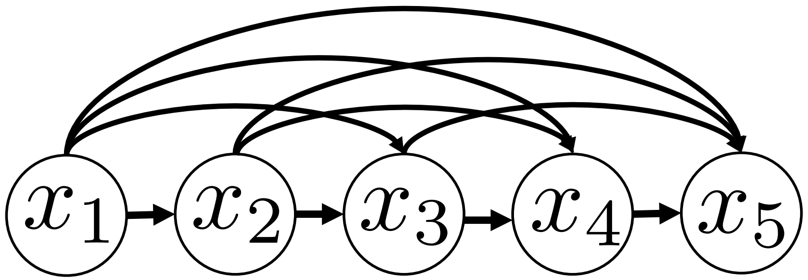
Cascaded Text Generation with Markov Transformers

Yuntian Deng, Alexander Rush

Harvard University, Cornell University

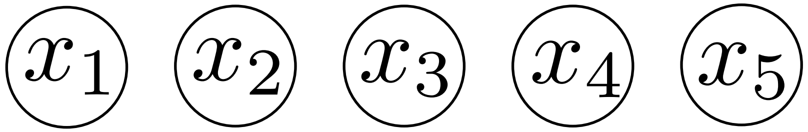
Fully Autoregressive vs Nonautoregressive

Fully Autoregressive



- Decoding: beam search
- Fluent but serial

Nonautoregressive [Gu et al 2018]



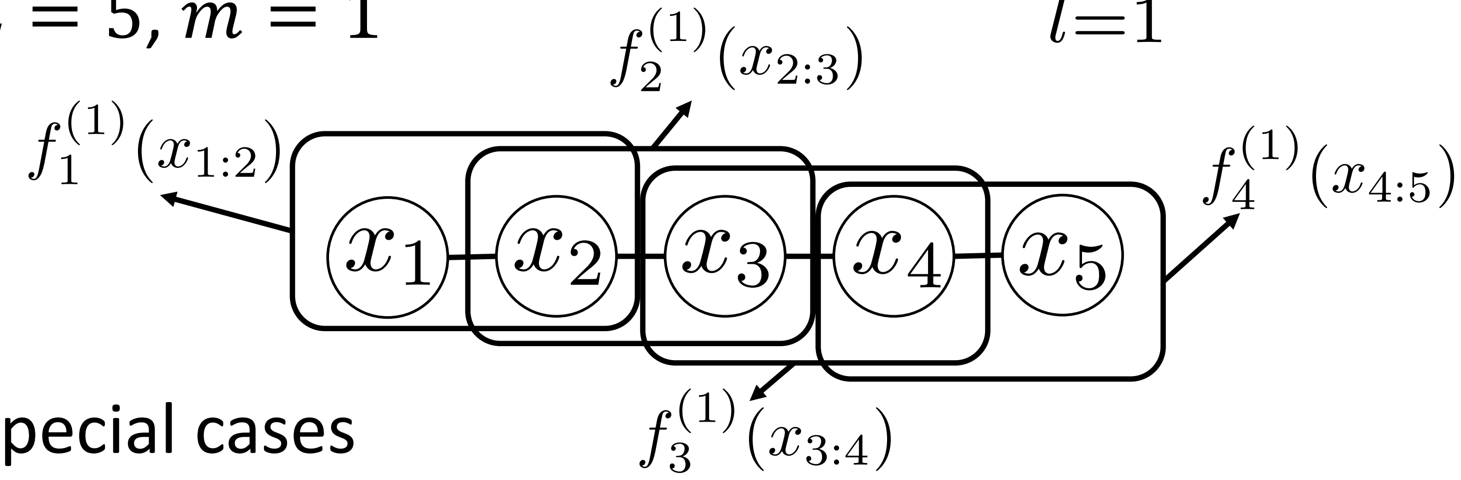
- Decoding: Argmax at each position
- Parallel but disfluent

Markov Random Field Framework

- An m-th order MRF

$$P^{(m)}(x_{1:L};\theta) \propto \exp \sum_{l=1}^{L-m} f_l^{(m)}(x_{l:l+m};\theta)$$

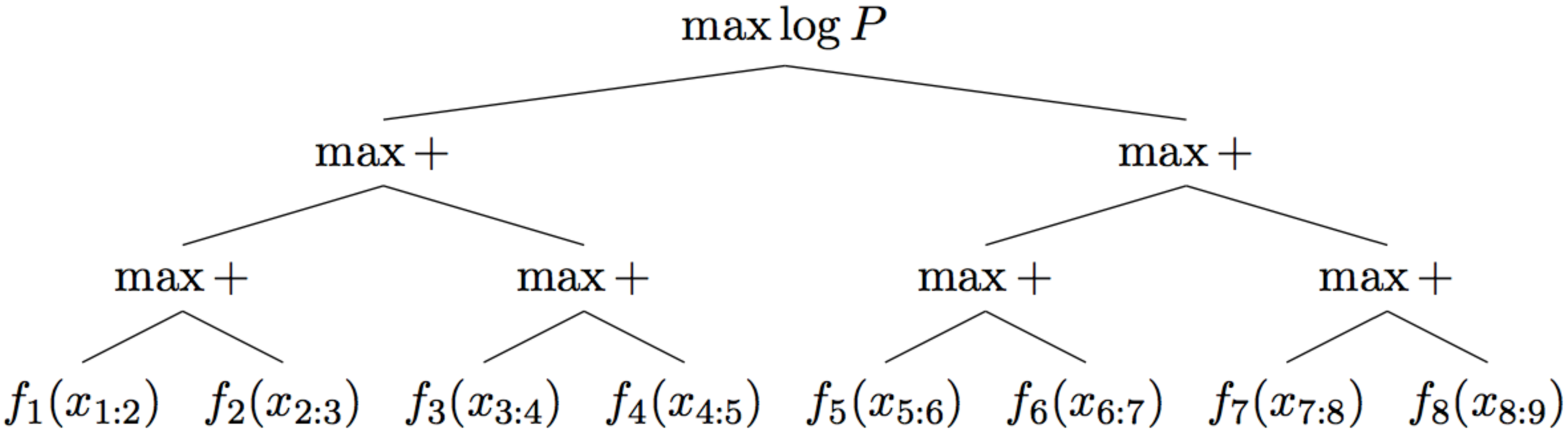
- $L = 5, m = 1$



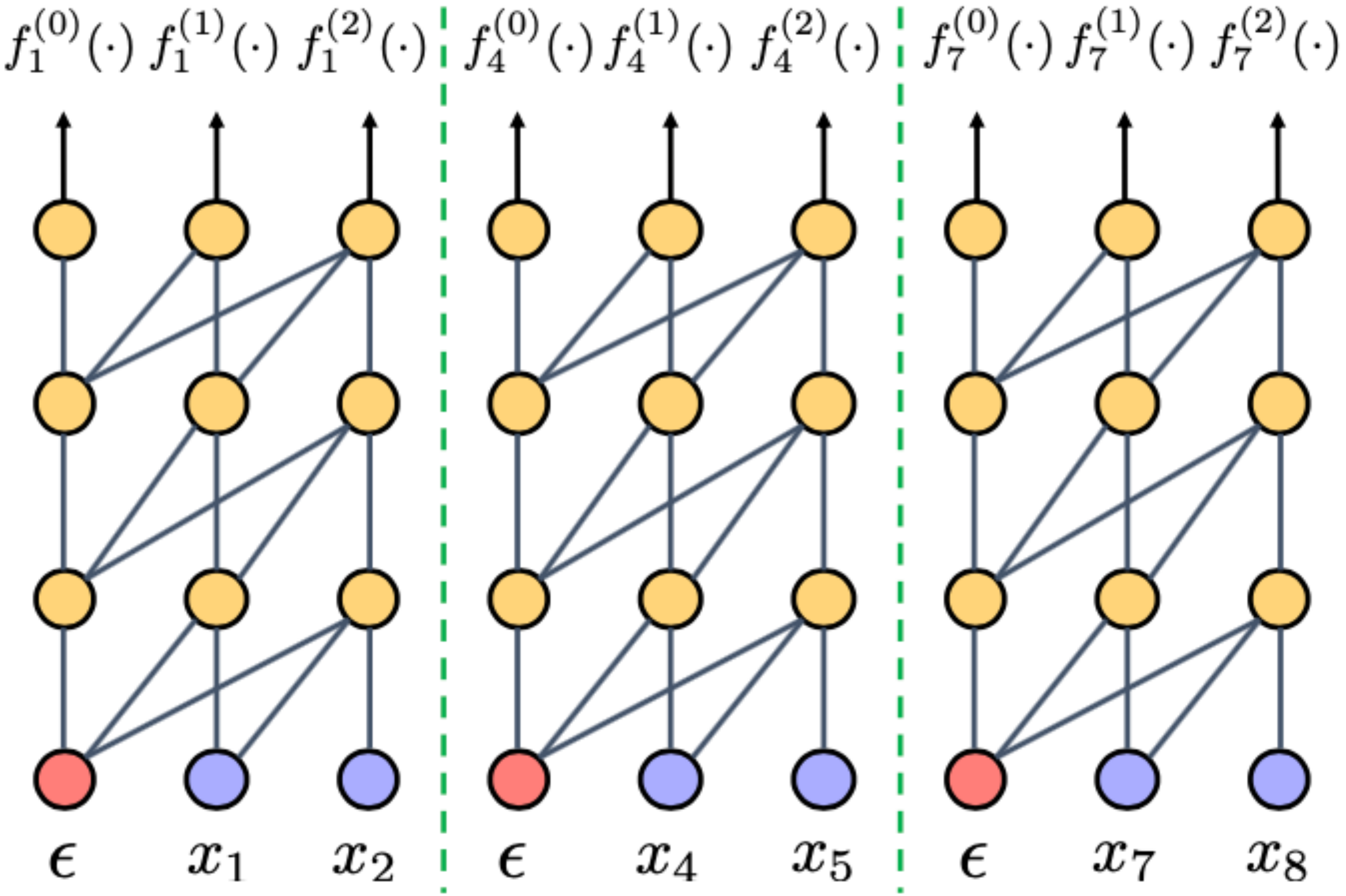
- Special cases
  - $m = 0$: nonautoregressive
  - $m = L - 1$: fully autoregressive
  - $0 < m < L - 1$: bounded-order models (this work)

Bounded-order models

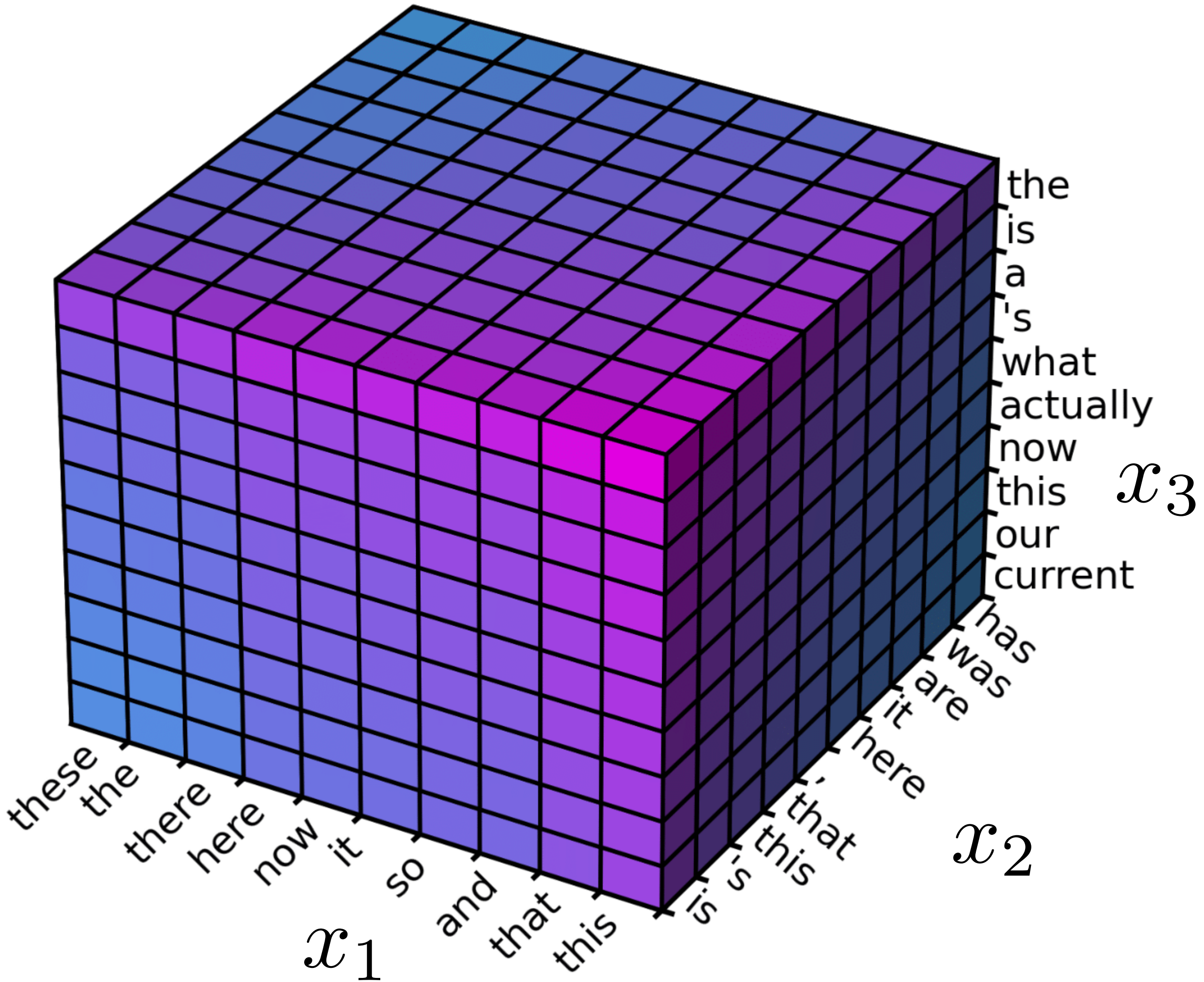
- Faster than fully autoregressive: Parallel Decoding [Rush 2020, Simo et al 2019]
- More fluent than nonautoregressive
- Allows accuracy/speed tradeoff



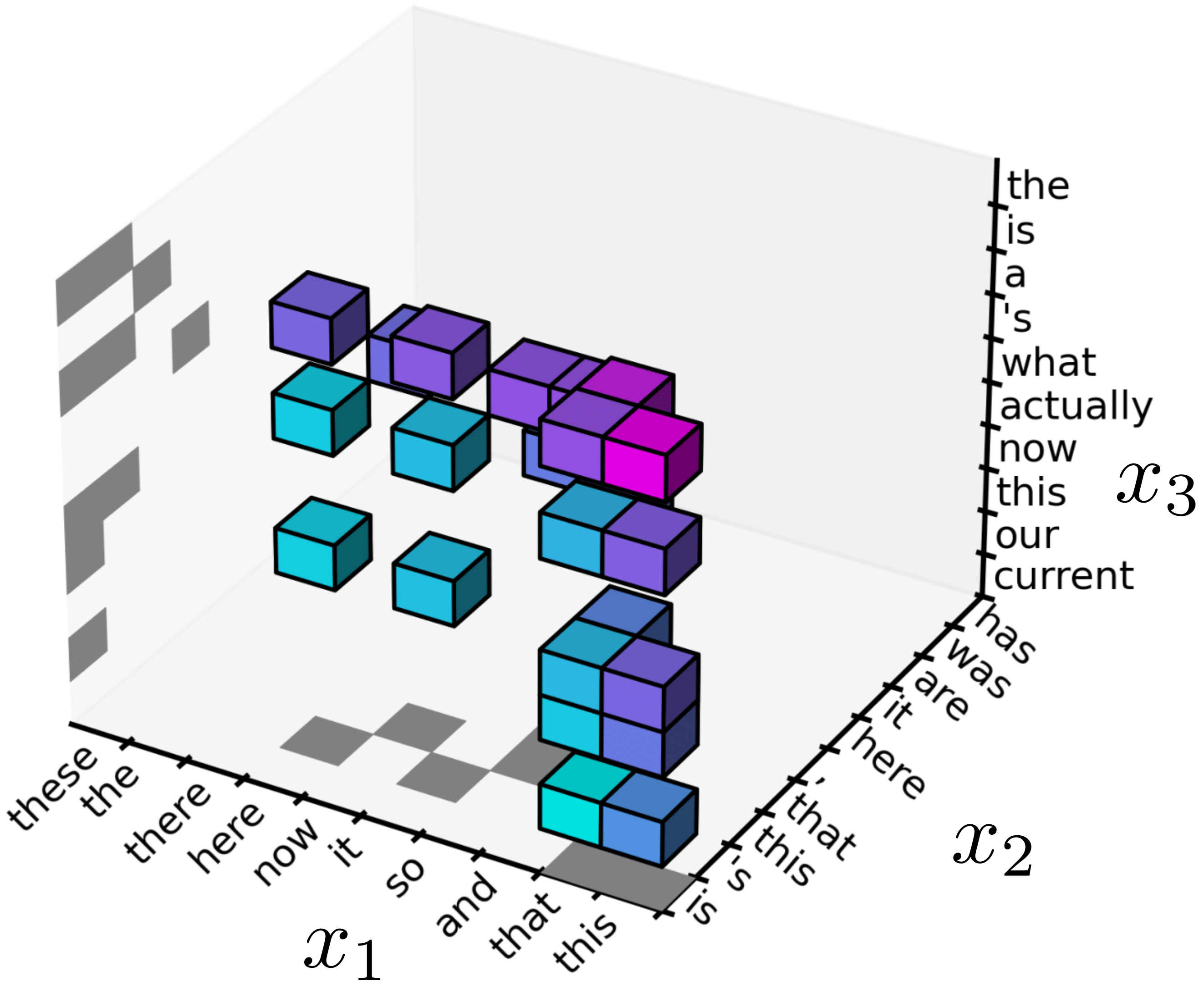
Parameterization: Markov Transformer



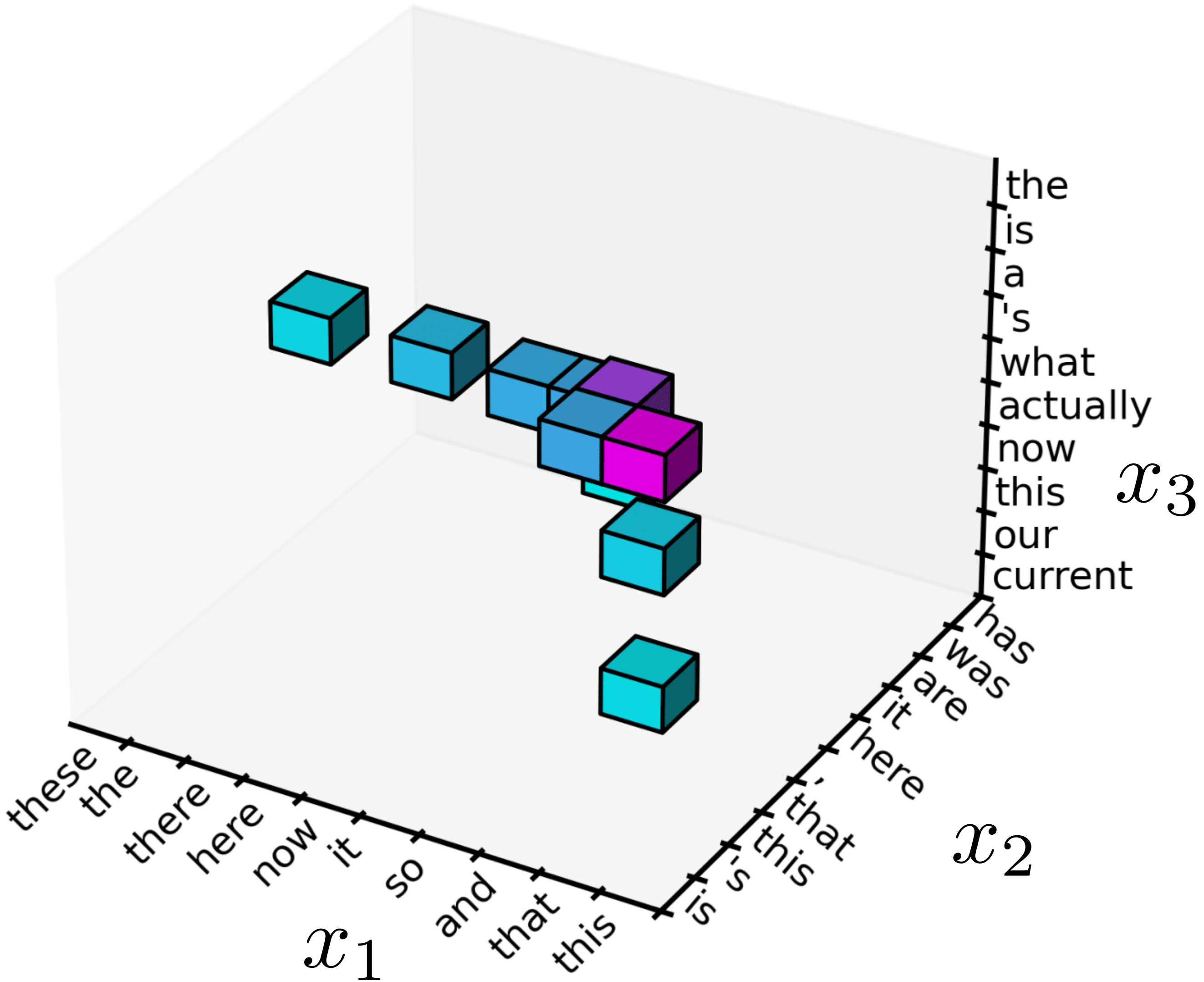
Cascaded Decoding



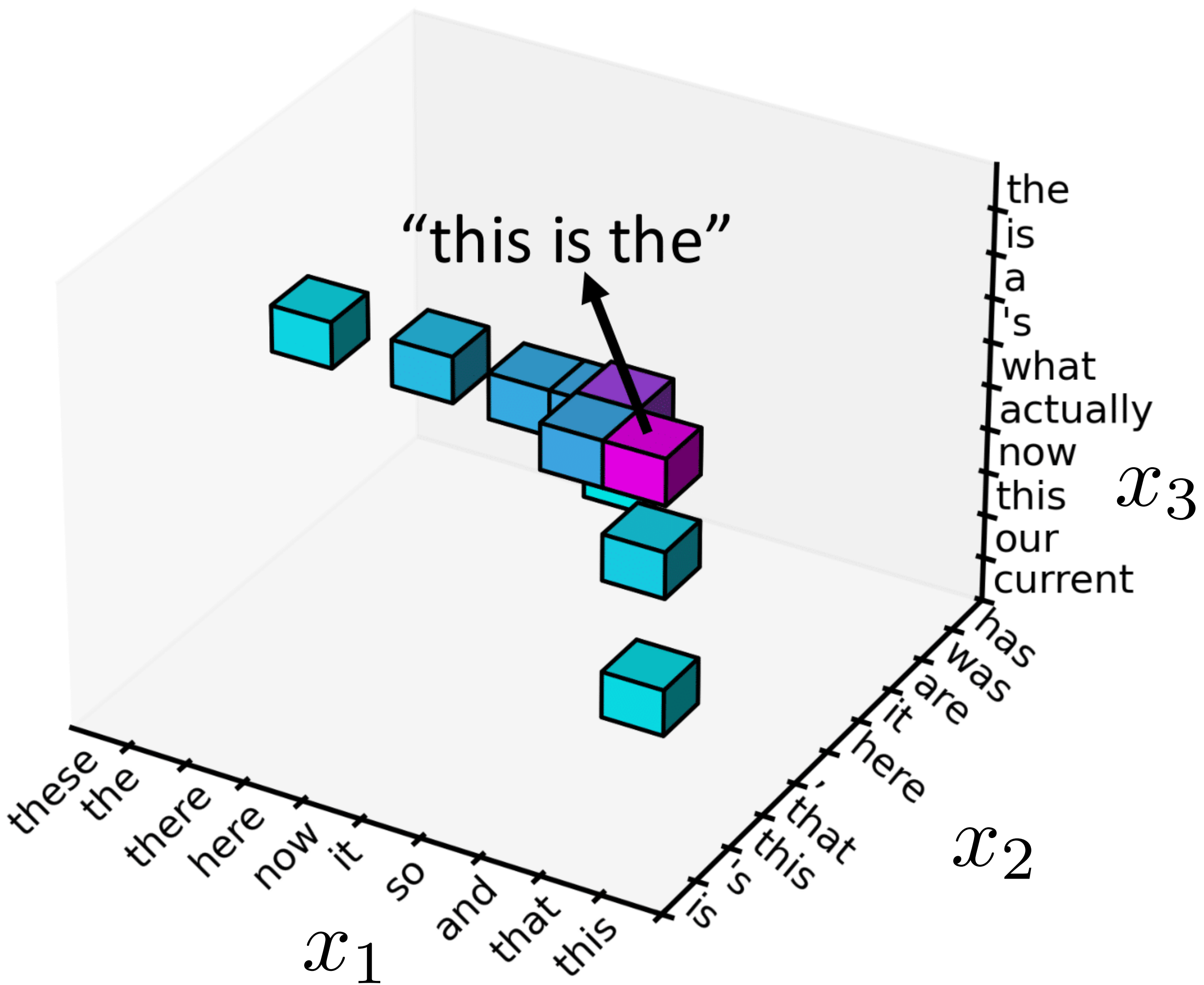
Cascaded Decoding



Cascaded Decoding



Cascaded Decoding



Parallel Time Complexity

- $f_l^{(m)}(x_{l:l+m})$ can be computed in parallel $O(1)$
- Parallel tree decoding takes $O(\log L)$ time
- In practice, as fast as nonautoregressive
  - parallel tree decoding takes <1% total time

Extension: Variable Length Generation

- Nonautoregressive Decoding needs to specify length L
  - An inappropriate length limits the achievable score
- MRF allows considering multiple length values
  - Specify the maximum possible length
  - Introduce a special padding symbol
  - End-of-sentence/padding always transition to padding

A Real Example

$K = 5$, max length 8, Source: eine erstaunliche frau .

m	$x_{1:1+m}$	$x_{2:2+m}$	$x_{3:3+m}$	$x_{4:4+m}$	$x_{5:5+m}$	$x_{6:6+m}$	$x_{7:7+m}$	$x_{8:8+m}$
	an	amazing	woman	woman	eos	eos	eos	pad
	amazing	woman	amazing	.	.	pad	pad	-
0	incredible	an	an	amazing	woman	.	.	-
	this	remarkable	.	eos	amazing	woman	woman	-
	remarkable	incredible	women	an	women	women	women	-

A Real Example ($K = 5$)

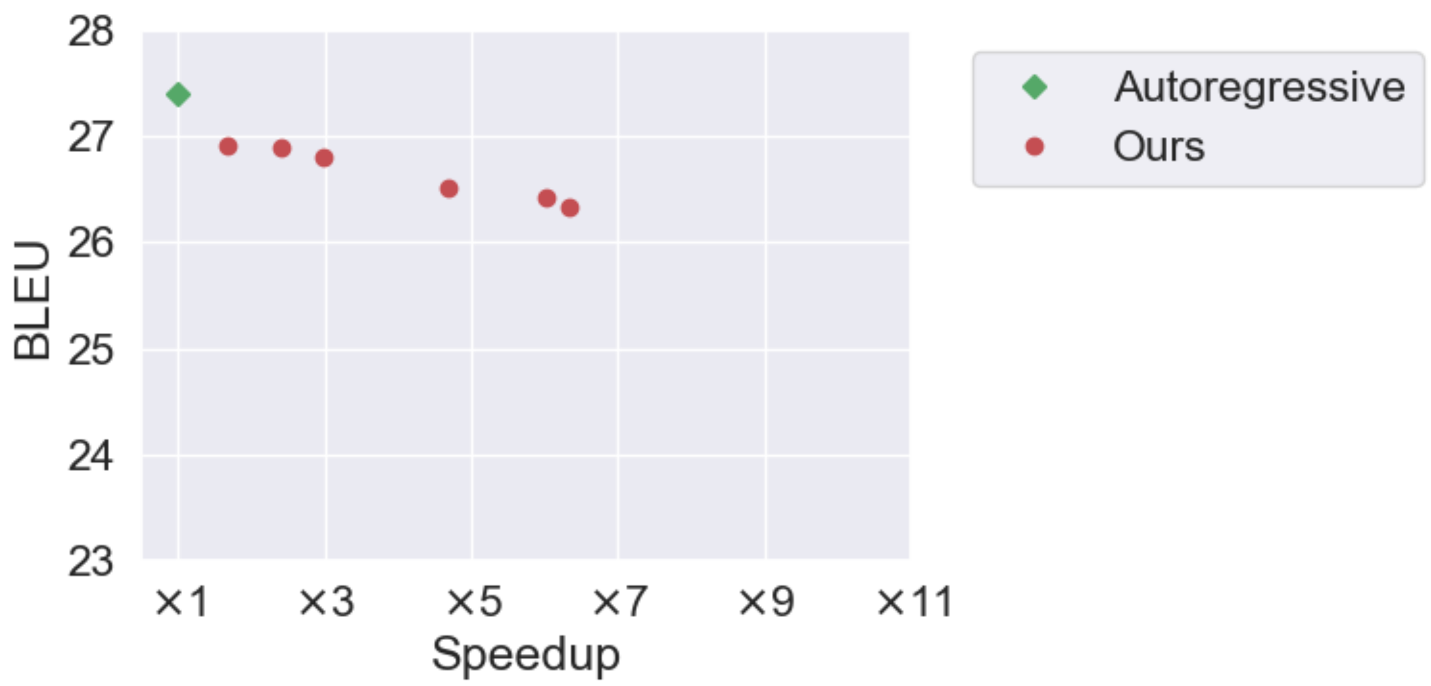
m	$x_{1:1+m}$	$x_{2:2+m}$	$x_{3:3+m}$	$x_{4:4+m}$	$x_{5:5+m}$	$x_{6:6+m}$	$x_{7:7+m}$
	an amazing	amazing woman	woman .	. eos	eos pad	pad pad	pad pad
	an incredible	incredible woman	amazing woman	woman .	. eos	eos pad	eos pad
1	this amazing	remarkable woman	women .	amazing woman	woman .	. eos	-
	an remarkable	woman amazing	woman woman	. .	women .	woman eos	-
	amazing woman	amazing women	an amazing	. woman	. .	-	-

A Real Example ($K = 5$)

m	$x_{1:1+m}$	$x_{2:2+m}$	$x_{3:3+m}$	$x_{4:4+m}$	$x_{5:5+m}$	$x_{6:6+m}$
	an amazing woman	amazing woman .	woman . eos	. eos pad	eos pad pad	pad pad pad
	an incredible woman	incredible woman .	women . eos	woman . eos	. eos pad	eos pad pad
2	this amazing woman	remarkable woman .	woman woman .	. . eos	woman . eos	. eos pad
	an remarkable woman	amazing women .	woman . .	. woman .	. . eos	-
	an amazing women	amazing woman woman	woman . woman	woman . .	-	-

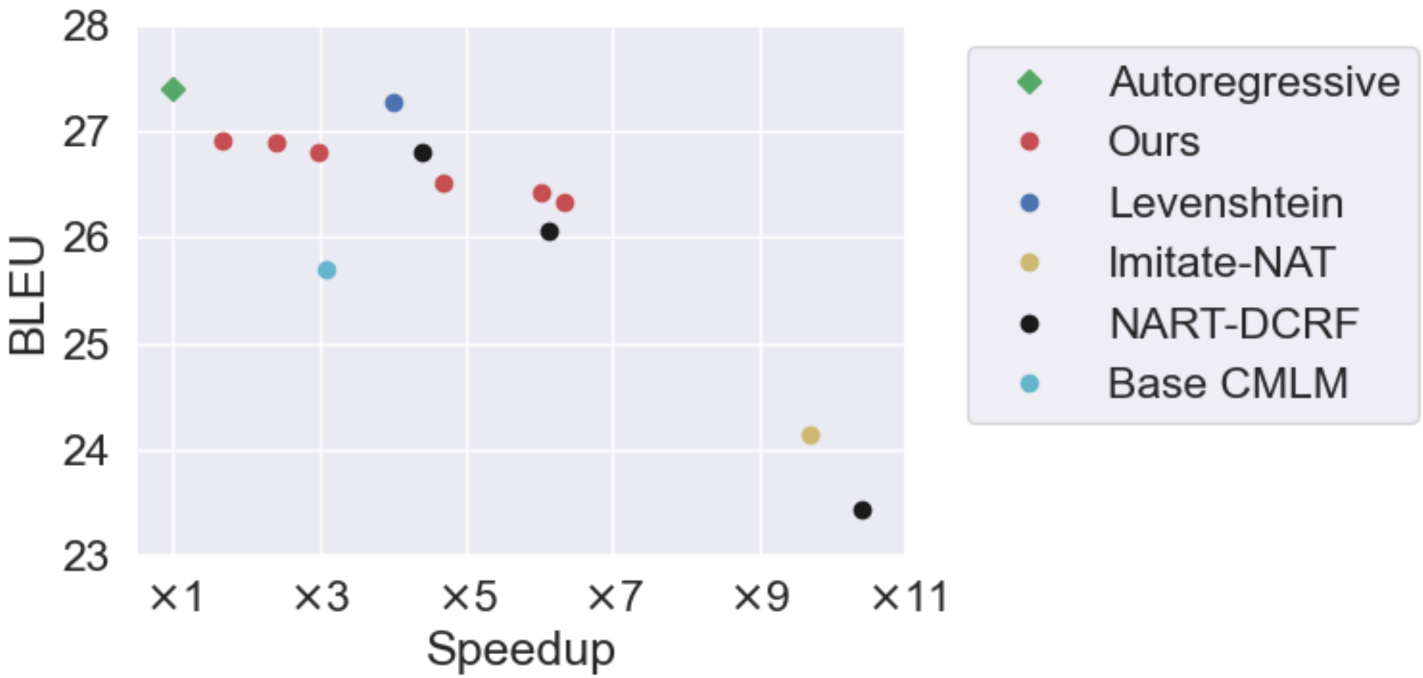
Results I: Speed/Accuracy Tradeoff

- Translation on IWSLT (w/ distillation [5])



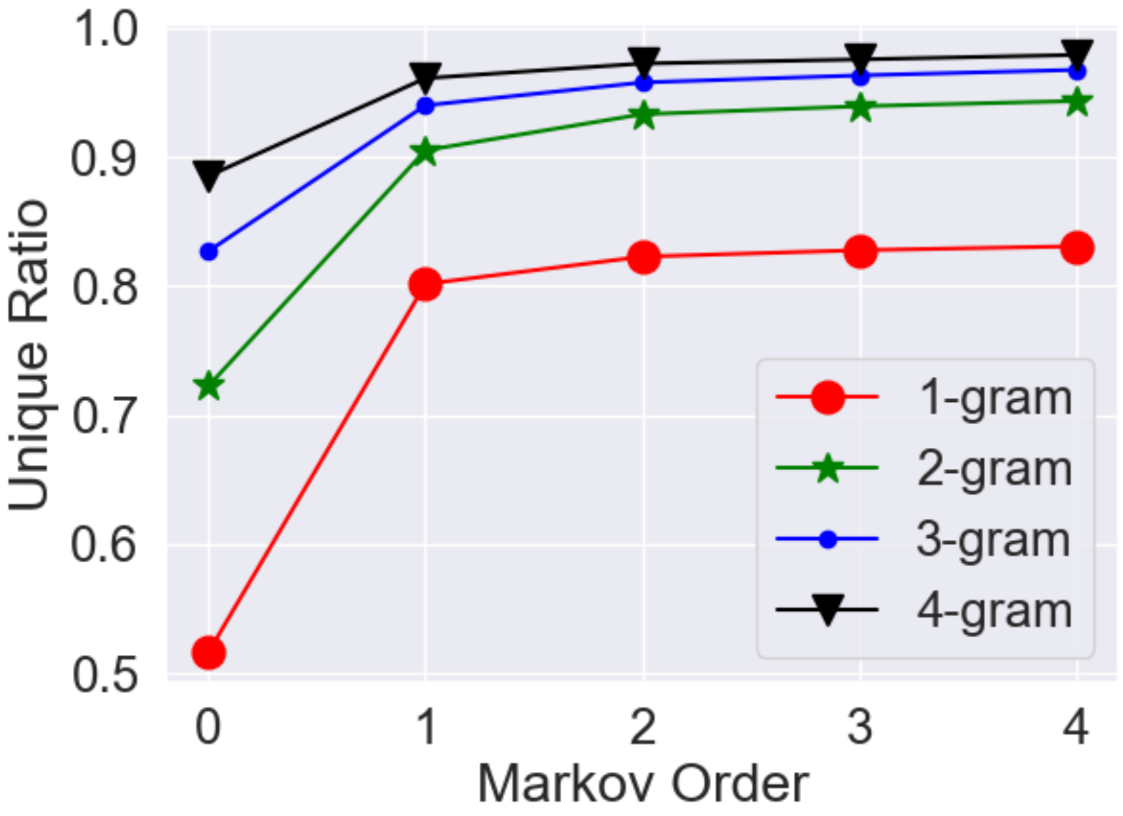
Results I: Speed/Accuracy Tradeoff

- Translation on IWSLT (w/ distillation [5])



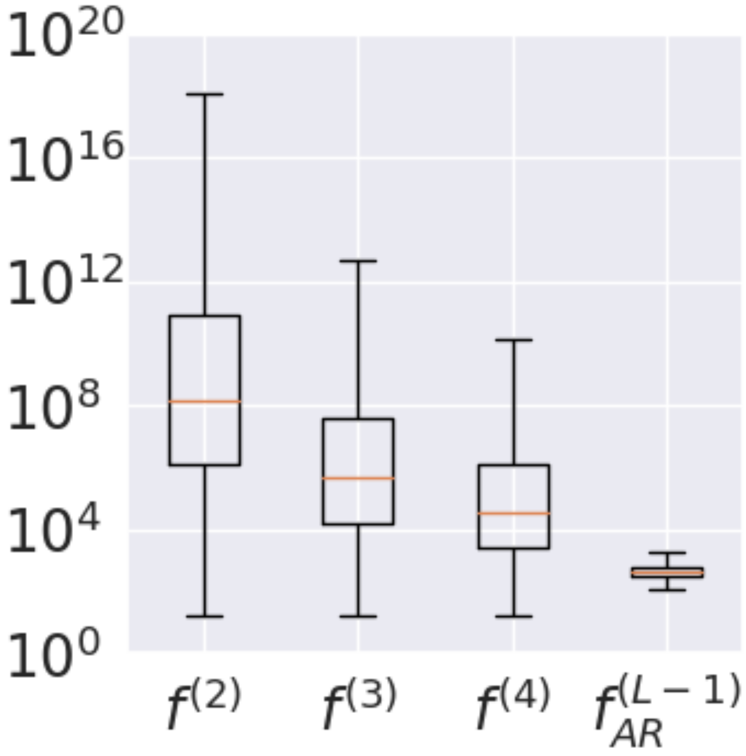
Results II: Repetitions

- Fewer Repetitions



Results III: Number of Sequences Scored

- Autoregressive (AR) w/ Beam search only scores KL sequences
- Cascaded decoding can score exponential number of sequences



Conclusions

- Nonautoregressiveness is sufficient but unnecessary for fast text generation
- Bounded-order MRFs enable parallel decoding
  - Faster than fully autoregressive
  - More fluent than nonautoregressive
- Cascaded search efficiently decodes a bounded-order MRF
- Markov transformer can parameterize the entire cascade

More Details

- Code, pretrained models, logs: https://github.com/harvardnlp/cascaded-generation
- Paper: https://arxiv.org/pdf/2006.01112.pdf

References

[1]: Gu, Jiatao, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. "Non-autoregressive neural machine translation." arXiv preprint arXiv:1711.02281 (2017).

[2]: Rush, Alexander M. "Torch-Struct: Deep Structured Prediction Library." arXiv preprint arXiv:2002.00876 (2020).

[3]: Särkkä, Simo, and Ángel F. García-Fernández. "Temporal parallelization of bayesian filters and smoothers." arXiv preprint arXiv:1905.13002 (2019).

[4]: Weiss, David, and Benjamin Taskar. "Structured prediction cascades." In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 916-923. 2010.

[5]: Kim, Yoon, and Alexander M. Rush. "Sequence-level knowledge distillation." arXiv preprint arXiv:1606.07947(2016).