COS 584

Advanced Natural Language Processing

P11: Pre-training

Spring 2021

NeurIPS 2019

XLNet: Generalized Autoregressive Pretraining for Language Understanding

Zhilin Yang[*1], **Zihang Dai**[*12], **Yiming Yang**[1], **Jaime Carbonell**[1], **Ruslan Salakhutdinov**[1], **Quoc V. Le**[2]
[1]Carnegie Mellon University, [2]Google AI Brain Team
{zhiliny,dzihang,yiming,jgc,rsalakhu}@cs.cmu.edu, qvl@google.com

From BERT to XLNet

- **Masked Language modeling (MLM)**: mask mask out 15% of the input words, and then predict the masked wordsout 15% of the input words, and then predict the masked words

store gallon

↑ ↑

the man went to the [MASK] to buy a [MASK] of milk

- **Next sentence prediction (NSP):** predict whether a sentence is followed after the next sentence

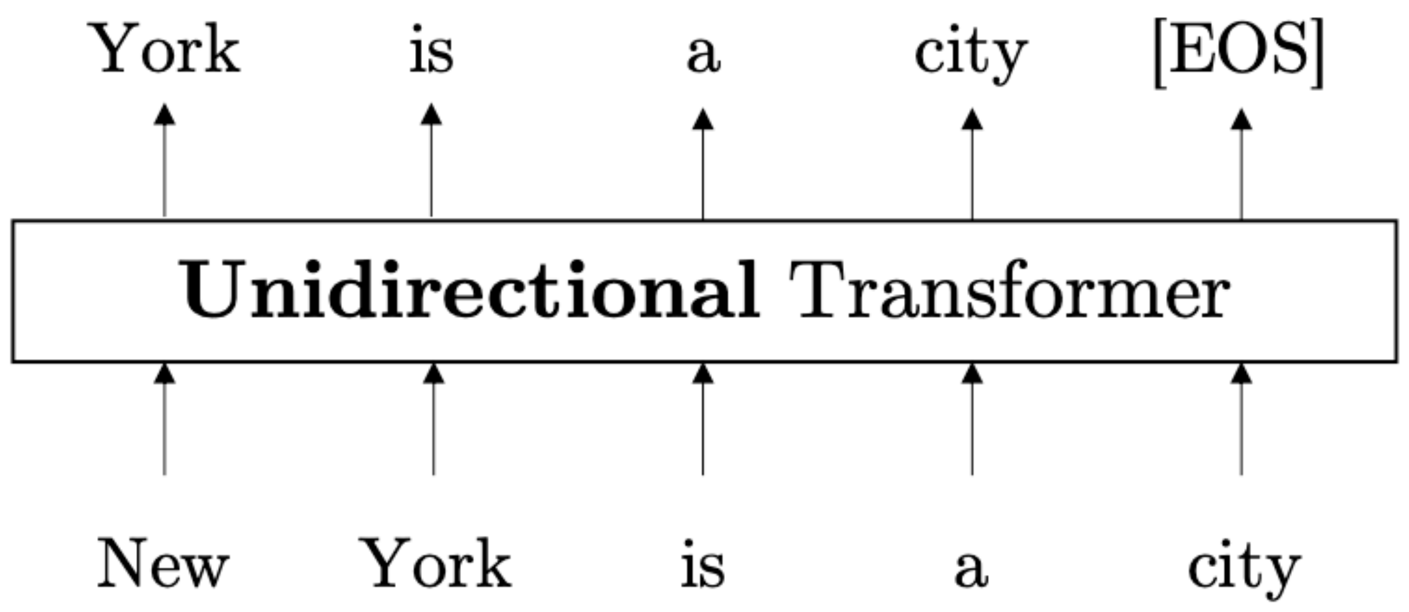
Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

Label = IsNext

Limitations of BERT

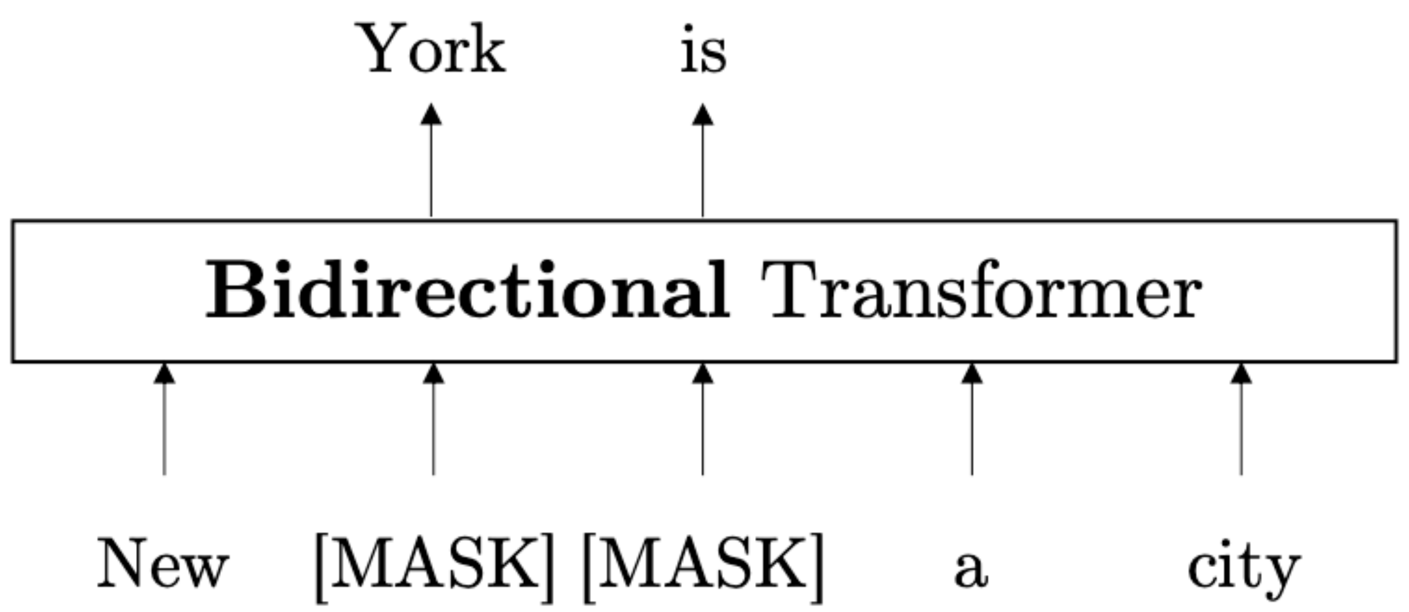
Auto-regressive Language Modeling



$$\log p(\mathbf{x}) = \sum_{t=1}^{T} \log p(x_t | \mathbf{x}_{<t})$$

- **Next-token prediction**

Denoising Auto-encoding (BERT)



$$\log p(\bar{\mathbf{x}}|\hat{\mathbf{x}}) = \sum_{t=1}^{T} \text{mask}_t \log p(x_t | \hat{\mathbf{x}})$$

- **Reconstruct masked tokens**

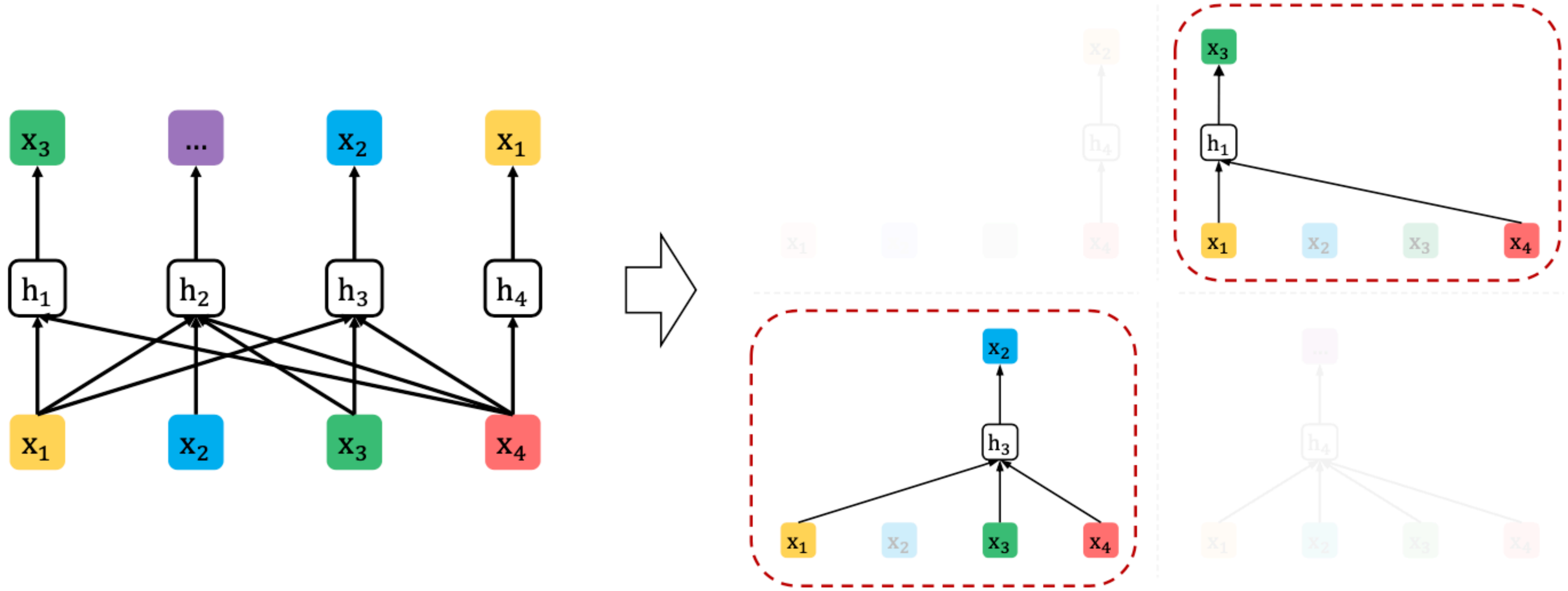
- The predictions are independent
- The [MASK] tokens add artificial noise - you never see them at testing time!

XLNet

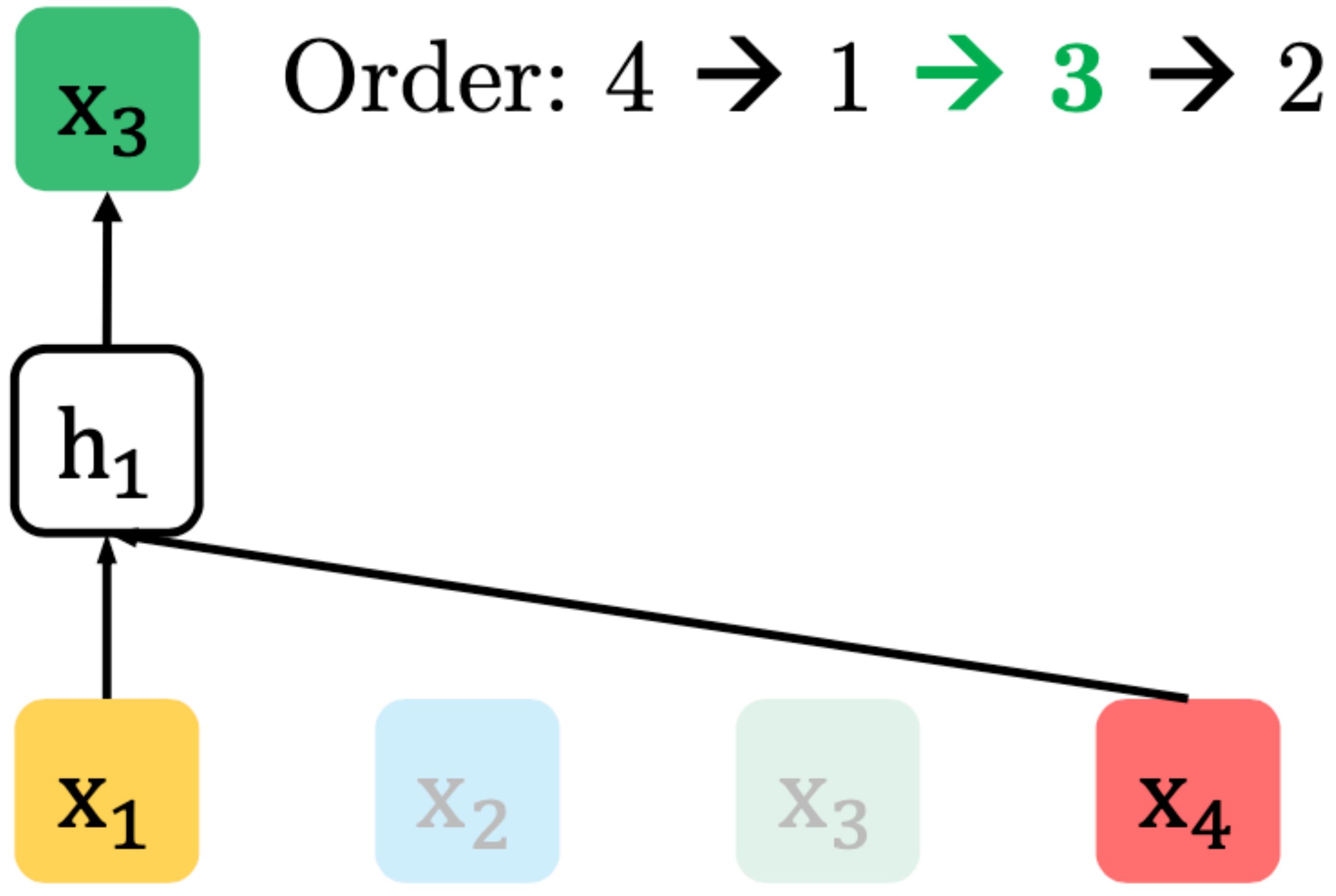
- XLNet = an autoregressive model that captures bidirectional contexts
- Key idea: **permutation language modeling** - sample a factorization order from all possible permutations and predict words by the factorization order one by one

Change the Factorization order to: 4 → 1 → 3 → 2

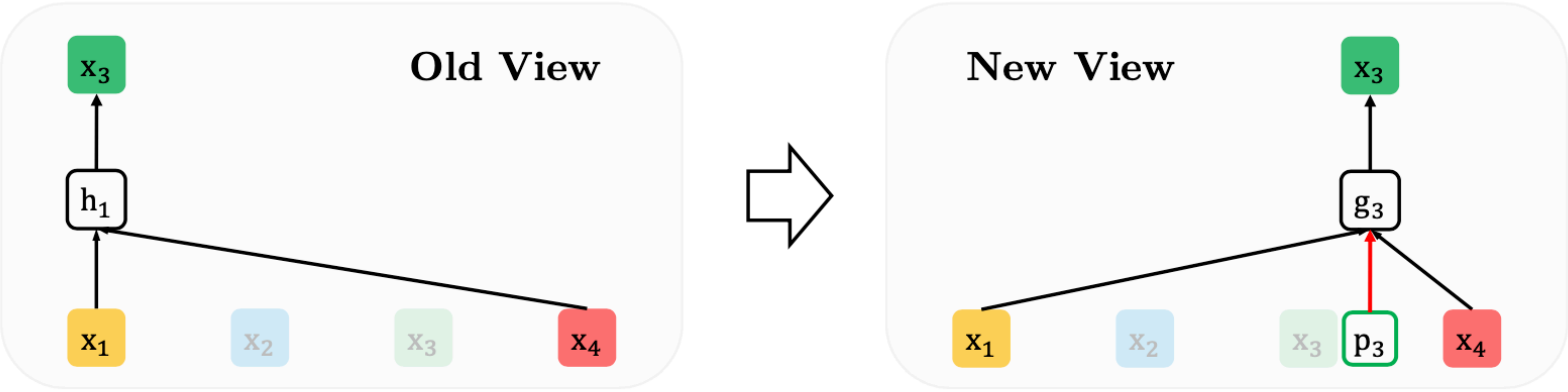
$$P(\mathbf{x}) = P(x_4)P(x_1 \mid \mathbf{x}_4)P(x_3 \mid \mathbf{x}_{1,4})P(x_2 \mid \mathbf{x}_{1,2,4}) \cdots$$



A Technical Challenge

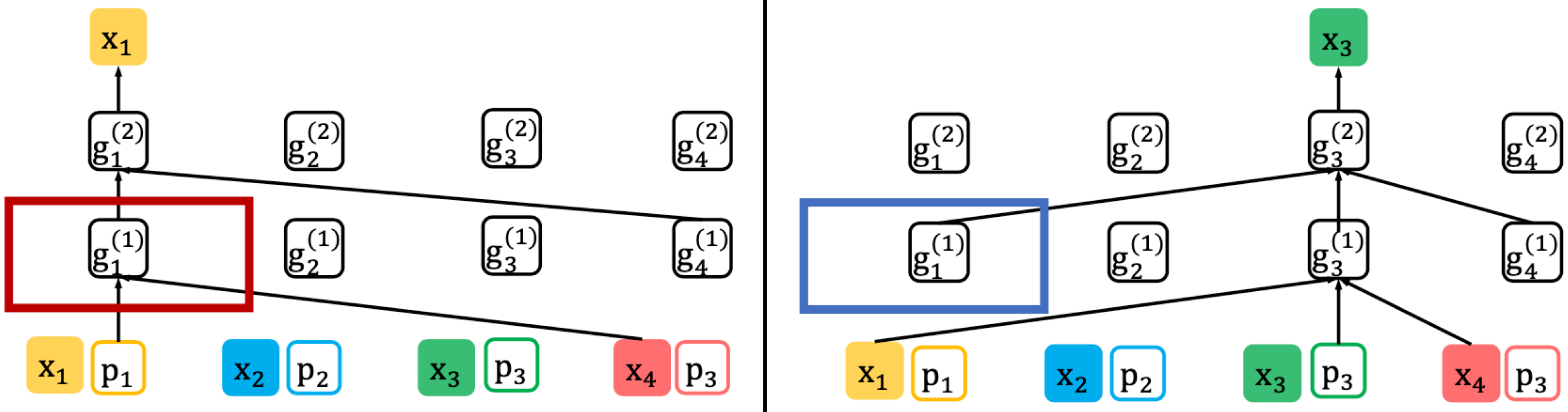


Conditioned on x_1, x_4, we want to predict x_3, how we can we know that we would want to predict the 3rd word even?



A Technical Challenge

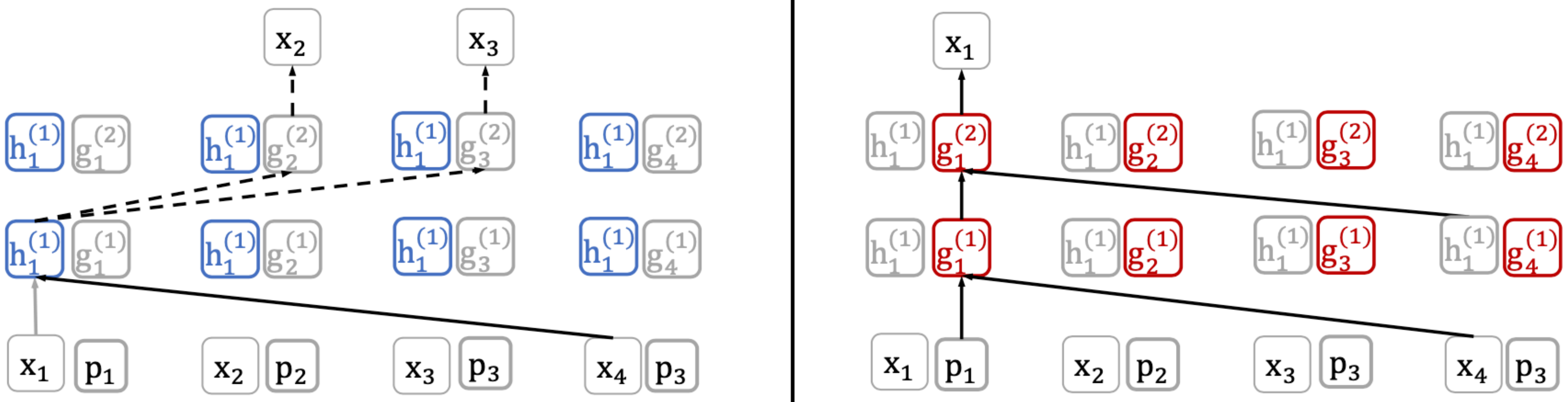
Use $g_1^{(1)}$ to predict x_1 (self) | Use $g_1^{(1)}$ to predict x_3 (other)



Should not encode x_1 | Should encode x_1

Solution: Two-stream Attention

Encoding. Predicting x_2 and x_3 (others). | **Decoding.** Predicting $\boldsymbol{x_1}$ (self).

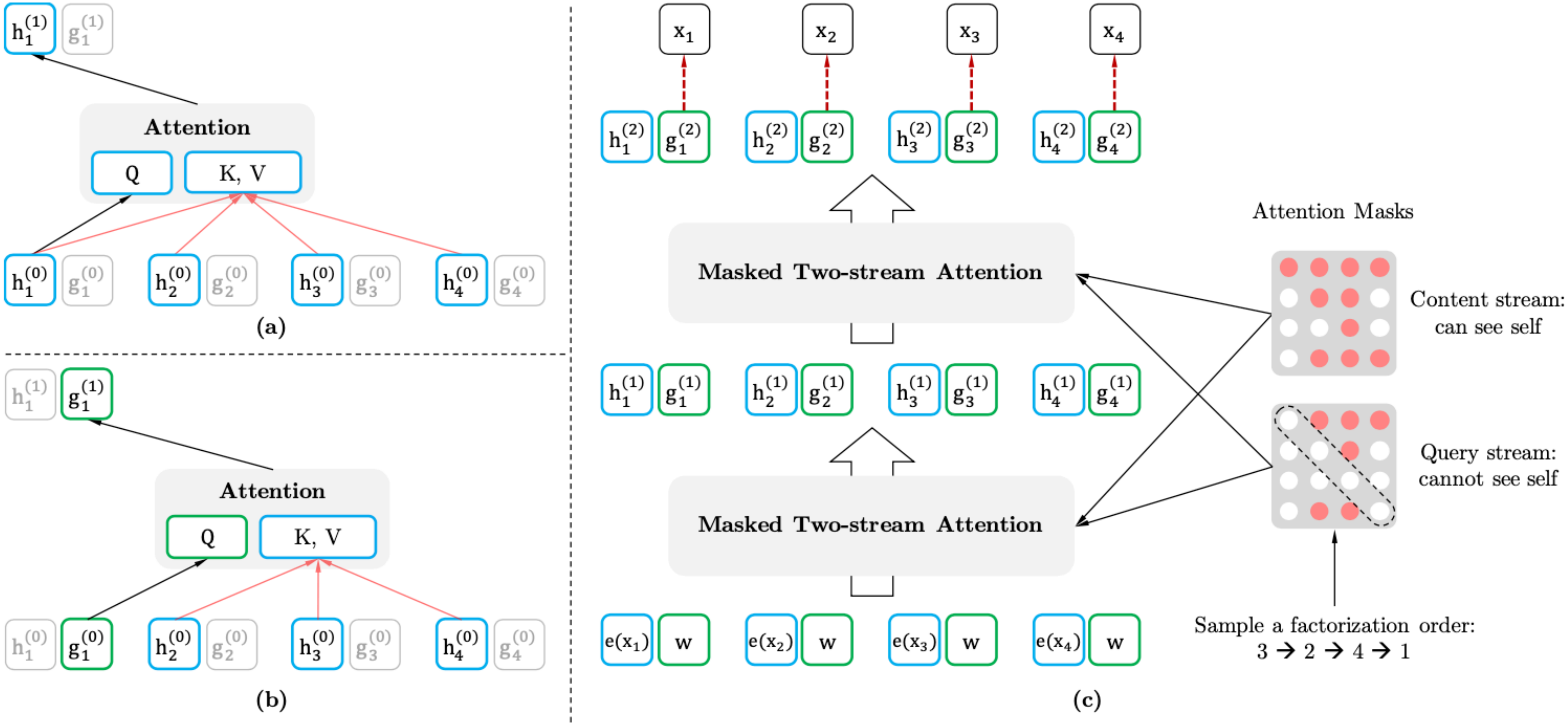


$\boldsymbol{h_1}$ encodes $\boldsymbol{x_1}$ | $\boldsymbol{g_1}$ does not encode $\boldsymbol{x_1}$

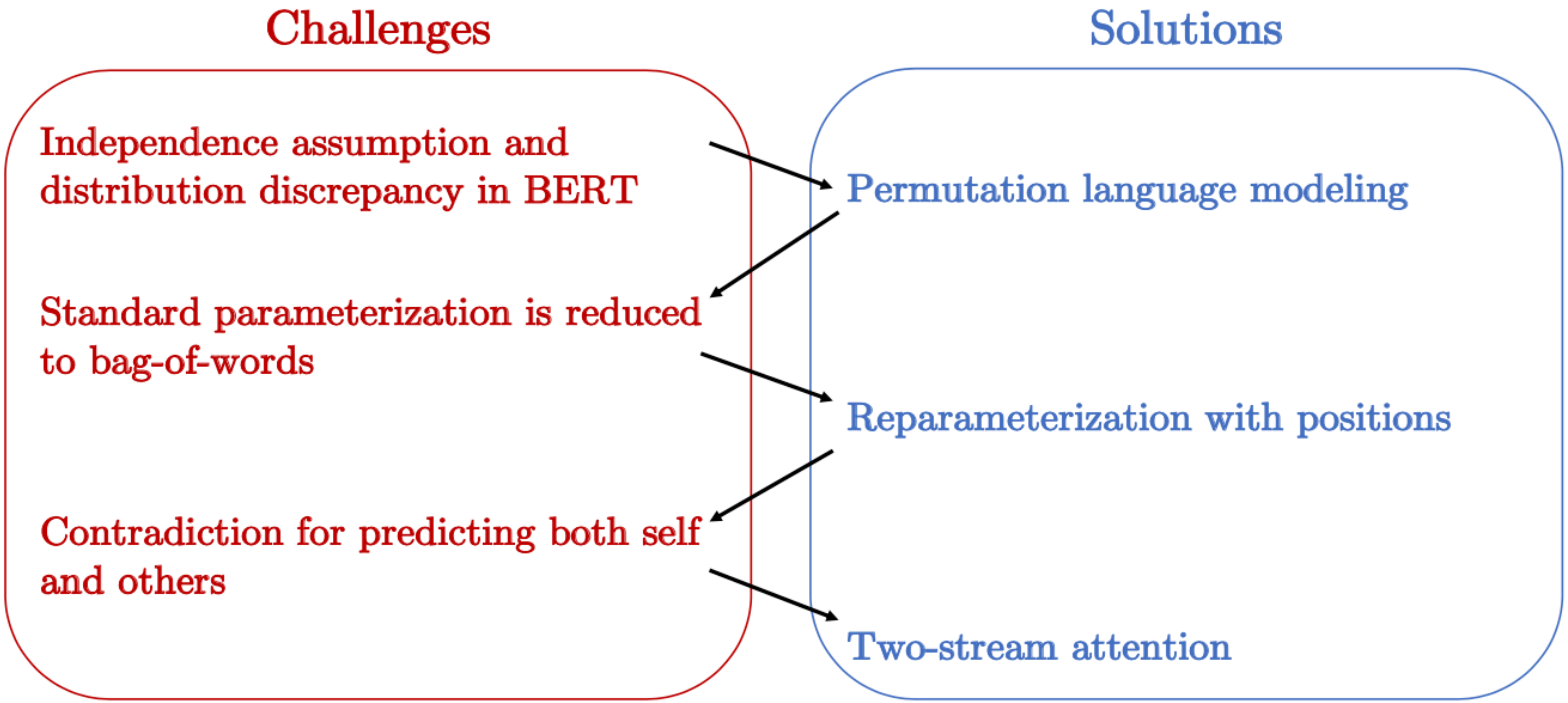
$$g_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = g_{z_t}^{(m-1)}, \mathrm{KV} = \mathbf{h}_{\mathbf{z}<t}^{(m-1)}; \theta), \quad \text{(query stream: use } z_t \text{ but cannot see } x_{z_t})$$

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = h_{z_t}^{(m-1)}, \mathrm{KV} = \mathbf{h}_{\mathbf{z}\leq t}^{(m-1)}; \theta), \quad \text{(content stream: use both } z_t \text{ and } x_{z_t}).$$

Solution: Two-stream Attention



Summary of XLNet



Other important details

- Partial predictions
    - It is very difficult to make predictions when seeing too little context
    - Solution: they only predict the last 1/K words (K = 6 or 7, similar to the 15% in BERT!!!)
- Re-use ideas from Transformer-XL (Dai et al., 2019):
    - Segment recurrence
    - Relative positional encodings
- Span-based prediction

(Dai et al., 2019) Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

XLNet: Experiments

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	**86.6**	96.4	**90.9**	68.0	92.4	-
XLNet	**90.8/90.8**	**94.9**	**92.3**	85.9	**97.0**	90.8	**69.0**	**92.5**	-
Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	**90.9/90.9**†	**99.0**†	**90.4**†	**88.5**	**97.1**†	**92.9**	**70.2**	**93.0**	**92.5**

SQuAD2.0	EM	F1	SQuAD1.1	EM	F1
Dev set results (single model)					
BERT [10]	78.98	81.77	BERT† [10]	84.1	90.9
RoBERTa [21]	86.5	89.4	RoBERTa [21]	88.9	94.6
XLNet	**87.9**	**90.6**	XLNet	**89.7**	**95.1**
Test set results on leaderboard (single model, as of Dec 14, 2019)					
BERT* [10]	80.005	83.061			
RoBERTa [21]	86.820	89.795			
XLNet	**87.926**	**90.689**			

They trained on 10x data and longer…

XLNet: Experiments

A fair comparison to BERT: still consistent gains but not impressive as before

Model	SQuAD1.1	SQuAD2.0	RACE	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
BERT-Large (Best of 3)	86.7/92.8	82.8/85.5	75.1	87.3	93.0	91.4	74.0	94.0	88.7	63.7	90.2
XLNet-Large-wikibooks	88.2/94.0	85.1/87.8	77.4	88.4	93.9	91.8	81.2	94.4	90.0	65.2	91.1

XLNet vs RoBERTa

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

SQuAD2.0	EM	F1	SQuAD1.1	EM	F1
Dev set results (single model)					
BERT [10]	78.98	81.77	BERT† [10]	84.1	90.9
RoBERTa [21]	86.5	89.4	RoBERTa [21]	88.9	94.6
XLNet	**87.9**	**90.6**	XLNet	**89.7**	**95.1**
Test set results on leaderboard (single model, as of Dec 14, 2019)					
BERT* [10]	80.005	83.061			
RoBERTa [21]	86.820	89.795			
XLNet	**87.926**	**90.689**			

XLNet: ablation studies

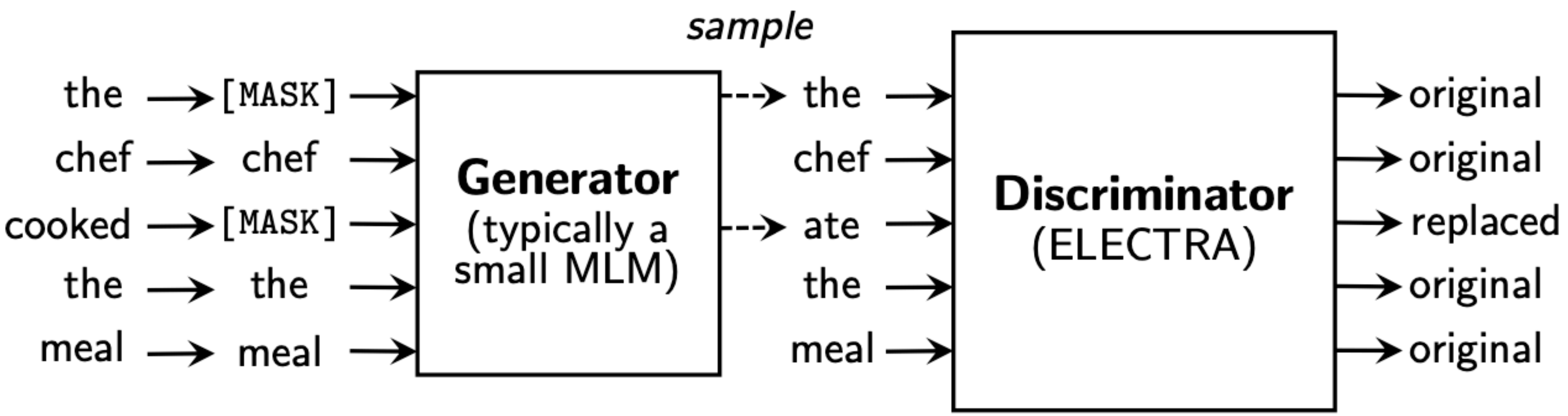
#	Model	RACE	SQuAD2.0 F1	SQuAD2.0 EM	MNLI m/mm	SST-2
1	BERT-Base	64.3	76.30	73.66	84.34/84.65	92.78
2	DAE + Transformer-XL	65.03	79.56	76.80	84.88/84.45	92.60
3	XLNet-Base ($K = 7$)	66.05	**81.33**	**78.46**	**85.84/85.43**	92.66
4	XLNet-Base ($K = 6$)	66.66	80.98	78.18	85.63/85.12	**93.35**
5	- memory	65.55	80.15	77.27	85.32/85.05	92.78
6	- span-based pred	65.95	80.61	77.91	85.49/85.02	93.12
7	- bidirectional data	66.34	80.65	77.87	85.31/84.99	92.66
8	+ next-sent pred	**66.76**	79.83	76.94	85.32/85.09	92.89

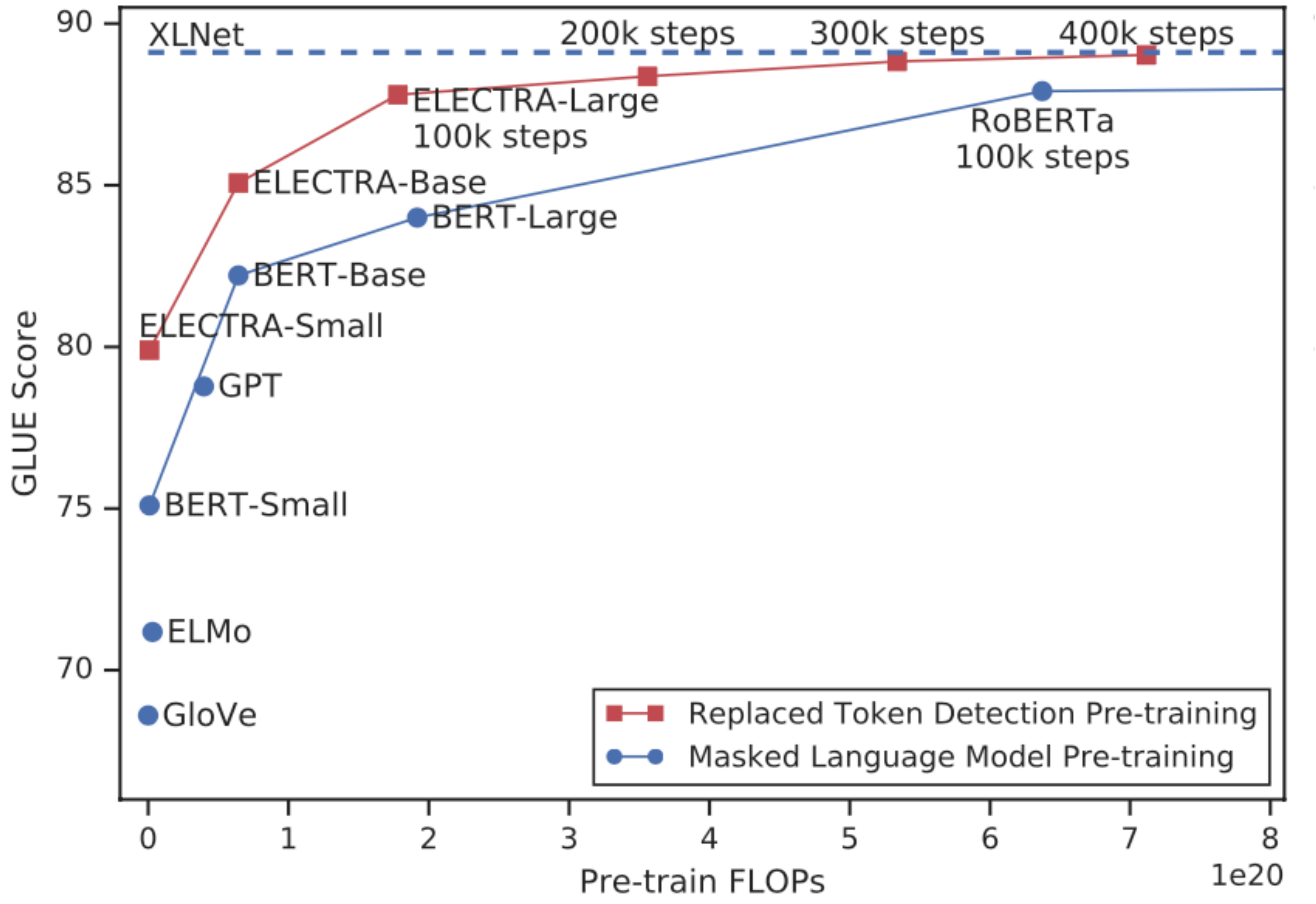
Breakout discussion

- Group 1 (Danqi)
- Group 2 (Chris)
- Group 3 (Kaiyu)
- Group 4 (Shunyu)

- Q1: What are the limitations of BERT that XLNet attempts to solve?
- Q2: How does XLNet address them?
- Q3: What do you think are the key factors that are mostly contributing to the superior performance of XLNet?
- Q4: What are the limitations of XLNet?
- Q5: Anything else you find interesting?

ELECTRA





ELECTRA is a much more efficient training method,
it predicts 100% of tokens (instead of 15%) every time

(Clark et al., 2020): ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators