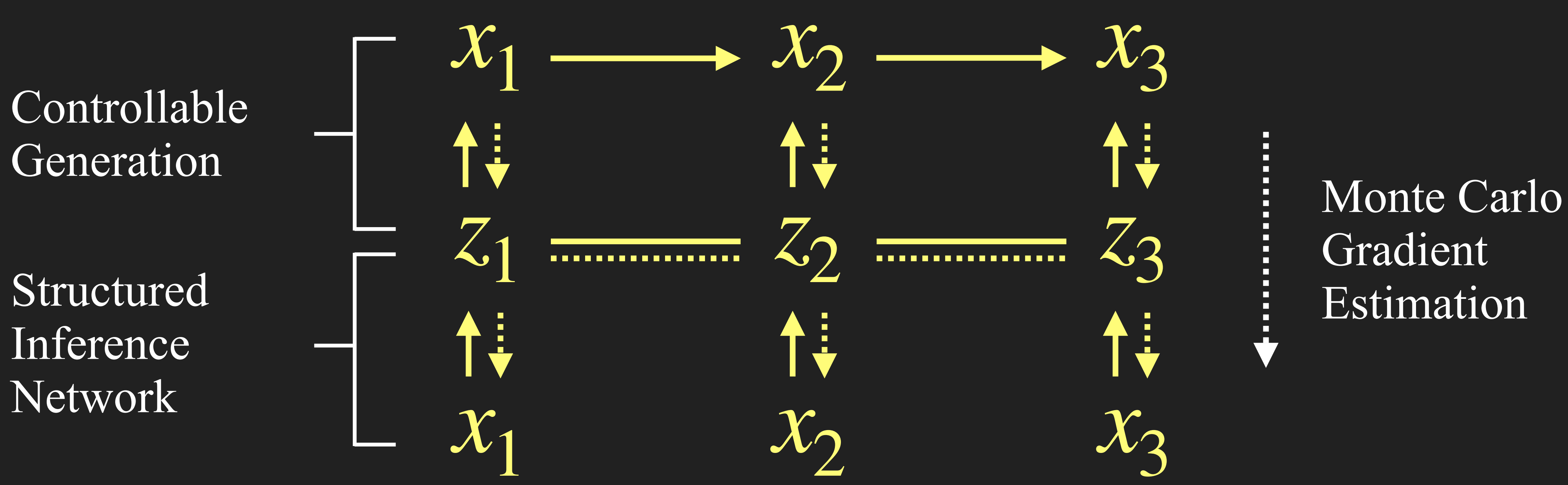
Latent Template Induction with Gumbel-CRFs

Yao Fu[1], Chuanqi Tan[2], Bin Bi[2], Mosha Chen[2], Yansong Feng[3], Alexander M. Rush[4]
[1]University of Edinburgh, [2]Alibaba Group, [3]Peking University, [4]Cornell University

NeurIPS 2020

Goal: Controllable Text Generation with Latent Templates



- Use templates z to control the structure of sentence x
- Infer z with linear-chain CRF
- Efficiently train z with reparameterized MC grad

Monte Carlo Gradient Estimation

$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log p(x\,|\,z)]$ # Goal of MC grad.

$= \mathbb{E}_{q_\phi(z|x)}[\underbrace{\log p(x\,|\,z)}_{\text{Reward}} \cdot \underbrace{\nabla_\phi \log q(z\,|\,x)}_{\text{Score}}]$ # High var. hard to train

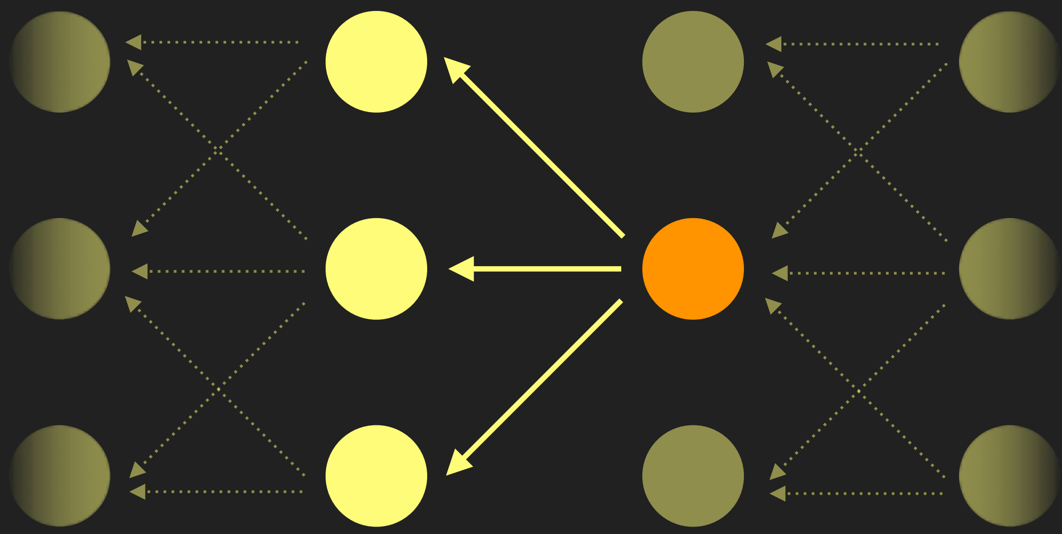
$= \mathbb{E}_{g(\epsilon)}[\nabla_\phi \log p(x\,|\,\underbrace{z(\epsilon, \phi)}_{\text{Reparameterization}})]$ # Lower var. more stable training

$= \mathbb{E}_{g(\epsilon)}[\nabla_z \log p(x\,|\,z) \odot \underbrace{\nabla_\phi \tilde{z}(\epsilon, \phi)}_{\text{Continuous Relaxation}}]$ # How to?

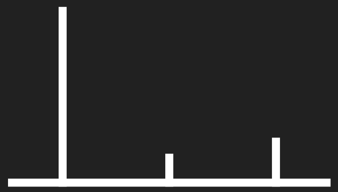
Large recent ML/ NLP trend

Quite challenging from many aspects

Gumbel-CRF Reparameterization



$$\hat{z}_t \sim p(z_t \mid \hat{z}_{t+1})$$



$$\tilde{z}_t = \text{Gumbel-Softmax}\,(p(z_t \mid \hat{z}_{t+1}))$$

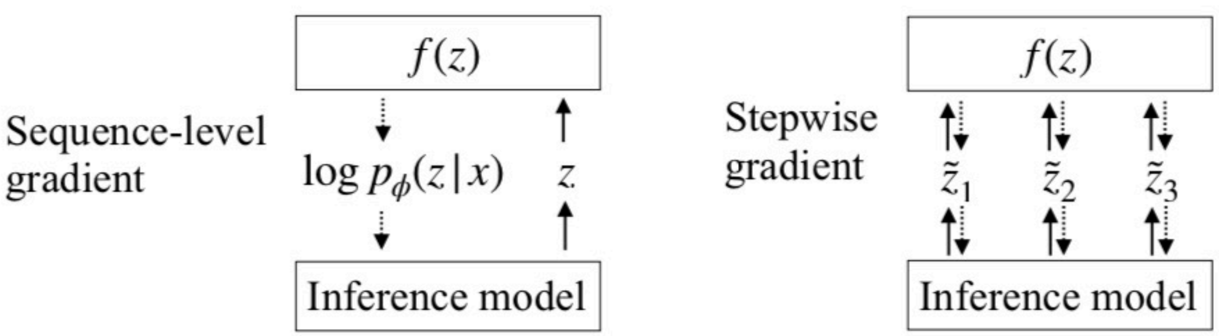
$$\hat{z}_t = \text{Argmax}\,(\tilde{z}_t)$$

- Apply Gumbel to each FFBS step to get soft sample $\tilde{z}_t$
- Use Argmax to recover hard sample $\hat{z}_t$

Gumbel-CRF

Algorithm 1 Forward Filtering Backward Sampling

1: **Input:** $\Phi(z_{t-1}, z_t, x_t), t \in \{1, .., T\}, \alpha_{1:T}, Z$
2: Calculate $p(z_T|x) = \alpha_T / Z$
3: Sample $\hat{z}_T \sim p(z_T|x)$
4: **for** $t \leftarrow T-1, 1$ **do**
5: $\quad p(z_t|\hat{z}_{t+1}, x) = \frac{\Phi(z_t, \hat{z}_{t+1}, x_{t+1})\alpha_t(z_t)}{\alpha_{t+1}(\hat{z}_{t+1})}$
6: $\quad$ Sample $\hat{z}_t \sim p(z_t|\hat{z}_{t+1}, x)$
7: **end for**
8: **Return** $\hat{z}_{1:T}$

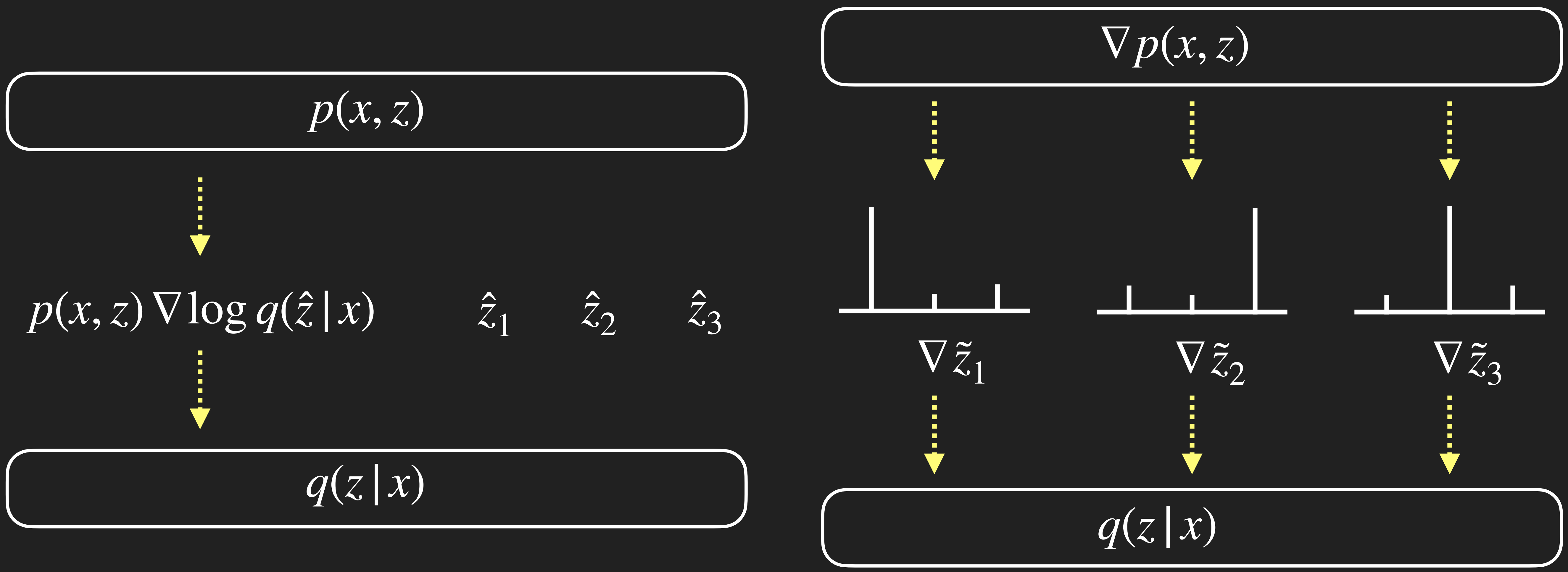


Algorithm 2 Gumbel-CRF (Forward Filtering Backward Sampling with Gumbel-Softmax)

1: **Input:** $\Phi(z_{t-1}, z_t, x_t), t \in \{1, .., T\}, \alpha_{1:T}, Z$
2: Calculate:
3: $\quad \pi_T = \alpha_T / Z$
4: $\quad \tilde{z}_T = \text{softmax}((\log \pi_T + g)/\tau), g \sim \text{G}(0)$
5: $\quad \hat{z}_T = \text{argmax}(\tilde{z}_T)$
6: **for** $t \leftarrow T-1, 1$ **do**
7: $\quad \pi_t = \frac{\Phi(z_t, \hat{z}_{t+1}, x_{t+1})\alpha_t(z_t)}{\alpha_{t+1}(\hat{z}_{t+1})}$
8: $\quad \tilde{z}_t = \text{softmax}((\log \pi_t + g)/\tau), g \sim \text{G}(0)$
9: $\quad \hat{z}_t = \text{argmax}(\tilde{z}_t)$
10: **end for**
11: **Return** $\hat{z}_{1:T}, \tilde{z}_{1:T}$ $\quad \triangleright$ $\tilde{z}$ is a relaxation for $\hat{z}$

- Conventional FFBS: each sample step (Alg. 1, line 3, 6) in FFBS is a categorical sample
- Gumbel-CRF: relax each categorical sample step w. Gumbel-Softmax (Alg. 2, line 4, 8)
- Recover exact hard sample: same as Gumbel-Max (Alg. 2, line 5, 9)
- Stepwise gradient: backprop. through each z_t v.s. seq level grad. in REINFORCE

Stepwise Gradients w. Gumbel-CRF



REINFORCE gives seq level grad

Gumbel-CRF induces stepwise grad

Alternative: REINFORCE

Estimators	Score /Reparam.	Seq. Level/ Stepwise	Unbiased MC Sample	Unbiased Grad.
REINFORCE-MS	Score	Seq.	Unbiased	Unbiased
REINFORCE-MS-C	Score	Seq.	Unbiased	Unbiased
PM-MRF	Reparam.	Step	Biased	Biased
PM-MRF-ST	Reparam.	Step	Biased	Biased
Gumbel-CRF	Reparam.	Step	Biased	Biased
Gumbel-CRF-ST	Reparam.	Step	Unbiased	Biased

(A) Characteristics of the estimators we compare

More estimators and detailed comparison in Appendix

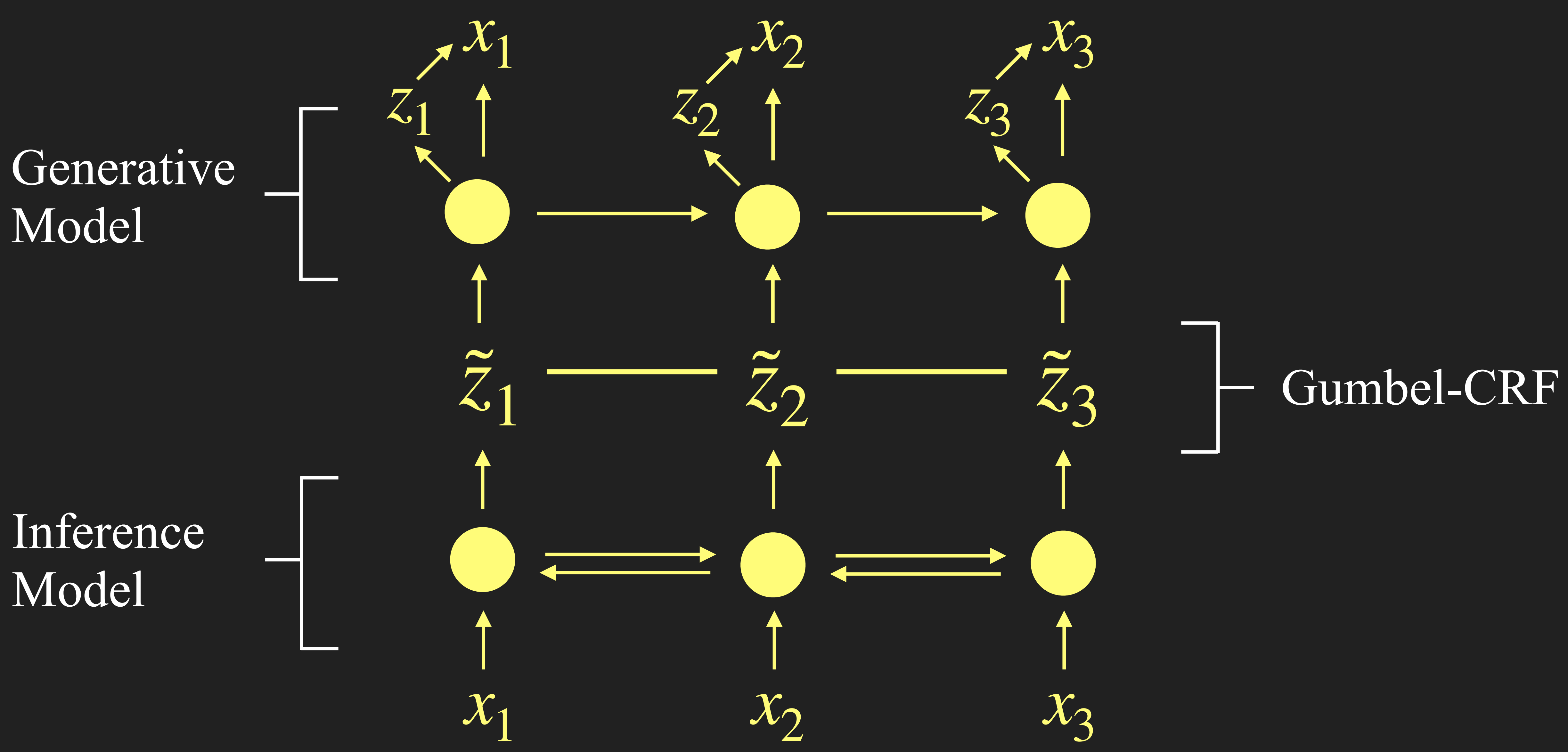
Gradient of REINFORCE:

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log p(x, z)] = \mathbb{E}_{q_\phi(z|x)}[\log p(x, z)\, \nabla_\phi \log q_\phi(z \mid x)]$$

Gradient of Gumbel-CRF:

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log p(x, z)] = \mathbb{E}_{q_\phi(z|x)}[\Sigma_i \nabla_{\tilde{z}_i} \log p(x, z) \odot \nabla_\phi \tilde{z}_i]$$

VAE w. Gumbel-CRF



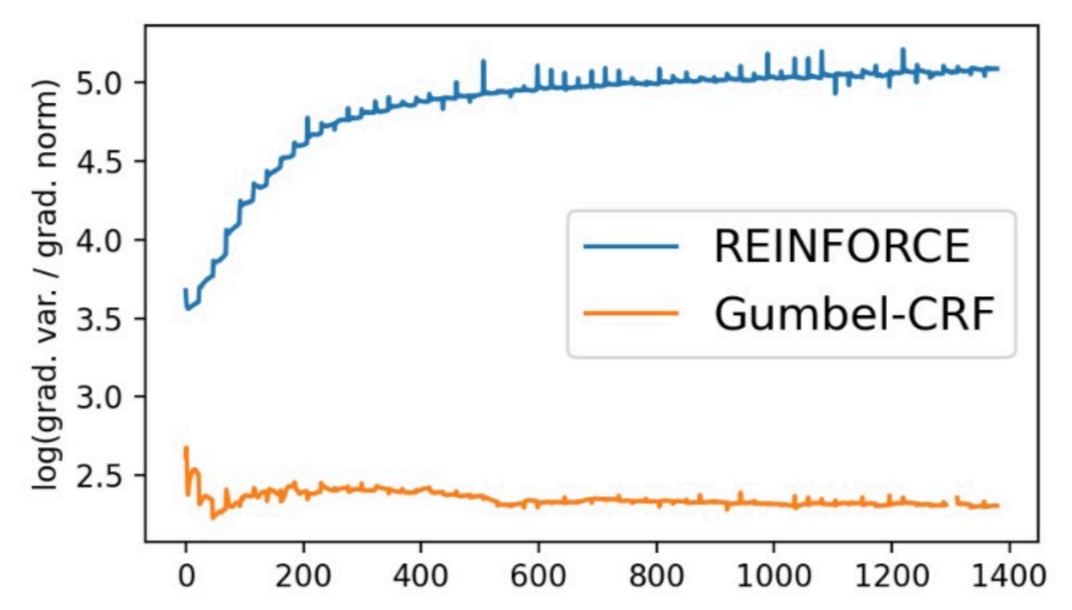
- Generative model autoregressive w.r.t. x and z (Li and Rush 2020)
- Inference model relaxed w. Gumbel-CRF

As a reparam.ed grad. estimator: density estimation

Table 1: Density Estimation Results. NLL is estimated with 100 importance samples. Models are selected from 3 different random seeds based on validation NLL. All metrics are evaluated on the discrete (exact) model.

Model	Neg. ELBO	NLL	PPL	Ent.	#sample
RNNLM	34.69	4.94	-	-	
PM-MRF	69.15	50.22	10.41	4.11	1
PM-MRF-ST	53.16	37.03	5.48	2.04	1
REINFORCE-MS	35.11	34.50	4.84	3.48	5
REINFORCE-MS-C	34.35	33.82	4.71	3.34	5
Gumbel-CRF (ours)	38.00	35.41	4.71	3.03	1
Gumbel-CRF-ST (ours)	34.18	33.13	4.54	3.26	1

- Gumbel-CRF ST version achieves best NLL and PPL w. less sample than baseline REINFORCE.



- Less variance than REINFORCE, more stable training

As a structured inference network: paraphrasing and data-to-text

Table 2: Paraphrase Generation. Upper: supervised models, Lower: unsupervised models. Models are selected from 5 random seeds based validation iB4.

Model	iB4	B2	B3	B4	R1	R2	RL
LBOW [15]	-	51.14	35.66	25.27	42.08	16.13	38.16
Gaussian VAE[7]	7.48	24.90	13.04	7.29	22.05	4.64	26.05
CGMH [40]	7.84	-	-	11.45	32.19	8.67	-
UPSA [36]	9.26	-	-	14.16	37.18	11.21	-
Ours trained w. REINFORCE	11.20	41.29	26.54	17.10	32.57	10.20	34.97
Ours trained w. Gumbel-CRF	10.20	38.98	24.65	15.75	31.10	9.24	33.60

Table 3: Data-to-text generation results. Upper: neural models, Lower: template-related models. Models are selected from 5 different random seeds based on validation BLEU.

Model	BLEU	NIST	ROUGE	CIDEr	METEOR
D&J[13]	65.93	8.59	68.50	2.23	44.83
KV2Seq[14]	74.72	9.30	70.69	2.23	46.15
SUB[13]	43.78	6.88	54.64	1.39	37.35
HSMM[62]	55.17	7.14	65.70	1.70	41.91
HSMM-AR[62]	59.80	7.56	65.01	1.95	38.75
SM-CRF PC [33]	67.12	8.52	68.70	2.24	45.40
Ours trained w. REINFORCE	60.41	7.99	62.54	1.78	38.04
Ours trained w. Gumbel-CRF	65.83	8.43	65.06	1.98	41.44

- Our model trained w. Gumbel-CRF performs similar to or better than REINFORCE
- Either way our model outperforms baselines

Practical Benefits and Controllable Generation

Table 4: Practical benefits of using Gumbel-CRF. Typically, REINFORCE has a long list of parameters to tune: h entropy regularization, b_0 constant baseline, b baseline model, r reward scaling, $\#s$ number of MC sample. Gumbel-CRF reduces the engineering complexity with significantly less parameters to tune (h entropy regularization, τ temperature annealing), less sample required (thus less memory consumption), and less time consumption. Models tested on Nvidia P100 with batch size 100.

Model	Hyperparams.	#s	GPU mem	Sec. per batch
REINFORCE	$h, b_0, b, r, \#s$	5	1.8G	1.42
Gumbel-CRF	h, τ	1	1.1G	0.48

name: clowns | eattype: coffee shop | food: chinese | customer_rating: 1 out of 5 | area: riverside | near: clare hall

1. [there is a]$_{20}$ [coffee shop]$_{35}$ [in the]$_{9}$ [riverside]$_{35}$ [area ,]$_{12}$ [serves]$_{20}$ [chinese]$_{35}$ [food]$_{12}$ [. it is]$_{20}$ [called]$_{35}$ [clowns]$_{44}$ [. is]$_{20}$ [near]$_{35}$ [clare hall]$_{44}$ [. It has a customer rating]$_{20}$ [of 1 out of 5]$_{8}$ [.]$_{20}$
2. [clowns]$_{44}$ [is a]$_{20}$ [expensive]$_{12}$ [coffee shop]$_{35}$ [located]$_{12}$ [in]$_{9}$ [riverside]$_{35}$ [area]$_{12}$ [.]$_{20}$
3. [clowns]$_{44}$ [is a]$_{20}$ [coffee shop]$_{35}$ [in the riverside]$_{9}$ [. it is]$_{20}$ [family friendly]$_{12}$ [and has a]$_{20}$ [1]$_{45}$ [out of 5]$_{8}$ [stars]$_{12}$ [rating .]$_{20}$

name: browns cambridge | eattype: coffee shop | food: chinese | customer_rating: 1 out of 5 | area: riverside | familyfriendly: yes | near: crowne plaza hote

1. [browns cambridge]$_{44}$ [offers]$_{12}$ [chinese]$_{35}$ [food]$_{12}$ [near]$_{35}$ [crowne plaza hotel]$_{44}$ [in]$_{35}$ [riverside]$_{9}$ [. it is a]$_{20}$ [coffee shop]$_{35}$ [, not children friendly]$_{12}$ [and has a]$_{20}$ [5]$_{45}$ [out of 5]$_{8}$ [rating .]$_{20}$
2. [there is a]$_{20}$ [moderately priced restaurant]$_{12}$ [that serves]$_{20}$ [chinese]$_{35}$ [food]$_{12}$ [called]$_{35}$ [browns cambridge]$_{44}$ [coffee]$_{9}$ [. it has a customer rating]$_{20}$ [of 5 out of 5.]$_{8}$ [it is]$_{20}$ [not family-friendly]$_{12}$ [. it is]$_{20}$ [located]$_{12}$ [near]$_{35}$ [crowne plaza]$_{44}$
3. [browns cambridge]$_{44}$ [is a]$_{20}$ [chinese coffee shop]$_{35}$ [located]$_{12}$ [in]$_{9}$ [riverside near]$_{35}$ [crowne plaza hotel]$_{44}$ [. it has a]$_{20}$ [customer rating]$_{20}$ [of 5 out of]$_{8}$ [5]$_{44}$ [and is]$_{20}$ [not family-friendly]$_{12}$ [.]$_{20}$

Figure 4: Controllable generation with templates.

Template Interpretability

Bigram	Sentence Segments	4gram	Sentence Segments	
(A) 12-35	1. located near	(D) 35-44-12-20	1. near the city center	ngrams w. semantically similar segments
	2. restaurant near		2. near café rouge, there is a	
	3. restaurant located near		3. in the city center, it is	
(B) 20-8	1. has a customer rating of	(E) 44-20-35-20	1. french food at a moderate	
	2. has a customer rating of 5 out of		2. french food for a moderate	
	3. and with a customer rating of		3. fast food restaurant with a moderate	
(C) 20-12	1. is located	(F) 12-20-12-20	1. food with a price range of	ngrams w. semantically different segments
	2. is a family friendly		2. price range and family friendly	

Figure 5: Analysis of state ngrams. State ngrams correlate to sentence meaning. In cases (A, B, D, E), semantically similar sentence segments are clustered to the same state ngrams: (A) "location" (B) "rating" (D) "location" (E) "food" and "price". Yet there are also cases where state ngrams correspond to sentence segments with different meaning: (C1) "location" v.s. (C2) "comments"; (F1) "price" v.s. (F2) "price" and "comments".

Conclusion

Learning latent templates for controllable text generation w. Gumbel-CRFs

Faster, more stable training than REINFORCE w. better density estimation

Models trained w. Gumbel-CRF perform similar to or better than same model trained w. REINFORCE w.r.t. end performance, interpretability and controllability

Applicable to models involving middle-layer CRFs, anther interesting direction would be joint entity recognition and relation classification

Code: https://github.com/FranxYao/Gumbel-CRF

Thanks!