General-purpose, long-context autoregressive modeling with

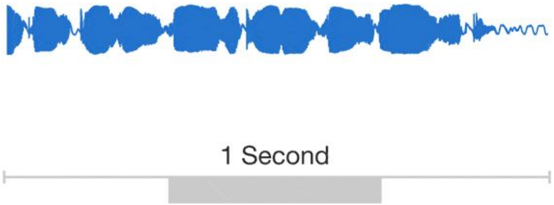
Perceiver AR

Curtis Hawthorne Drew Jaegle Cătălina Cangea Sebastian Borgeaud Charlie Nash Mateusz Malinowski Sander Dieleman

Oriol Vinyals Matt Botvinick Ian Simon Hannah Sheahan Neil Zeghidour Jean-baptiste Alayrac Joao Carreira Jesse Engel

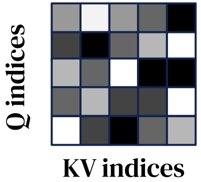
Motivation: Autoregressive Transformer for Long Sequences

- Transformers are great for autoregressive modeling (PaLM, Chinchilla)
- Self-attention is typically $O(n^2)$ in compute and memory
- Real-world sequences are long!
- Example: modeling full pieces of music

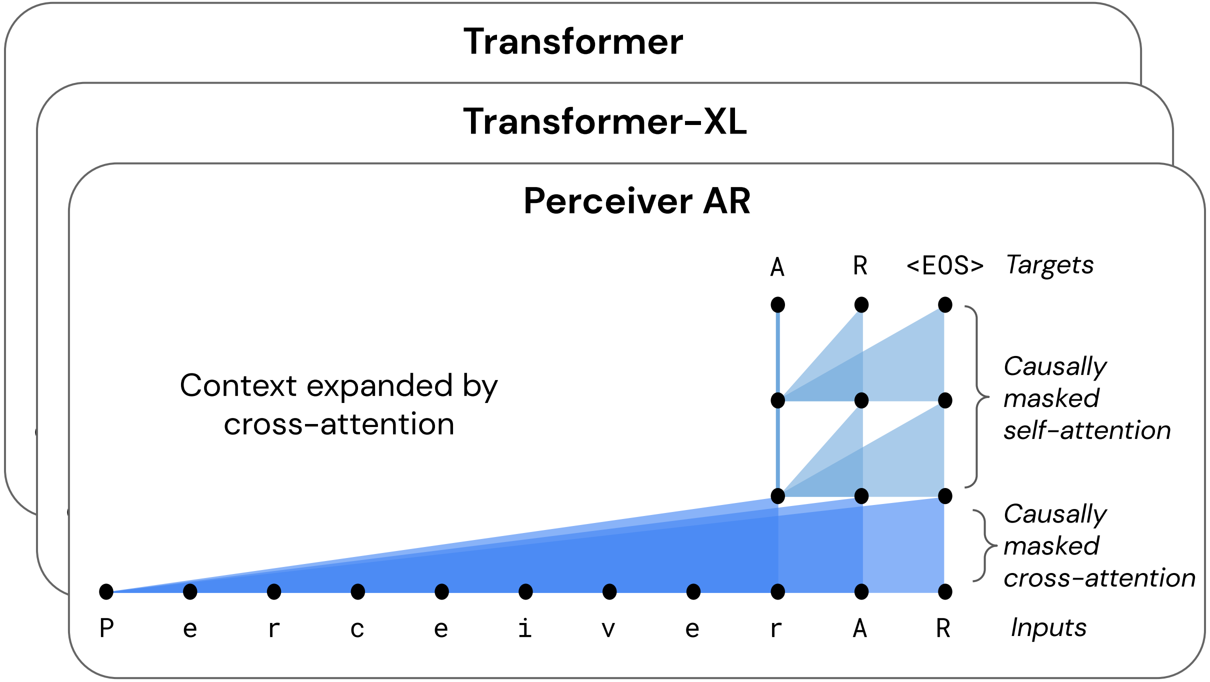


MxM

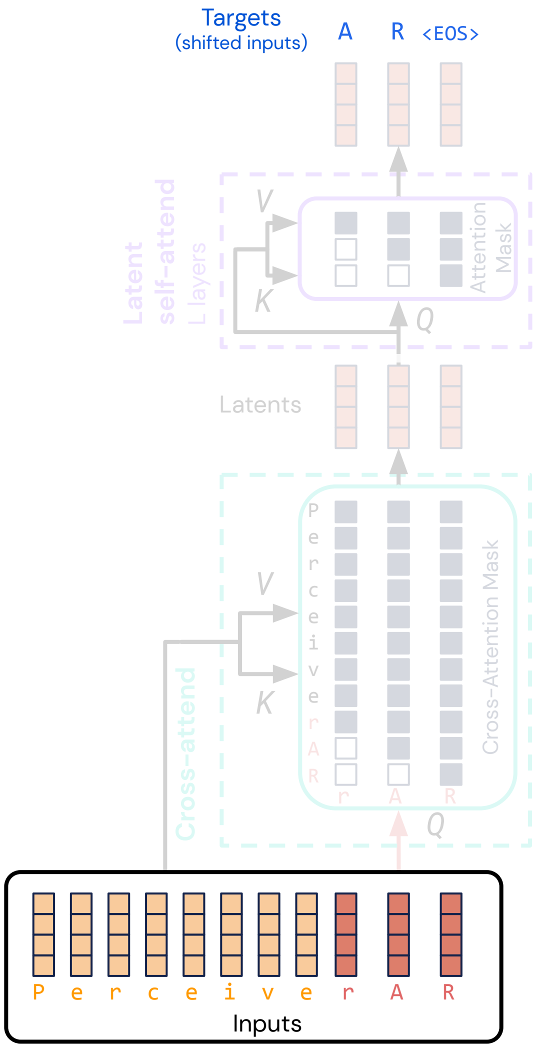
$A=\mathrm{softmax}(QK^T)$



Solution: Decouple sequence length from compute

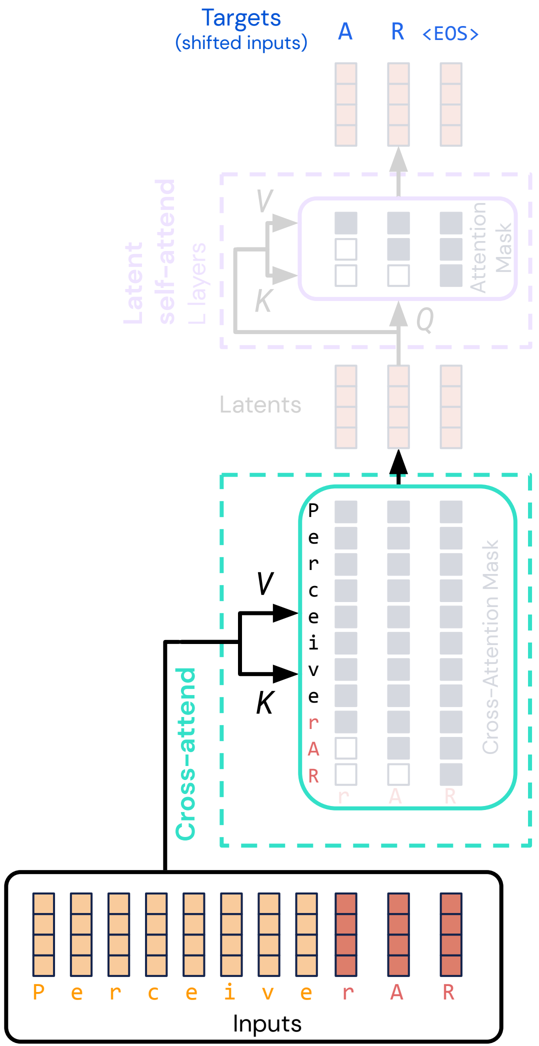


Architecture: Inputs



Architecture: Cross-attend

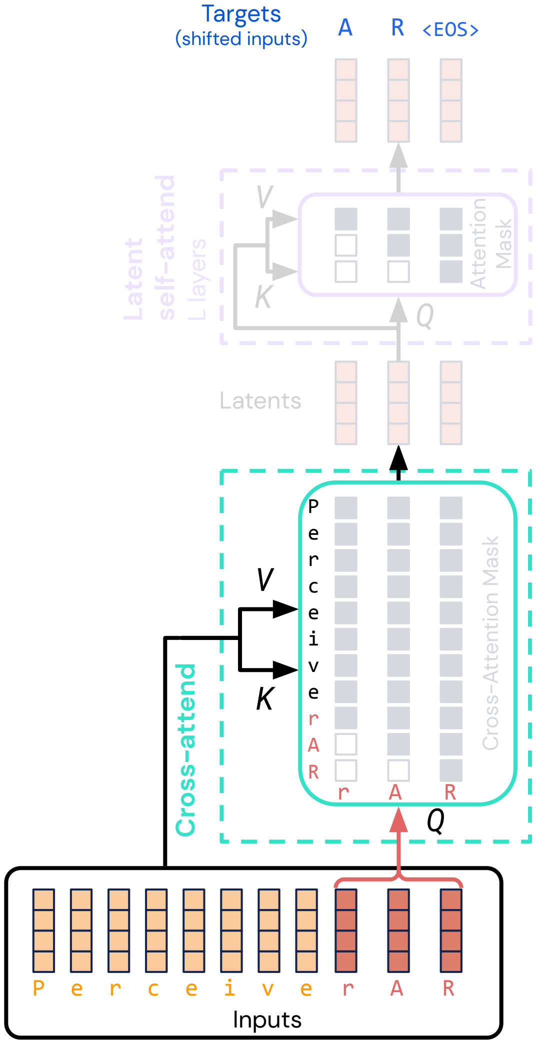
Keys/Values are all M inputs (PerceiverAR)



Architecture: Cross-attend

Keys/Values are all M inputs (`PerceiverAR`)

Queries are last N inputs (`rAR`)



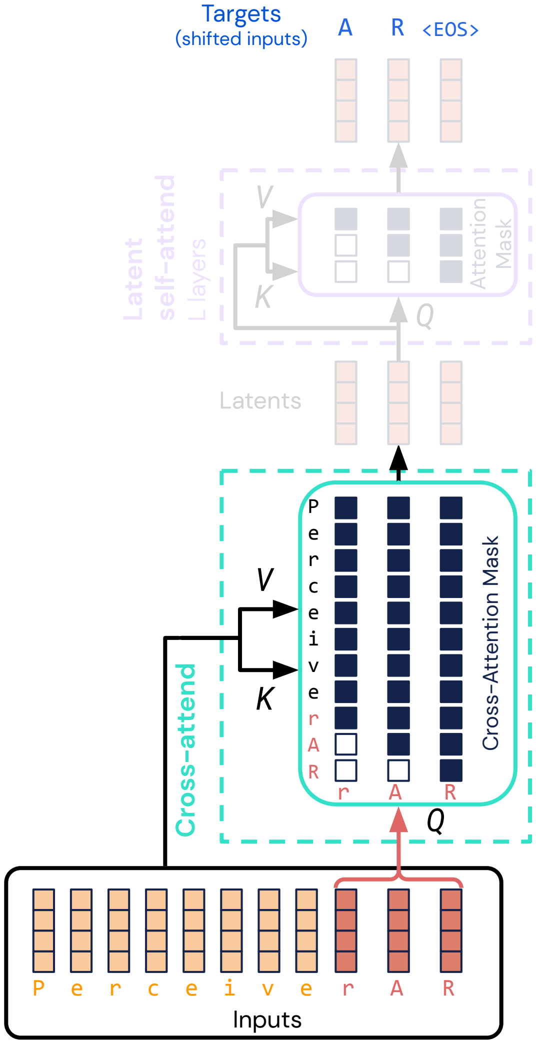
Architecture: Cross-attend

Keys/Values are all M inputs (`PerceiverAR`)

Queries are last N inputs (`rAR`)

Inputs in the “future” are masked

- `A` sees only `PerceiverA`
- `A` predicts `R`



Architecture: Cross-attend

Keys/Values are all M inputs (`PerceiverAR`)

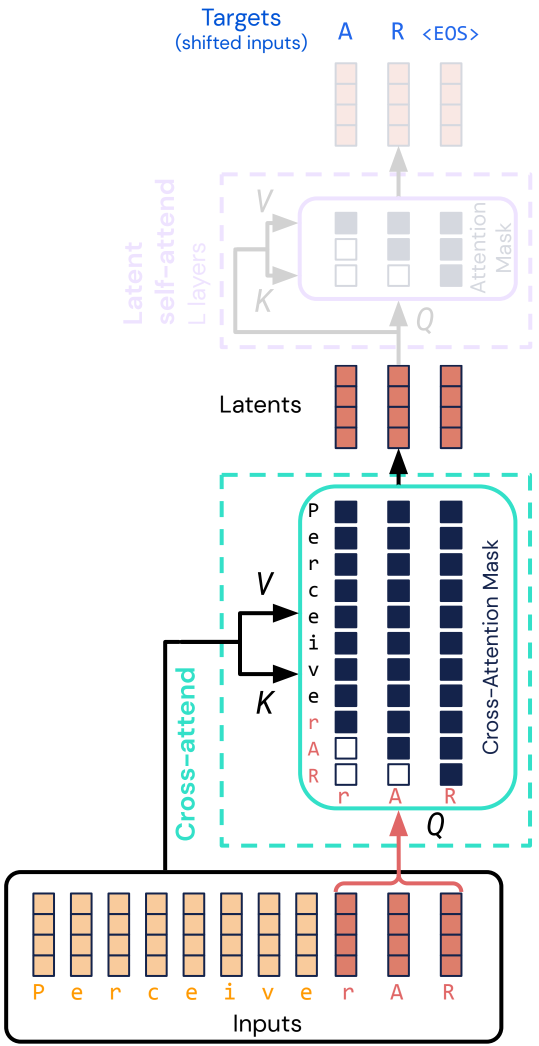
Queries are last N inputs (`rAR`)

Inputs in the “future” are masked

- `A` sees only `PerceiverA`
- `A` predicts `R`

⟶ N “causally correct” latents

⟶ Can self-attend with $O(n^2)$

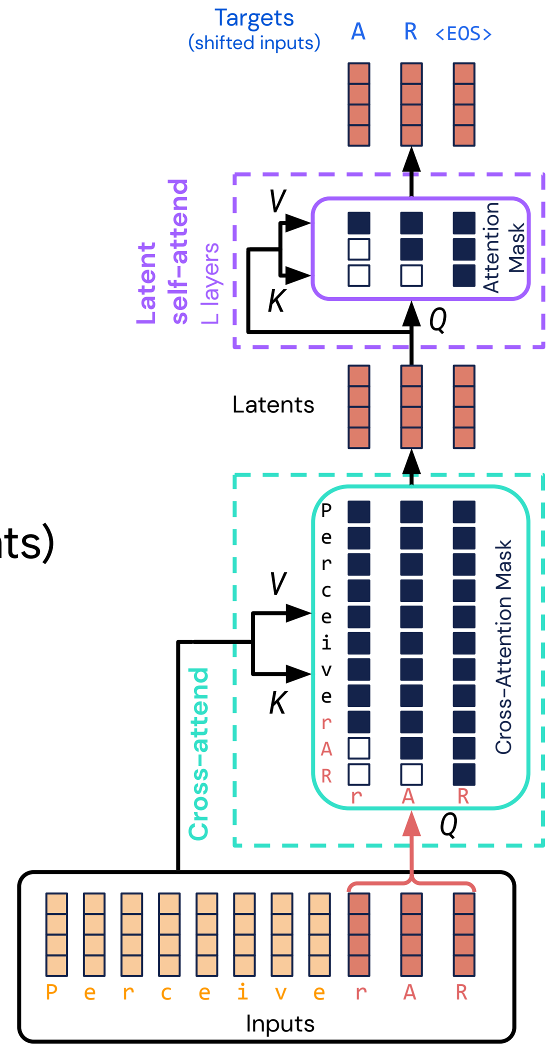


Architecture: Self-attention

Decoder-only style causal masking

Complexity is dependent on N (number of latents)

Independent of M (actual input length)



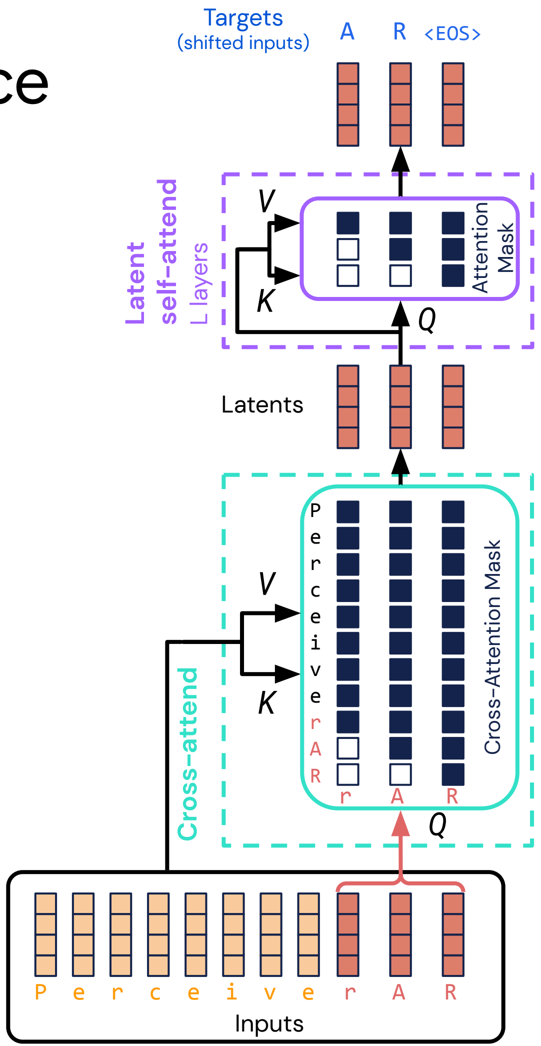
Architecture: Training and Inference

Model outputs cover only last N positions

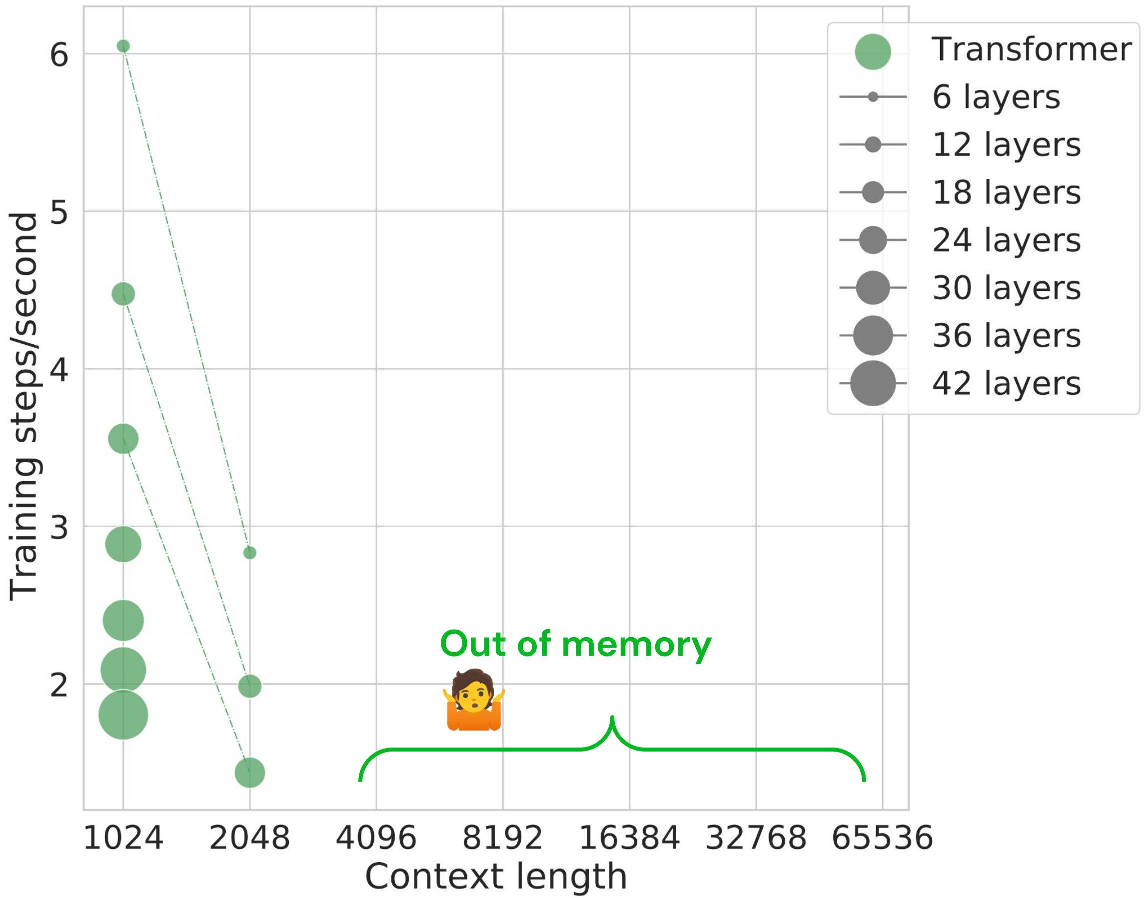
Training: Use random crops

Inference: Queries slide forward

Always cover last N positions



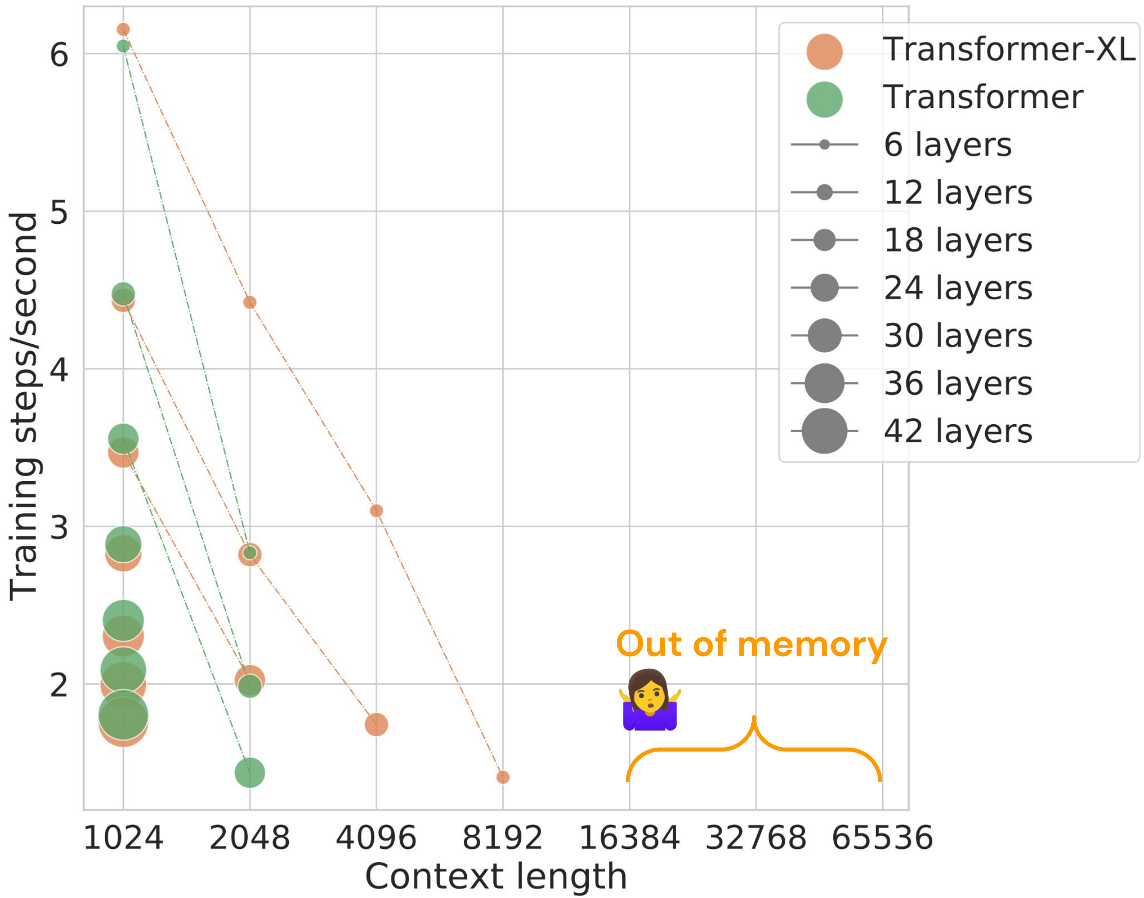
Perceiver AR scales to long contexts and large depth



Results on TPUv3, batch size 1

SPS: measure of real compute requirements

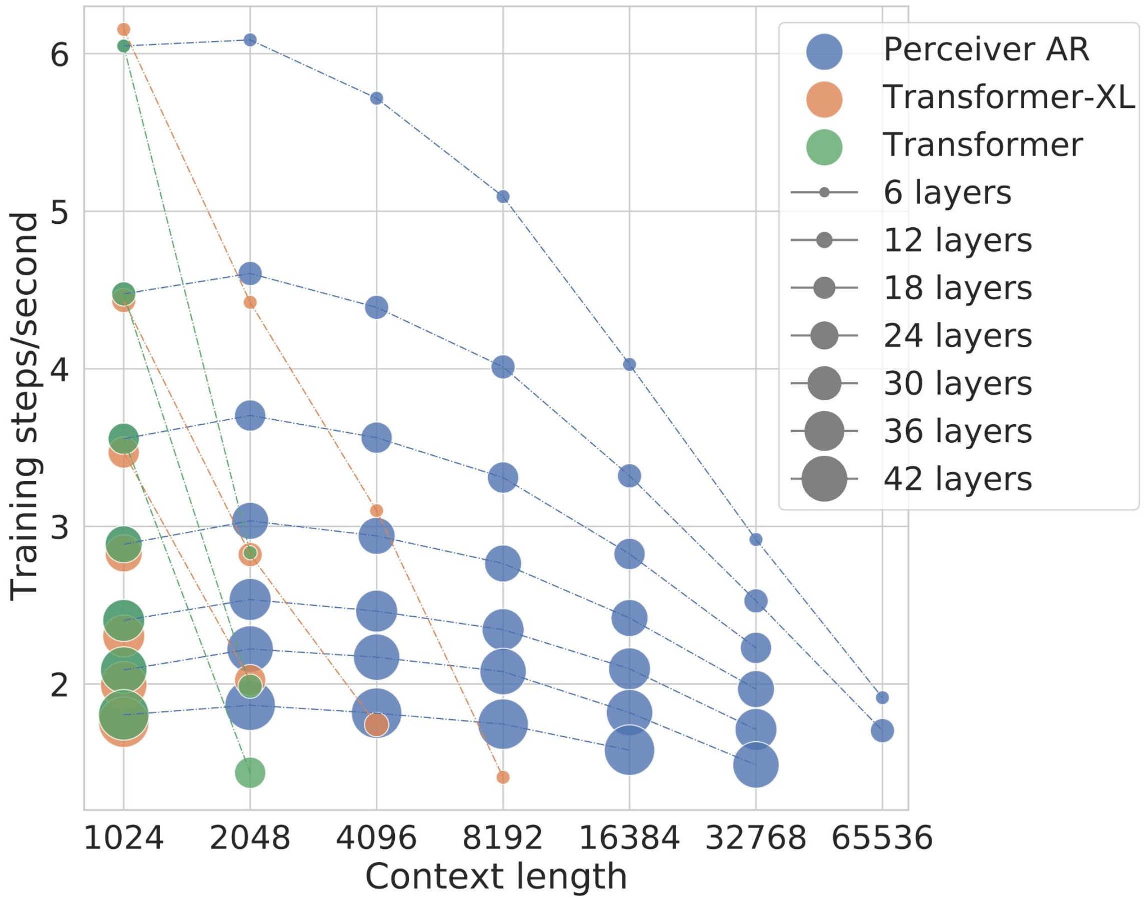
Perceiver AR scales to long contexts and large depth



Results on TPUv3, batch size 1

SPS: measure of real compute requirements

Perceiver AR scales to long contexts and large depth



Results on TPUv3, batch size 1

SPS: measure of real compute requirements

Results on long-context images and text

ImageNet 64x64
64x64x3 = 12,288 elements

Model	Type	Bits/Dim
PixelCNN	AR	3.57
Sparse Transformer	AR	3.44
Routing Transformer	AR	3.43
Combiner	AR	3.42
VDM	**Diff**	**3.40**
Perceiver AR (ours)	**AR**	**3.40**

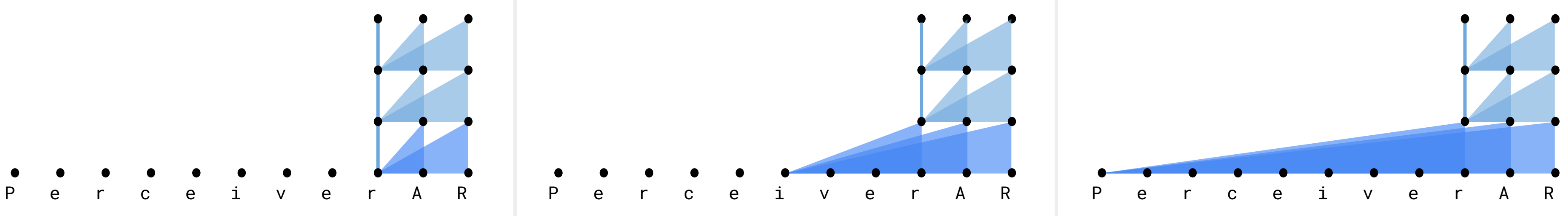
Project Gutenberg books (PG-19)

Model	Context length	# layers	Val ppl.	Test ppl.
Transformer-XL (Rae et al., 2019)	512+1024	36	45.5	36.3
Compressive Transformer (Rae et al., 2019)	512+512+2x512	36	43.4	33.6
Routing Transformer (Roy et al., 2021)	8192	22	-	33.2
Perceiver AR (ours)	2048	60	45.9	**28.9**
Perceiver AR (ours)	4096	60	45.9	29.0

Perceiver overfits for larger context on PG-19 (only ~28k training books)

Context scaling in the large data regime

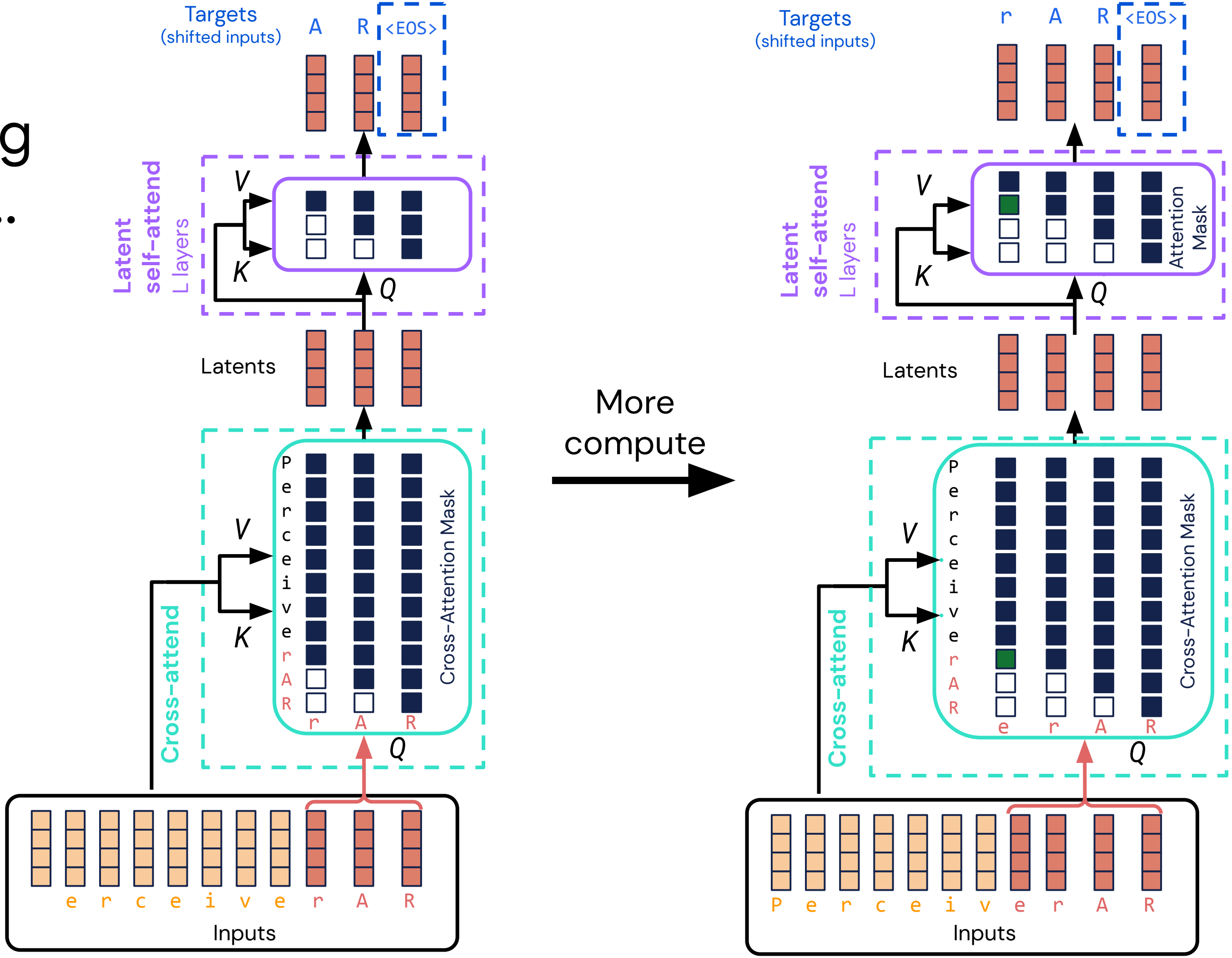
Same parameter count (~500M), expanding context



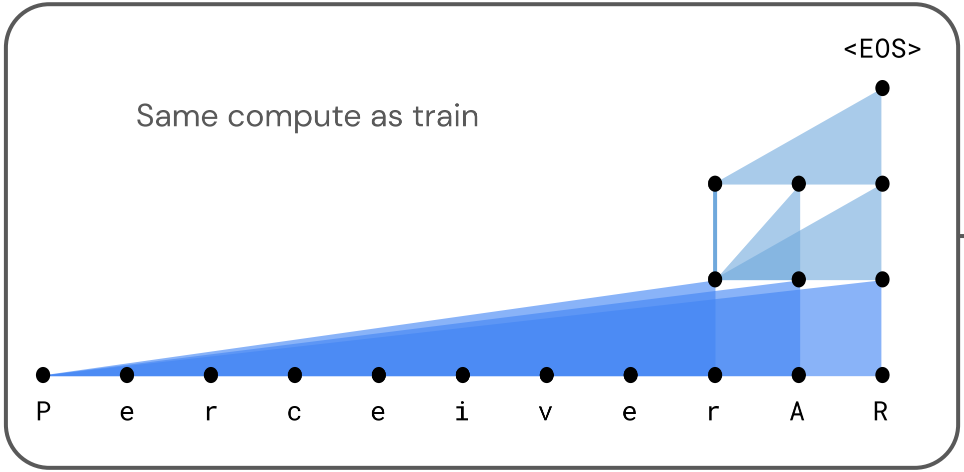
Model	Context	Eval ppl.	Train Steps/sec
Perceiver AR	1024	14.88	2.19
Perceiver AR	4096	14.60	2.09
Perceiver AR	8192	14.57	1.95
Perceiver AR	16384	14.56	1.75

16x longer context at 20% reduced speed.

Without retraining params...



Varying compute at eval

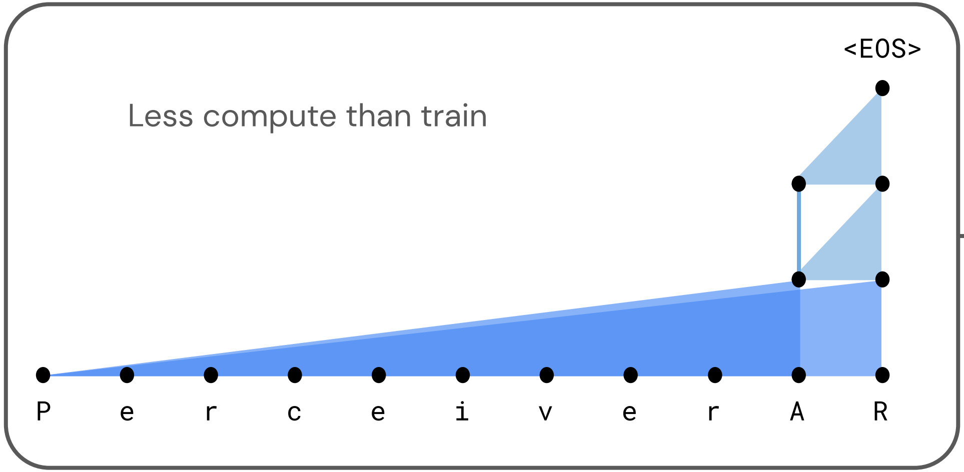


1024 latents

3.402 bits/dim, 3.7 mins/sample

Same parameters always used

Varying compute at eval



16 latents

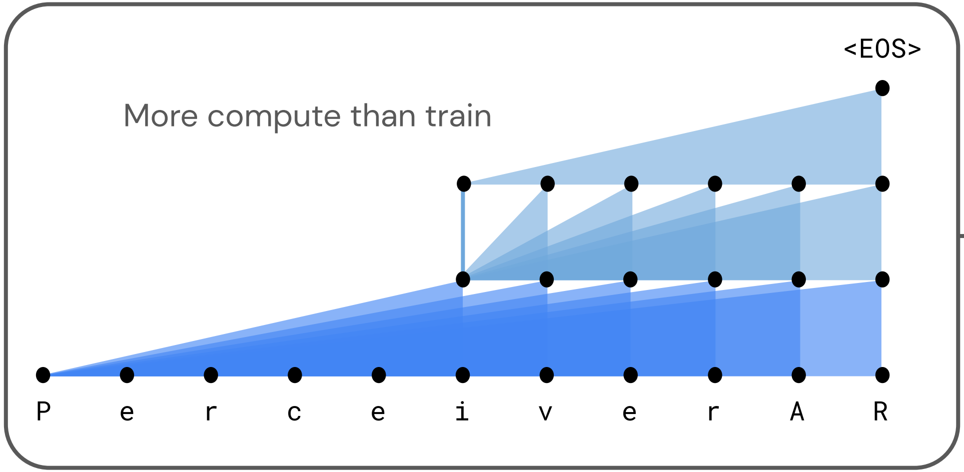
3.576 bits/dim, 2.0 mins/sample

1024 latents

3.402 bits/dim, 3.7 mins/sample

Same parameters always used

Varying compute at eval



16 latents
3.576 bits/dim, 2.0 mins/sample

1024 latents
3.402 bits/dim, 3.7 mins/sample

1536 latents
3.399 bits/dim, 4.7 mins/sample

Same parameters always used

Conclusion

- Retains all the benefits of typical decoder-only Transformers
- Decouples input length from compute/memory requirements
- Demonstrated efficacy across modalities
- Simple to implement
    - Replace bottom self-attend layer with cross-attend

Blog w/ audio examples: magenta.tensorflow.org/perceiver-ar

Author notes: dpmd.ai/dm-perceiver-ar

Code: github.com/google-research/perceiver-ar

Contact: fjord@google.com, drewjaegle@deepmind.com