Fast-weight Product Key Memory

Tianyu Zhao Llion Jones
tianyu@sakana.ai
sakana.ai

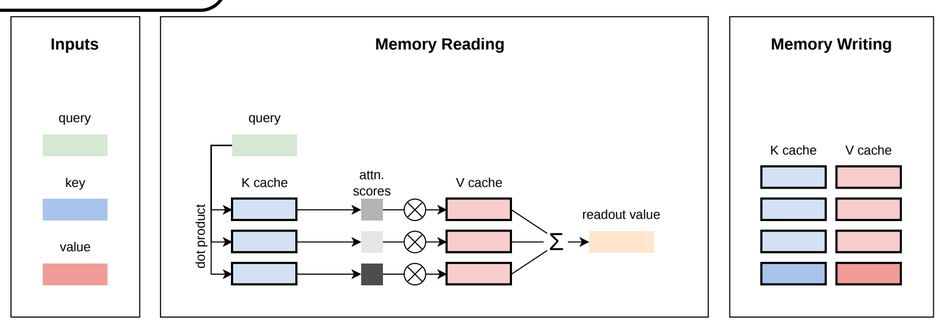
PAPER CODE

Weights or States?

	Examples	Optimization	Plasticity	Content
Weights	Params. of attention Params. of MLP Params. of embedding	E2E training + SGD Training time	Persistent Frozen after training	Info. of training data
States	Attention KV cache RNN hidden states SSM space states	Local rules Training + inference time	Plastic Always changing	Info. of input context

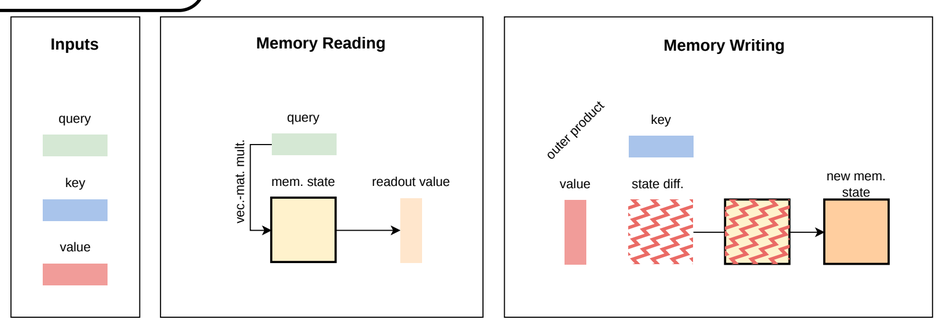
Methodology

Softmax Attention



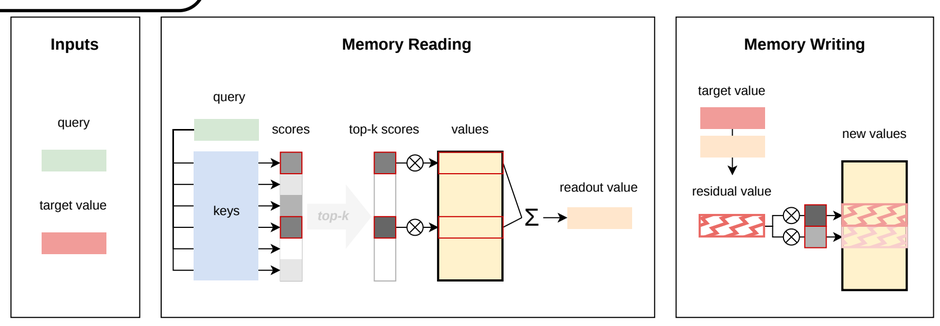
Memory arch.
- KV cache
State capacity
- unbounded
State resolution
- high-fidelity
Per-token cost
- O(Ld)

Linear Attention



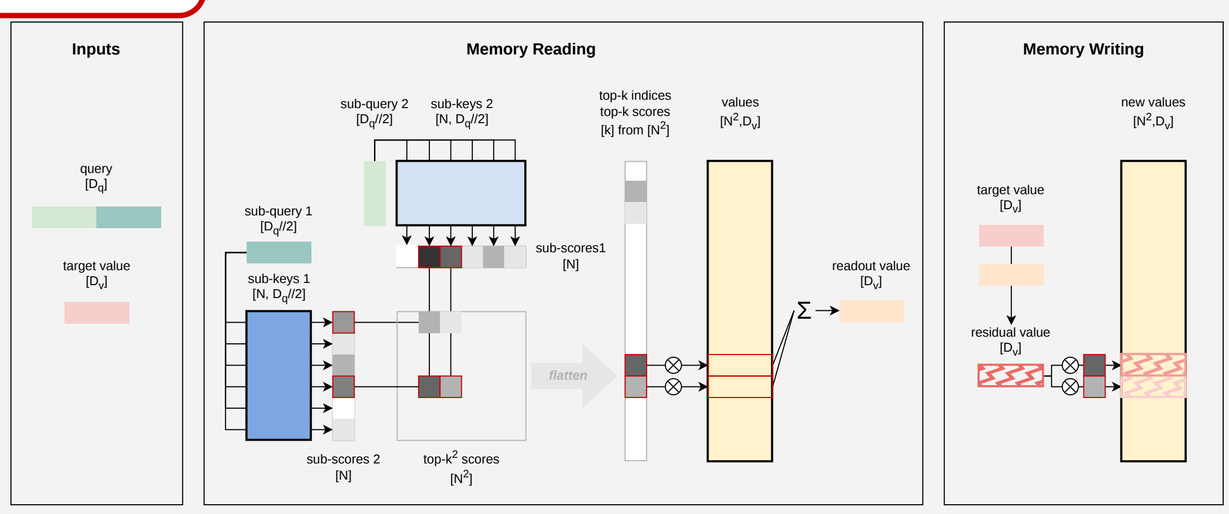
Memory arch.
- matrix
State capacity
- limited
State resolution
- compressed
Per-token cost
- $O(d^2)$

Sparse Memory



Memory arch.
- sparse memory
State capacity
- limited but larger O(N)
State resolution
- (less) compressed
Per-token cost
- O(Nd)

FwPKM



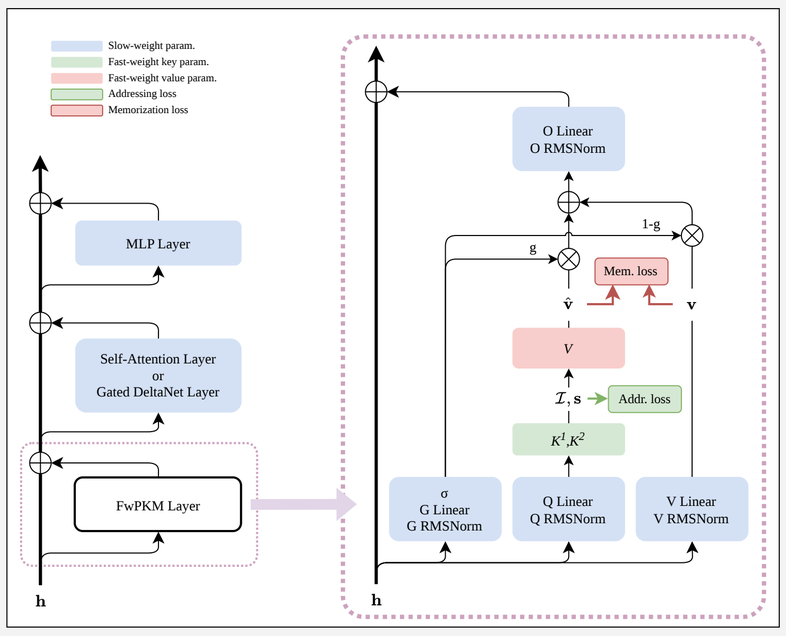
Memory arch.
- product key memory

State capacity
- large $O(N^2)$

State resolution
- high-fidelity

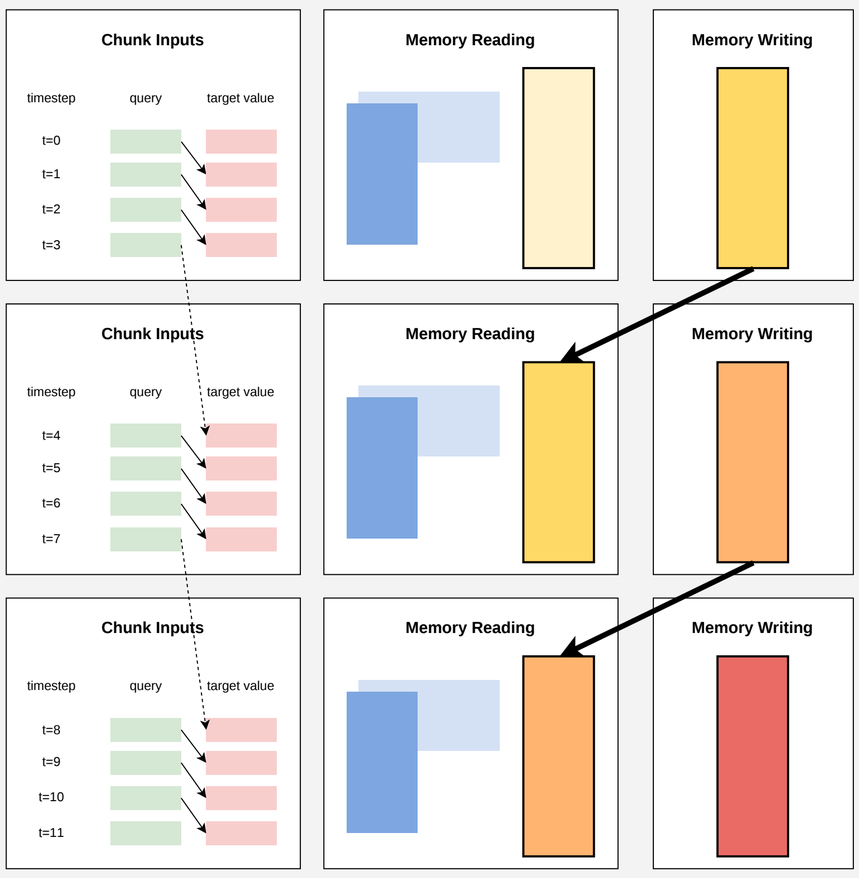
Per-token cost
- O(Nd)

FwPKM: augmented token mixer



Learning FwPKM states

Chunkwise Update + Lookahead QV association



Results

Training:

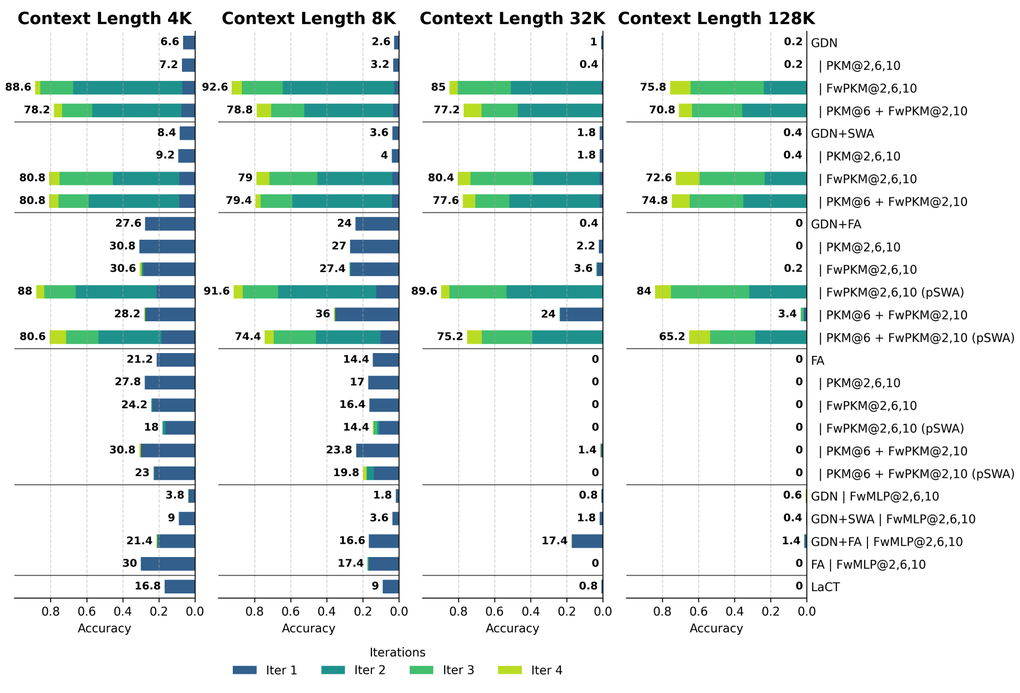
> 130M backbone LM - hybrid of attention and GDN
> PKM or FwPKM inserted at layer 2, 6, 10
> 10B training tokens; 4K seq len.

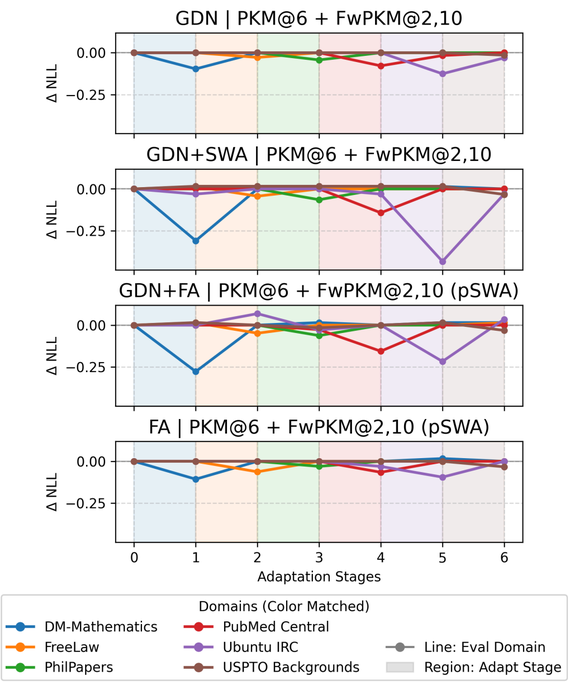
Eval: Gating visual (SakanaAI wikipedia)

> stronger memory use for rare entities

Eval: Needle-In-A-Haystack (5 needles; 6 digits)

> strong memorization: 70-90% acc
> extrapolate to long context: 128K tokens



Eval: CL/Adaptation (6 OOD the Pile domains)

> PPL reduction w/o weight update
> fail to retain memory