Wouter Kool, Herke van Hoof, Max Welling

GUMBEL Mathemagic

University of Amsterdam

ORTEC
OPTIMIZE YOUR WORLD

AMLAB
Amsterdam
Machine Learning Lab

- "This is how you randomize a beam search!"

Wouter Kool, Herke van Hoof, Max Welling

*- "You will **never** have duplicate samples again!"*

STOCHASTIC BEAMS

AND WHERE TO FIND THEM

The Gumbel-Top-*k* Trick for Sampling Sequences Without Replacement

★★★★★

ICML 2019
BEST PAPER
HONORABLE MENTION

University of Amsterdam
ORTEC
OPTIMIZE YOUR WORLD
AMLAB
Amsterdam Machine Learning Lab

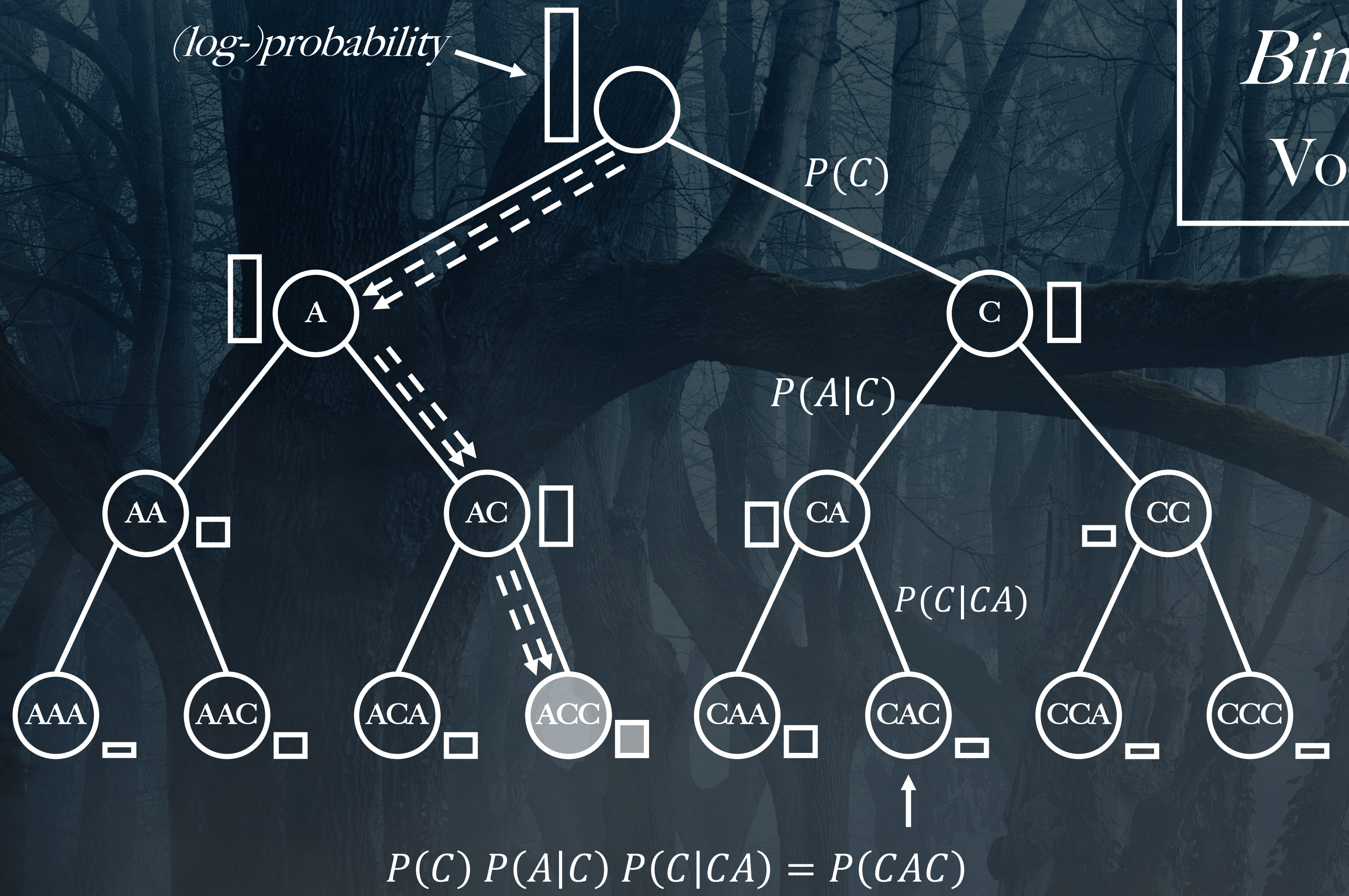
TL;DR

Stochastic Beam Search finds a set of unique samples (without replacement) from a sequence model.

Example

Binarese language model

Vocabulary: {**A**bra, **C**adabra}



*What if we want a **sample** from our model?*

The Gumbel-Max Trick

"Prof. Gumbeldore"

(Gumbel, 1945; Maddison et al., 2014)

$\phi_i = \log p_i$ $\quad$ $G_i \sim \mathrm{Gumbel}(0)$ $\quad$ $G_{\phi_i} \sim \mathrm{Gumbel}(\phi_i)$

log-probability $\quad$ Gumbel noise $\quad$ *perturbed log-probability*

The Gumbel-Max Trick

"Prof. Gumbeldore"

(Gumbel, 1945; Maddison et al., 2014)

$$\max_i G_{\phi_i} \sim \text{Gumbel}\left(\log \sum_i \exp \phi_i\right)$$

max and argmax are *independent*

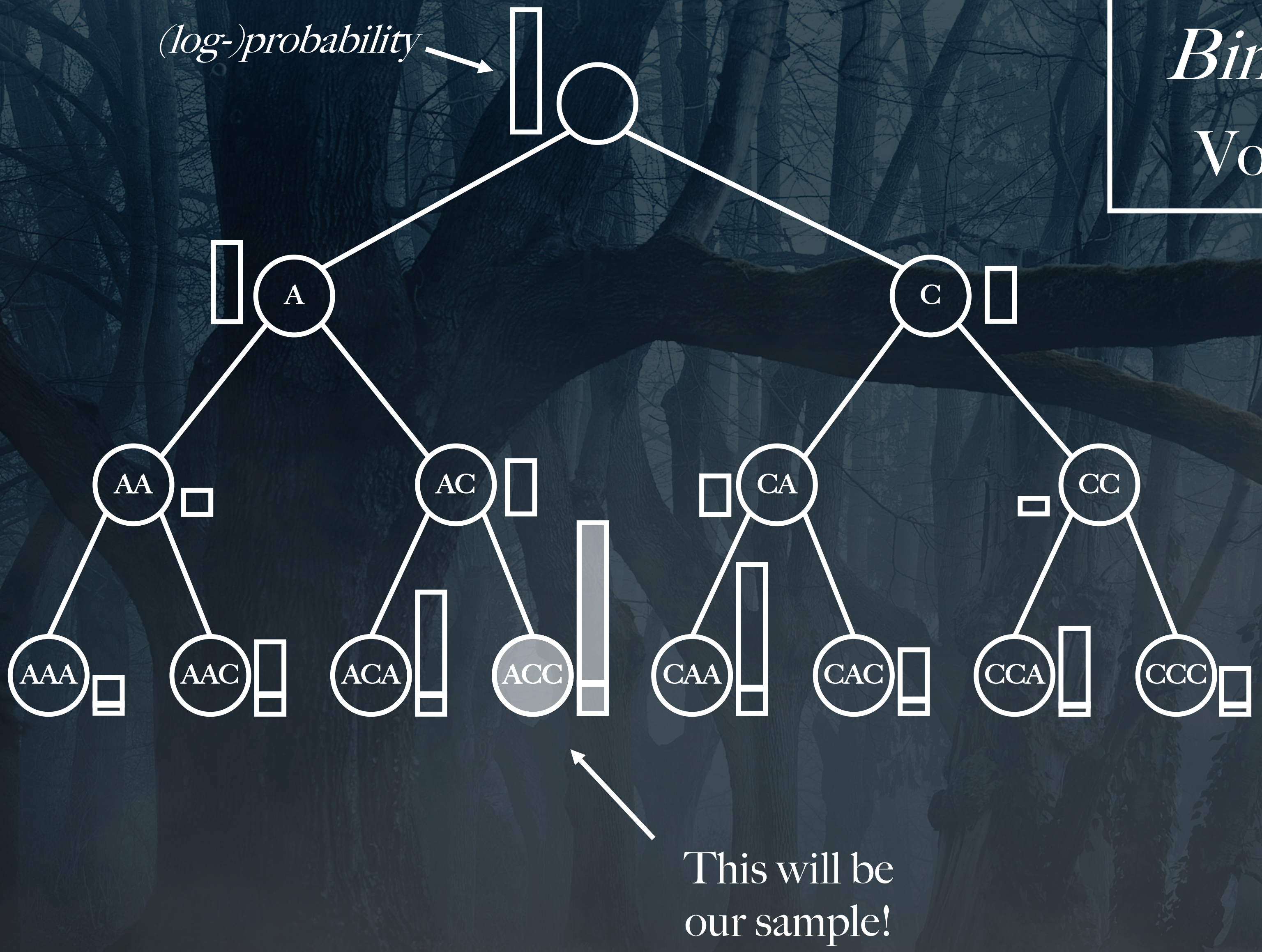
$$I = \operatorname*{argmax}_i G_{\phi_i} \sim \text{Categorical}(p_i)$$

$$P(I = i) = p_i$$

Example

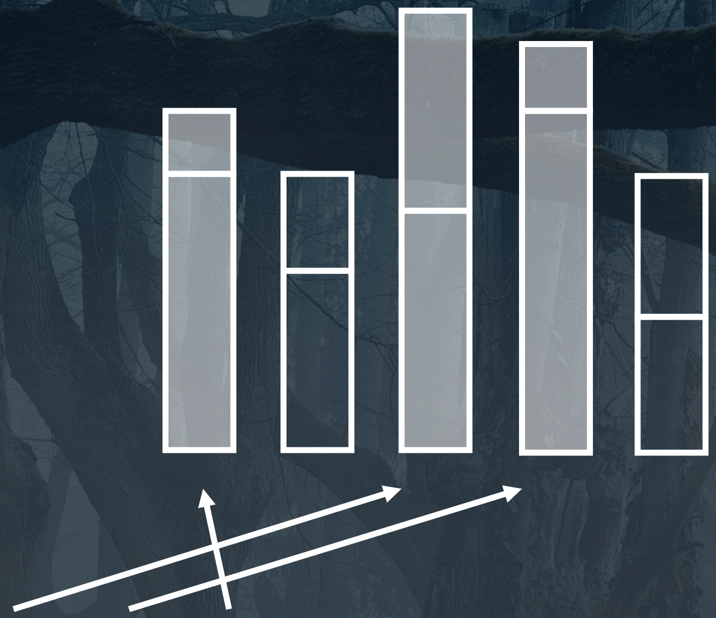
Binarese language model

Vocabulary: {**A**bra, **C**adabra}



*What if we want a **sample** from our model?*

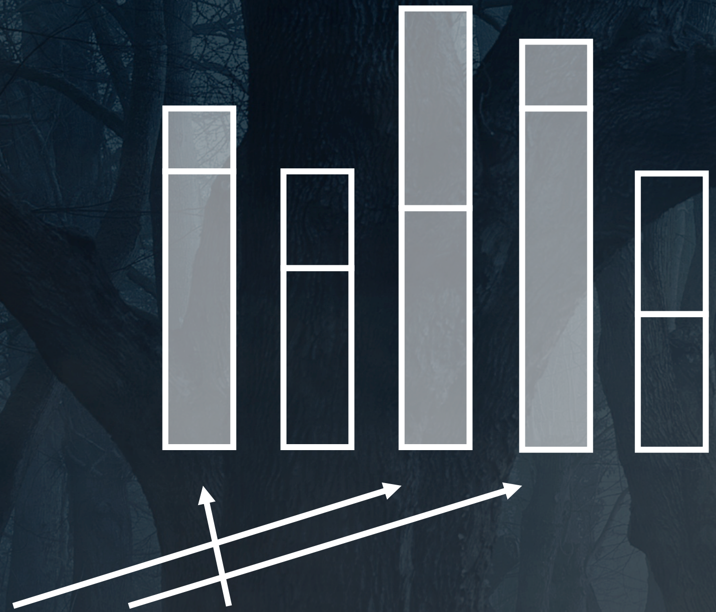
What happens if, instead of 1 (one), we take the k largest elements (top k)?



$k = 3$

$$I_1, \ldots, I_k = \arg \operatorname*{top}_{i} k \, G_{\phi_i}$$

The 'Gumbel-Top-k' Trick



$$I_1, \dots, I_k = \arg \operatorname*{top}_i k\, G_{\phi_i}$$

$$P(I_1 = i_1, \dots, I_k = i_k)$$
$$= p_{i_1} \cdot \frac{p_{i_2}}{1 - p_{i_1}} \cdot \dots \cdot \frac{p_{i_k}}{1 - \sum_{\ell=1}^{k-1} p_{i_\ell}}$$
$$= \prod_{j=1}^{k} \frac{p_{i_j}}{1 - \sum_{\ell=1}^{j-1} p_{i_\ell}}$$

Also known as Plackett-Luce

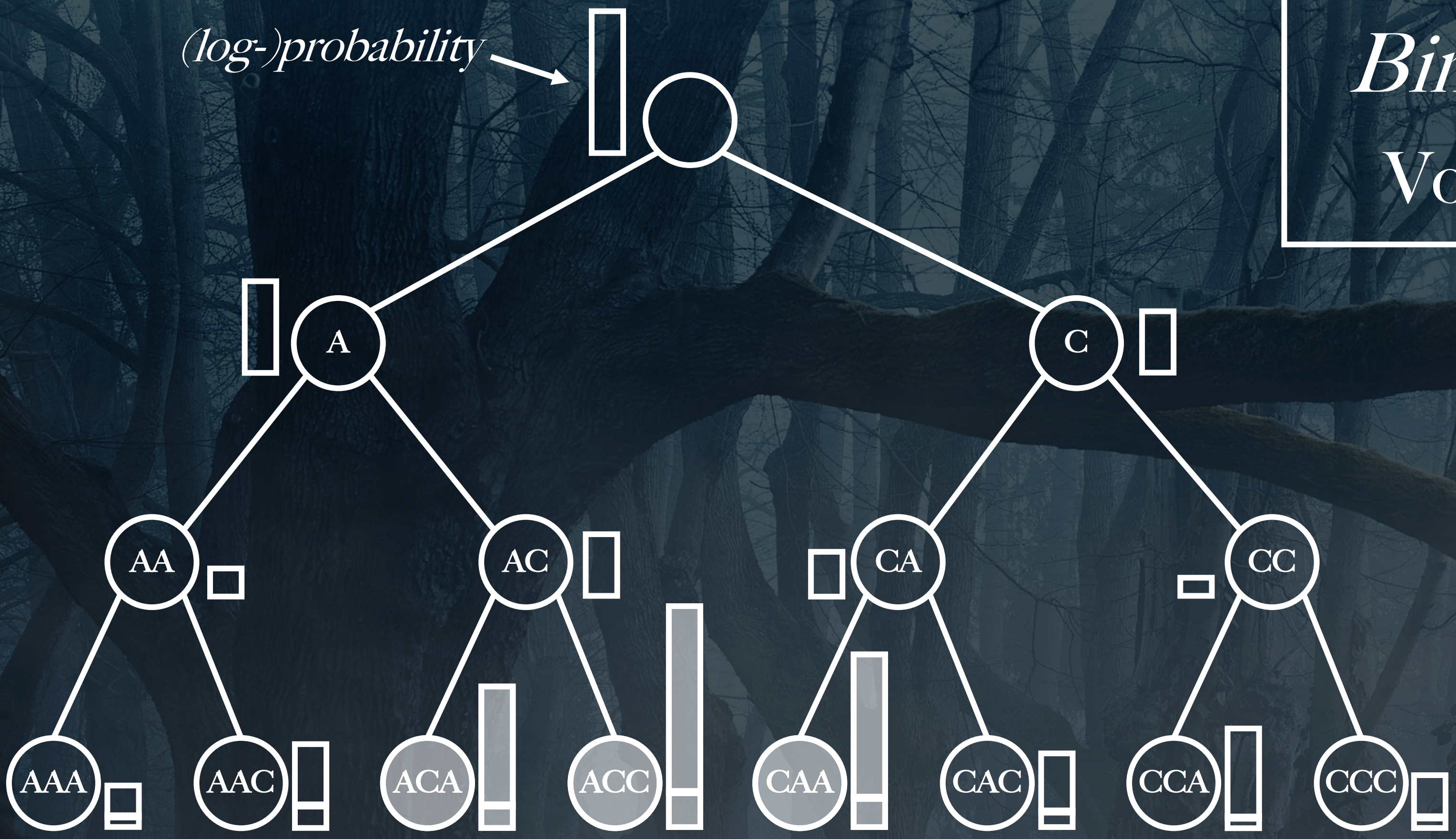
This is equivalent to repeated sampling without replacement!

(Vieira, 2014)

Example

Binarese language model

Vocabulary: {**A**bra, **C**adabra}



This will be our set of samples!

We can get a set of unique samples from our model!

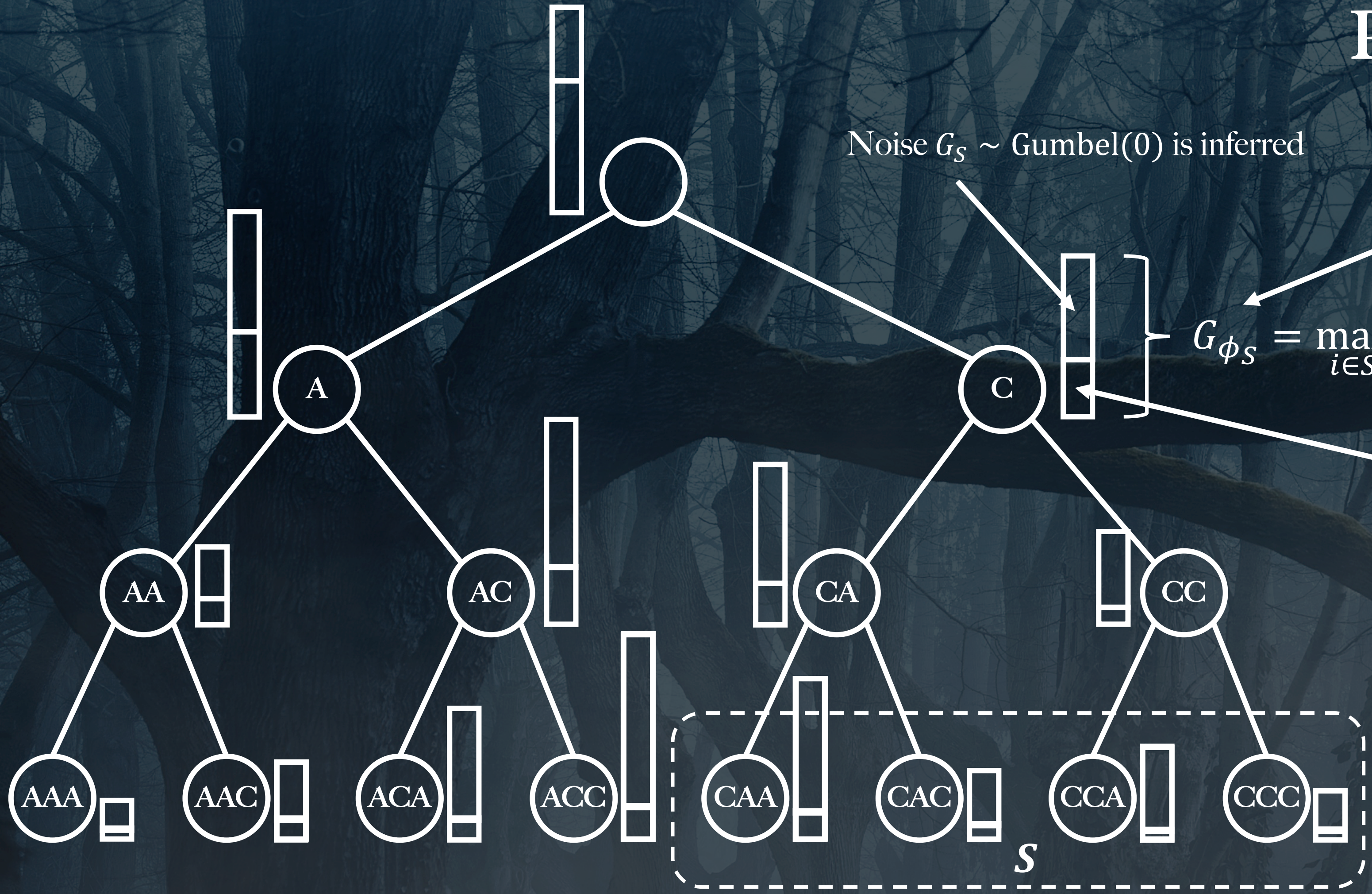
PROBLEM

In general, constructing the full tree is not possible...

... but we don't have to!

Perturbed log-probability of partial sequence (“C”)

Noise $G_S \sim \text{Gumbel}(0)$ is inferred



$$G_{\phi_S} = \max_{i \in S} G_{\phi_i} \sim \text{Gumbel}\left(\log \sum_{i \in S} \exp \phi_i\right)$$

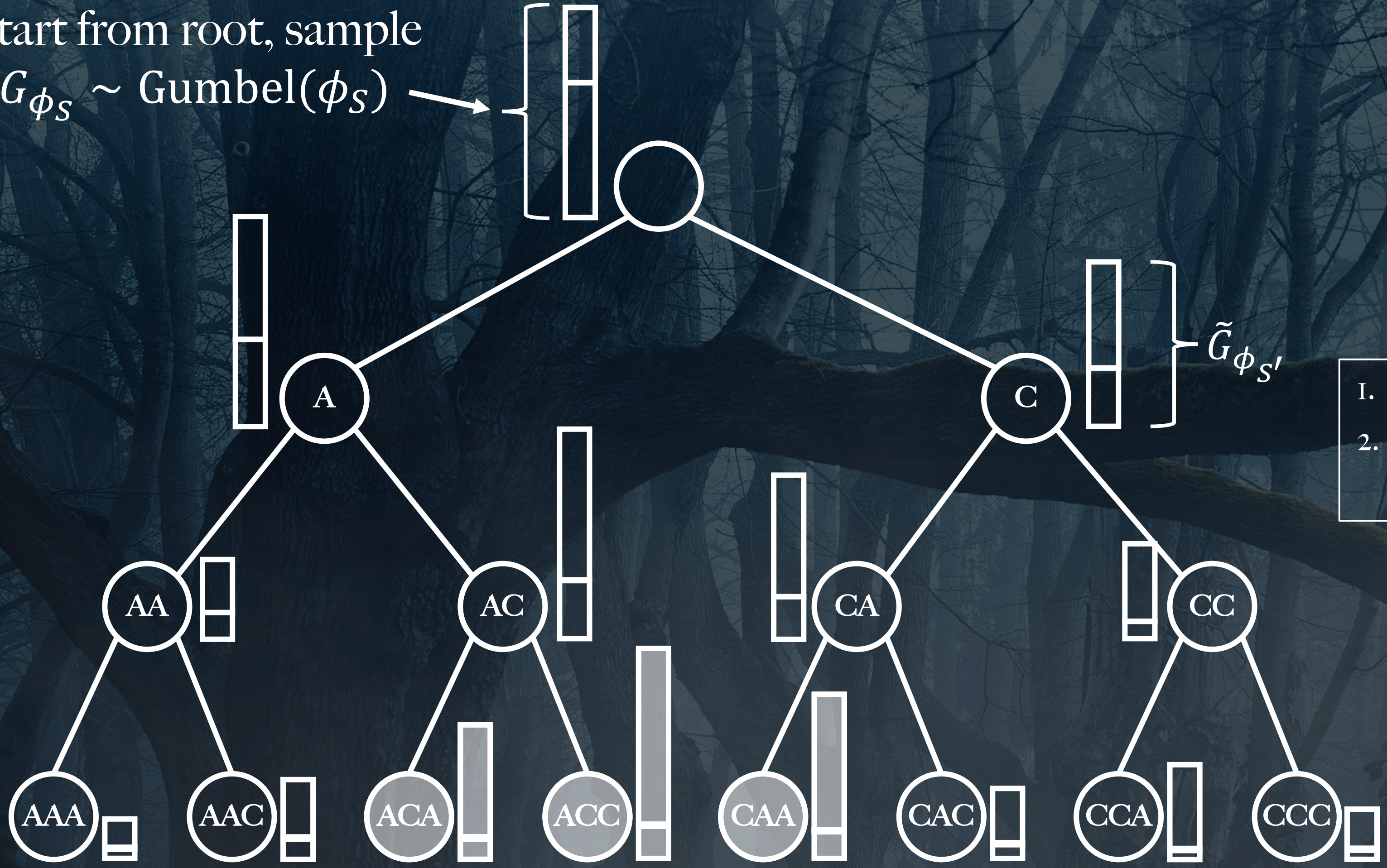
ϕ_S = log-probability of “C”

We can sample
$G_{\phi_S} \sim \text{Gumbel}(\phi_S)$
directly

Look at maximum of perturbed log-probabilities in subtree

Top-down sampling

Start from root, sample $G_{\phi_S} \sim \text{Gumbel}(\phi_S)$



Sample children $G_{\phi_{S'}}$ *conditionally on*

$$\max_{S' \in \text{Children}(S)} G_{\phi_{S'}} = G_{\phi_S}$$

1. sample $G_{\phi_{S'}}$ independently, compute $Z = \max_{S'} G_{\phi_{S'}}$
2. 'shift' Gumbels in (negative) exponential space: $\tilde{G}_{\phi_{S'}} = -\log\left(\exp(-G_{\phi_S}) - \exp(-Z) + \exp\left(-G_{\phi_{S'}}\right)\right)$

... the result is **equivalent** to sampling G_{ϕ_i} for leaves directly!

(Maddison et al., 2014)

We only need to expand the top k nodes at each level in the tree

Threshold

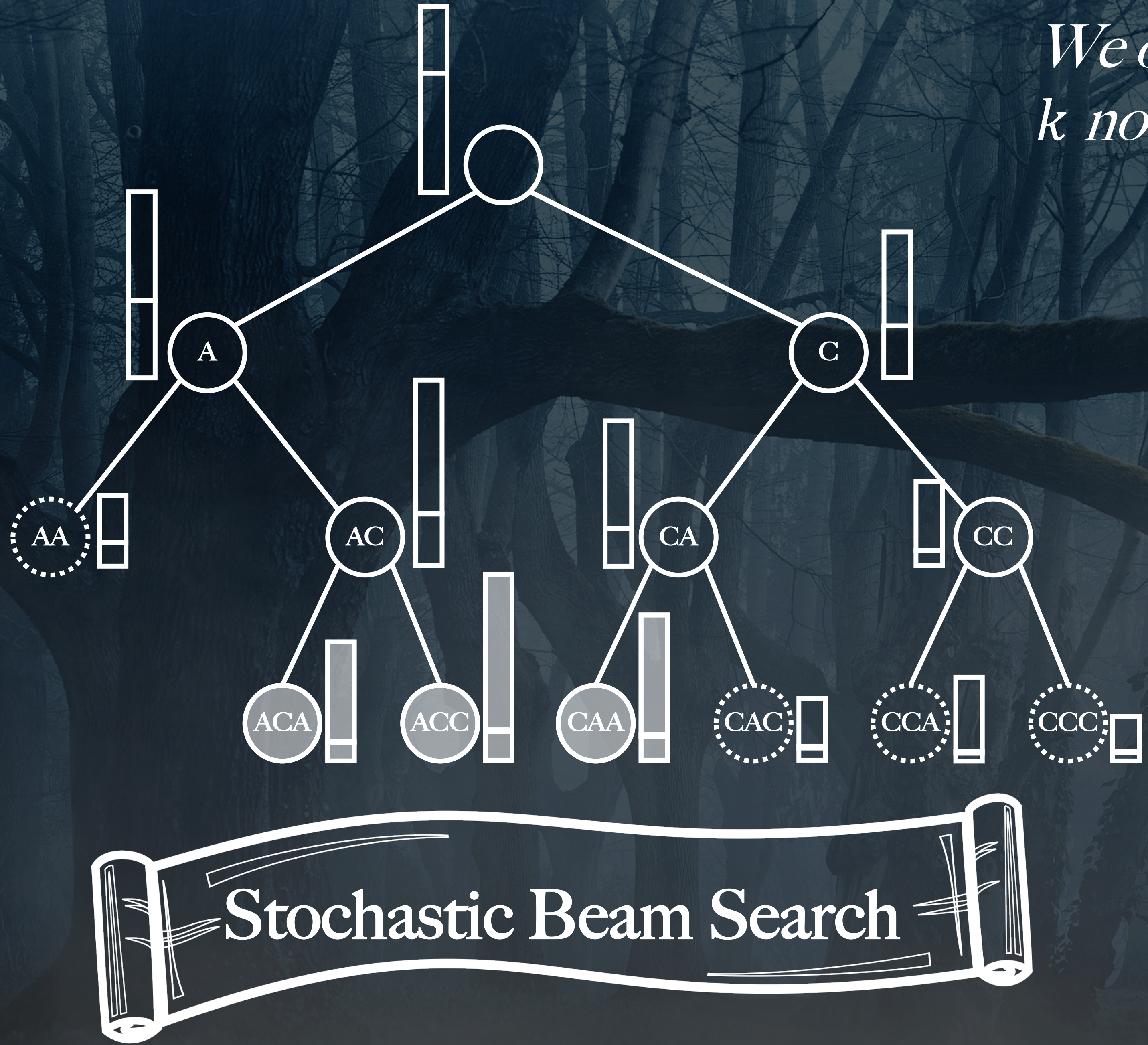
Each top k node generates (at least) one leaf (maximum) above threshold

At least k leafs will be above threshold

Other nodes only generate leafs below threshold

No need to expand

The Key Insight



We only need to expand the top k nodes at each level in the tree

↑

This is a *beam search*

Top k according to perturbed log-probability

= ← Gumbel-Top-k trick

Sampling (without replacement)

Stochastic Beam Search

- A beam search that *samples* the nodes to expand

Important!

- But... samples children *conditionally* on parent

- The result is a sample without replacement from the full sequence model

- Is a generalization of ancestral sampling ($k = 1$)

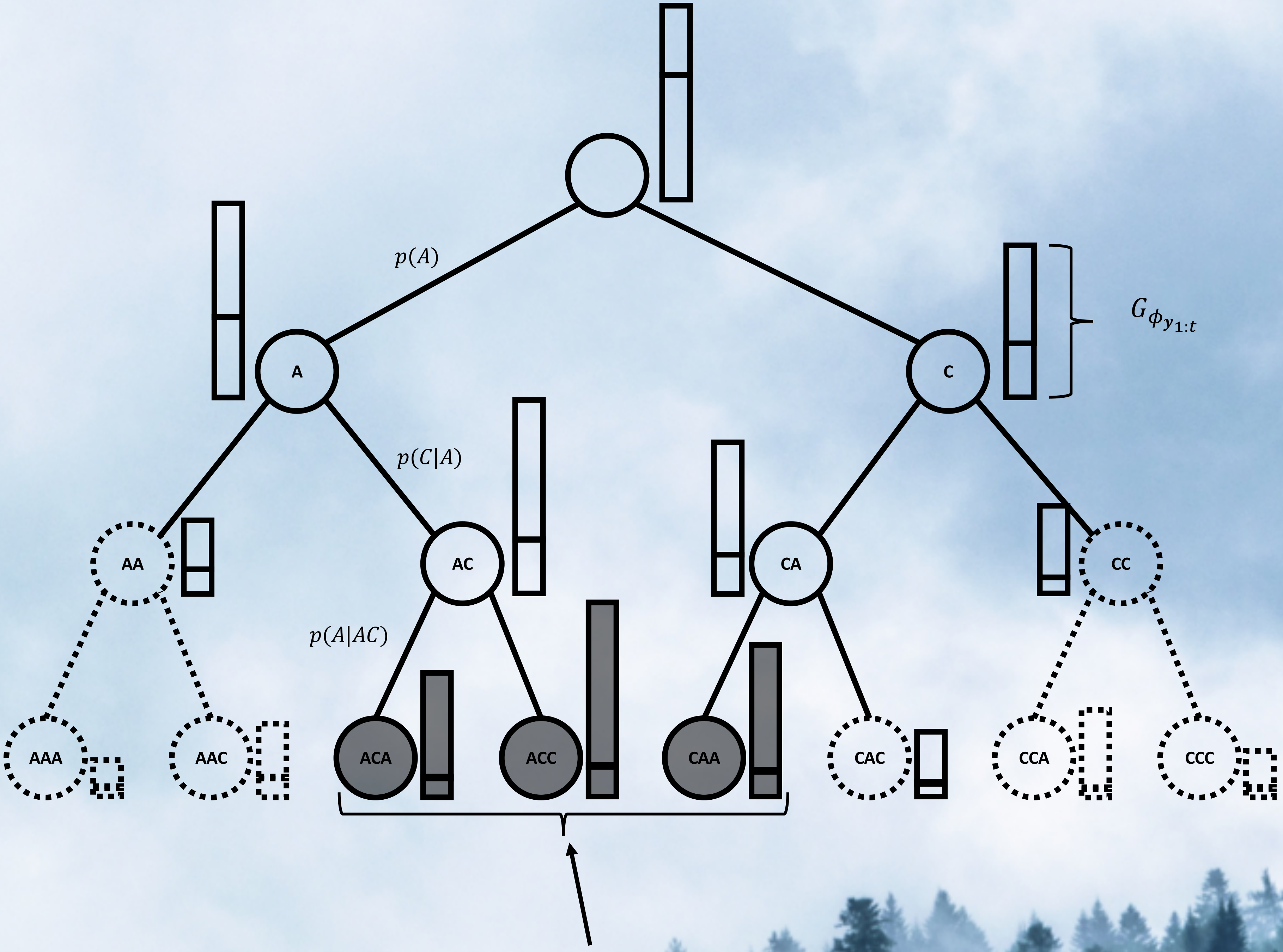
Ancestral Gumbel-Top-k Sampling

Ancestral Gumbel-Top-k Sampling

Generalizes Stochastic Beam Search

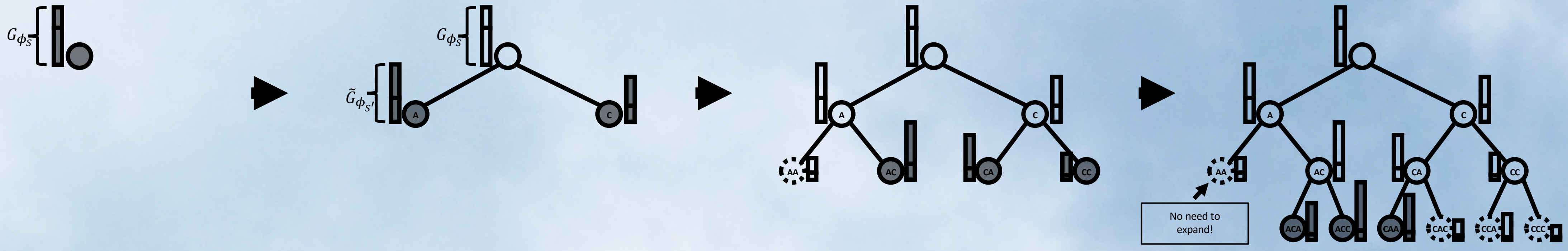
Expands $1 \leq m \leq k$ nodes per iteration

Applies to discrete valued Bayes networks

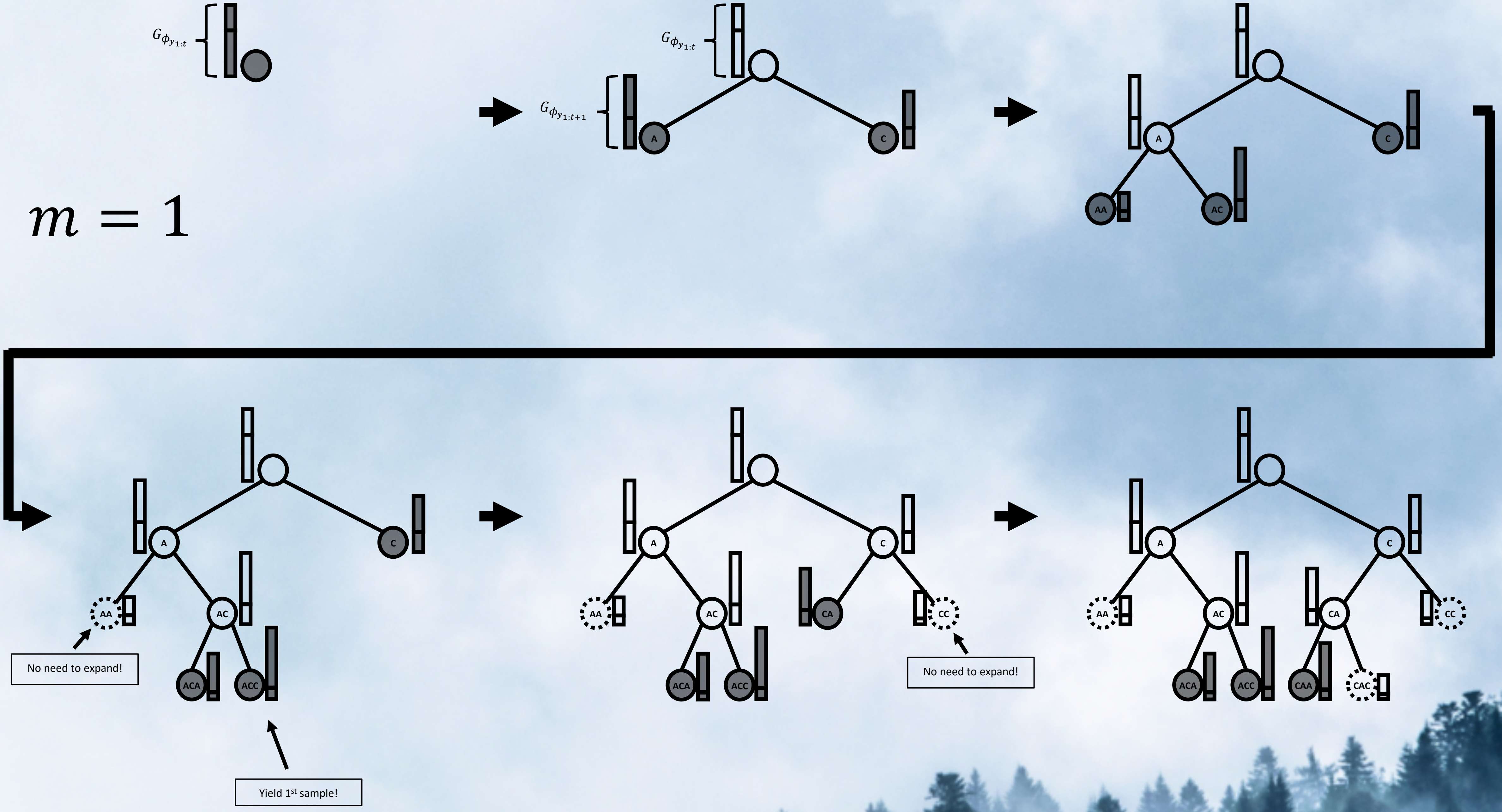


$p(A)$
$G_{\phi_{y_{1:t}}}$
A
C
$p(C|A)$
AA
AC
CA
CC
$p(A|AC)$
AAA
AAC
ACA
ACC
CAA
CAC
CCA
CCC

Stochastic Beam Search

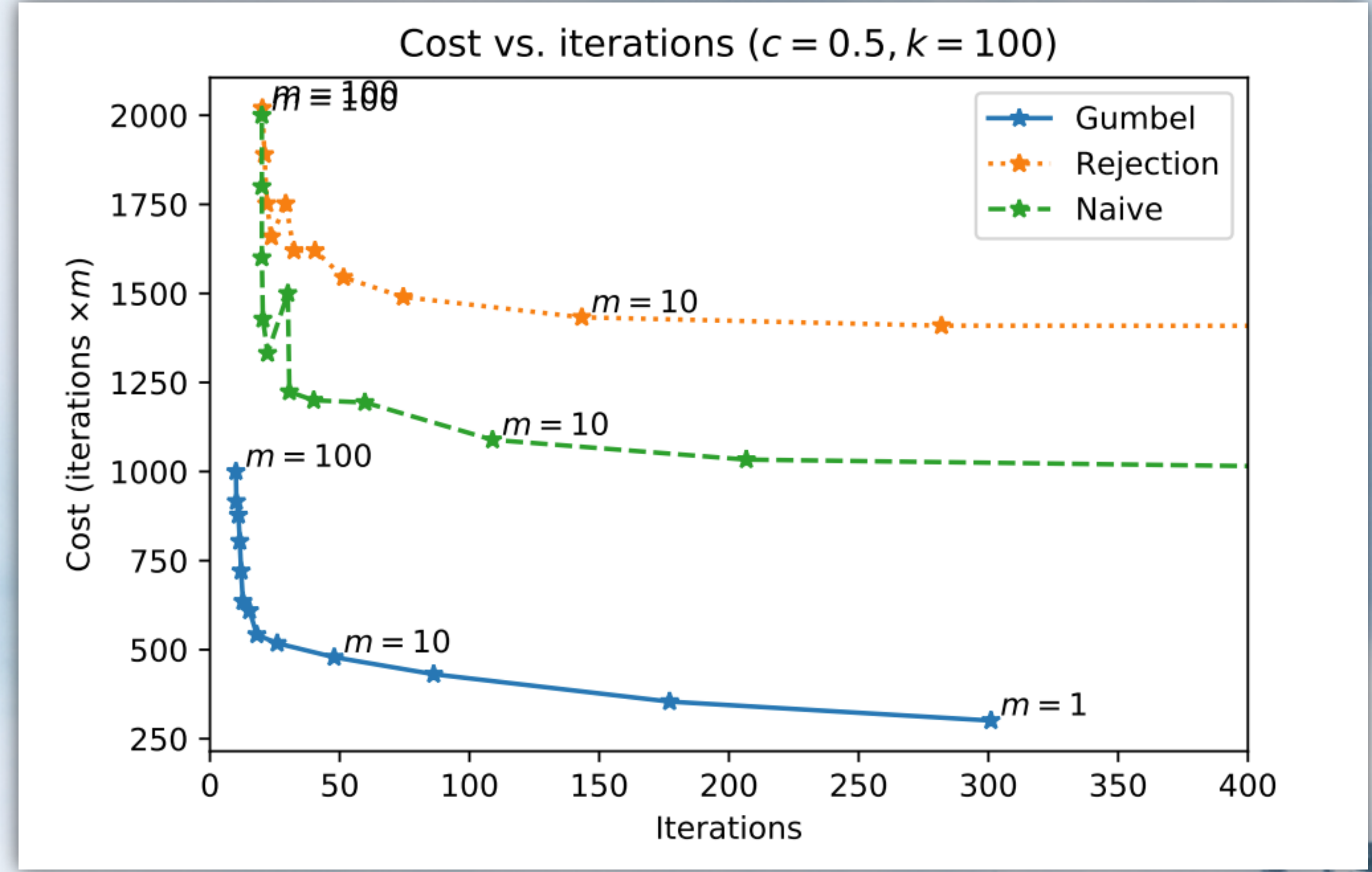


$$m = k \ (= 3)$$


$m = 1$
$G_{\phi_{y_{1:t}}}$
$G_{\phi_{y_{1:t+1}}}$
A
C
AA
AC
ACA
ACC
CA
CC
CAA
CAC
No need to expand!
Yield 1st sample!
No need to expand!

Ancestral Gumbel-Top-*k* Sampling

$m = 1$		$m = k$
Sequential	⟷	Parallel
Incremental		Batch
More iterations		Fewer iterations
Less computation		More computation



Cost vs. iterations ($c = 0.5, k = 100$)
Gumbel
Rejection
Naive
$m = 100$
$m = 100$
$m = 10$
$m = 10$
$m = 100$
$m = 10$
$m = 1$
Cost (iterations $\times m$)
Iterations

Ancestral Gumbel-Top-k Sampling

Generalizes Stochastic Beam Search

Expands $1 \leq m \leq k$ nodes per iteration

Applies to discrete valued Bayes networks

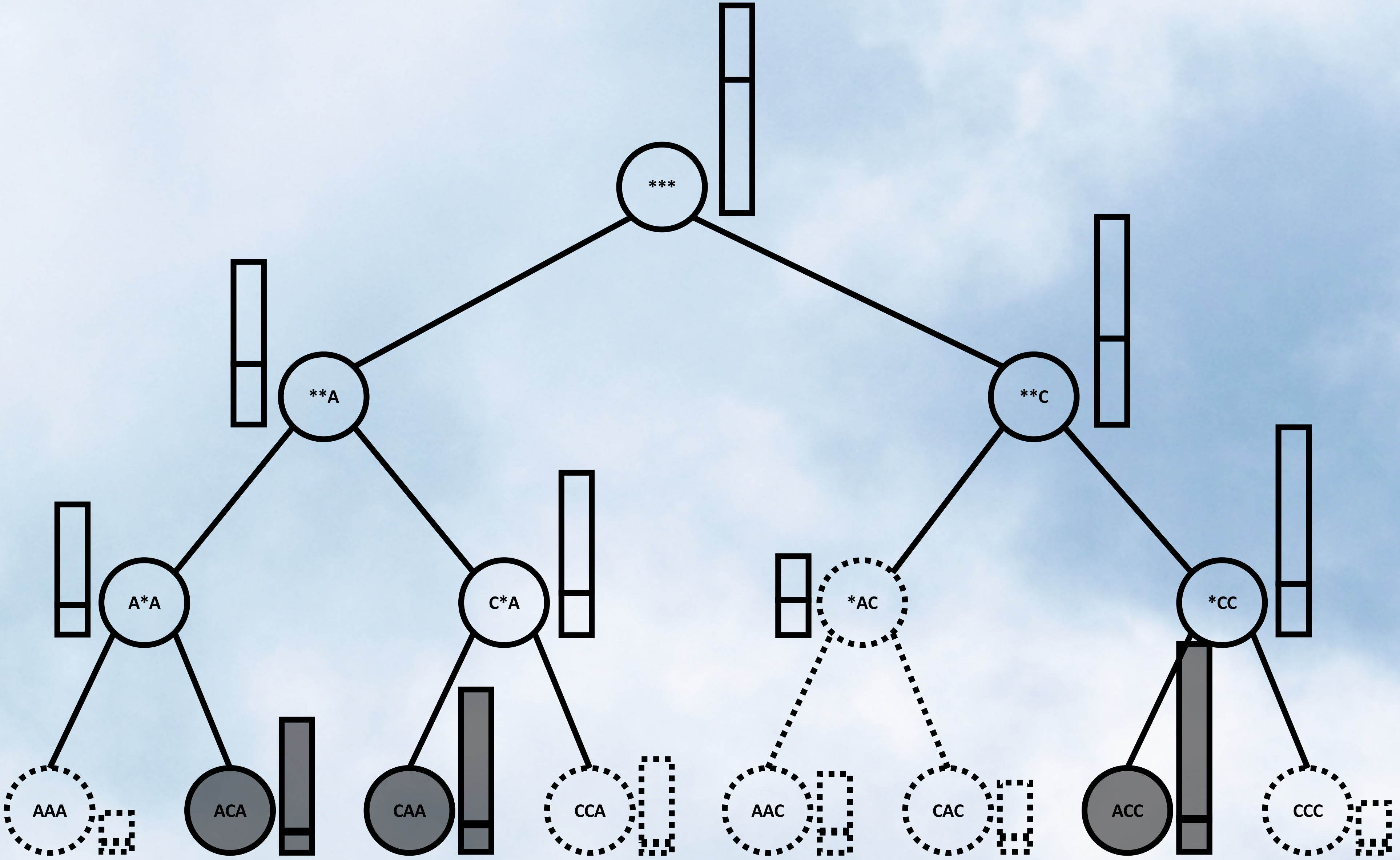
(a) Independent
$p(\boldsymbol{y}) = \prod_{v} p(y_v)$

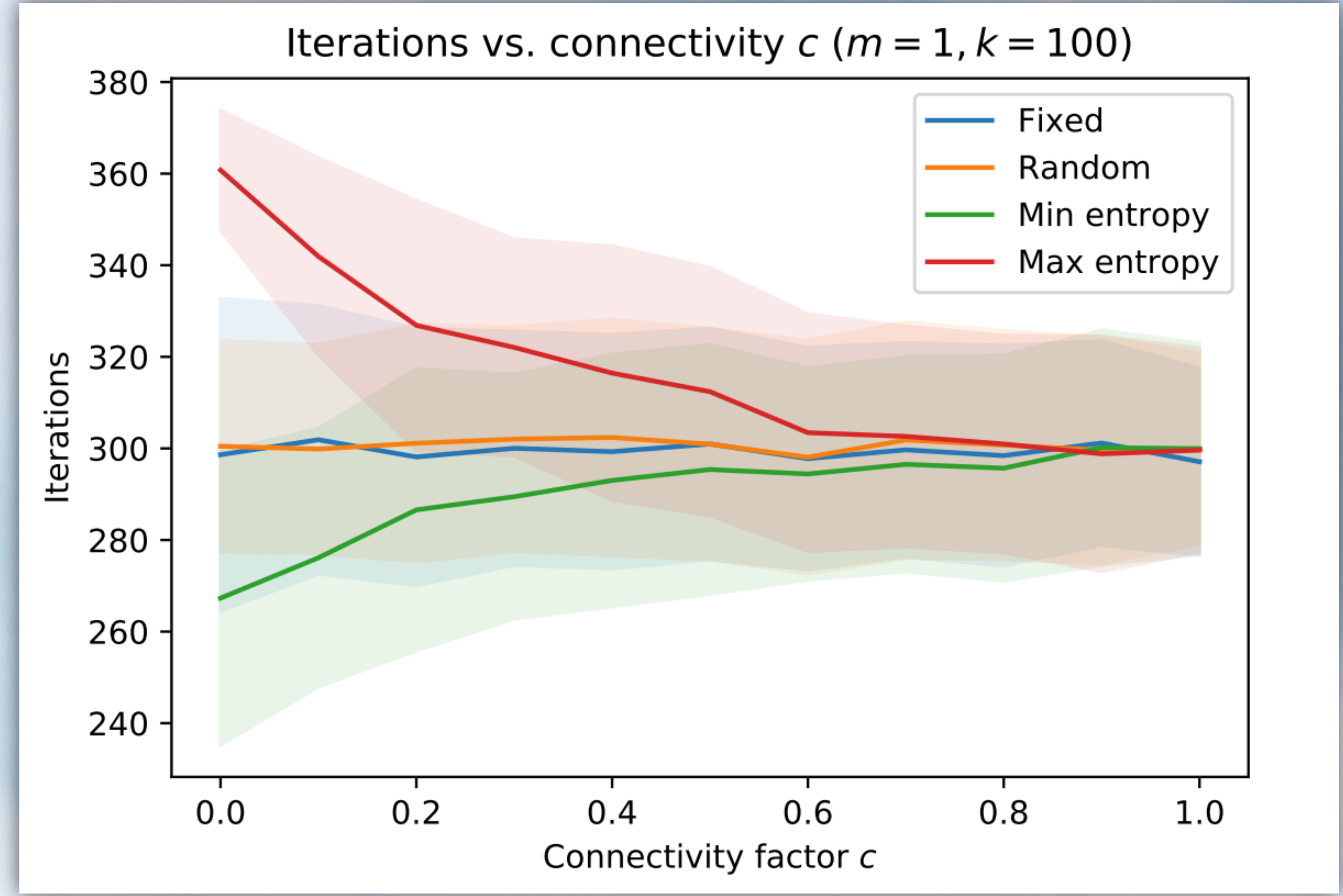
(b) Markov chain
$p(\boldsymbol{y}) = p(y_1) \prod_{t>1} p(y_t | y_{t-1})$

(c) General network
$p(\boldsymbol{y}) = \prod_{v} p\left(y_v | \boldsymbol{y}_{\mathrm{pa}(v)}\right)$

(d) Sequence model
$p(\boldsymbol{y}) = \prod_{t} p\left(y_t | \boldsymbol{y}_{1:t-1}\right)$

Figure 1: Examples of Bayesian networks.


***
**A
**C
A*A
C*A
*AC
*CC
AAA
ACA
CAA
CCA
AAC
CAC
ACC
CCC

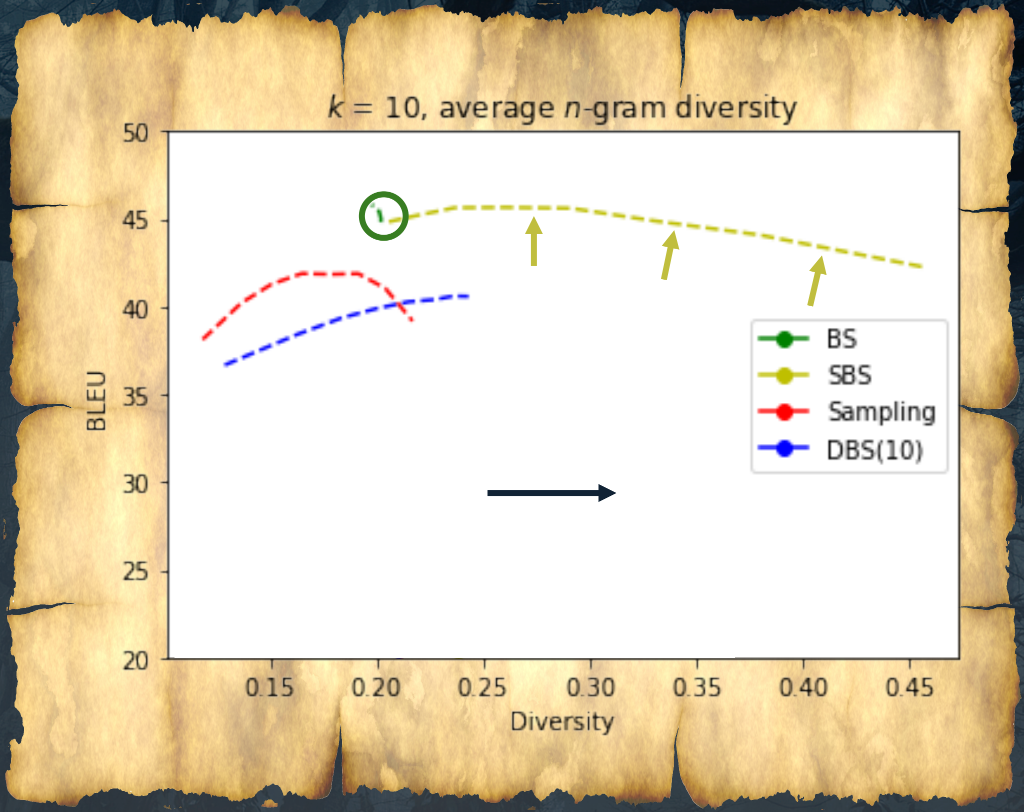

Iterations vs. connectivity c ($m = 1, k = 100$)
Iterations
Connectivity factor c
380
360
340
320
300
280
260
240
0.0
0.2
0.4
0.6
0.8
1.0
Fixed
Random
Min entropy
Max entropy

Is this useful?

Experiments

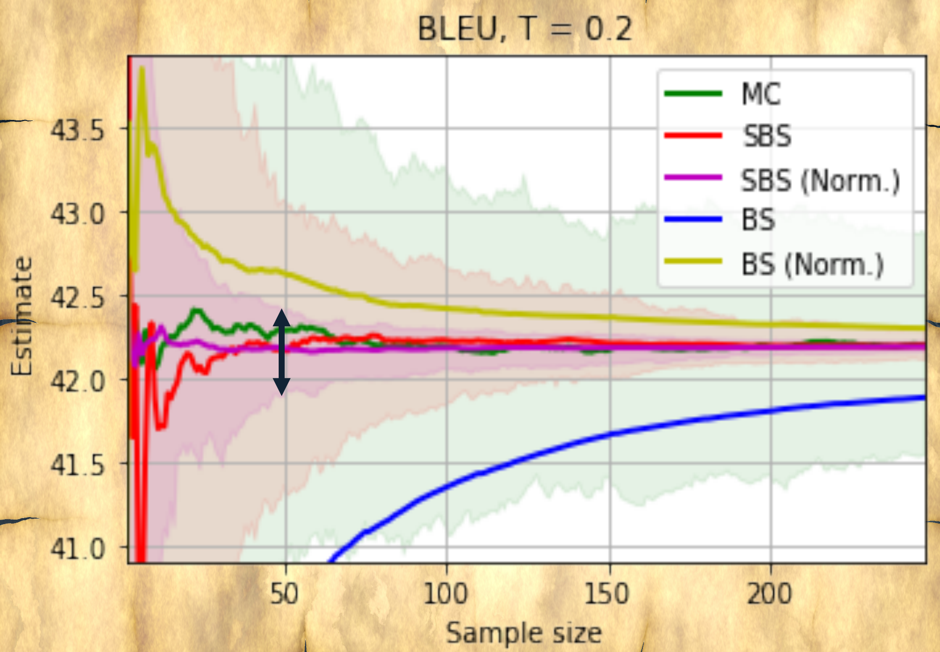
Translation Diversity

- Generate *k* translations
- Plot BLEU against diversity
- Vary softmax temperature
- Compare:
  - Beam Search
  - Stochastic Beam Search
  - Sampling
  - Diverse Beam Search (Vijayakumar et al., 2018)



BLEU Score Estimation

- Estimate expected sentence-level BLEU
- Plot mean and 95% interval vs. num samples
- Compare:
  - Monte Carlo Sampling
  - Stochastic Beam Search with (normalized) Importance Weighted estimator
  - Beam Search with deterministic estimate



Can we use it for training?

Estimating Gradients for Discrete Distributions by Sampling Without Replacement

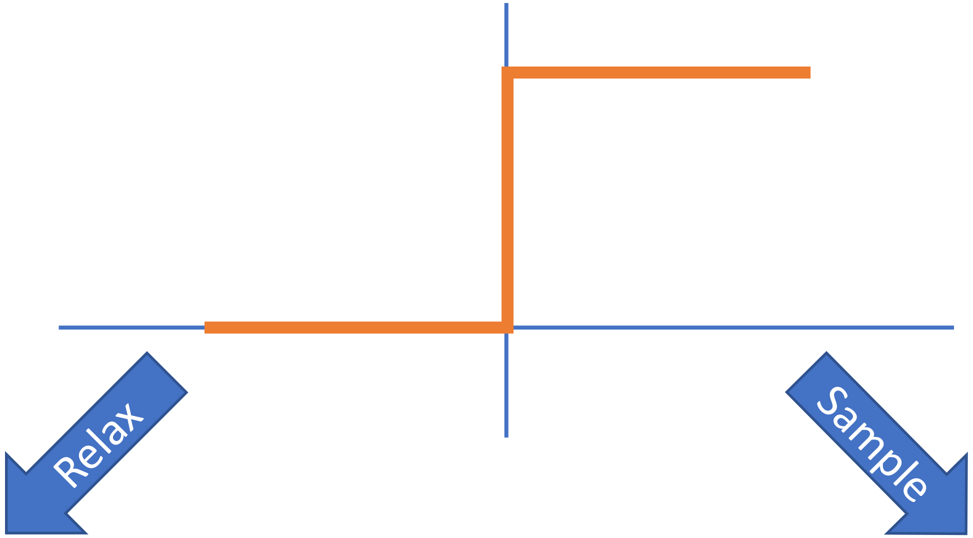
Wouter Kool, Herke van Hoof & Max Welling

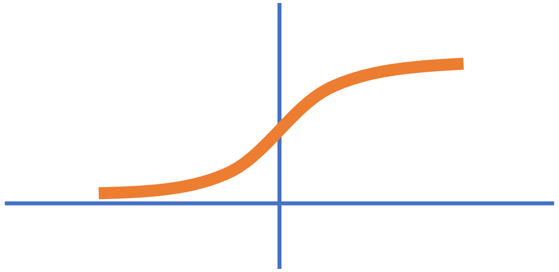
International Conference on Learning Representations (ICLR) 2020

Problems of discrete nature

- Reinforcement Learning
- Machine Translation / Image Captioning
- Discrete Latent Variable Modelling
- (Hard) Attention

Gradient of discrete operation





Biased

"REINFORCE"

$$\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)\, f(x)$$

High variance

REINFORCE

$$\nabla_{\boldsymbol{\theta}} E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(x)}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)\, f(x)]$$

REINFORCE

$$\nabla_{\boldsymbol{\theta}} E_{p_{\boldsymbol{\theta}}(x)}[f(x)] \approx \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)\, f(x)$$

REINFORCE with multiple samples

$$\nabla_{\boldsymbol{\theta}} E_{p_{\boldsymbol{\theta}}(x)}[f(x)] \approx \frac{1}{k} \sum_{i=1}^{k} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x_i)\, f(x_i)$$

REINFORCE with baseline

$$\nabla_{\boldsymbol{\theta}} E_{p_{\boldsymbol{\theta}}(x)}[f(x)] \approx \frac{1}{k} \sum_{i=1}^{k} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x_i) \left(f(x_i) - \underbrace{\frac{\sum_{j \neq i} f(x_j)}{k-1}}_{\text{Baseline}} \right)$$

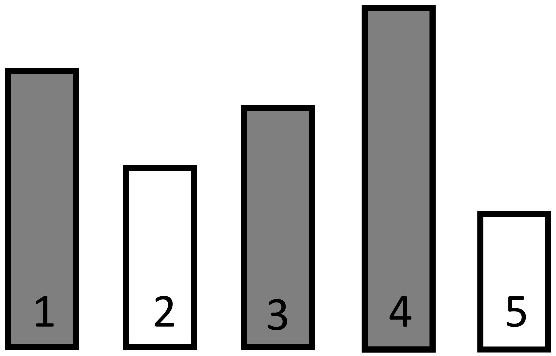
Mnih & Rezende, 2016

Sampling *without* replacement

Since duplicate samples are uninformative!

*In a deterministic setting

Sampling without replacement



$B = (3, 4, 1)$

$$p(B) = p(b_1) \times \frac{p(b_2)}{1 - p(b_1)} \times \frac{p(b_3)}{1 - p(b_1) - p(b_2)}$$

Ordered samples without replacement

$$p(B) = \prod_{i=1}^{k} \frac{p(b_i)}{1 - \sum_{j<i} p(b_j)}$$

Sequence $B = (3,4,1)$

Unordered samples without replacement

$$p(B) = \prod_{i=1}^{k} \frac{p(b_i)}{1 - \sum_{j<i} p(b_j)}$$

Set $S = \{1,3,4\}$

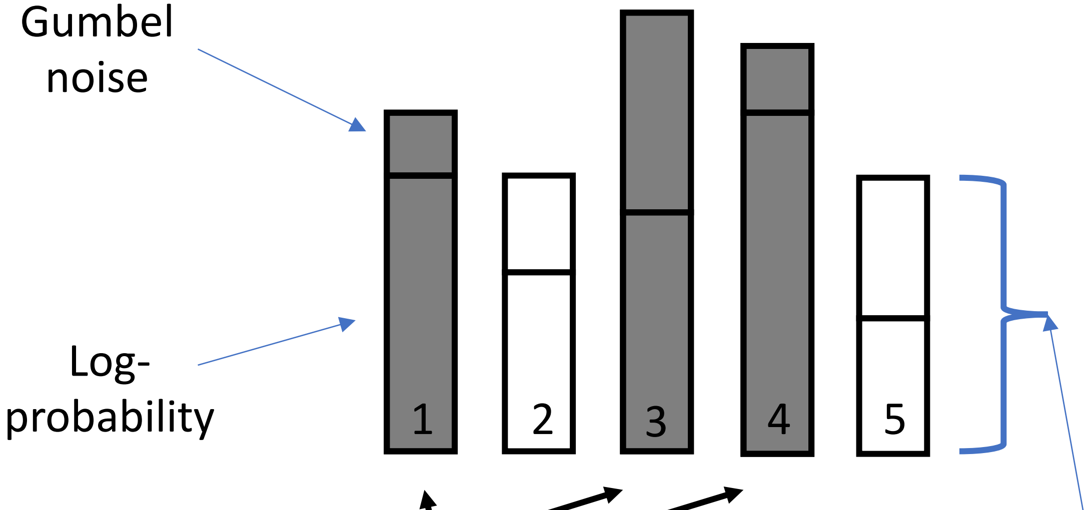
Unordered samples without replacement

$$p(S) = \sum_{B \in \mathcal{B}(S)} p(B) = \sum_{B \in \mathcal{B}(S)} \prod_{i=1}^{k} \frac{p(b_i)}{1 - \sum_{j<i} p(b_j)}$$

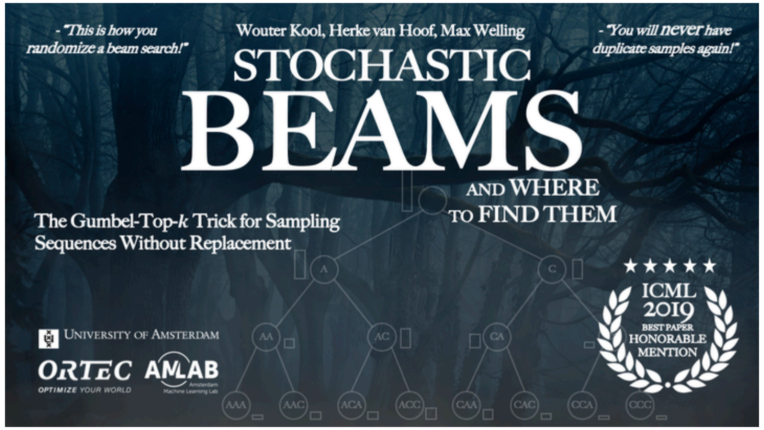
Set $S = \{1,3,4\}$

Sum over $k!$
permutations

Gumbel-Top-k sampling



$$B = b_1, \ldots, b_k = \underset{i}{\arg\operatorname{top} k}\, G_{\phi_i}$$



https://arxiv.org/abs/1903.06059

http://www.jmlr.org/papers/v21/19-985.html

$$B = (3,4,1)$$
$$S = \{1,3,4\}$$

Back to our problem

$$\nabla_{\boldsymbol{\theta}} E_{p_{\boldsymbol{\theta}}(x)}[f(x)]$$

Estimating the expectation

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)]$$

The single sample estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(B)}[f(b_1)]$$

Separating the expectation

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(B|S)}[f(b_1)]\right]$$

Set of unordered samples

Conditional distribution of their order

Separating the expectation

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)]\right]$$

Rao-Blackwellizing the estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)]\right]$$

$$E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)] = \sum_{s\in S} P(b_1 = s|S)\, f(s)$$

Rao-Blackwellizing the estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)]\right]$$

$$E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)] = \sum_{s \in S} P(b_1 = s|S) f(s)$$

$$P(b_1 = s|S) = \frac{P(S|b_1 = s) P(b_1 = s)}{P(S)}$$

Rao-Blackwellizing the estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)]\right]$$

$$E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)] = \sum_{s \in S} P(b_1 = s|S) f(s)$$

$$P(b_1 = s|S) = \underbrace{\frac{P(S|b_1 = s)}{P(S)}}_{\text{Leave-one-out ratio } R(S,s)} \underbrace{P(b_1 = s)}_{p_{\boldsymbol{\theta}}(s)}$$

Rao-Blackwellizing the estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)]\right]$$

$$E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)] = \sum_{s \in S} P(b_1 = s|S) f(s)$$

$$P(b_1 = s|S) = R(S, s) p_{\boldsymbol{\theta}}(s)$$

Rao-Blackwellizing the estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)]\right]$$

$$E_{p_{\boldsymbol{\theta}}(b_1|S)}[f(b_1)] = \sum_{s \in S} R(S, s) p_{\boldsymbol{\theta}}(s) f(s)$$

Rao-Blackwellizing the estimator

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(S)}\left[\underbrace{\sum_{s \in S} R(S,s)p_{\boldsymbol{\theta}}(s)f(s)}_{\text{Unordered set estimator}}\right]$$

Murthy 1957

Combining with REINFORCE

$$E_{p_{\boldsymbol{\theta}}(x)}[f(x)]$$

$$= E_{p_{\boldsymbol{\theta}}(S)}\left[\sum_{s \in S} R(S, s) p_{\boldsymbol{\theta}}(s) f(s)\right]$$

Combining with REINFORCE

$$E_{p_{\boldsymbol{\theta}}(x)}[\nabla_\theta \log p_{\boldsymbol{\theta}}(s)\, f(x)]$$

$$= E_{p_{\boldsymbol{\theta}}(S)}\left[\sum_{s\in S} R(S,s) p_{\boldsymbol{\theta}}(s) \nabla_\theta \log p_{\boldsymbol{\theta}}(s)\, f(s)\right]$$

Combining with REINFORCE

$$\nabla_\theta E_{p_{\boldsymbol{\theta}}(x)}[f(x)] = E_{p_{\boldsymbol{\theta}}(x)}[\nabla_\theta \log p_{\boldsymbol{\theta}}(s)\, f(x)]$$

$$= E_{p_{\boldsymbol{\theta}}(S)}\left[\sum_{s \in S} R(S,s) \underbrace{p_{\boldsymbol{\theta}}(s) \nabla_\theta \log p_{\boldsymbol{\theta}}(s)}_{\nabla_\theta p_{\boldsymbol{\theta}}(s)} f(s)\right]$$

Combining with REINFORCE

$$\nabla_\theta E_{p_{\boldsymbol{\theta}}(x)}[f(x)]$$

$$= E_{p_{\boldsymbol{\theta}}(S)}\left[\underbrace{\sum_{s \in S} R(S,s) \nabla_\theta p_{\boldsymbol{\theta}}(s) f(s)}_{\text{Unordered set policy gradient estimator}}\right]$$

Include a baseline

$$\nabla_\theta E_{p_{\boldsymbol{\theta}}(x)}[f(x)]$$

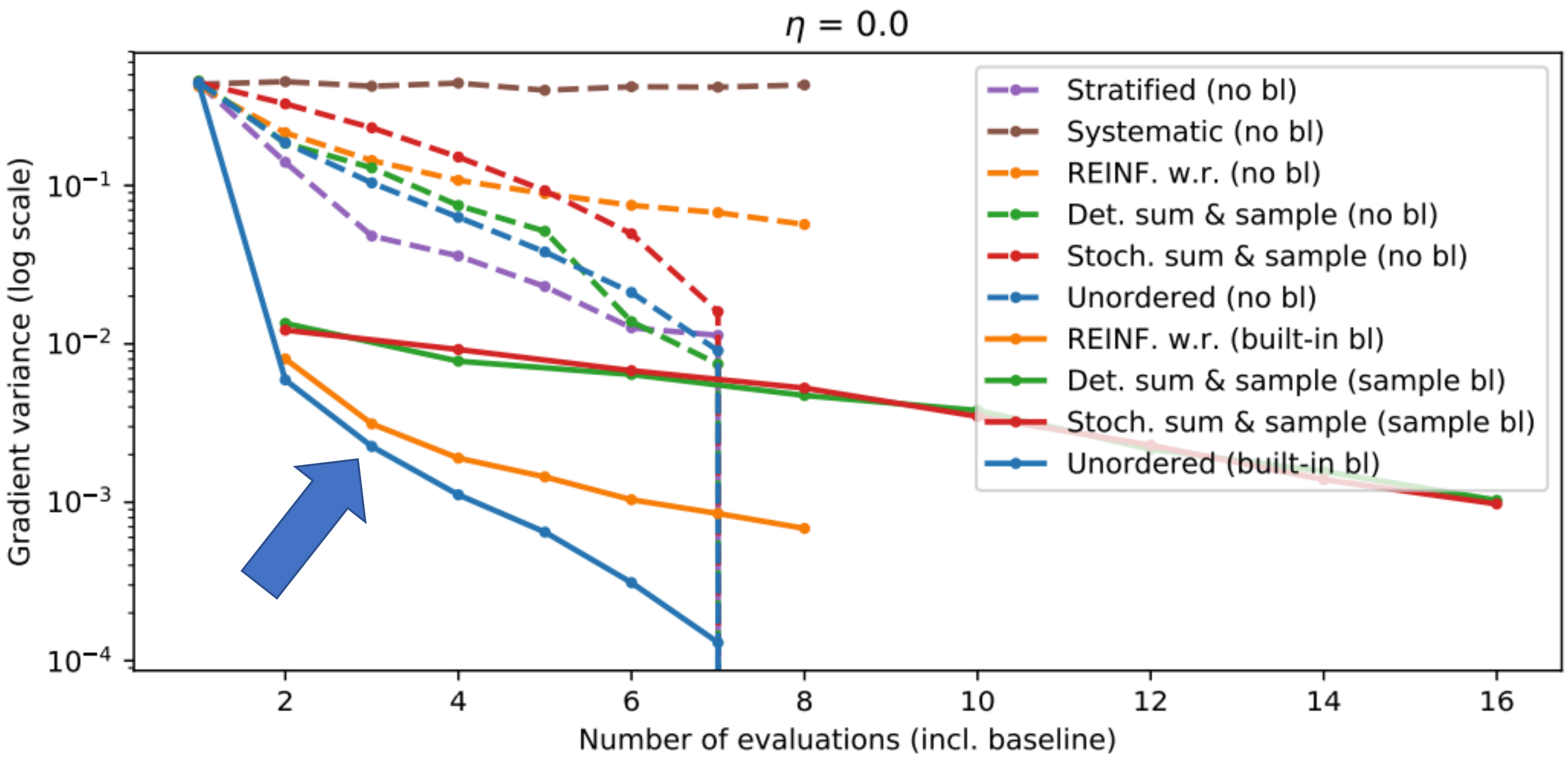
$$= E_{p_{\boldsymbol{\theta}}(S)}\left[\sum_{s\in S} R(S,s)\nabla_\theta p_{\boldsymbol{\theta}}(s)\left(f(s) - \underbrace{\sum_{s'\in S} R^{\setminus s}(S,s')p_{\boldsymbol{\theta}}(s')f(s')}_{\text{‘Baseline’}}\right)\right]$$

Second order leave-one-out ratio

Unbiased!

Experiments

Bernoulli gradient variance



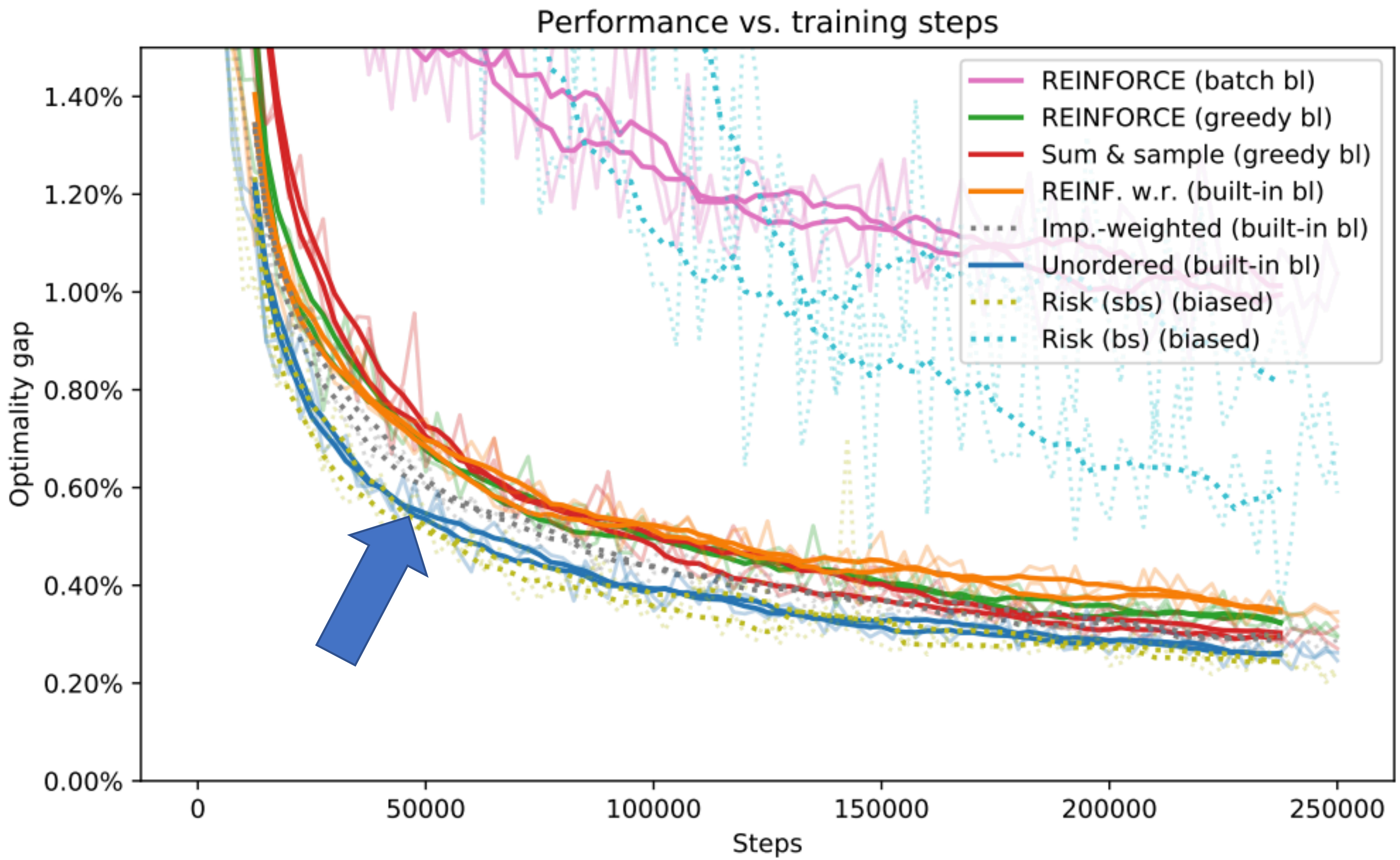
(a) High entropy ($\eta = 0$)

Categorical Variational Auto-Encoder (grad. var.)

Table 1: VAE gradient log-variance of different unbiased estimators with $k = 4$ samples.

Domain	ARSM	RELAX	REINFORCE (no bl)	REINFORCE (sample bl)	Sum & sample (no bl)	Sum & sample (sample bl)	REINF. w.r. (built-in bl)	Unordered (built-in bl)
Small 10^2	13.45	11.67	11.52	7.49	**6.29**	**6.29**	6.65	**6.29**
Large 10^{20}	15.55	15.86	13.81	8.48	13.77	8.44	**7.06**	**7.05**

Travelling Salesman Problem



Take away

The unordered set estimator

- Low-variance
- Unbiased
- Alternative to Gumbel-Softmax

End of story