# InfLLM: Training-Free Long-Context Extrapolation for LLMs with an Efficient Context Memory

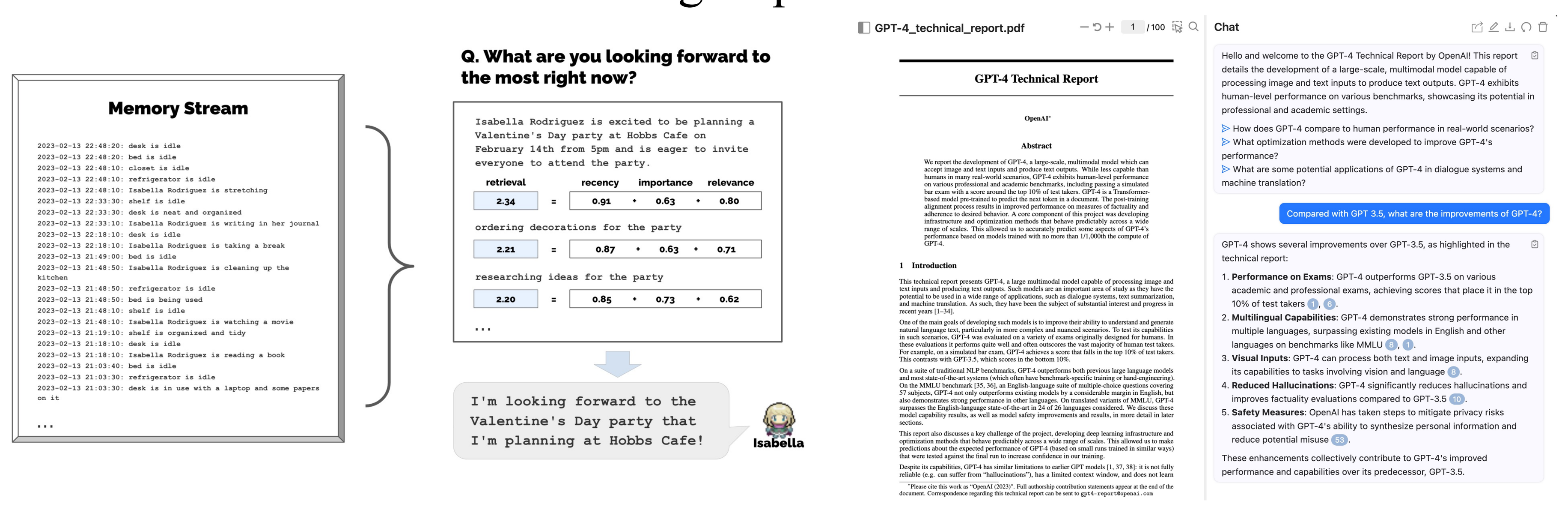

Chaojun Xiao*, Pengle Zhang*, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun

Tsinghua University    Massachusetts Institute of Technology    Renmin University of China


## Background

- With the blooming of LLM-driven applications, such as agent construction and embodied robotics, enhancing the capability of LLMs to process streaming long sequences become increasingly crucial.

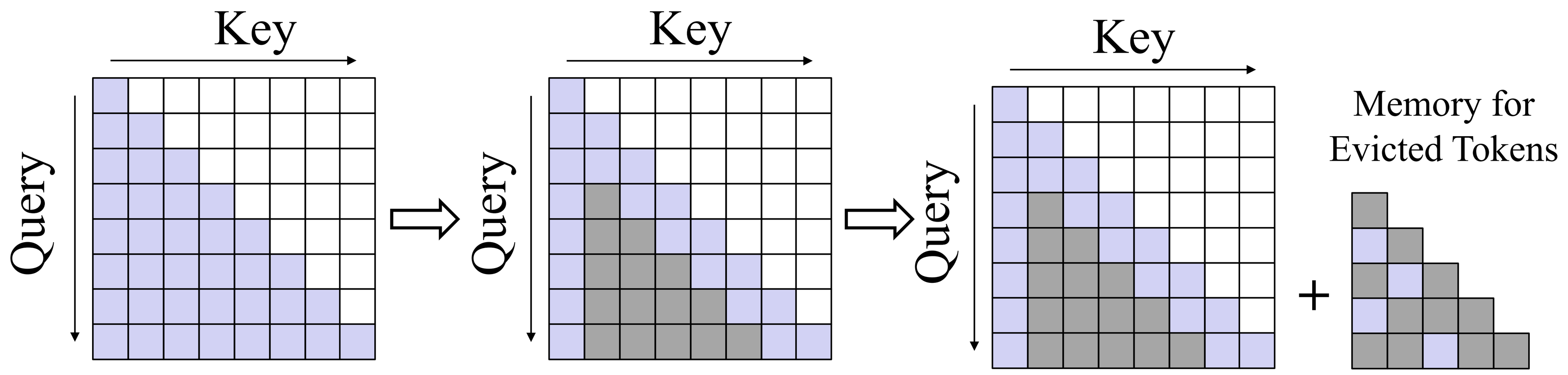
Many real-world applications require LLMs to process extremely long sequences




LLM-driven agents make decisions based on long historical memories

Reading academic papers spanning hundreds of pages with LLMs

- Sliding window attention can enable LLMs to process streaming long sequences. However, as it discards all distant context, sliding window attention will suffer from catastrophic forgetting issue.

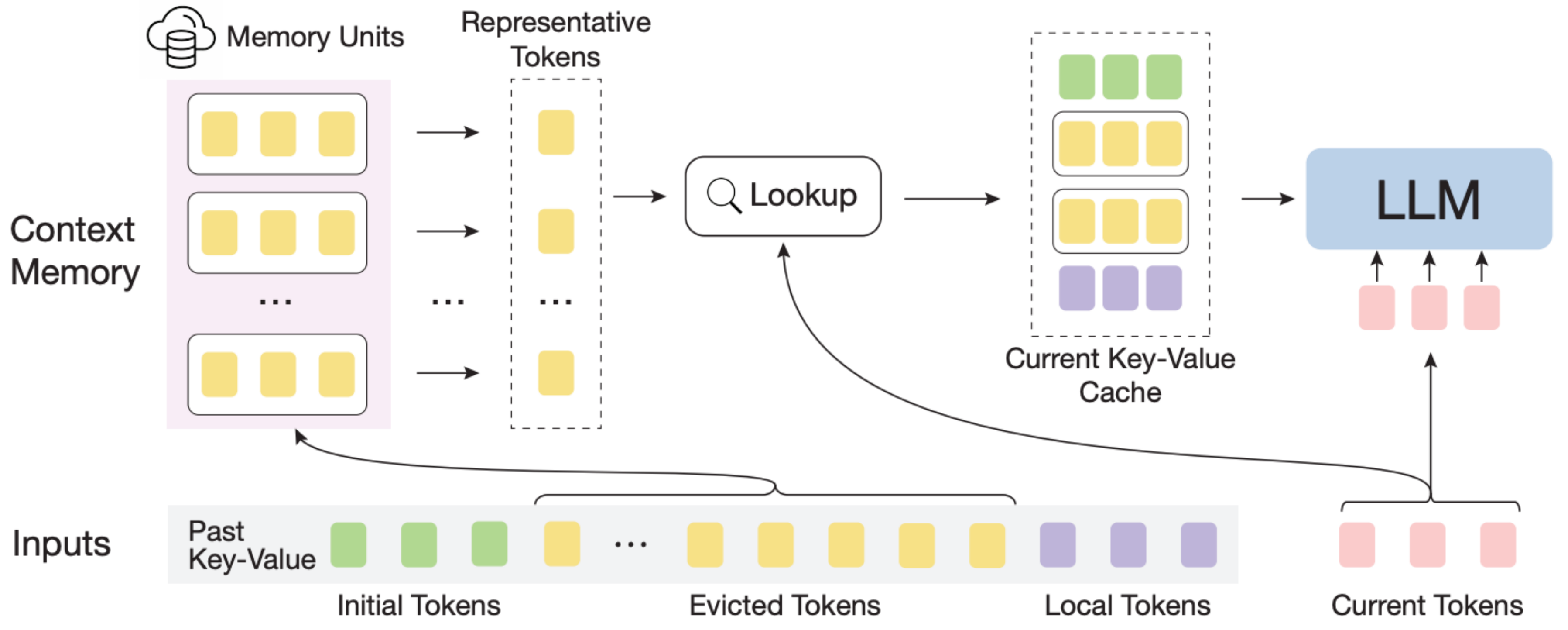



Full Attention    Sliding Window    InfLLM

- **Our Goal**: Building a context memory to save evicted tokens, training-free extending the context window without forgetting distant contexts.

## Methodology

- **InfLLM = Sliding Window + Block-Level Context Memory**
  - InfLLM organizes past key-value vectors into blocks, named as memory unit, each containing a continuous token sequence
  - **Representative Tokens**: Within each block, the semantically most significant tokens that receive the highest attention scores are selected as the unit representation for subsequent relevance computation in memory lookup
  - **Offloading**: InfLLM offloads all units on CPU memory and dynamically retains the frequently used units on GPU memory




## Main Results

- **InfLLM can achieve superior performance with limited computation.**

| | Window | Streaming | R.PK | R.Num | R.KV | Choice | QA | Sum | Math.F | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| Mistral-based Models (7B) | | | | | | | | | | |
| Mistral | 32K | ✗ | 28.8 | 28.8 | 14.8 | 44.5 | 12.9 | 25.9 | 20.6 | 25.2 |
| NTK | 128K | ✗ | 100.0 | 86.8 | 19.2 | 40.2 | 16.9 | 20.3 | 26.9 | 44.3 |
| SelfExtend | 128K | ✗ | 100.0 | 100.0 | 15.6 | 42.8 | 17.3 | 18.8 | 19.1 | 44.8 |
| Infinite | 32K | ✓ | 28.8 | 28.8 | 0.4 | 42.8 | 11.4 | 22.5 | 16.3 | 21.6 |
| Streaming | 32K | ✓ | 28.8 | 28.5 | 0.2 | 42.4 | 11.5 | 22.1 | 16.9 | 21.5 |
| H2O | 32K | ✓ | 8.6 | 4.8 | 2.6 | 48.0 | 15.6 | 24.4 | 26.9 | 18.7 |
| InfLLM | 16K | ✓ | 100.0 | 96.1 | 96.8 | 43.7 | 15.7 | 25.8 | 25.7 | 57.7 |
| Llama-3-based Models (8B) | | | | | | | | | | |
| Llama-3 | 8K | ✗ | 8.5 | 7.8 | 6.2 | 44.1 | 15.5 | 24.7 | 21.7 | 18.4 |
| NTK | 128K | ✗ | 0.0 | 0.0 | 0.0 | 0.0 | 0.4 | 6.4 | 2.6 | 1.3 |
| SelfExtend | 128K | ✗ | 100.0 | 100.0 | 0.2 | 19.7 | 8.6 | 14.7 | 22.6 | 38.0 |
| Infinite | 8K | ✓ | 6.8 | 7.6 | 0.2 | 41.5 | 14.6 | 20.8 | 20.6 | 16.0 |
| Streaming | 8K | ✓ | 8.5 | 8.3 | 0.4 | 40.6 | 14.3 | 20.4 | 21.4 | 16.3 |
| H2O | 8K | ✓ | 2.5 | 2.4 | 0.0 | 0.0 | 0.7 | 2.8 | 6.0 | 2.1 |
| InfLLM | 8K | ✓ | 100.0 | 99.0 | 5.0 | 43.7 | 19.5 | 24.3 | 23.7 | 45.0 |

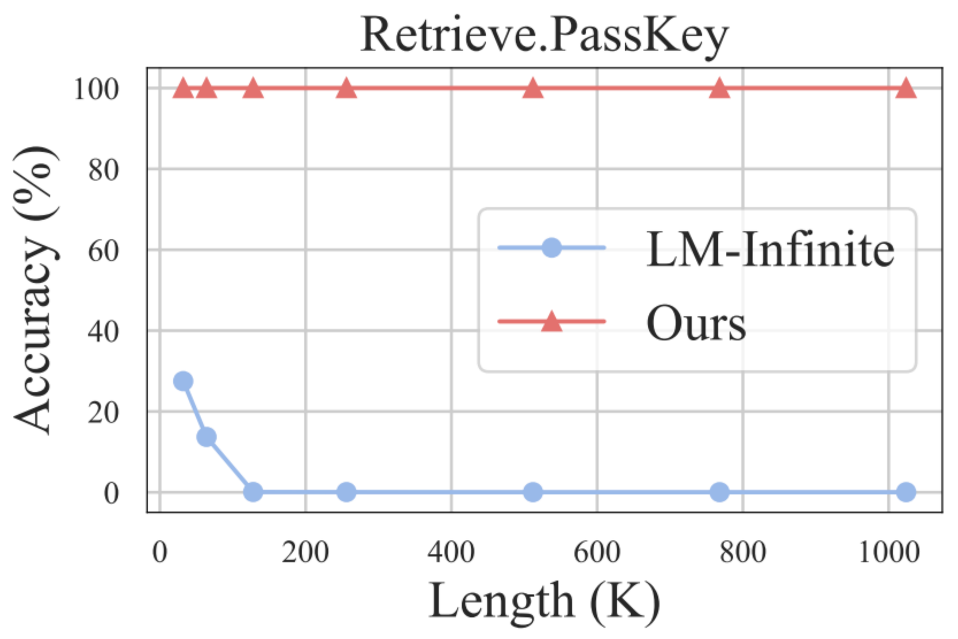
## Comparing to Models with Continual Training

- Compared to Llama-3-8B-Instruct-Gradient-1048k (Llama-1M), InfLLM can achieve comparable without any additional training.
- InfLLM achieves **a 34% decrease in time consumption while using only 34% of the GPU memory** compared to the Llama-1M.
- InfLLM can be directly combined with Llama-1M to further improve the performance.

| | Train-Free | R.PK | R.Num | R.KV | Choice | QA | Sum | Math.F | VRAM | Time |
|---|---|---|---|---|---|---|---|---|---|---|
| Llama-1M | ✗ | **100.0** | **99.8** | **23.2** | **51.5** | 13.6 | 18.5 | 18.3 | 76.6G | 40.4s |
| InfLLM | ✓ | **100.0** | 99.0 | 5.0 | 43.7 | **19.5** | **24.3** | **23.7** | **26.3G** | **26.7s** |
| Llama-1M+InfLLM | ✗ | 100.0 | 100.0 | 55.8 | 39.3 | 20.3 | 17.1 | 31.4 | 26.3G | 26.7s |

## Scaling to 1024K Context

- InfLLM can extend the context window size of Mistral and **achieve 100% accuracy on passkey retrieval task.**




## Comparing to RAG

InfLLM has following advantages:

- **Training-Free**: RAG requires additional retrieval data to train a retrieval model.
- **Broader Applicability**: RAG models are usually limited by the performance of their retrieval components. Besides, existing retrieval models will suffer from out-of-distribution issues.

| Task | R.PK | R.Num | R.KV |
|---|---|---|---|
| RAG-E5 | 89.2 | 65.4 | 13.2 |
| InfLLM | **100.0** | **96.1** | **96.8** |

Paper    Github


Email: xcjthu@gmail.com