Not All Memories are Created Equal: Learning to Forget by Expiring

Sainbayar Sukhbaatar, Da JU, Spencer Poff, Stephen Roller, Arthur Szlam, Jason Weston, Angela Fan

ICML 2021

FACEBOOK AI

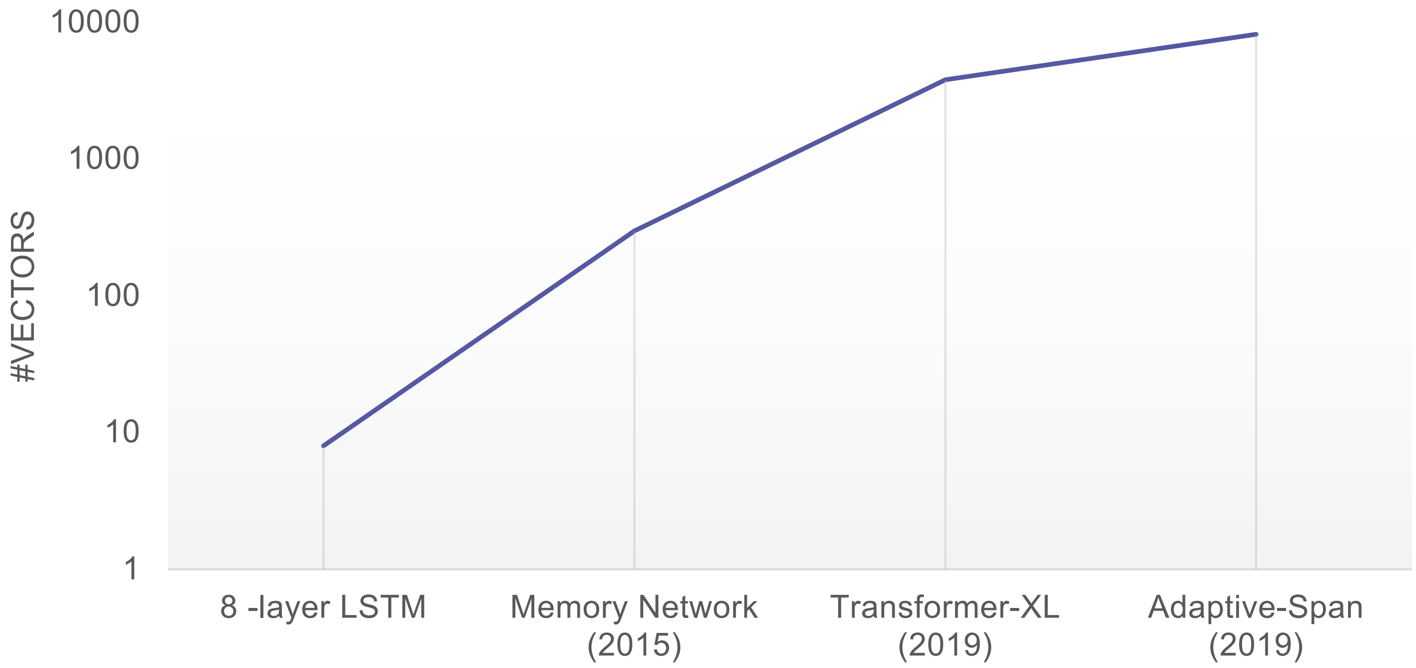
Motivation

- External memory (e.g. Transformer) allows access to **past states**
  - Selective reading via the attention mechanism
  - Important for NLP, Reinforcement Learning

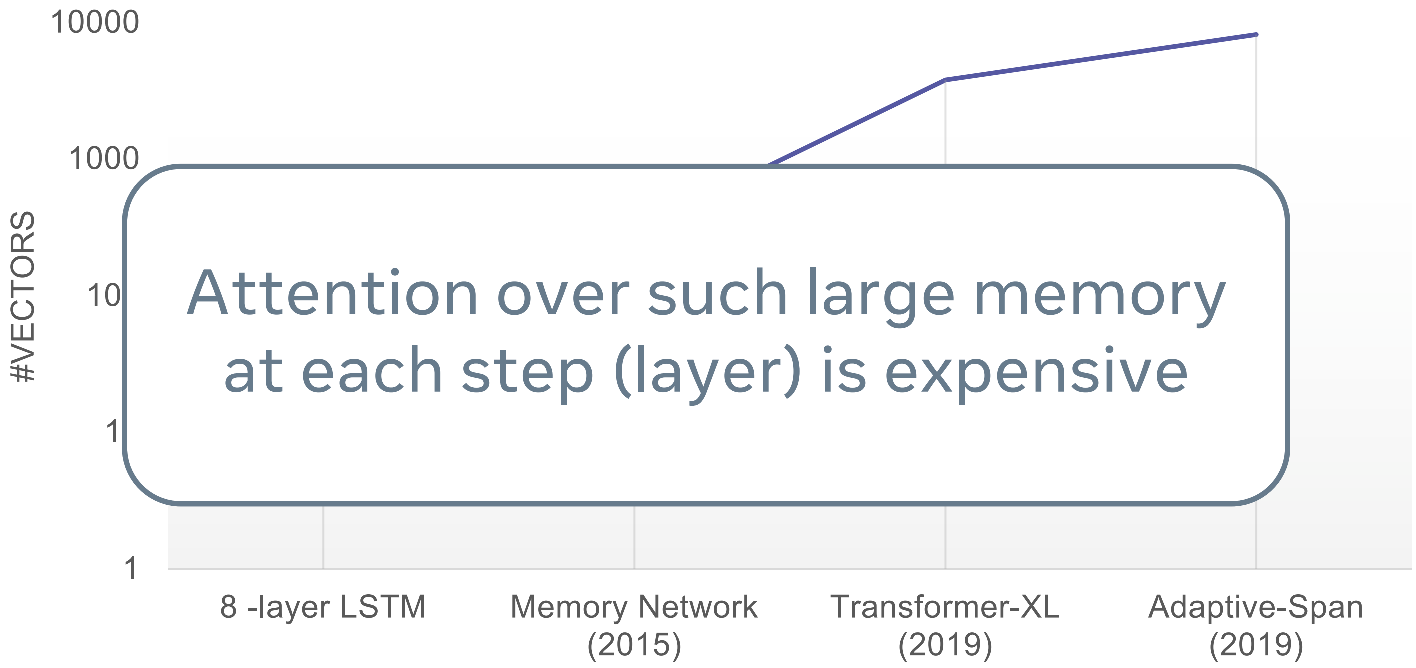
- Scaling problem: all memories stored in the same way
  → irrelevant memories take up space and compute
  → high computational cost when scaling

- Can we learn to **forget irrelevant** memories?

Related I: Memory Size Growth



Related I: Memory Size Growth

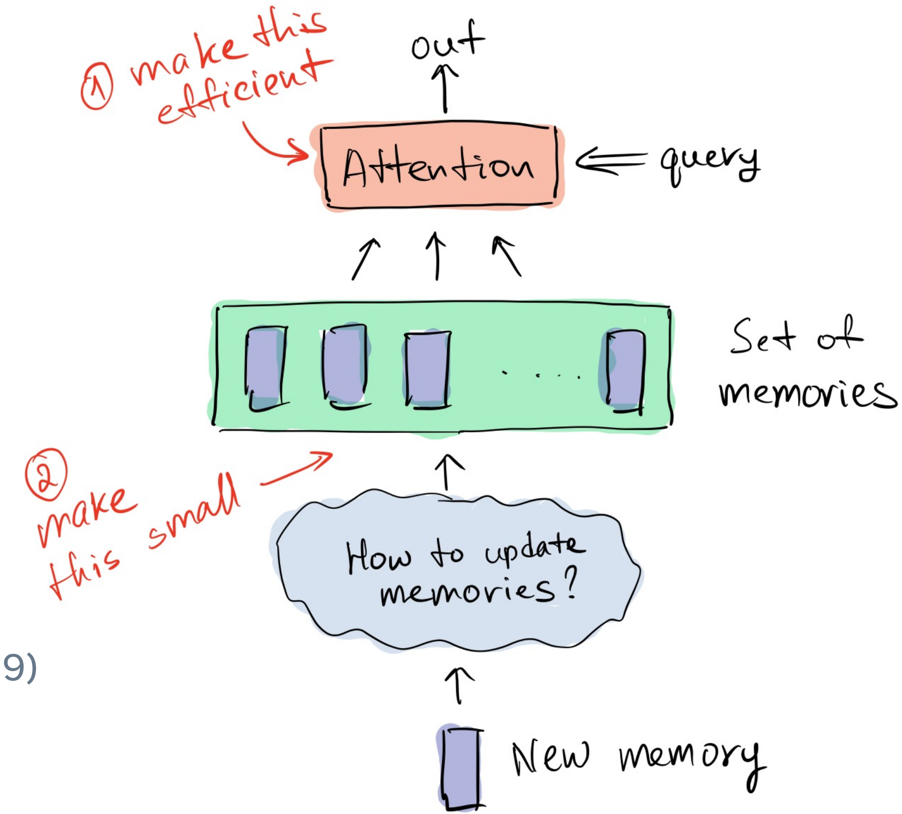


Attention over such large memory at each step (layer) is expensive

Related II: Two (orthogonal) Approaches

1. **Faster search:** given a query, efficiently attend over memories
    - Ex) Routing (Roy et al.) ,
      Linear Trans. (Katharaopoulos et al.),
      Performer (Choromanski et al.),
      Reformer (Kitaev et al.) all in 2020.

2. **Small memory:** keep the number of memories small
    - Transformer-XL (Dai et al., 2019)
    - Adaptive-span (Sukhbaatar et al., 2019)
    - Compressive (Rae et al., 2020)



Related: Reducing Memory Size

Method	How memory is handled	Complexity T tokens
Transformer	Never forgets	$\mathcal{O}(T^2)$
Fixed-span (e.g. Transformer-XL)	Memory is forgotten after L steps	$\mathcal{O}(TL)$ $L \ll T$
Adaptive-span	Learn L from data per layer → most layers have small L'	$\mathcal{O}(TL')$ $L' \ll L$
Compressive Trans.	Merge c memories into a single vector	$\mathcal{O}(TL/c)$

Related: Reducing Memory Size

Method	How memory is handled	Complexity T tokens
Transformer	Never forgets	$\mathcal{O}(T^2)$
Fixed-span (e.g. Transformer-X		$\mathcal{O}(TL)$ $L \ll T$
Adaptive-span		$\mathcal{O}(TL')$ $L' \ll L$
Compressive Trans.	Merge c memories into a single vector	$\mathcal{O}(TL/c)$

All memories are treated equally regardless of their importance!

Method: Expire-Span

Learn to forget irrelevant memories

- Assign an **expiration** date to each memory
  - Depends on context
- Memory is **removed** after that date → free space for important memories
- Memories are gradually decayed → learning by backpropagation

Some equations

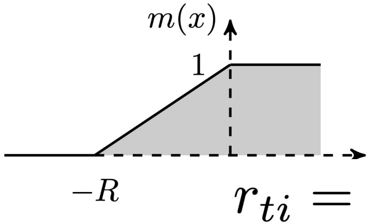
- Compute Expire-spans from the hidden state

$$e_i = L\sigma(\mathbf{w}^\top \mathbf{h}_i + b)$$

- Mask attention weights

$$a'_{ti} = \frac{m_{ti} a_{ti}}{\sum_j m_{tj} a_{tj}}$$

- Soft masking function over the remaining span

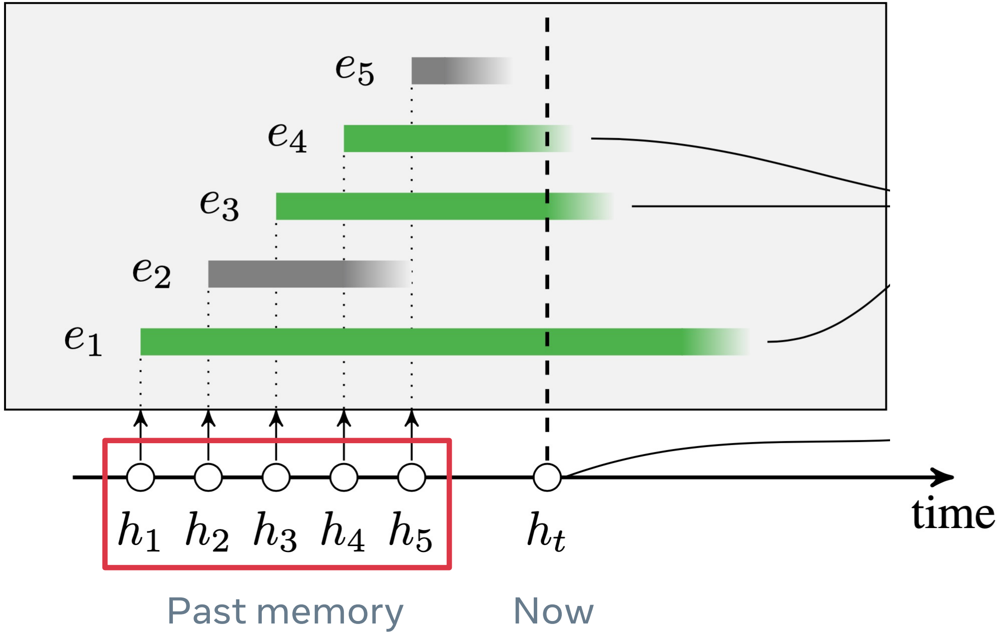


$$r_{ti} = e_i - (t - i)$$

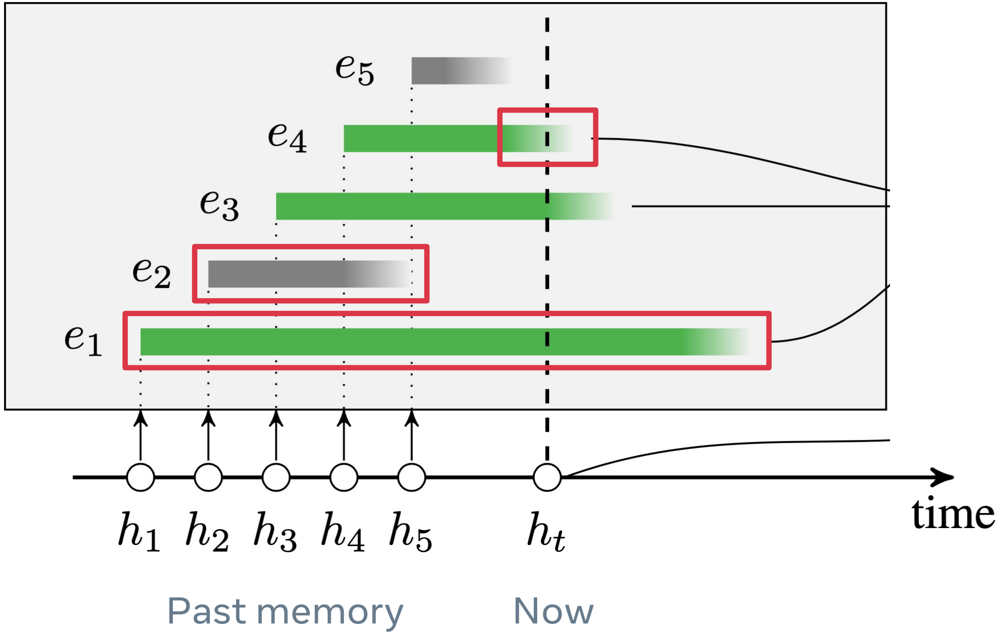
- Auxiliary loss term for reducing the memory size

$$L_{\text{total}} = L_{\text{task}} + \alpha \sum_i e_i / T$$

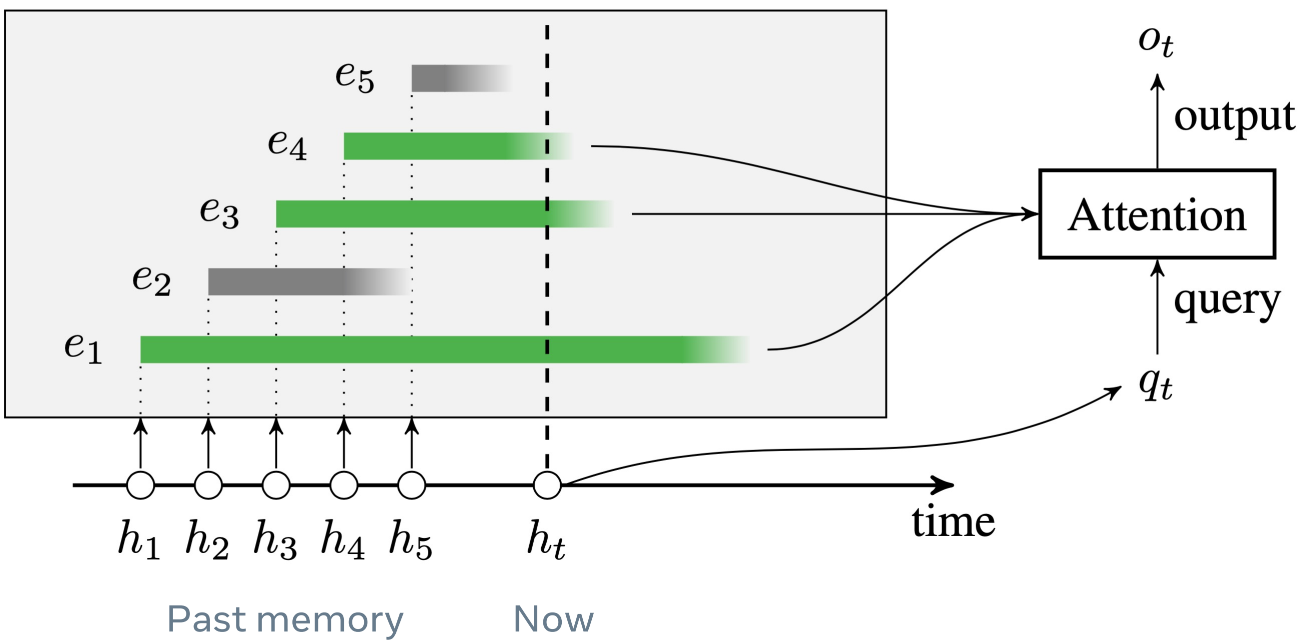
Expire-Span example



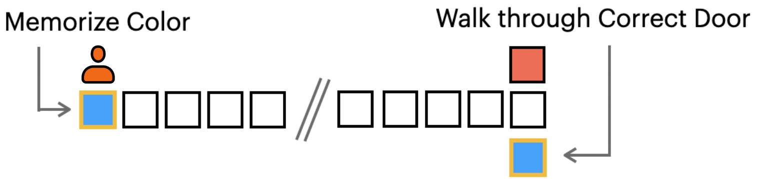
Expire-Span example



Expire-Span example

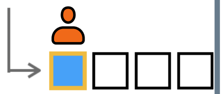


Corridor Task



1. At start, the agent sees a color
2. Cross a long corridor
3. Open the door of the same color

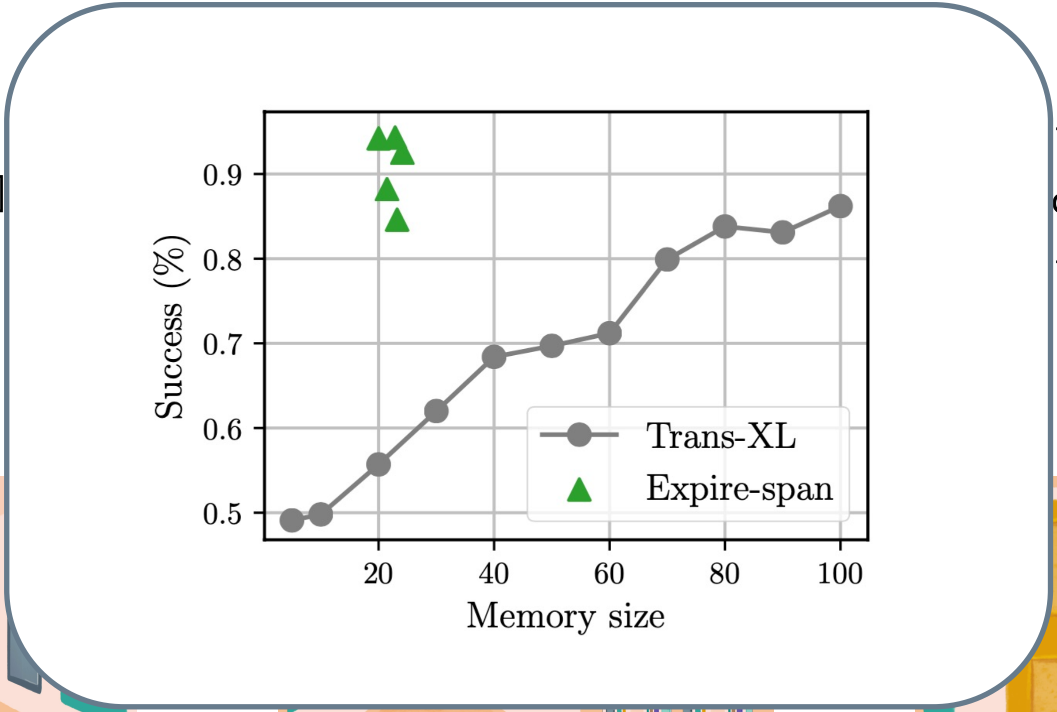
Corridor Task

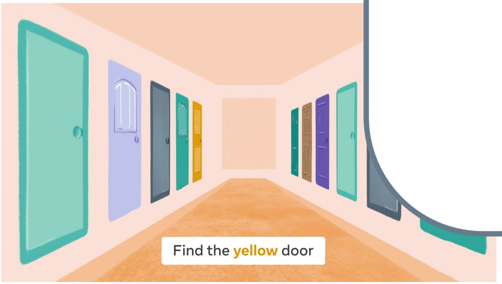


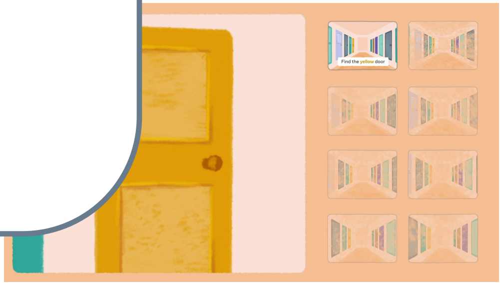
t sees a color

dor

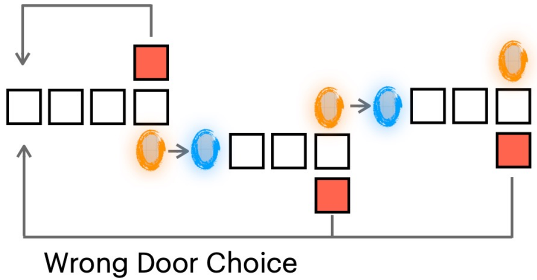
the same color

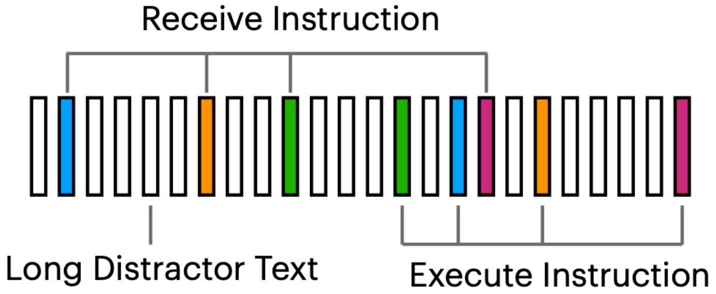


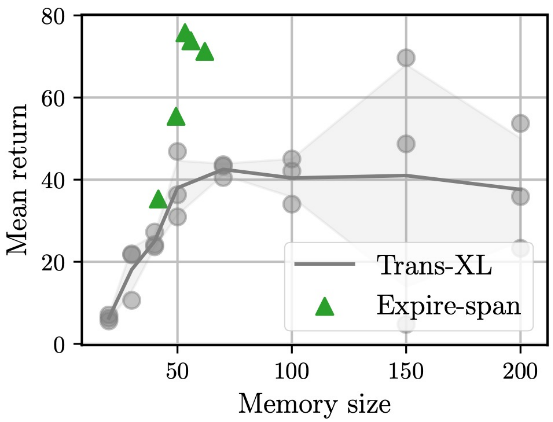


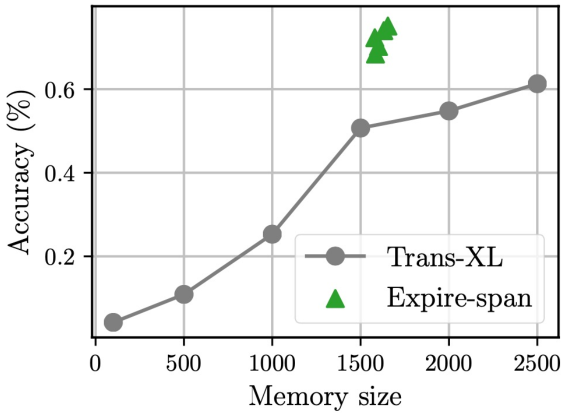


Portal and Instruction Tasks

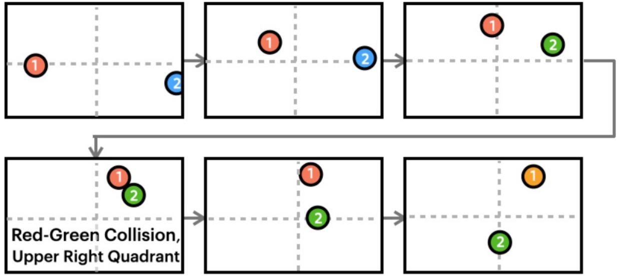


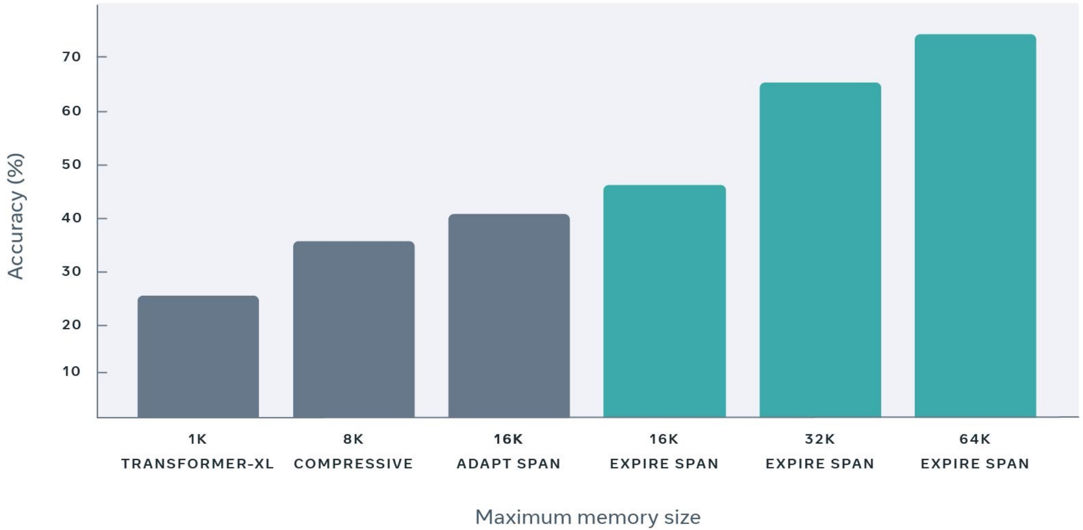






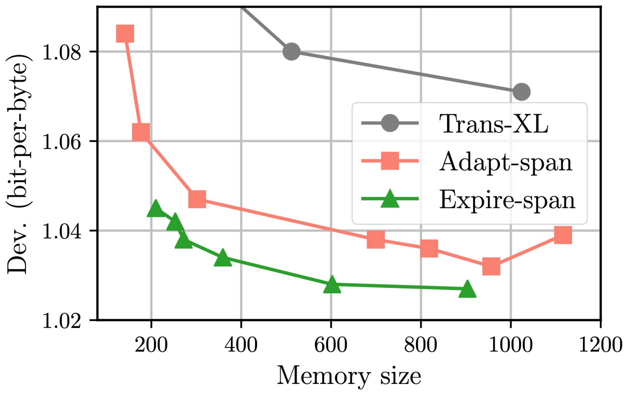
Object Collision Task



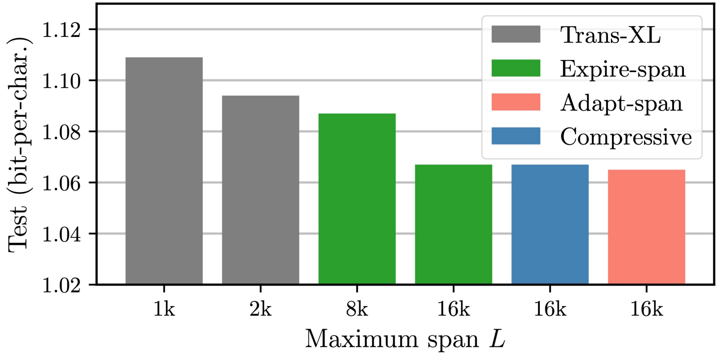


Language Modeling Task

- Character-level Enwik8
  - SoTA performance
  - Spans max=22k mean=1.2k



- Character-level PG19
  - Comparable performance
  - 3x smaller memory size than adaptive-span

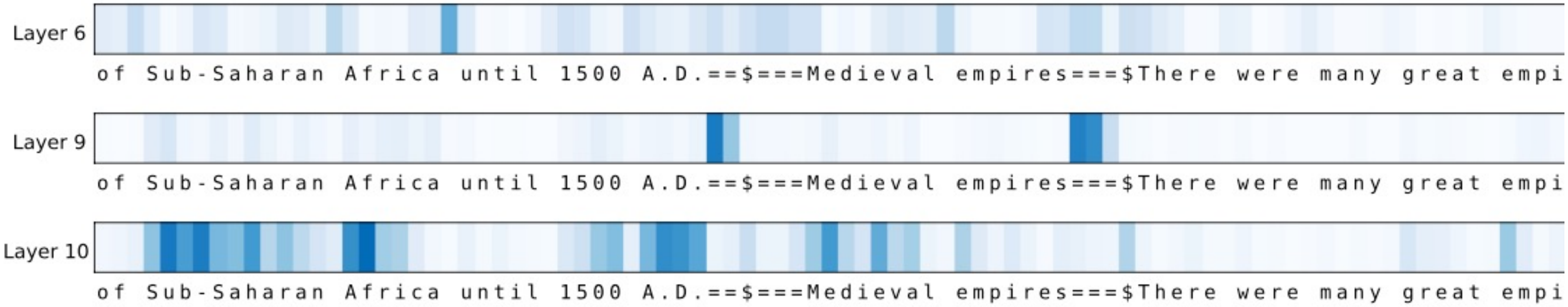


Model efficiency

TASK	MODEL	PERFORMANCE	GPU MEMORY (GB)	TIME/BATCH (MS)
Enwik8	Compressive Transformer	1.05 bpb	21	838
	Adaptive-Span	1.05 bpb	20	483
	Expire-Span	**1.03 bpb**	**15**	**408**
Char-level PG-19	Compressive Transformer	1.07 bpc	17	753
	Adaptive-Span	1.07 bpc	**13**	427
	Expire-Span	**1.07 bpc**	15	**388**
Object collision	Compressive Transformer	63.8% error	**12**	327
	Adaptive-Span	59.8% error	17	365
	Expire-Span	**52.2% error**	**12**	**130**

Expiration in Expire-Span

Different Layers focus on different things



Expire-spans at different layers (enwik8):

Layer 6: space tokens have long spans → word-level

Layer 9: newlines, section titles → sentence, section level

Layer 10: named entities

Conclusion

- A new method for learning to forget at scale
  - What to forget is learnt from data itself
  - End-to-end training with backpropagation
- Successful forgetting Reinforcement Learning tasks
- In real-world Language Modeling tasks
  - Most memories can be forgotten
  - Improved efficiency and performance

Thank You