# Adapting Language Models to Compress Contexts

Alexis Chevalier* Alexander Wettig* Anirudh Ajith Danqi Chen

Princeton Language and Intelligence, Princeton University

* equal contribution

Paper Code

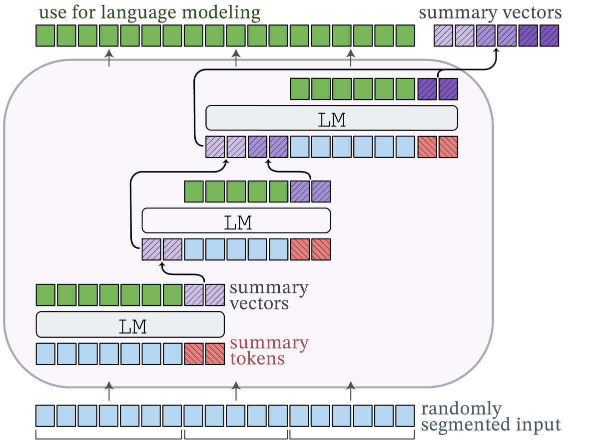
## Summary

Goals:

- Speed up attention over long contexts for both training and inference
- Reduce memory cost of storing past key-value cache

Introducing ***AutoCompressors***:

- Compress text into compact ***summary vectors*** which the model can read as soft prompts
- Summary vectors can be ***accumulated*** to compress long documents
- Simple training objective: Fine-tune from pre-trained model with next-token prediction
- Summary vectors of a document can be ***cached*** and re-used as a concise context to relevant prompts

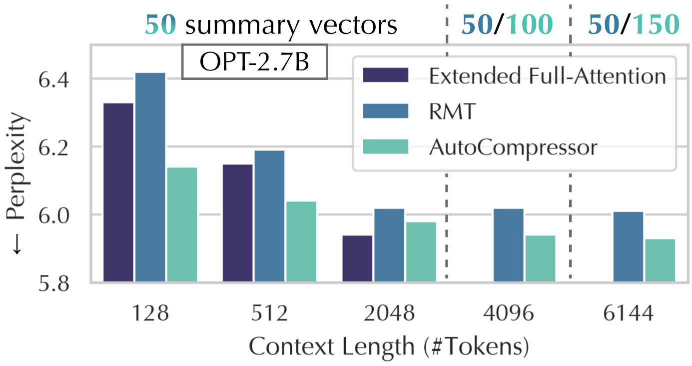



Fine-tuning an AutoCompressor from a pre-trained LM

# Evaluation

## Language Modeling

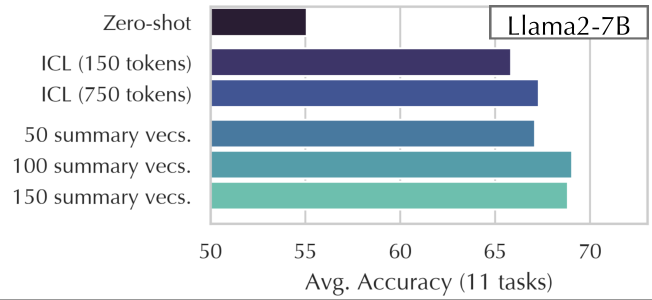
- We present AutoCompressors based on **OPT-1.3B**, **OPT-2.7B** and **Llama-2-7B,** fine-tuned on sequences of up to **30k tokens** (OPT) and 6k tokens (Llama)
  All models available at huggingface.co/princeton-nlp!
- Recursively **compress** up to **2048 tokens** into **50 tokens**
- Measure gains in perplexity from adding raw context tokens vs. summary vectors
- Baselines:
  - **RMT** [Bulatov et al., 2022] (no summary accumulation, fixed segment length)
  - **Extended Full-Attention** (initialize new position embeddings / RoPE interpolation)




## In-Context Learning

Can summary vectors encode task semantics?

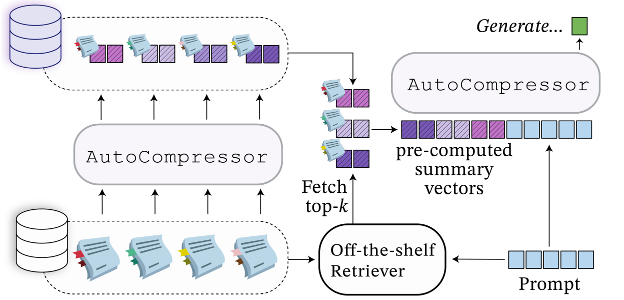
- Encoding demonstrations as summary vectors can be seen as *zero-shot soft prompt tuning*
- On 5/7 SuperGLUE tasks, conditioning on 150 summary vectors outperforms 750 tokens worth of plain-text demonstrations



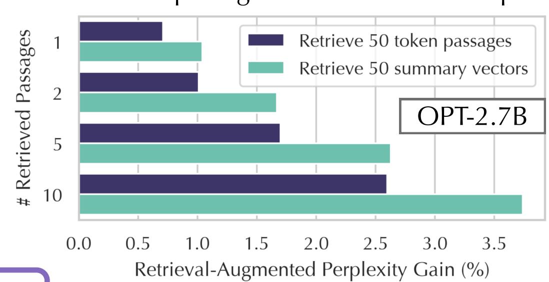

# Applications

## Retrieval-Augmented LM

1. Compress all the documents in corpus to summary vectors
2. Retrieve passages and use their summary vectors for efficient inference




- Using summary vectors outperforms retrieving same-size token passages at same inference speed




## Passage Re-ranking

- Re-rank passages based on *p(query | passage)*
- Preprocess: 512 token passage → 50 summary vectors
- Inference with large models based on summary vectors is superior to small models based on full passages