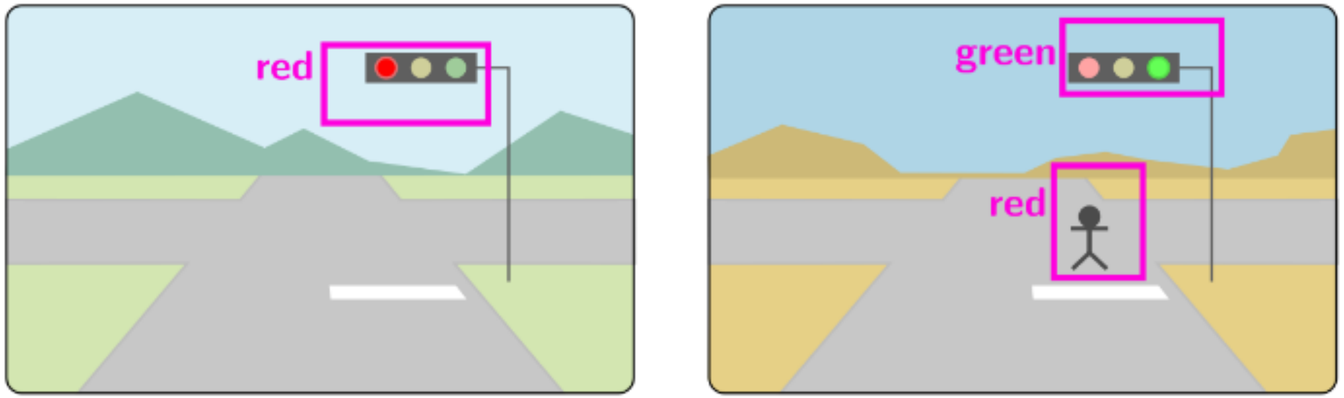


Constraints: preserve #atoms, #electrons … in chemical reactions

Why do we need constraints?

AI safety

$$K_1 = (\text{pedestrian} \vee \text{red} \Rightarrow \text{stop})$$

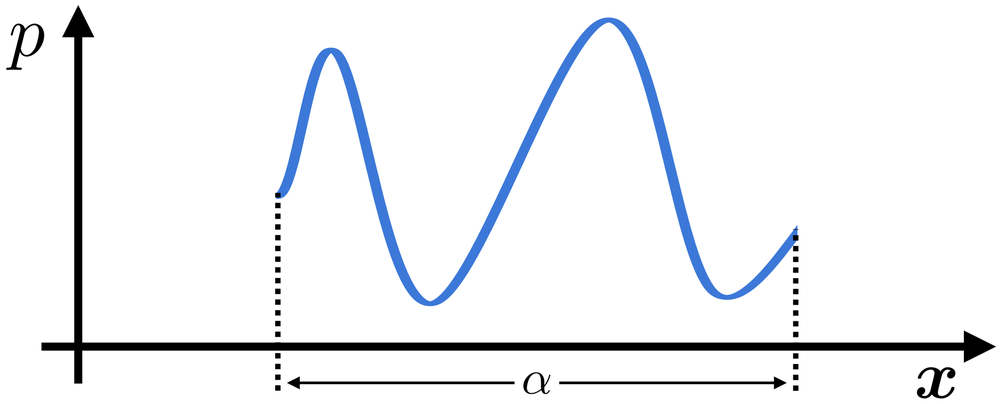


Constraints: traffic rules, scene understanding …

Marconato, Emanuele, et al. "Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts." *NeurIPS 2023*

Goal

Given constraint α , the generated samples should be **realistic** and **compliant**



Model distribution p

Guaranteed to satisfy constraints

$$p(\boldsymbol{x} \mid \alpha) = \begin{cases} 0, & if \;\; \boldsymbol{x} \not\models \alpha \\ \frac{p(\boldsymbol{x})}{p(\alpha)}, & otherwise \end{cases}$$

Approximate data distribution

But how bad are purely neural models when dealing with hard constrains in the real world?

Fixed-sum constraint

Variational Autoencoders

- **Dataset**: modified MNIST, each image has the same pixel sum $\sum_i x_i = k$

Variational Autoencoders

- **Dataset**: modified MNIST, each image has the same pixel sum $\sum_i x_i = k$

- **Results:**

MODEL	LL ↑	VIOLATION ↓
VAE	-22.42 $\pm$ 0.29	0.30 $\pm$ 0.06
Ladder VAE	-24.25 $\pm$ 0.07	0.38 $\pm$ 0.02
Graph VAE	-22.74 $\pm$ 0.11	0.29 $\pm$ 0.09

Note: constraint violation rate is measured with some tolerance

Diffusion Models

- **Dataset**: modified CIFAR/CelebA/LSUN Church/LSUN Cat
- **Results:**

DATASET	MODEL	FID ↓	IS ↑	VIOLATION ↓
CIFAR	DDPM	4.173	9.278 ± 0.116	0.999
	DDIM	8.123	8.646 ± 0.084	1.0
CelebA	DDPM	10.345	2.358 ± 0.030	0.999
	DDIM	12.417	2.366 ± 0.033	0.999
LSUN Church	DDPM	4.945	2.460 ± 0.028	1.0
	DDIM	6.702	2.584 ± 0.027	0.999
LSUN Cat	DDPM	12.913	4.705 ± 0.047	1.0
	DDIM	18.958	4.849 ± 0.062	0.999

Note: constraint violation rate is measured with some tolerance

But how bad are purely neural models when dealing with hard constrains in the real world?

Learn from data ❌ **Constraint integration** ✅

Challenges in constraint integration

1) Constraints are hard to represent in a unified way

Logical constraints

- Propositional logic

$$(a \wedge b) \vee d \Rightarrow c$$

- First-order logic (FOL)

$$\forall a \exists b : R(a, b) \wedge Q(b) \Rightarrow C(b)$$

- Satisfiability modulo theory (SMT)

$$(x_i - \beta x_j \leq 100) \vee (x_j + x_k \geq 0) \Rightarrow (x_j x_k \leq x_i)$$

Challenges in constraint integration

1) Constraints are hard to represent in a unified way
2) How to integrate constraints and probabilities in a single architecture

Challenges in constraint integration

1) Constraints are hard to represent in a unified way
2) How to integrate constraints and probabilities in a single architecture
3) Logical constraints are piecewise constant functions (non-differentiable)

Solution

Probabilistic neuro-symbolic AI

Solution

Probabilistic neuro-symbolic AI

Integrate probabilistic reasoning

Solution

Probabilistic neuro-symbolic AI

Deep neural nets

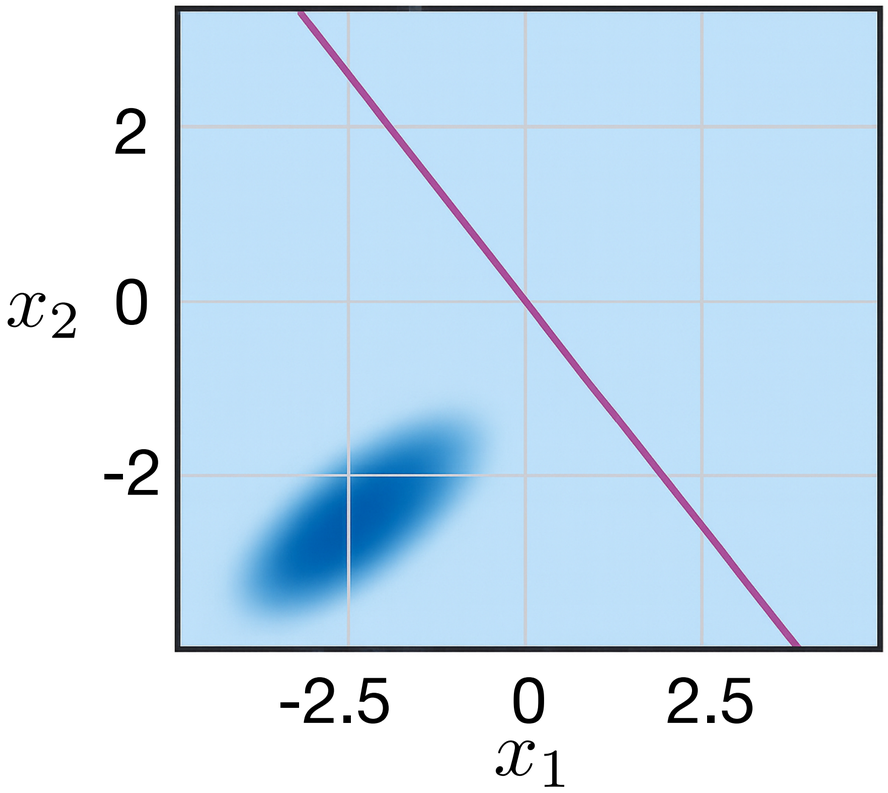
Solution

Probabilistic neuro-symbolic AI

Explicit hard constraints

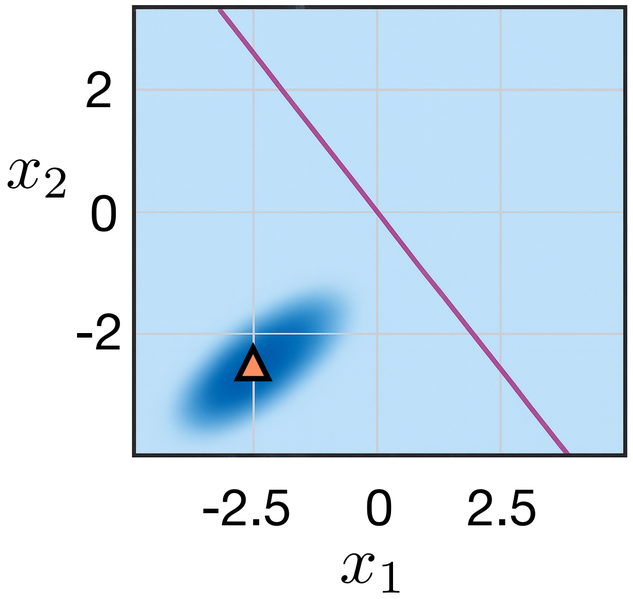
Motivating example

- Given an unconstrained model distribution, and a constraint $x_1 + x_2 = 0$
- Question: how to *sample* and *optimize*?



Motivating example

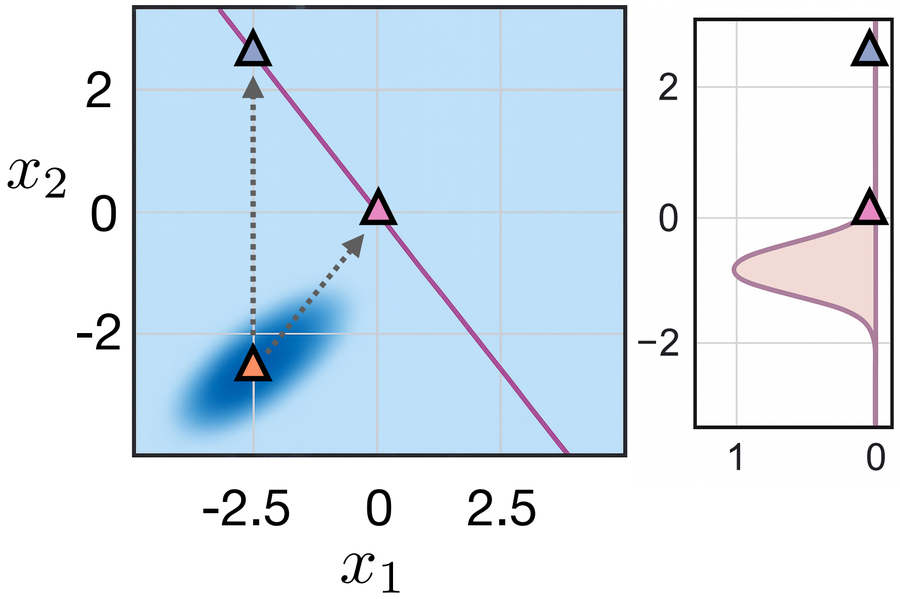
- Given an unconstrained model distribution, and a constraint $x_1 + x_2 = 0$
- Question: how to *sample* and *optimize*?



	▲ MAP
Sample	$\boldsymbol{x} \sim p(\boldsymbol{x})$
Optimize	$\mathbb{E}_{p(\boldsymbol{x})}[\ell(\boldsymbol{x})]$
Constraint	violate
Likelihood	high

Motivating example

- Given an unconstrained model distribution, and a constraint $x_1 + x_2 = 0$
- Question: how to *sample* and *optimize*?

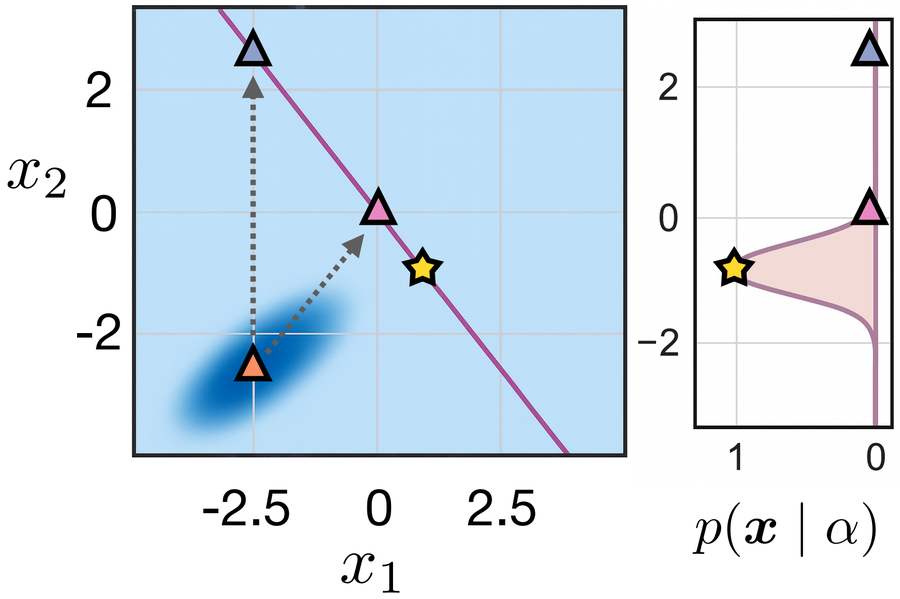


	MAP	L1 L2
Sample	$\boldsymbol{x} \sim p(\boldsymbol{x})$	$\boldsymbol{x}' \sim p(\boldsymbol{x}')$ $\boldsymbol{x} = \arg\min_{\boldsymbol{x}} D(\boldsymbol{x}, \boldsymbol{x}')$
Optimize	$\mathbb{E}_{p(\boldsymbol{x})}[\ell(\boldsymbol{x})]$	$\mathbb{E}_{p(\boldsymbol{x}')}[\ell(\boldsymbol{x}(\boldsymbol{x}'))]$
Constraint	violate	satisfy
Likelihood	high	low

Cătălina Stoian, Mihaela, et al. "How Realistic Is Your Synthetic Data? Constraining Deep Generative Models for Tabular Data." *ICLR 2024*

Motivating example

- Given an unconstrained model distribution, and a constraint $x_1 + x_2 = 0$
- Question: how to *sample* and *optimize*?



	▲ MAP	▲ L1 ▲ L2	★ Ours
Sample	$\boldsymbol{x} \sim p(\boldsymbol{x})$	$\boldsymbol{x}' \sim p(\boldsymbol{x}')$ $\boldsymbol{x} = \arg\min_{\boldsymbol{x}} D(\boldsymbol{x}, \boldsymbol{x}')$	$\boldsymbol{x} \sim p(\boldsymbol{x} \mid \alpha)$
Optimize	$\mathbb{E}_{p(\boldsymbol{x})}[\ell(\boldsymbol{x})]$	$\mathbb{E}_{p(\boldsymbol{x}')}[\ell(\boldsymbol{x}(\boldsymbol{x}'))]$	$\mathbb{E}_{p(\boldsymbol{x}\mid\alpha)}[\ell(\boldsymbol{x})]$
Constraint	violate	satisfy	satisfy
Likelihood	high	low	high

Cătălina Stoian, Mihaela, et al. "How Realistic Is Your Synthetic Data? Constraining Deep Generative Models for Tabular Data." *ICLR 2024*

Gradient Estimator Design

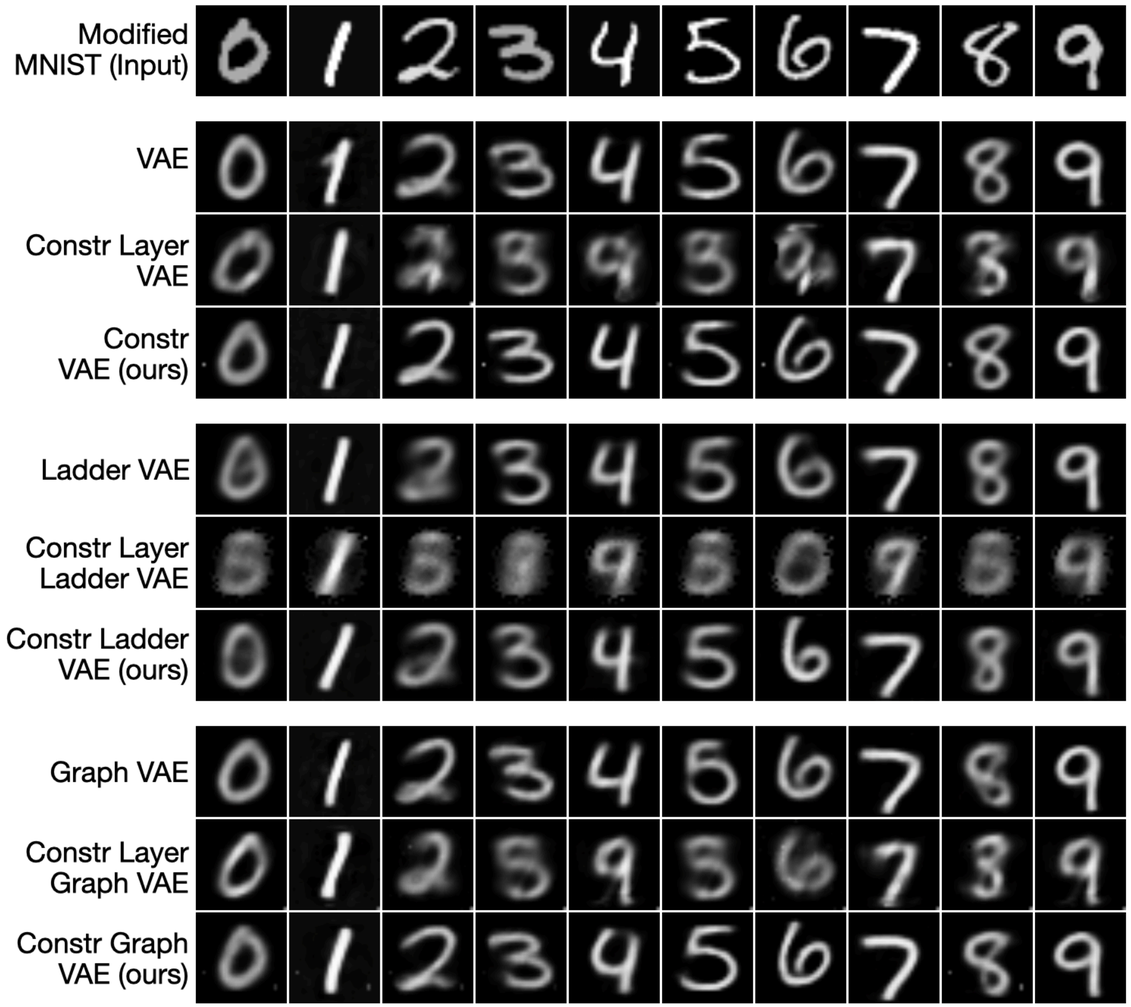
- Constrained sampling $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \alpha)$
- Optimization $\nabla_\theta \mathbb{E}_{p_\theta(\boldsymbol{x}|\alpha)}[\ell(\boldsymbol{x})]$

Gradient Estimator Design

- Constrained sampling $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \alpha)$
- Optimization $\nabla_\theta \mathbb{E}_{p_\theta(\boldsymbol{x}|\alpha)}[\ell(\boldsymbol{x})]$
  - Approximation: $\nabla_\theta \mathbb{E}_{p_\theta(\boldsymbol{x}|\alpha)}[\ell(\boldsymbol{x})] \approx \partial_\theta \boldsymbol{\mu}(\theta) \nabla_{\boldsymbol{x}} \ell(\boldsymbol{x})$

*=> It allows the constraint to be integrated at **any** layer*

Constrained-VAE

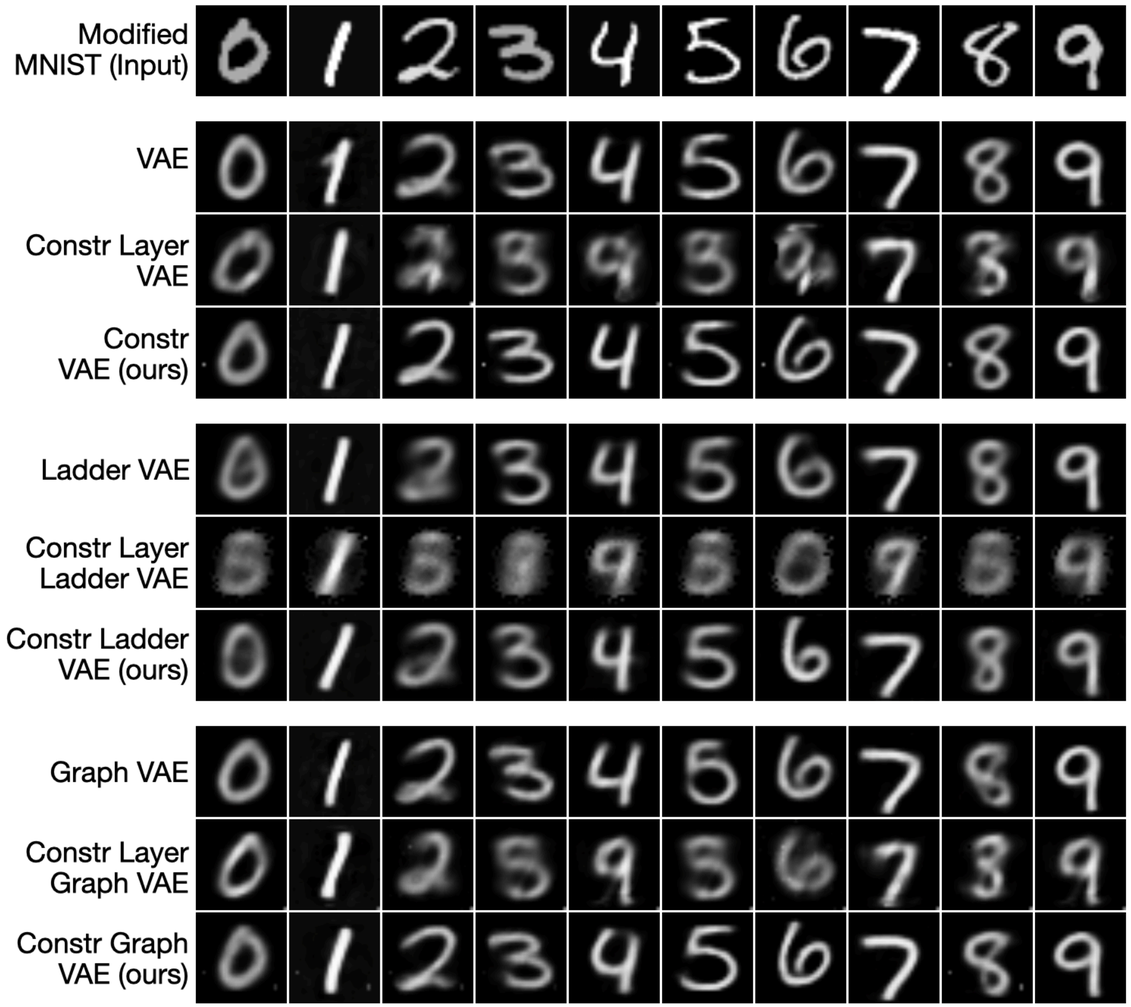


Realistic but not compliant

Compliant but not realistic

Realistic and compliant!

Constrained-VAE



MODEL	LL ↑	ELBO ↑	RL ↓	VIOLATION ↓
VAE	-22.42 ± 0.29	-23.41 ± 0.22	15.00 ± 0.46	0.30 ± 0.06
VAE + Constr Layer	-34.45 ± 2.64	-40.89 ± 9.37	37.11 ± 9.35	**0.00 ± 0.00**
VAE + Constr Reparam	-22.33 ± 0.48	-23.83 ± 0.49	14.54 ± 0.58	**0.00 ± 0.00**
ours	**-21.48 ± 0.18**	**-22.62 ± 0.07**	**12.79 ± 0.11**	**0.00 ± 0.00**
Ladder VAE	-24.25 ± 0.07	-30.84 ± 0.51	**23.06 ± 0.54**	0.38 ± 0.02
Ladder VAE + Constr Layer	-36.83 ± 0.56	-39.59 ± 0.56	37.46 ± 0.55	**0.00 ± 0.00**
Ladder VAE + Constr Reparam	-25.27 ± 0.19	-31.56 ± 0.48	25.20 ± 0.62	**0.00 ± 0.00**
ours	**-23.86 ± 0.06**	**-30.78 ± 0.08**	**23.40 ± 0.16**	**0.00 ± 0.00**
Graph VAE	-22.74 ± 0.11	-23.54 ± 0.18	15.45 ± 0.41	0.29 ± 0.09
Graph VAE + Constr Layer	-33.27 ± 3.60	-33.27 ± 5.84	28.29 ± 6.40	**0.00 ± 0.00**
Graph VAE + Constr Reparam	-22.96 ± 0.80	-23.55 ± 0.86	16.08 ± 1.38	**0.00 ± 0.00**
ours	**-21.61 ± 0.20**	**-22.53 ± 0.06**	**12.73 ± 0.21**	**0.00 ± 0.00**

Constrained diffusion models

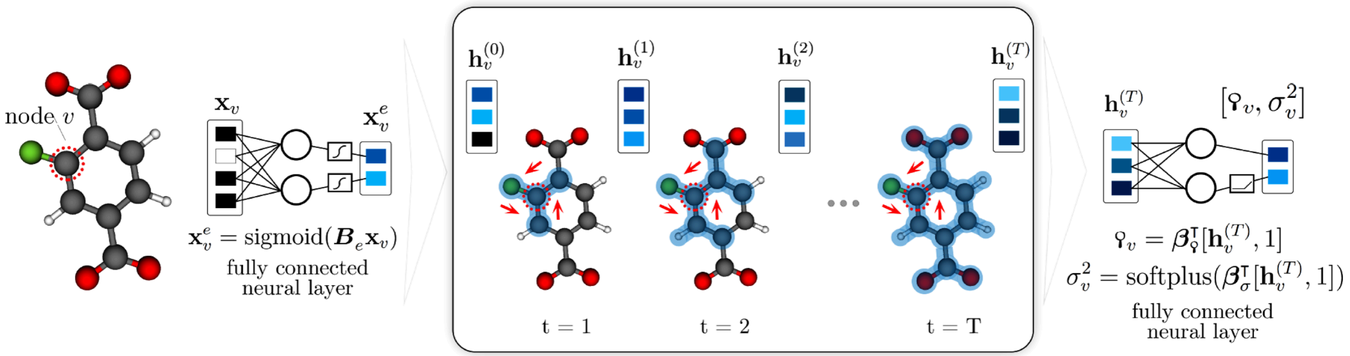
Dataset	Model	FID ↓	IS ↑	Violation ↓
CIFAR	Ours	**3.811**	9.223 ± 0.130	**0**
	Constr Layer	4.234	8.535 ± 0.157	**0**
	Constr Reparam	3.881	9.212 ± 0.122	**0**
	DDPM	4.173	**9.278 ± 0.116**	0.999
CelebA	Ours	**10.193**	**2.360 ± 0.016**	**0**
	Constr Layer	12.067	2.345 ± 0.039	**0**
	Constr Reparam	10.961	2.043 ± 0.028	**0**
	DDPM	10.345	2.358 ± 0.030	0.999
LSUN Church	Ours	**4.779**	**2.471 ± 0.020**	**0**
	Constr Layer	6.695	2.319 ± 0.028	**0**
	Constr Reparam	4.895	2.377 ± 0.032	**0**
	DDPM	4.945	2.460 ± 0.028	1.0
LSUN Cat	Ours	**12.489**	**4.711 ± 0.054**	**0**
	Constr Layer	13.472	4.642 ± 0.081	**0**
	Constr Reparam	12.696	4.707 ± 0.062	**0**
	DDPM	12.913	4.705 ± 0.047	1.0

Dataset	Model	FID ↓	IS ↑	Violation ↓
CIFAR	Ours	**7.972**	8.585 ± 0.118	**0**
	DDIM	8.123	**8.646 ± 0.084**	1.0
CelebA	Ours	**12.389**	**2.367 ± 0.023**	**0**
	DDIM	12.417	2.366 ± 0.033	0.999
LSUN Church	Ours	**6.557**	**2.596 ± 0.033**	**0**
	DDIM	6.702	2.584 ± 0.027	0.9999
LSUN Cat	Ours	**18.902**	**4.857 ± 0.037**	**0**
	DDIM	18.958	4.849 ± 0.062	0.9999

=> The integration of constraint improves generation quality!

Constrained Graph Neural Networks

Partial Charge Prediction to Metal-Organic Frameworks

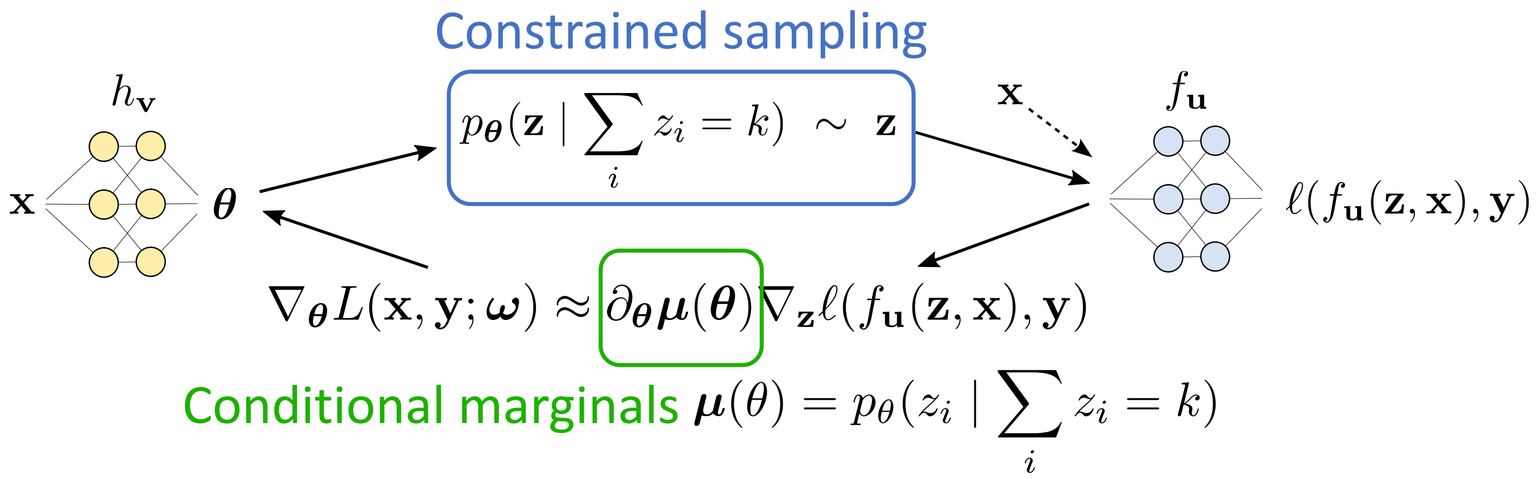


METHOD neutrality enforcement	MAD ↓ mean ± std	NLL ↓ mean ± std
Constrained Layer	0.327 ± 0.004	103.522 ± 3.018
Constant Prediction	0.324 ± 0.007	—
Element-mean (uniform)	0.154 ± 0.002	—
Element-mean (variance)	0.153 ± 0.002	—
MPNN (KKThPINN)	0.0260 ± 0.0008	109.8 ± 6.9
MPNN (Constr Reparam)	0.0256 ± 0.0007	334.03 ± 2398
MPNN (variance)	0.0251 ± 0.0010	-19.9 ± 71.1
Ours	**0.0248 ± 0.0008**	**-252 ± 24.7**
Constrained Layer (ens)	0.319 ± 0.002	99.236 ± 2.3
MPNN (ens, KKThPINN)	0.0244 ± 0.0006	57.29 ± 12.8
MPNN (ens, Constr Reparam)	0.0242 ± 0.0006	4883.46 ± 1695
MPNN (ens, variance)	0.0238 ± 0.0007	-45.2 ± 55.8
Ours (ens)	**0.0231 ± 0.0007**	**-180 ± 38.3**

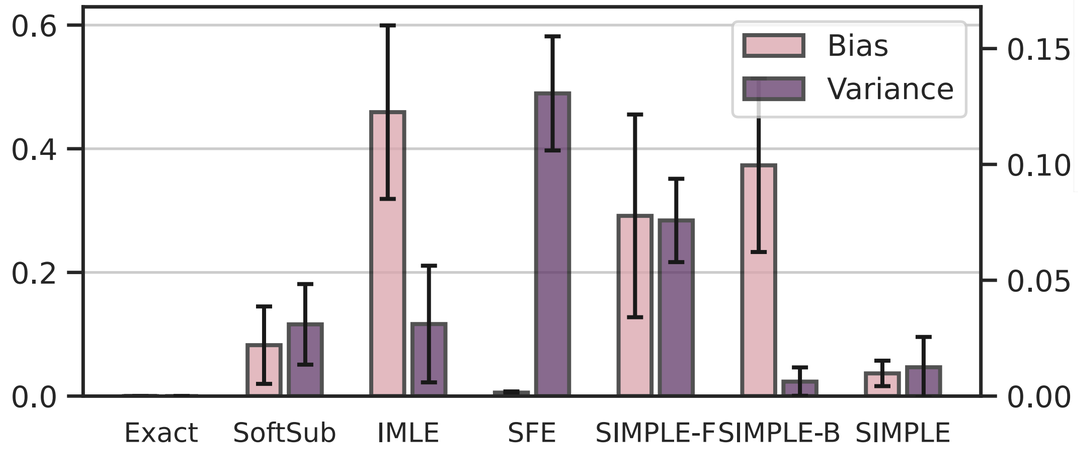
Raza, Ali, et al. "Message passing neural networks for partial charge assignment to metal–organic frameworks." *The Journal of Physical Chemistry C* 124.35 (2020): 19070-19082.

Gradient estimator for *k*-subset sampling

- **k-subset selection:** fixed-sum constraint in discrete domain with $z_i \in \{0, 1\}$
  - Enforcing $\sum_i z_i = k$ means selecting a subset of size exactly k
- Our proposed gradient estimator *SIMPLE*



Evaluation



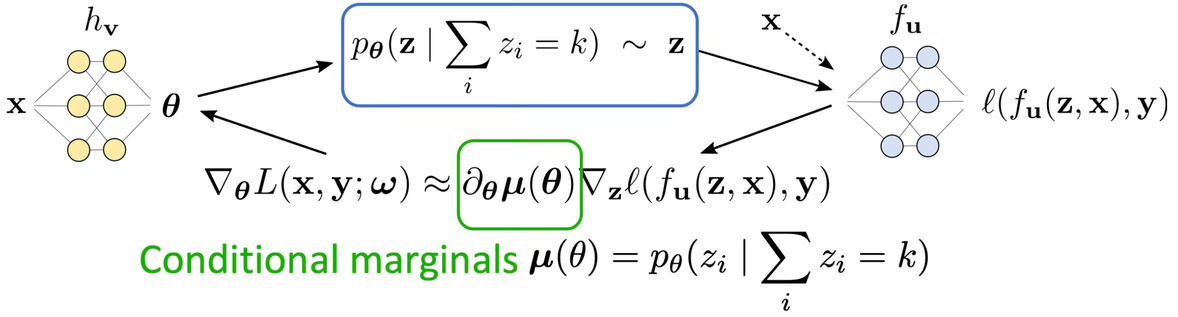
Gumbel Softmax

Perturb & MAP

Score Function

Ours

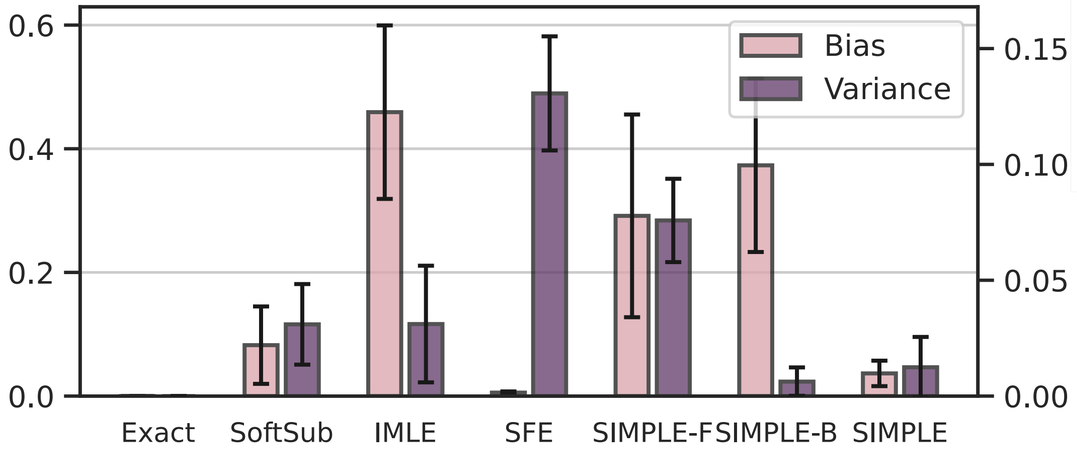
Constrained sampling



Conditional marginals $\boldsymbol{\mu}(\theta) = p_\theta(z_i \mid \sum_i z_i = k)$

Why does SIMPLE work well?

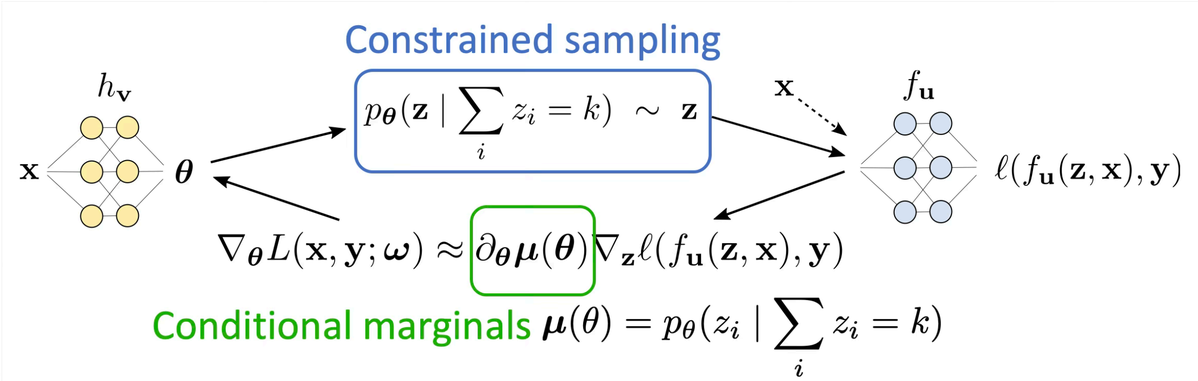
Ablation study



Perturb & MAP

Constrained sampling
+ P&M backward

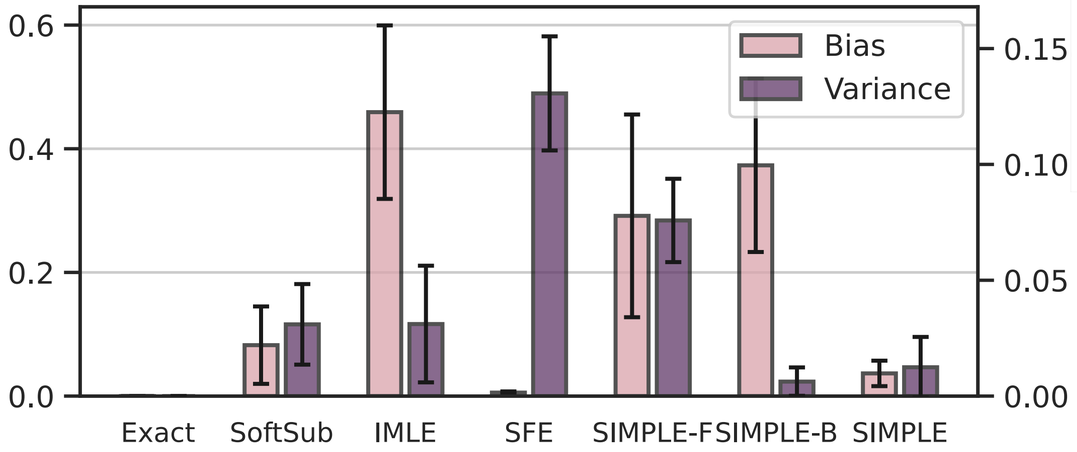
Ours



=> Exact constrained sampling reduces the **bias**

Why does SIMPLE work well?

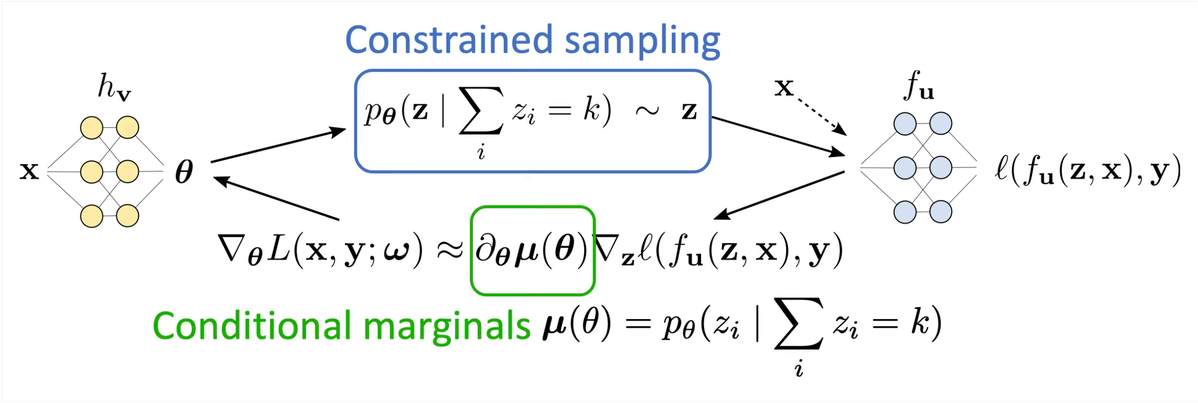
Ablation study



Perturb & MAP

P&M forward + conditional marginals

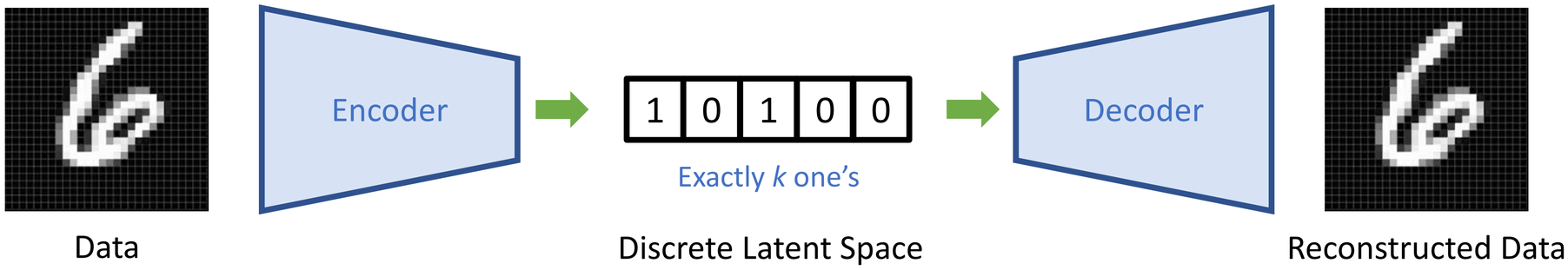
Ours



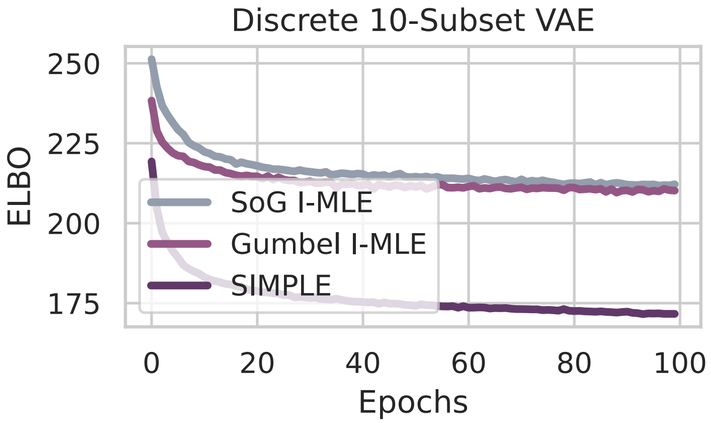
=> Exact constrained sampling reduces the **bias**

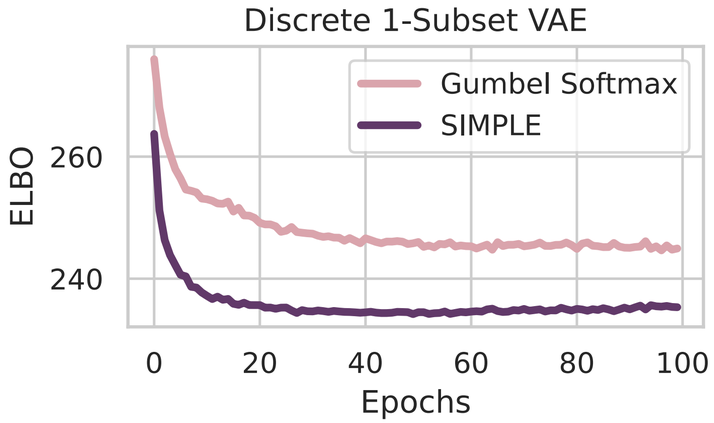
=> Conditional marginals reduces the **variance** by marginalizing over all possible samples

Evaluation — Discrete VAE



Metric: *exact* ELBO





Evaluation — Learn to explain (L2X)

Dataset

- Beer Review

Input (*review*)	Output
a little like wood and caramel with a hop finish. has a sort of fruity flavor like grapes or cherry that is sort of buried in there. mouth feel was lite, sort of bubbly.	0.75

- Human annotations of the words that best describe the review

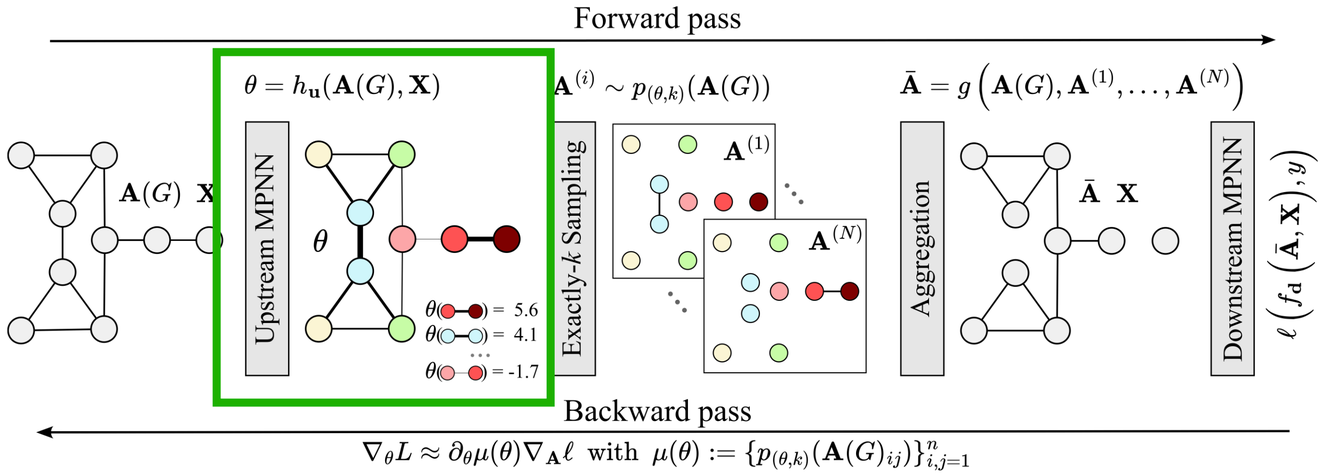
Predictive performance ↓

	Mean Squared Error ↓	Precision ↑
SIMPLE (Ours)	**2.11**	**42.31**
L2X	6.92	32.23
SoftSub	2.18	41.98
IMLE	2.38	41.38

↑ **How much the learned explanations match the ones from human**

Probabilistically rewired message-passing neural networks

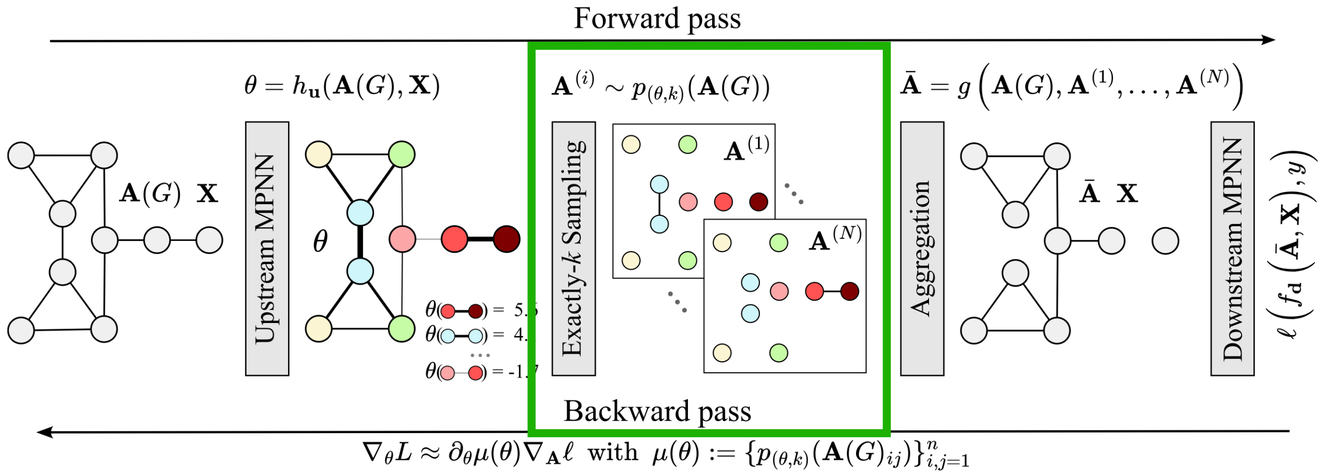
Goal: to learning graph structures to mitigate issues like over-squashing and under-reaching to improve learning efficiency



Qian, Chendi, et al. "Probabilistically rewired message-passing neural networks." *ICLR 2024*

Probabilistically rewired message-passing neural networks

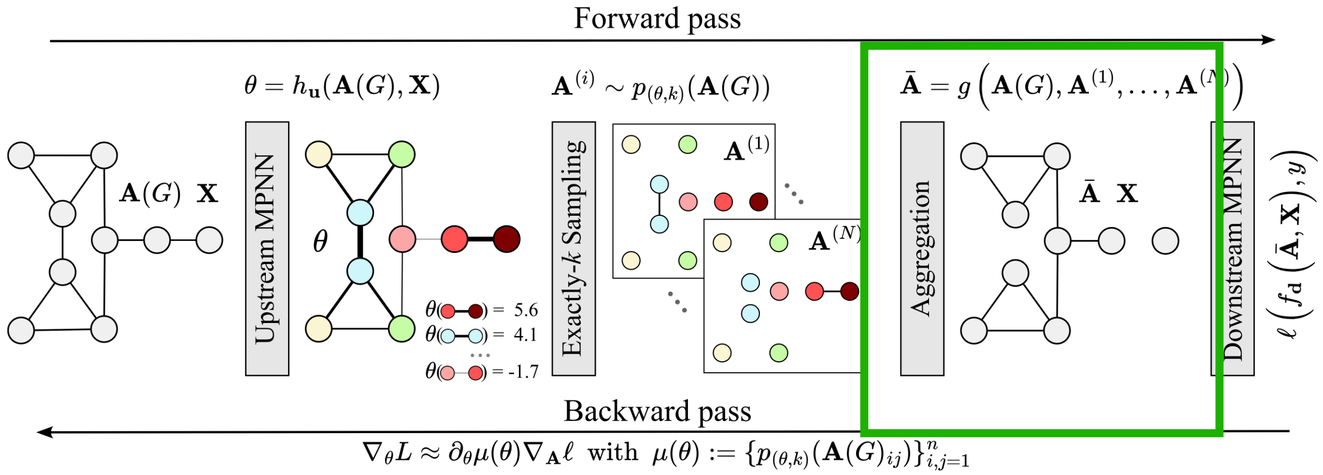
Goal: to learning graph structures to mitigate issues like over-squashing and under-reaching to improve learning efficiency



Qian, Chendi, et al. "Probabilistically rewired message-passing neural networks." *ICLR 2024*

Probabilistically rewired message-passing neural networks

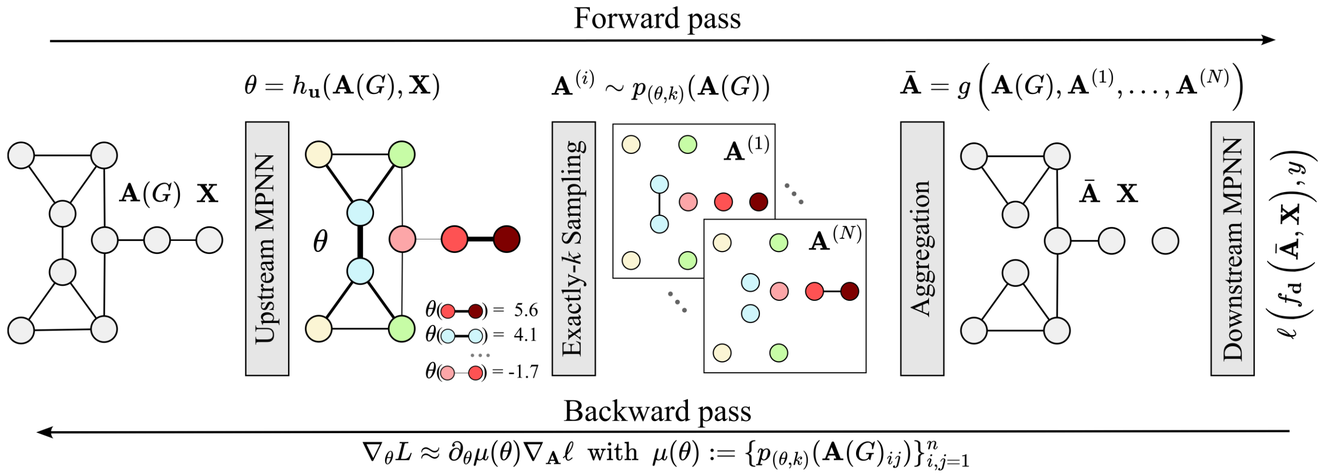
Goal: to learning graph structures to mitigate issues like over-squashing and under-reaching to improve learning efficiency

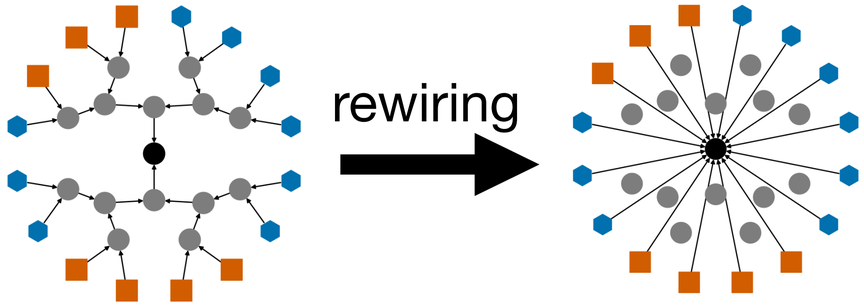


Qian, Chendi, et al. "Probabilistically rewired message-passing neural networks." *ICLR 2024*

Probabilistically rewired message-passing neural networks

Goal: to learning graph structures to mitigate issues like over-squashing and under-reaching to improve learning efficiency



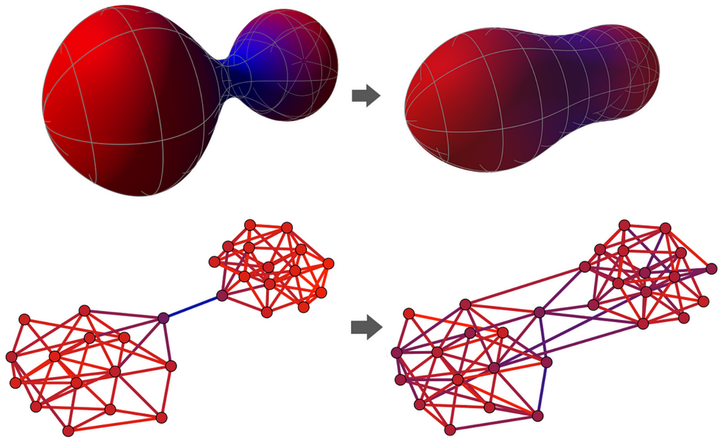


Task tree leaf counting

Qian, Chendi, et al. "Probabilistically rewired message-passing neural networks." *ICLR 2024*

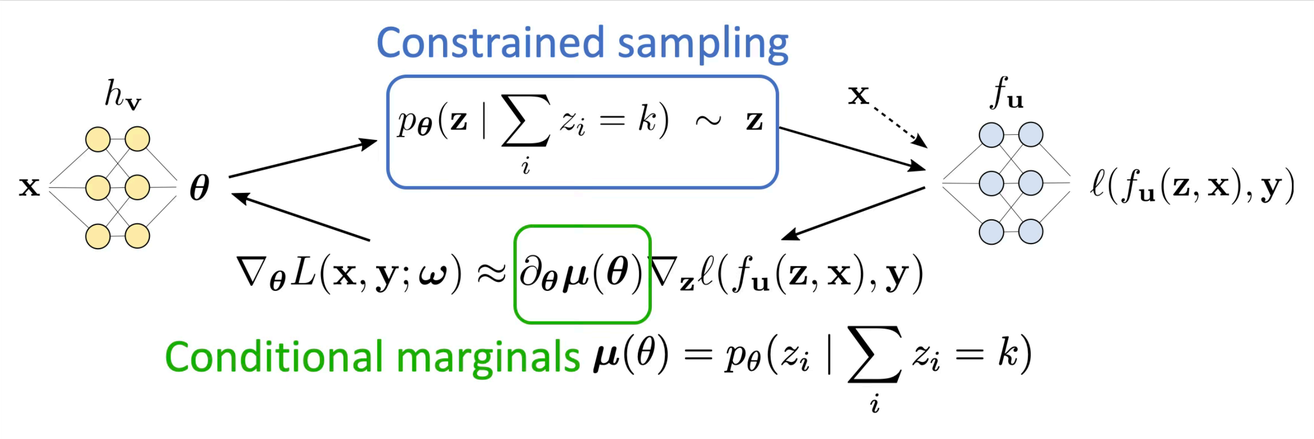
Probabilistically rewired message-passing neural networks

TUDataset molecular prediction datasets



Best 2nd-best 3rd-best

MODEL	MUTAG	PTC_MR	PROTEINS	NCI1	NCI109
GK ($k = 3$) (SHERVASHIDZE ET AL., 2009)	81.4±1.7	55.7±0.5	71.4±0.3	62.5±0.3	62.4±0.3
PK (NEUMANN ET AL., 2016)	76.0±2.7	59.5±2.4	73.7±0.7	82.5±0.5	N/A
WL KERNEL (SHERVASHIDZE ET AL., 2011)	90.4±5.7	59.9±4.3	75.0±3.1	**86.0**±1.8	N/A
DGCNN (ZHANG ET AL., 2018)	85.8±1.8	58.6±2.5	75.5±0.9	74.4±0.5	N/A
IGN (MARON ET AL., 2019B)	83.9±13.0	58.5±6.9	76.6±5.5	74.3±2.7	72.8±1.5
GIN (XU ET AL., 2019)	89.4±5.6	64.6±7.0	76.2±2.8	82.7±1.7	N/A
PPGNs (MARON ET AL., 2019A)	90.6±8.7	66.2±6.6	77.2±4.7	83.2±1.1	82.2±1.4
NATURAL GN (DE HAAN ET AL., 2020)	89.4±1.6	66.8±1.7	71.7±1.0	82.4±1.3	83.0±1.9
GSN (BOURITSAS ET AL., 2022)	92.2±7.5	68.2±7.2	76.6±5.0	83.5±2.0	83.5±2.3
CIN (BODNAR ET AL., 2021)	92.7±6.1	68.2±5.6	77.0±4.3	83.6±1.4	**84.0**±1.6
CAN (GIUSTI ET AL., 2023A)	**94.1**±4.8	**72.8**±8.3	**78.2**±2.0	84.5±1.6	83.6±1.2
CIN++ (GIUSTI ET AL., 2023B)	**94.4**±3.7	**73.2**±6.4	**80.5**±3.9	**85.3**±1.2	**84.5**±2.4
FOSR (KARHADKAR ET AL., 2022)	86.2±1.5	58.5±1.7	75.1±0.8	72.9±0.6	71.1±0.6
DROPGNN (PAPP ET AL., 2021)	90.4±7.0	66.3±8.6	76.3±6.1	81.6±1.8	80.8±2.6
PR-MPNN (10-FOLD CV)	**98.4**±2.4	**74.3**±3.9	**80.7**±3.9	**85.6**±0.8	**84.6**±1.2



Constrained sampling is hard in general

… and so is computing the conditional marginals

🤔

Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} = \text{"the weather is"})$

Zhang, Honghua, et al. "Adaptable logical control for large language models." *NeurIPS 2024*

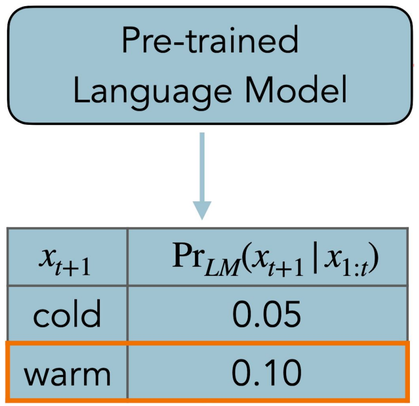
Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} =$ "the weather is")

Unconstrained



x_{t+1}	$\Pr_{LM}(x_{t+1} \mid x_{1:t})$
cold	0.05
warm	0.10

Zhang, Honghua, et al. "Adaptable logical control for large language models." *NeurIPS 2024*

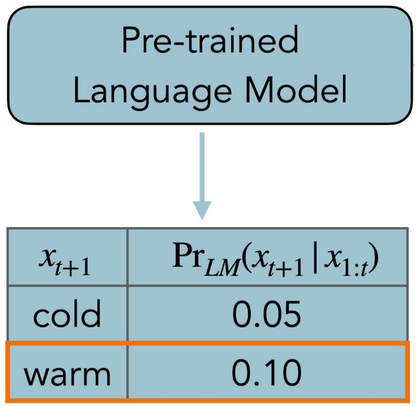
Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} =$ "the weather is")

Unconstrained



x_{t+1}	$\Pr_{LM}(x_{t+1} \mid x_{1:t})$
cold	0.05
warm	0.10

p_{LM}(next -token | α, prefix)

Zhang, Honghua, et al. "Adaptable logical control for large language models." *NeurIPS 2024*

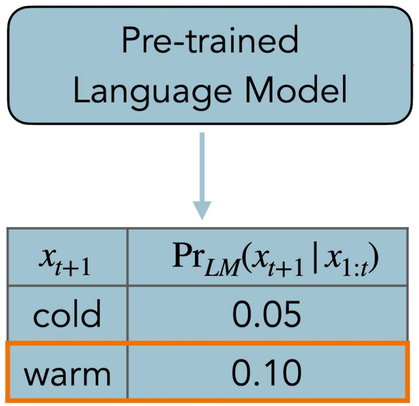
Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} =$ "the weather is")

Unconstrained



x_{t+1}	$\Pr_{LM}(x_{t+1} \mid x_{1:t})$
cold	0.05
warm	0.10

$$p_{LM}(\text{next-token} \mid \alpha, \text{prefix})$$

$$\propto p_{LM}(\text{next-token} \mid \text{prefix}) \cdot p_{LM}(\alpha \mid \text{next-token}, \text{prefix})$$

(Bayes rule)

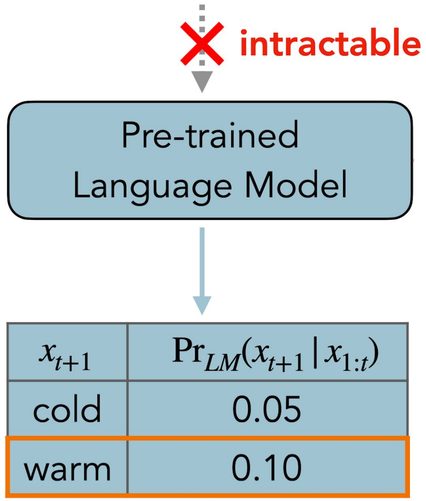
Zhang, Honghua, et al. "Adaptable logical control for large language models." *NeurIPS 2024*

Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} =$ "the weather is")



x_{t+1}	$\Pr_{LM}(x_{t+1} \mid x_{1:t})$
cold	0.05
warm	0.10

$$p_{LM}(\text{next-token} \mid \alpha, \text{prefix})$$

$$\propto p_{LM}(\text{next-token} \mid \text{prefix}) \cdot p_{LM}(\alpha \mid \text{next-token}, \text{prefix})$$

Intractable

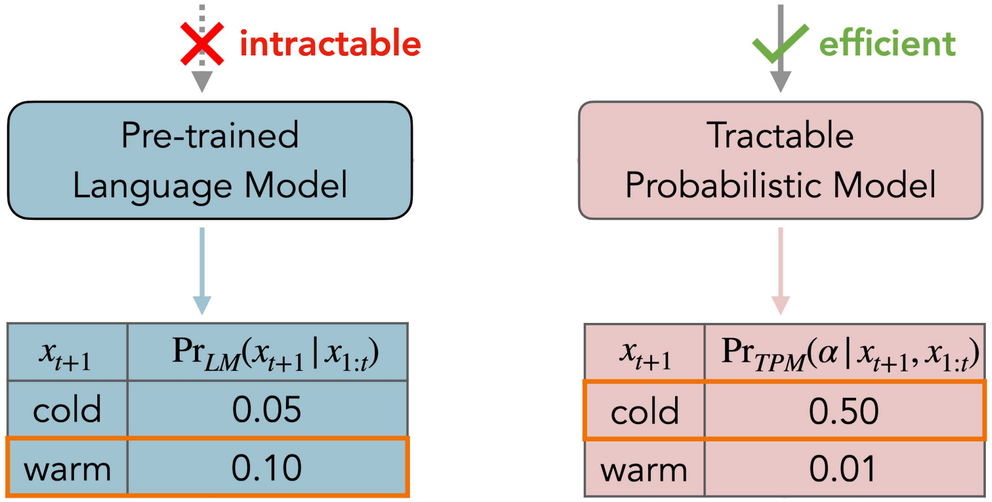
Zhang, Honghua, et al. "Adaptable logical control for large language models." *NeurIPS 2024*

Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} =$ "the weather is")



$$p_{LM}(\text{next-token} \mid \alpha, \text{prefix}) \propto p_{LM}(\text{next-token} \mid \text{prefix}) \cdot p_{LM}(\alpha \mid \text{next-token}, \text{prefix})$$

Intractable

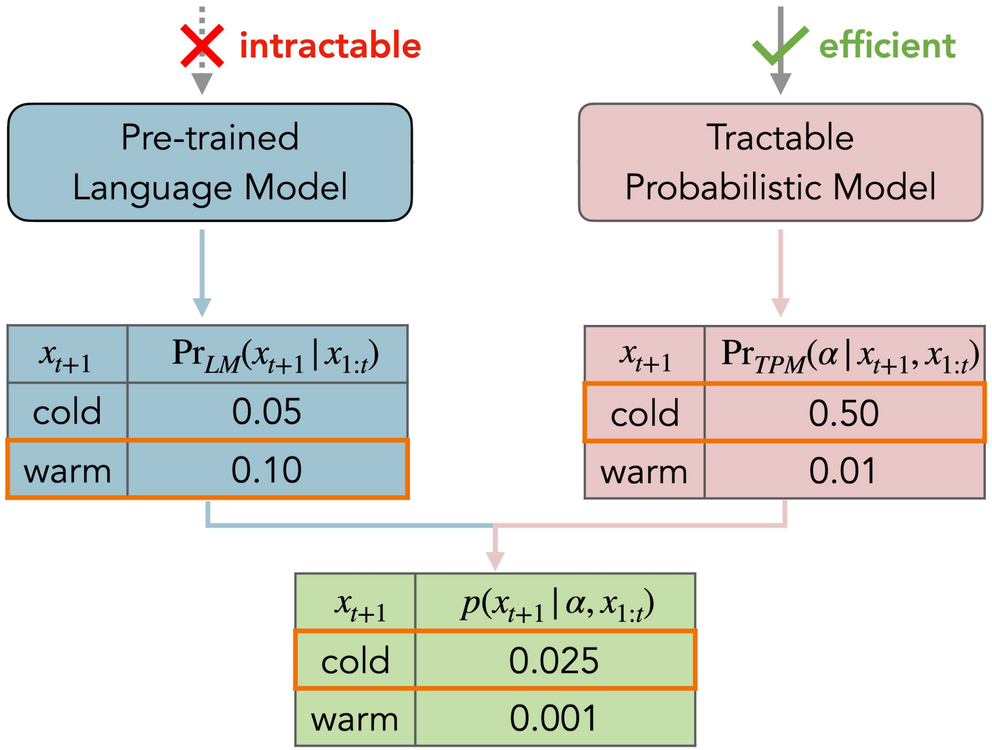
Choi, Y., Antonio Vergari, and Guy Van den Broeck. "Probabilistic circuits: A unifying framework for tractable probabilistic models."

Constrained text generation with LLMs

with *tractable probabilistic models*

Lexical Constraint α: sentence contains keyword “winter”

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} =$ "the weather is")



$p_{\text{CTRL-G}}$(next -token | α, prefix)

$\propto$ p_{LM}(next -token | prefix)

$\cdot$ p_{TPM}(α | next -token, prefix)

Zhang, Honghua, et al. "Adaptable logical control for large language models." *NeurIPS 2024*

CommonGen Benchmark

Generate a sentence using 3 to 5 concepts (keywords).

Input: snow drive car

α = ("car" ∨ "cars"…) ∧ ("drive" ∨ "drove"…) ∧

Reference 1: A car drives d̈ẅn a snow-covered road.

Reference 2: Two cars drove through the snow.

	BLEU-4		ROUGE-L		CIDEr		SPICE		Constraint	
	dev	*test*	*dev*	*test*	*dev*	*test*	*dev*	*test*	*dev*	*test*
supervised - base models trained with full supervision										
FUDGE	-	24.6	-	40.4	-	-	-	-	-	47.0%
A*esque	-	28.2	-	43.4	-	15.2	-	30.8	-	98.8%
NADO	30.8	-	44.4	-	16.1	-	32.0	-	88.8%	-
→ Ctrl-G	**35.1**	**34.4**	**46.7**	**46.4**	**17.4**	**17.6**	**32.7**	**33.3**	**100.0%**	**100.0%**
unsupervised - base models not trained with keywords as supervision										
A*esque	-	28.6	-	44.3	-	15.6	-	29.6	-	-
NADO	26.2	-	-	-	-	-	-	-	-	-
→ Ctrl-G	**32.1**	**31.5**	**45.2**	**44.8**	**16.0**	**16.2**	**30.8**	**31.2**	**100.0%**	**100.0%**

Challenges in constraint integration

1) Constraints are hard to represent in a unified way

=> tractability of constraint representation, solving …

2) How to integrate constraints and probabilities in a single architecture

=> constraint probability, approximation via TPMs

3) Logical constraints are piecewise constant functions (non-differentiable)

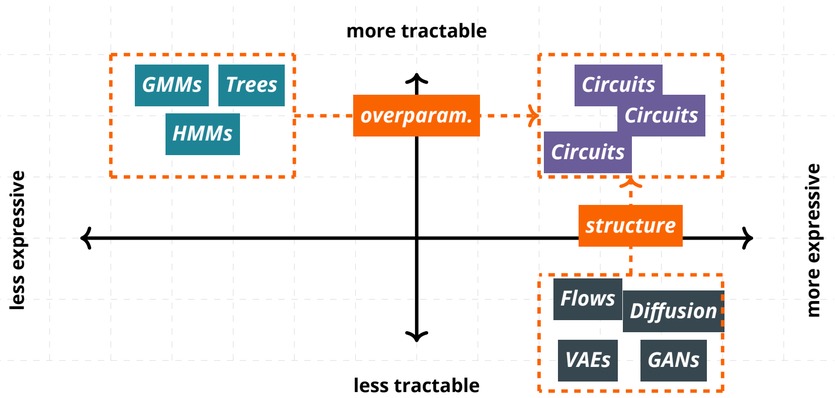


Image by Antonio Vergari

=> constrained marginals are informative proxy for gradient updates

=> expressiveness/efficiency of training TPMs

Thank you!