Online Adaptation of Language Models with a Memory of Amortized Contexts

Jihoon Tack[1], Jaehyung Kim[2], Eric Mitchell[3], Jinwoo Shin[1], Yee Whye Teh[4], Jonathan Richard Schwarz[5]

[1]KAIST, [2]Yonsei University, [3]Stanford University, [4]University of Oxford, [5]Harvard University & Thomson Reuters

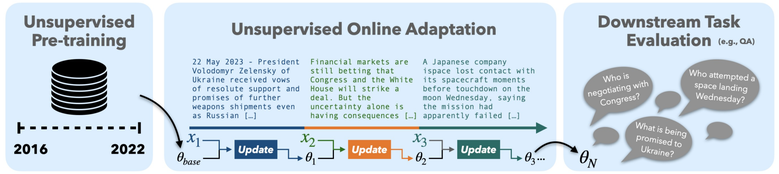
NEURAL INFORMATION PROCESSING SYSTEMS

TL;DR. We propose an efficient online adaptation method for language models by compressing document knowledge into Parameter Efficient Finetuning (PEFT) modulation and then learning to retrieve modulation from the memory bank based on the input query.

Problem of Interest: Online Adaptation

Making **language models up to date** is highly important

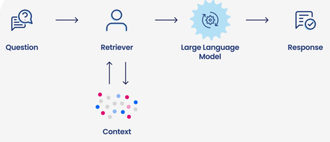
- → Need to extract new knowledge from documents and then update the old knowledge of the LLM
- → But recent LLMs are **getting larger** / Updating knowledge may induce **catastrophic forgetting** of other knowledge



For instance, after reading recent news articles, will ask, "What is the price of Bitcoin?"

Previous works?

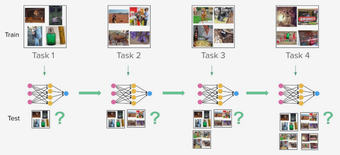
Retrieval augmentation: Save documents into a memory bank and later retrieve and prepend it



Pros
- Strong knowledge retention

Cons
- Inefficient inference
- Can retrieve a wrong document
- The given information is counterfactual to the learned knowledge

Online finetuning: Update the model's parameter with the stream of documents

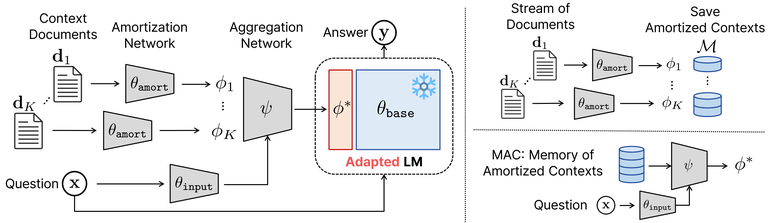


Pros
- Efficient inference

Cons
- Forgetting the learned knowledge
- Sensitive to the optimization hyper-parameter
- Need to calculate the gradients to update knowledge

MAC: Memory of Amortized Contexts

We propose **Memory of Amortized Contexts**



Training: Learning to Amortize and Aggregate

Inference: Online Adaptation

$$\min_{\theta_{\text{amort}},\theta_{\text{input}},\psi} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(\text{LM}_{\theta_{\text{base}}}(\mathbf{x}_i;\phi_i^*),\mathbf{y}_i\right) \quad \text{where} \quad \phi_i^* := h_\psi\left(g_{\theta_{\text{input}}}(\mathbf{x}_i),\{\phi_k\}_{k=1}^K\right)$$

$$\phi_k := g_{\theta_{\text{amort}}}(\mathbf{d}_k)$$

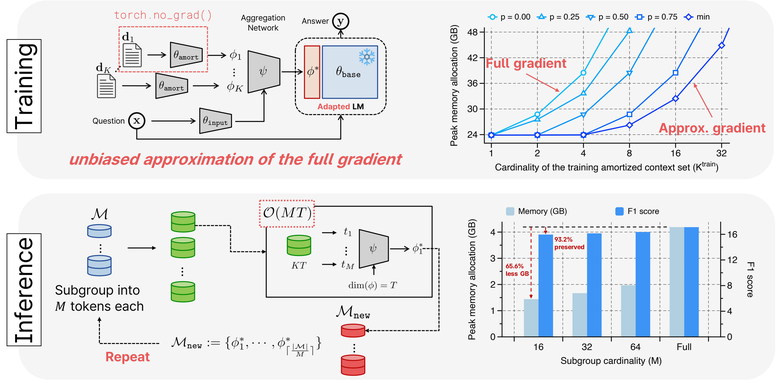
Training time

- Compress the stream of documents into **PEFT modulations** (or amortized contexts), such as Prefix tuning parameters.
- Aggregate a batch of PEFT modulation into a single PEFT based on the input (or question)

Inference: Only requires a forward pass

- Compress documents, then save them into the memory bank

Efficient Training and Inference of MAC



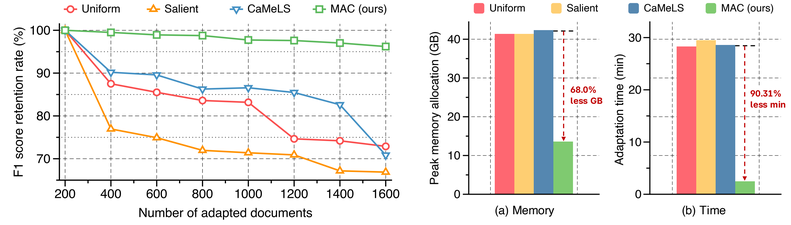
Experimental Results

Comparison with online finetuning methods

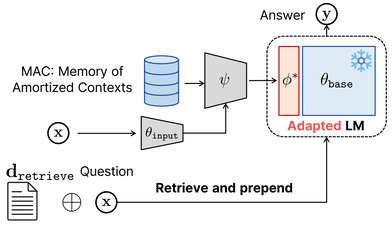
- Setup: Adapt on a stream of 1,665 documents, then perform QA

Model (# params)	Method	StreamingQA EM (↑)	StreamingQA F1 (↑)	SQuAD-Seq EM (↑)	SQuAD-Seq F1 (↑)	ArchivalQA-Seq EM (↑)	ArchivalQA-Seq F1 (↑)
DistilGPT2 (82M)	Uniform	1.62	3.76	1.24	2.54	4.86	4.08
	Salient Spans	1.44	4.67	1.03	2.47	4.52	3.76
	CaMeLS	1.62	5.79	1.47	3.08	4.62	6.19
	MAC (ours)	**5.59**	**10.18**	**2.01**	**6.85**	**7.55**	**10.58**
GPT2-Large (774M)	Uniform	4.74	7.00	3.64	4.97	7.66	8.71
	Salient Spans	4.86	8.54	4.03	6.48	9.75	11.19
	CaMeLS*	5.35	10.60	4.97	8.63	9.92	12.41
	MAC (ours)	**7.25**	**13.31**	**6.43**	**11.42**	**11.84**	**15.26**
GPT2-XL (1.5B)	Uniform	5.11	7.48	6.10	6.78	8.61	10.78
	Salient Spans	5.40	9.42	4.55	6.74	11.81	14.11
	CaMeLS*	6.55	11.67	6.70	10.15	13.87	15.74
	MAC (ours)	**8.99**	**15.38**	**7.10**	**12.55**	**14.01**	**17.12**
LLaMA-2 (7B)	Uniform	12.43	13.54	13.25	17.01	18.53	21.35
	Salient Spans	13.33	18.97	13.74	18.66	18.97	22.75
	CaMeLS			OOM			
	MAC (ours)	**14.29**	**21.79**	**15.07**	**21.14**	**20.12**	**23.90**

- knowledge retention (left) / Adaptation efficiency (right) is better than the online finetuning methods



MAC can be **jointly** used with **retrieval augmentation methods**



Online adaptation performance of LLaMA-2 7B with retrieved documents on ArchivalQA

	Top-1 EM	Top-1 F1	Top-3 EM	Top-3 F1	Top-5 EM	Top-5 F1
BM25	48.53	54.17	56.18	63.74	64.74	71.83
BM25 + MAC (ours)	**52.81**	**56.55**	**60.22**	**66.82**	**68.85**	**74.89**
Contriever	44.78	51.55	52.56	61.28	60.10	67.83
Contriever + MAC (ours)	**47.99**	**53.23**	**53.92**	**63.75**	**61.28**	**70.01**
DPR	48.98	55.01	57.02	64.27	65.07	72.24
DPR + MAC (ours)	**49.57**	**55.98**	**60.19**	**67.05**	**68.52**	**75.00**

More results can be found on our project page: https://jihoontack.github.io/MAC/