Reformer: The Efficient Transformer

Khoklin I., Kovrigin A., Shibaev E.

Introduction. Challenges with Traditional Transformer Models

- **Widespread Use and Growing Size of Transformer Models**: Transformer architecture is central to state-of-the-art results in NLP, leading to increasingly larger models.
- **Resource Strain and Accessibility Issues**: These large-scale models demand significant computational resources, to the extent that their training and fine-tuning are often restricted to well-equipped industrial research labs. This trend raises concerns about the sustainability and inclusivity of NLP research, as even basic tasks like fine-tuning cannot be performed on standard hardware setups.

Introduction. Key problems

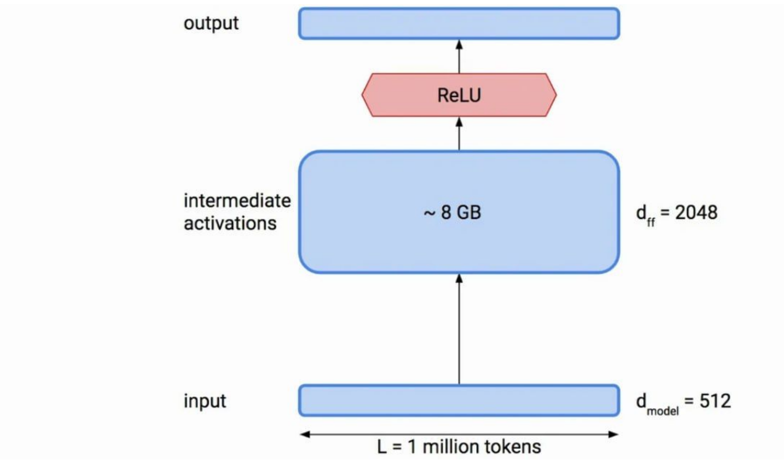
- Memory in a model with N layers is N-times larger than in a single-layer model due to the fact that activations need to be stored for back-propagation.
- Since the depth of intermediate feed-forward layers is often much larger than the depth of attention activations, it accounts for a large fraction of memory use.
- Attention on sequences of length L is $O(L^2)$ in both computational and memory complexity, so even for a single sequence of 64K tokens can exhaust accelerator memory

Introduction. Solutions

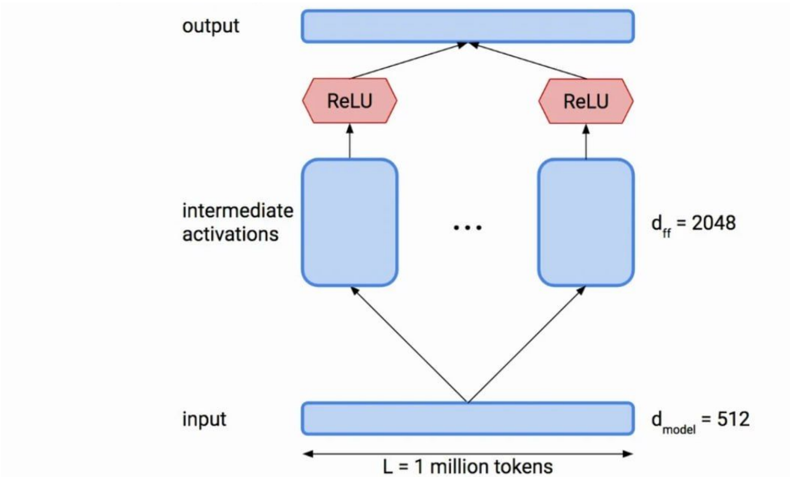
- Reversible layers enable storing only a single copy of activations in the whole model, so the N factor disappears.
- Splitting activations inside feed-forward layers and processing them in chunks saves memory inside feed-forward layers.
- Approximate attention computation based on locality-sensitive hashing replaces the $O(L^2)$ factor in attention layers with $O(L \log L)$ and so allows operating on long sequences.

Method. Chunking

Computations in feed-forward layers are completely independent across positions in a sequence, so the computation can be split into chunks. Operating on one chunk at a time can reduce memory.



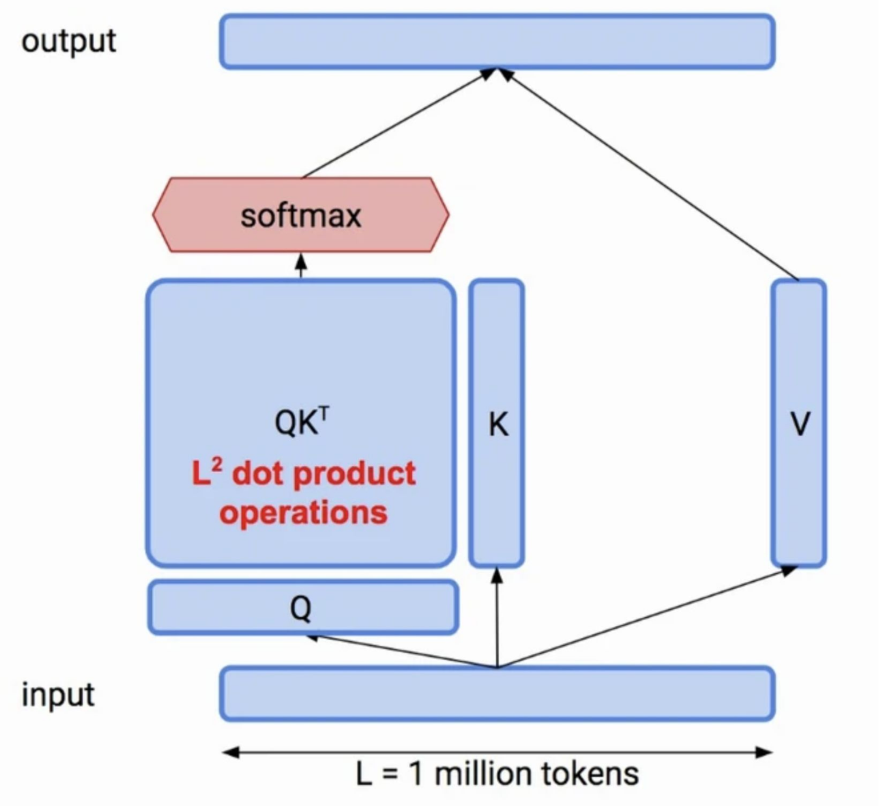
Traditional Transformer



Reformer

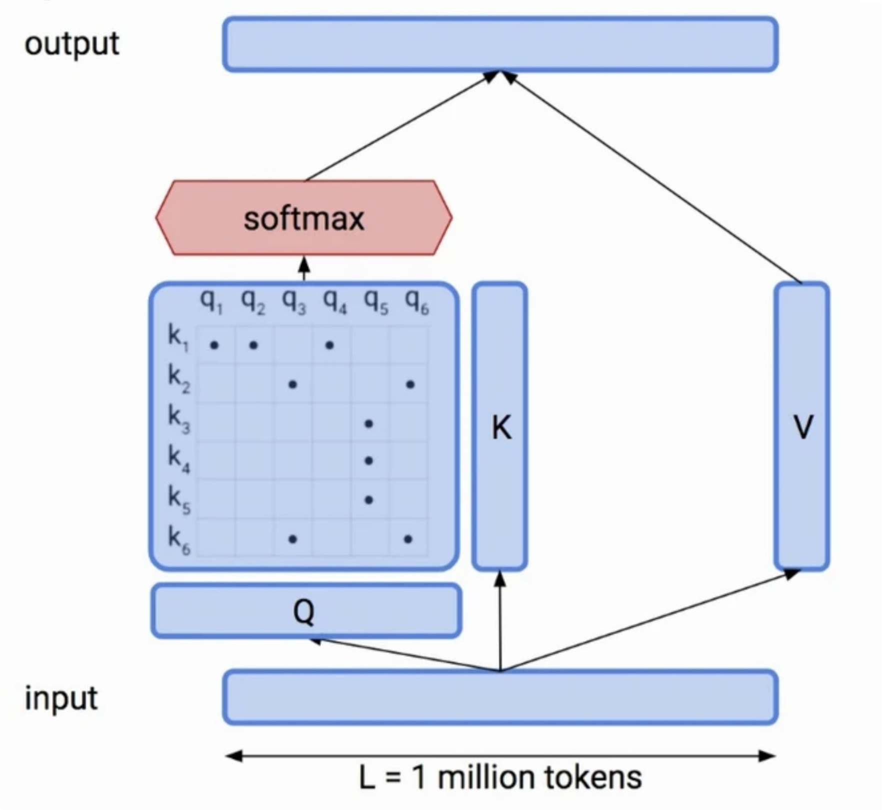
Time Complexity

- The **most time-consuming part** of a transformer is **attention**.
- It's the **only place** requiring **quadratic** time.



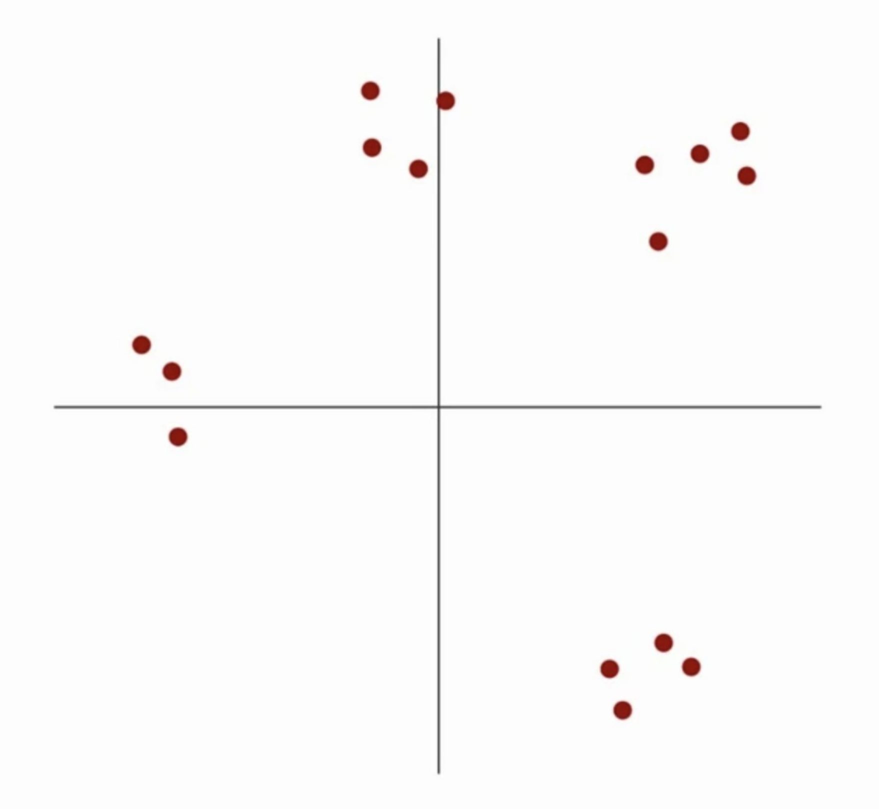
Time Complexity

- The **most time-consuming part** of a transformer is **attention**.
- It's the **only place** requiring **quadratic** time.
- Time wasted: softmax **picks only several** most alike keys for each query.



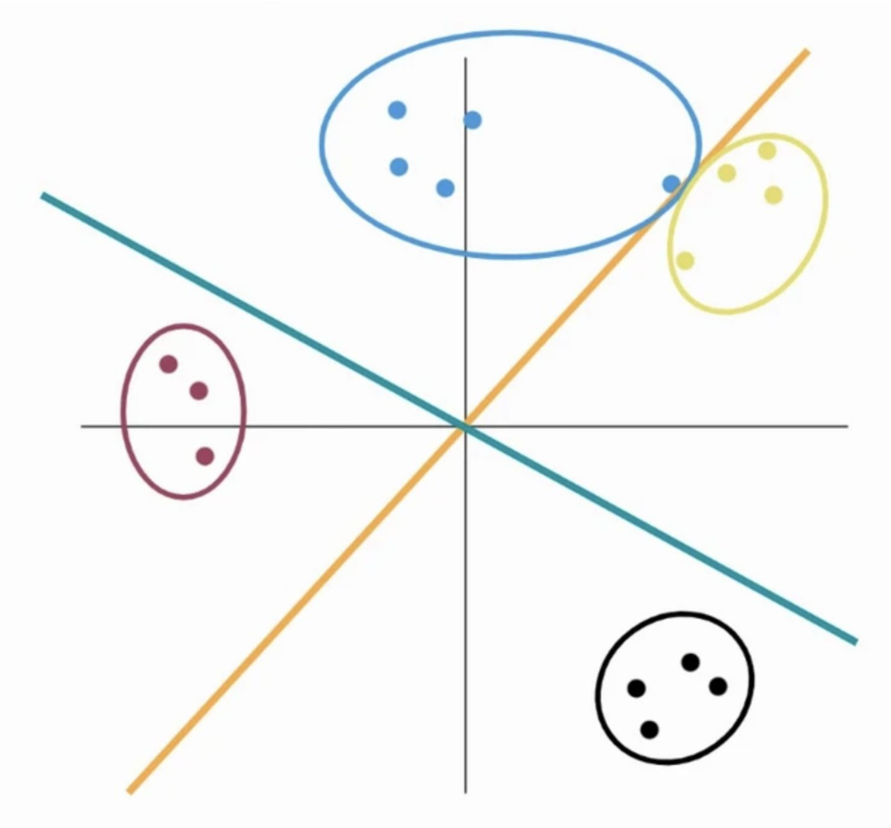
Locality Sensitive Hashing (LSH)

- We want to **attend only alike** vectors together.

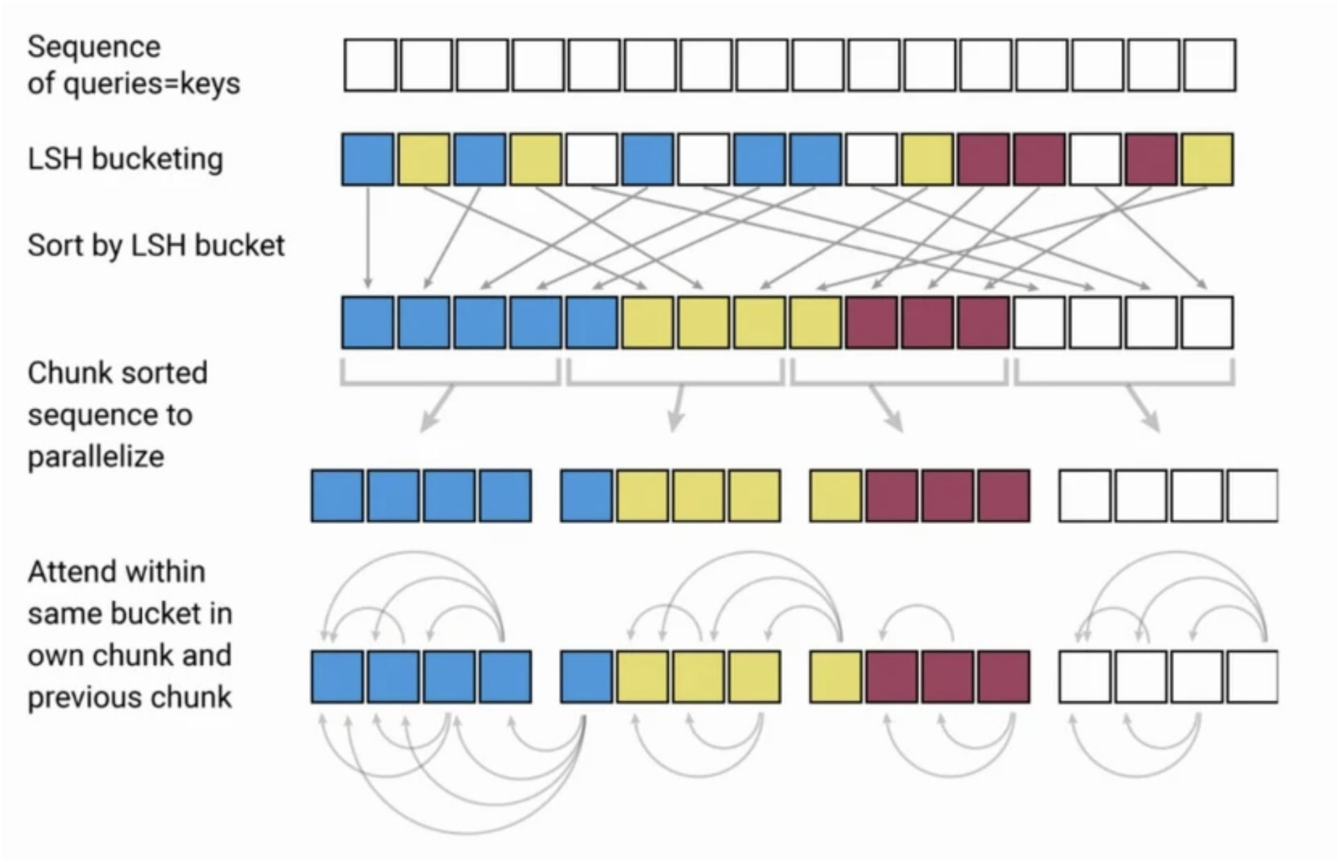


Locality Sensitive Hashing (LSH)

- We want to **attend only alike** vectors together.
- Let's drop some hyperplanes and put the vectors lying in **same parts** in the **same buckets**.



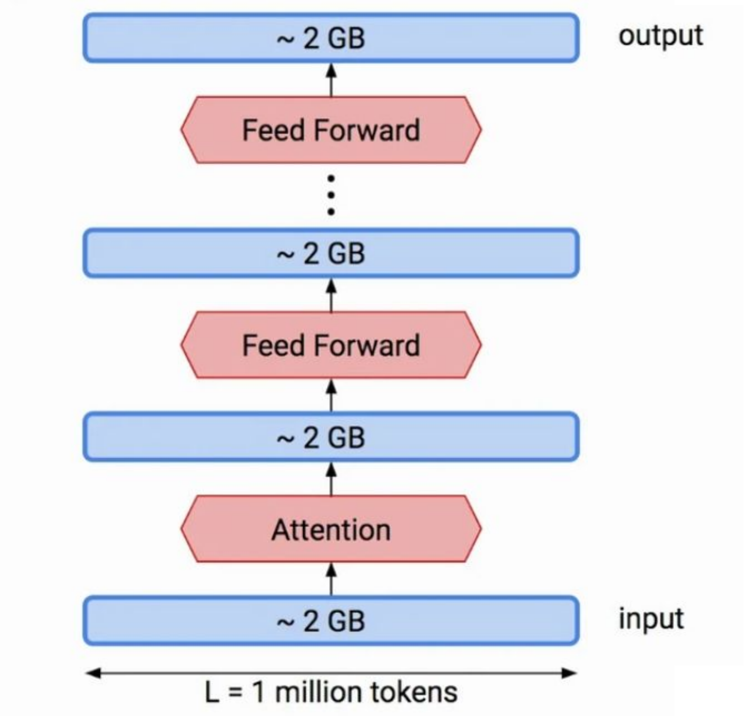
LSH Attention



Activations: Memory Efficiency

- Each layer has b * l * d_model activations
- For **backward propagation** we need to **store all of them**

$$X = X + Attention(X)$$
$$X = X + FeedForward(X)$$

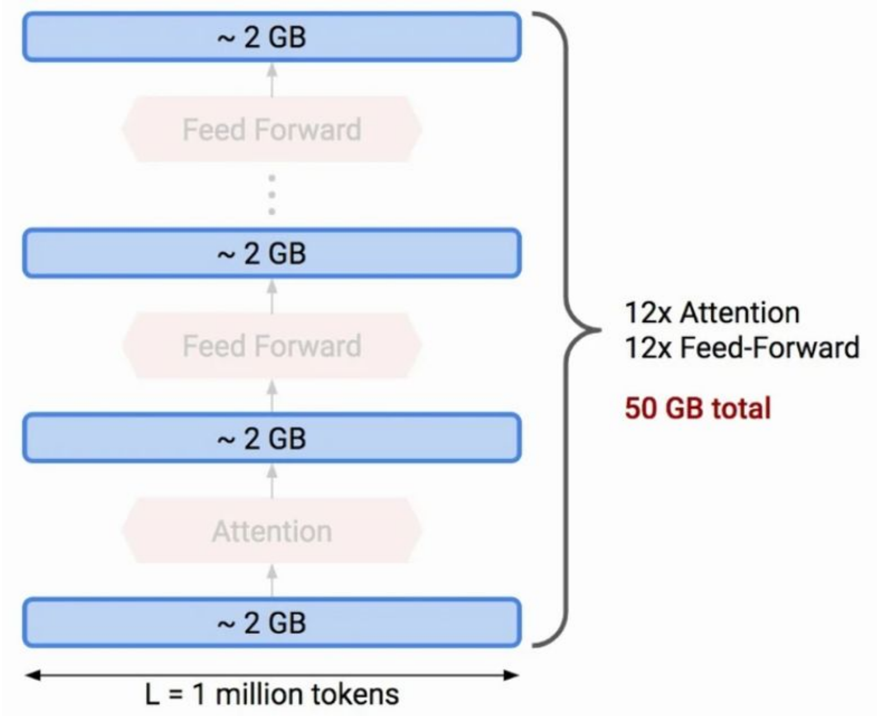


Activations: Memory Efficiency

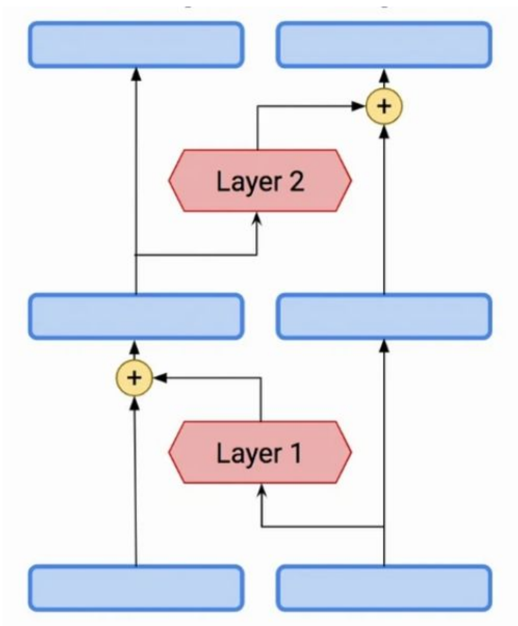
- Each layer has b * l * d_model activations
- For **backward propagation** we need to **store all of them**

$$X = X + Attention(X)$$
$$X = X + FeedForward(X)$$

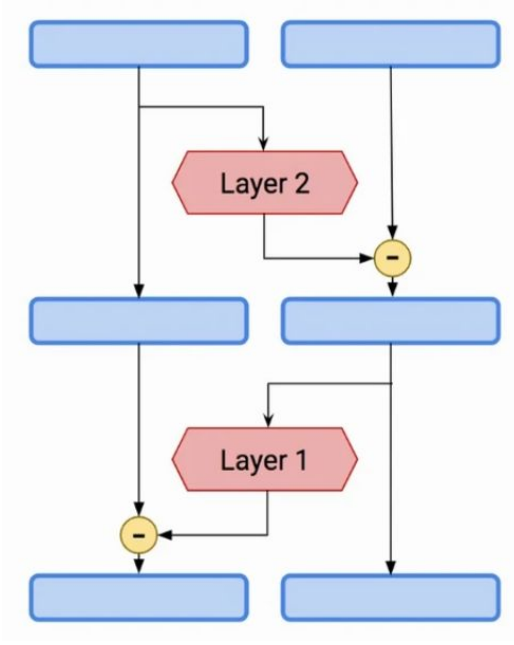
- This means b * l * d_model * n_layers memory!



Reversible Transformer



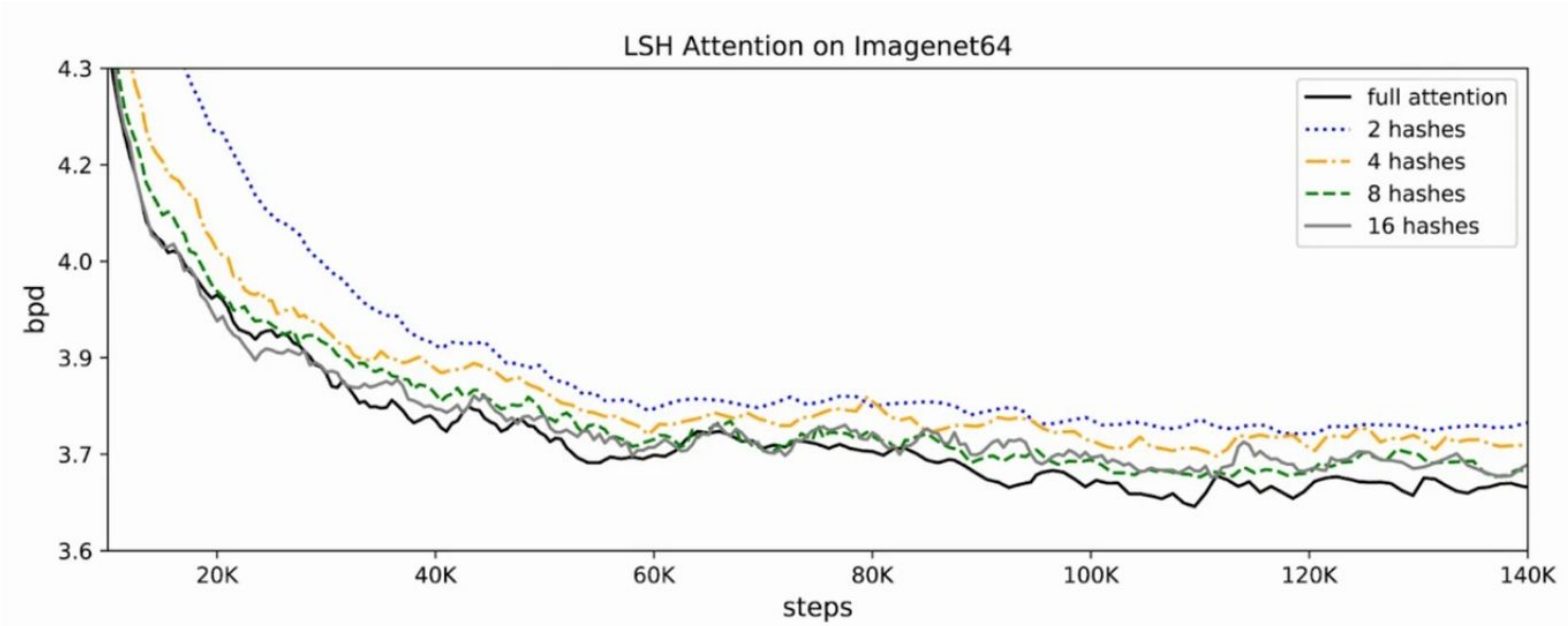
$$Y_1 = X_1 + Attention(X_2)$$
$$Y_2 = X_2 + FeedForward(Y_1)$$



$$X_2 = Y_2 - FeedForward(Y_1)$$
$$X_1 = Y_1 - Attention(X_2)$$

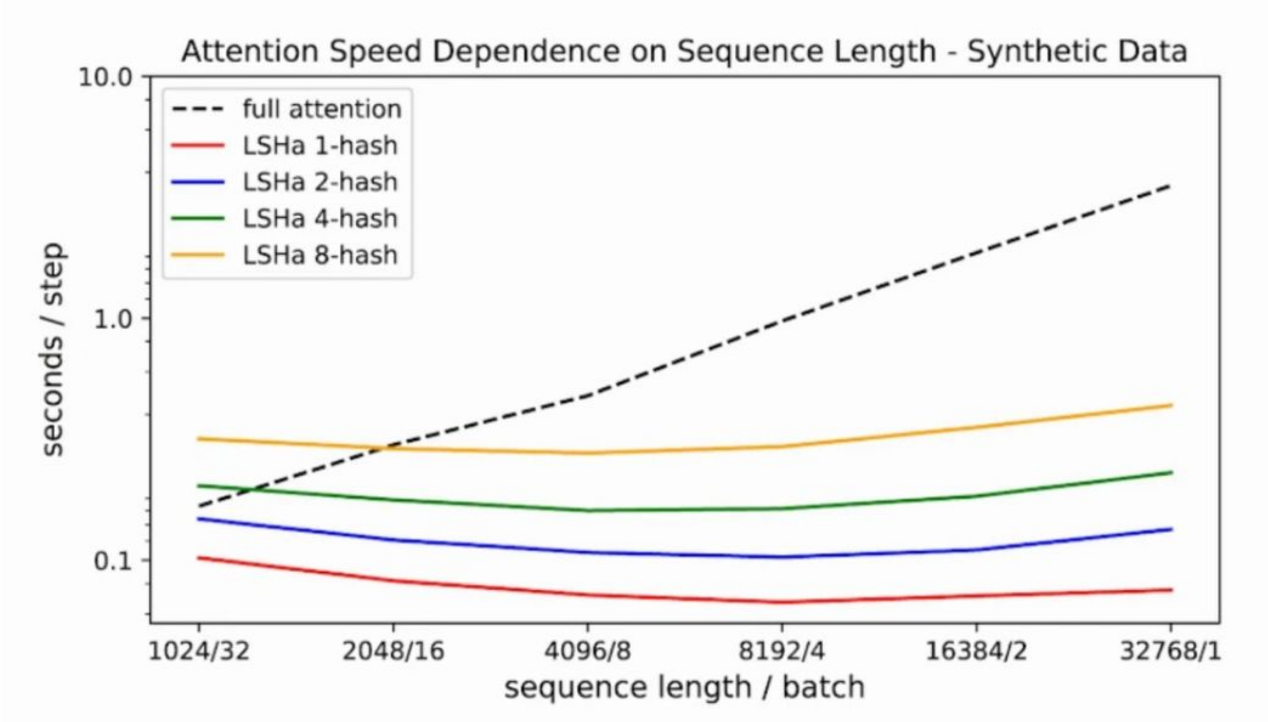
Experiments: LSH Attention Quality

- **Similar model quality** to full attention

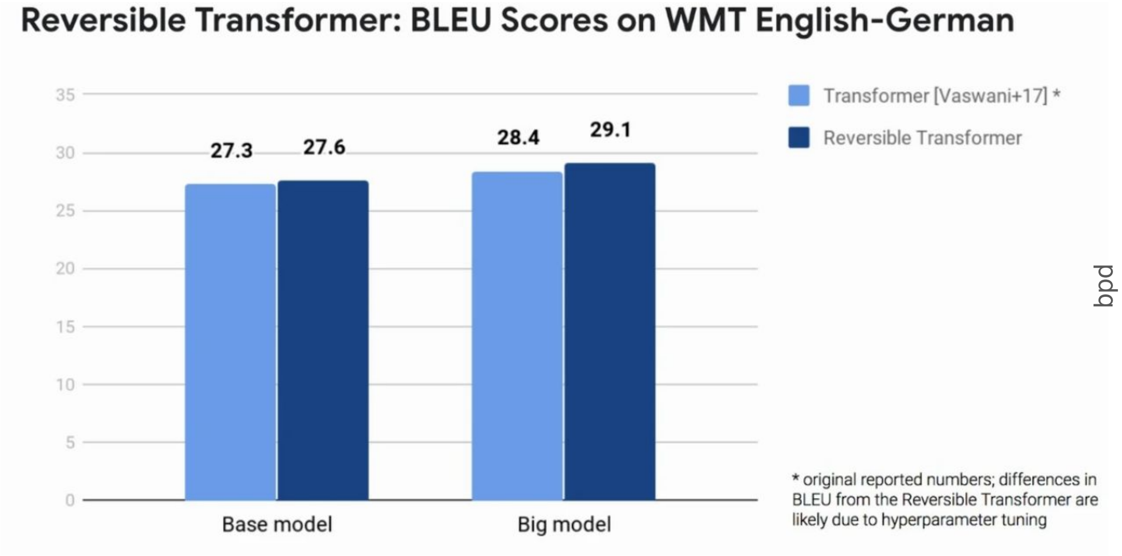


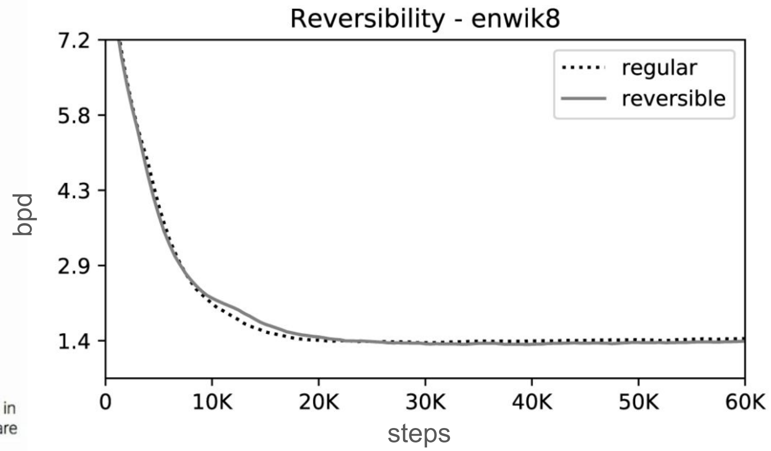
Experiments: LSH Attention Speed

- **Significant speed improvement** over full attention



Experiments: Reversible Transformer





Questions?