Large Memory Layers with Product Keys

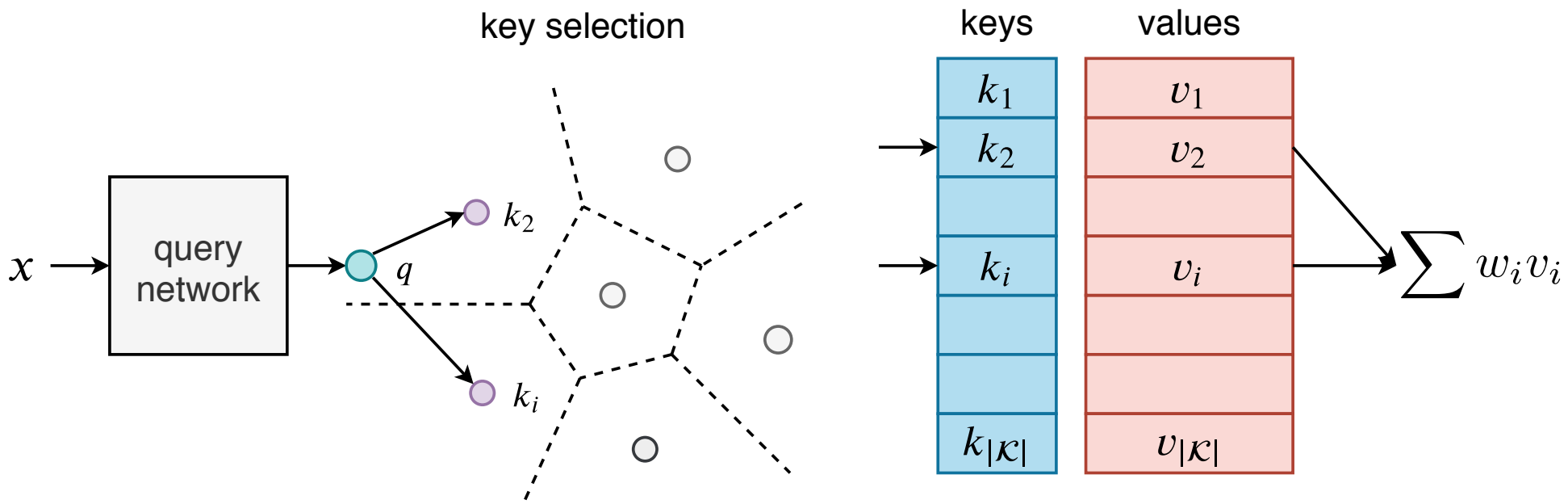
Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou

Code: https://github.com/facebookresearch/XLM

Motivation

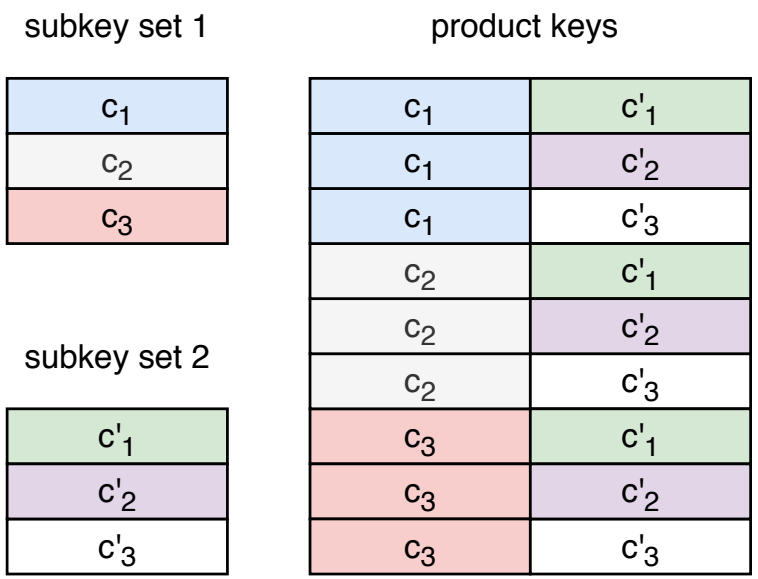
- Neural networks benefit from more capacity:
  - Increase the network depth
  - Increase the network width
- Performance keeps improving with network size
- Problems:
  - Training is too slow, as well as inference
  - Model size is limited by GPU capacity

Product—Key Memory Layers



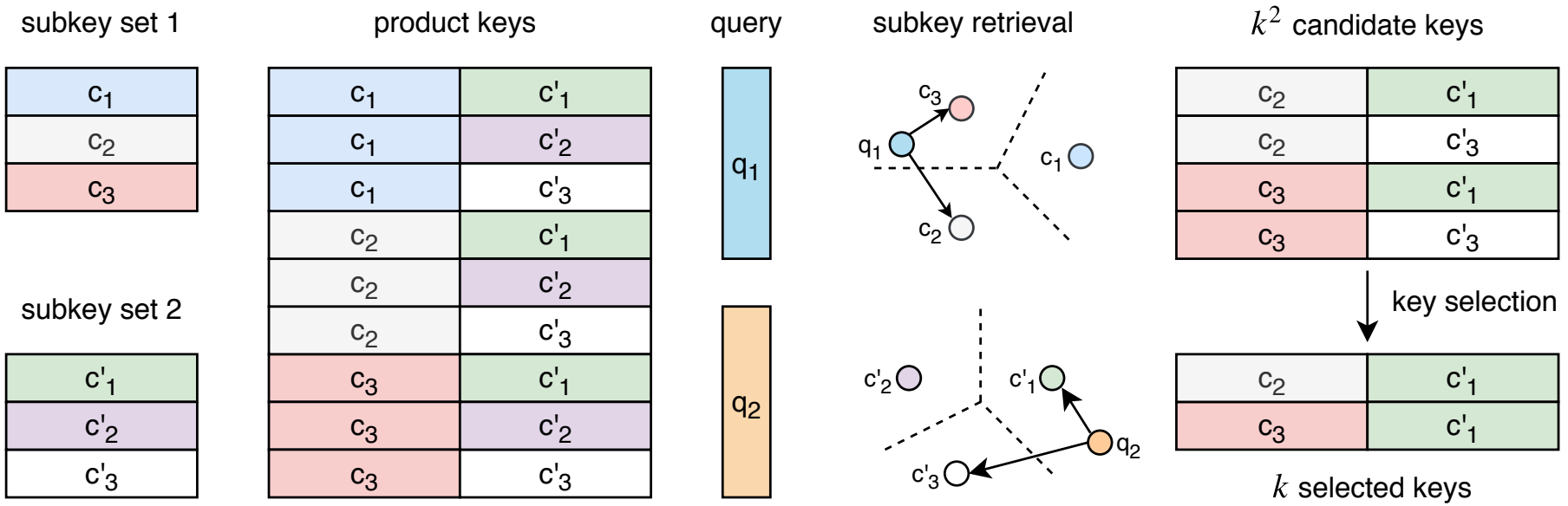
Challenge: the retrieval cost is linear in the number of keys/values

Product—Key Memory Layers



- *subkey set 1* and *subkey set 2* induce a much larger set of keys, never made explicit

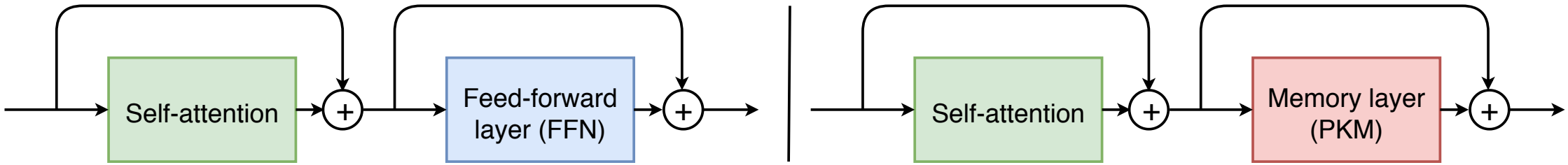
Product—Key Memory Layers



- *subkey set 1* and *subkey set 2* induce a much larger set of keys, never made explicit
- Retrieval over the product keys set can be done extremely efficiently

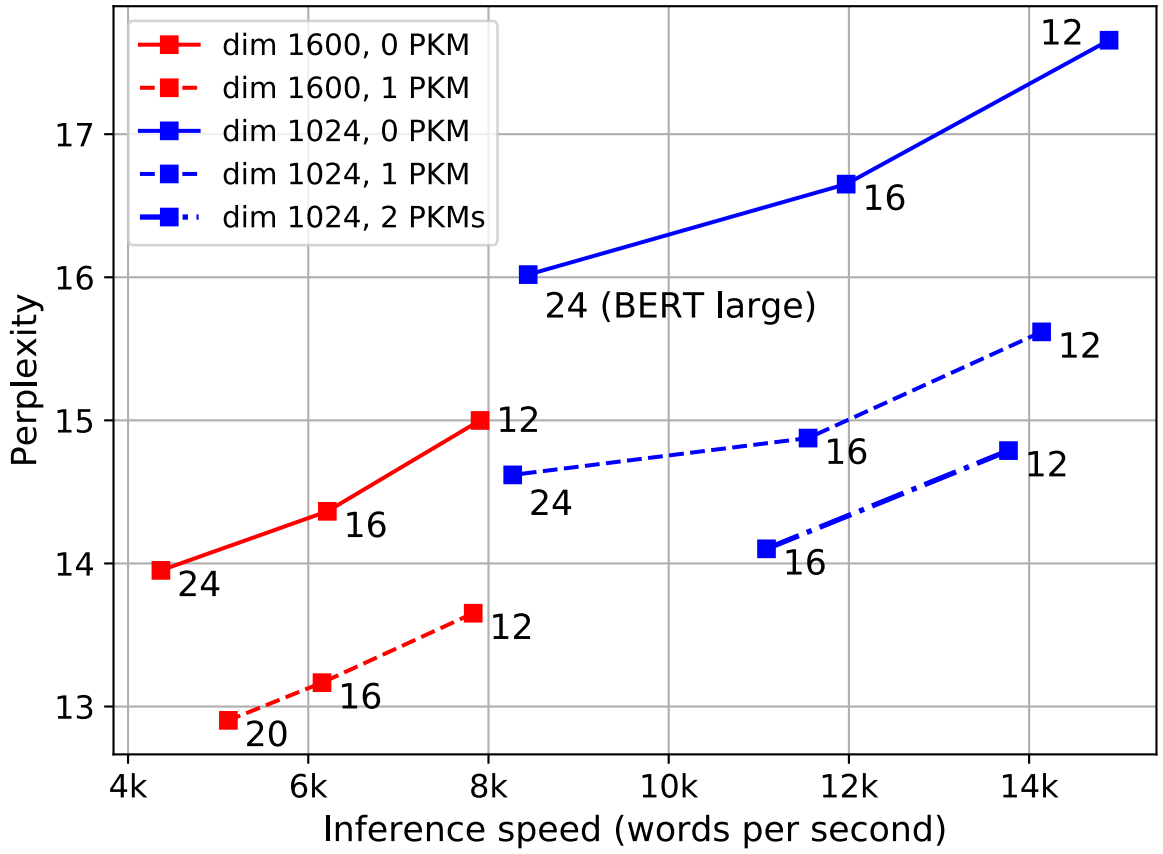
- Flat keys retrieval cost: $\mathcal{O}(|\mathcal{K}| \times d_{\mathrm{q}})$
- Product keys retrieval cost: $\mathcal{O}\left((\sqrt{|\mathcal{K}|} + k^2) \times d_{\mathrm{q}}\right)$

Product—Key Memory Layers (PKM)



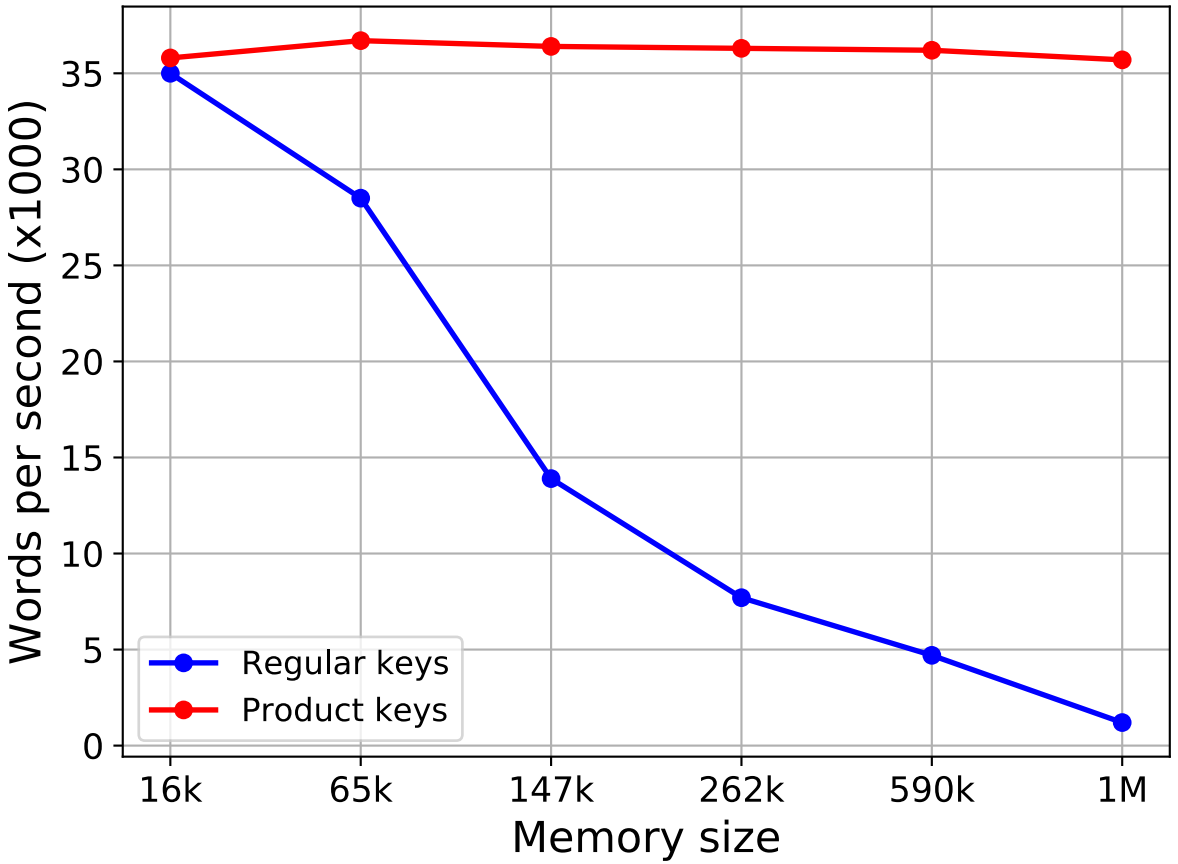
- PKM layer inside a transformer network
- Train on a very large corpus of news articles
  - 160 GB of text, 40 millions English news articles
- We evaluate the perplexity with/without memory on language modeling

Product—Key Memory Layers (PKM)



- A PKM layer gives a negligible computational overhead
- Models with 12 layers and a memory outperform models with 24 layers, for half the running time

Inference speed



- With regular keys, increasing the memory size significantly deteriorates the inference speed
- With product keys, the model speed is barely affected by the size of the memory

Thank you