Not All Heads Matter: A Head-Level KV Cache Compression Method with Integrated Retrieval and Reasoning

Yu Fu[1], Zefan Cai[2], Abedelkadir Asi[3], Wayne Xiong[3], Yue Dong[1] , Wen Xiao[3]

[1]University of California, Riverside [2]University of Wisconsin-Madison [3]Microsoft

Background – KV cache

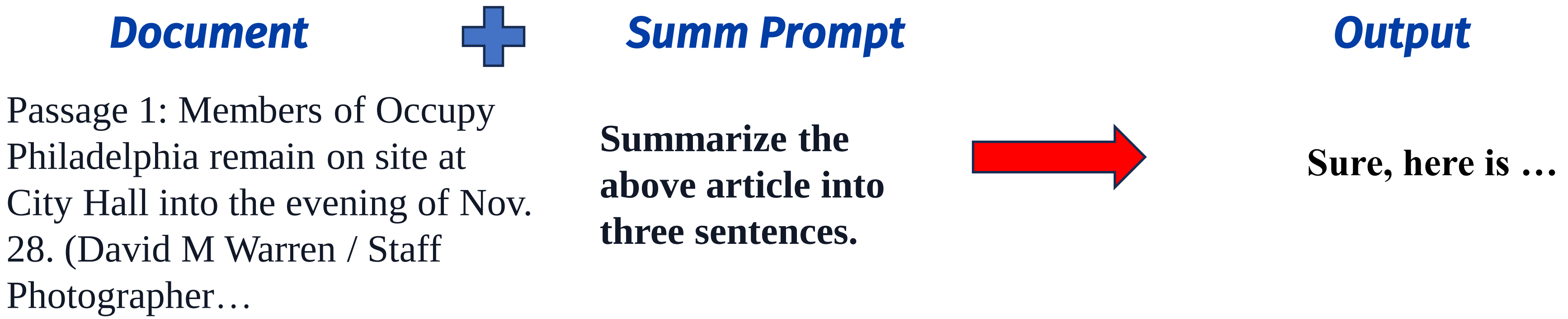
For LLM Generation:

Prefilling phase

$$X \in \mathbb{R}^{l\times d}; \quad K_{cache} = K \in \mathbb{R}^{l\times d}; \quad V_{cache} = V \in \mathbb{R}^{l\times d}$$

Decoding phase

$$x_{l+1} \in \mathbb{R}^{l\times d}; \quad K = [K_{cache}, k_{l+1}] \in \mathbb{R}^{(l+1)\times d}; \quad V = [V_{cache}, v_{l+1}] \in \mathbb{R}^{(l+1)\times d}$$



Document

Passage 1: Members of Occupy Philadelphia remain on site at City Hall into the evening of Nov. 28. (David M Warren / Staff Photographer…

Summ Prompt

Summarize the above article into three sentences.

Output

Sure, here is …

Prefilling Phrase

Decoding Phrase

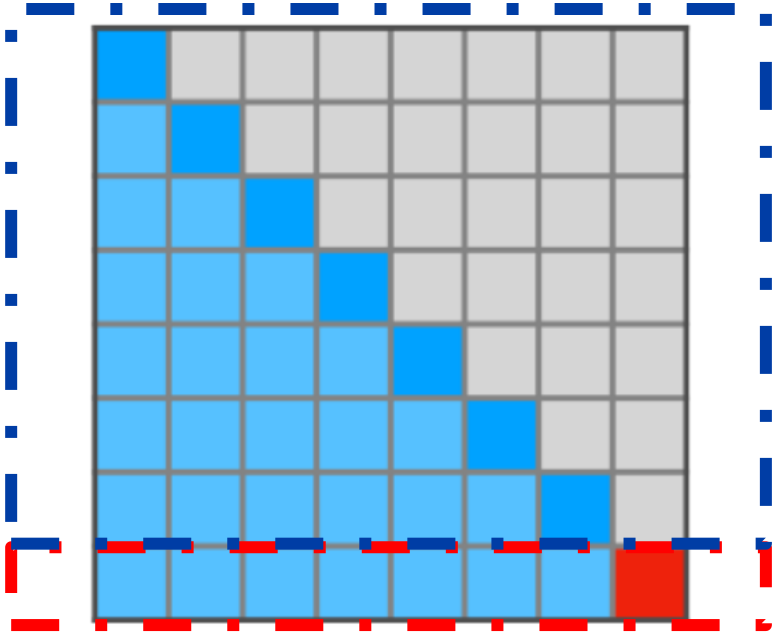
Background – KV cache

Prefilling phase

$l \times l$

Decoding phase

$1 \times (l+1)$



Calculation Matrix

Super Long Document?

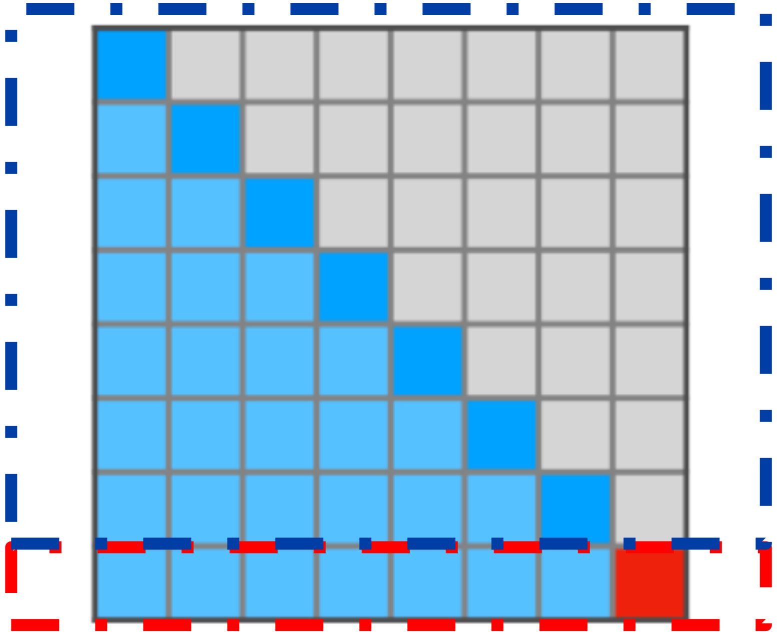
1. Memory. (Linear to the Input)

For each head: KV cache
*K + V: 2 * (batch_size, **seq_len**, head_dim)*

2. Efficiency.

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Method



Prefilling phase

$l \times l$

Decoding phase

$1 \times (l + 1)$

Calculation Matrix

Focus on the Prefilling phase

Motivation:
KV cache Compression during Prefilling.

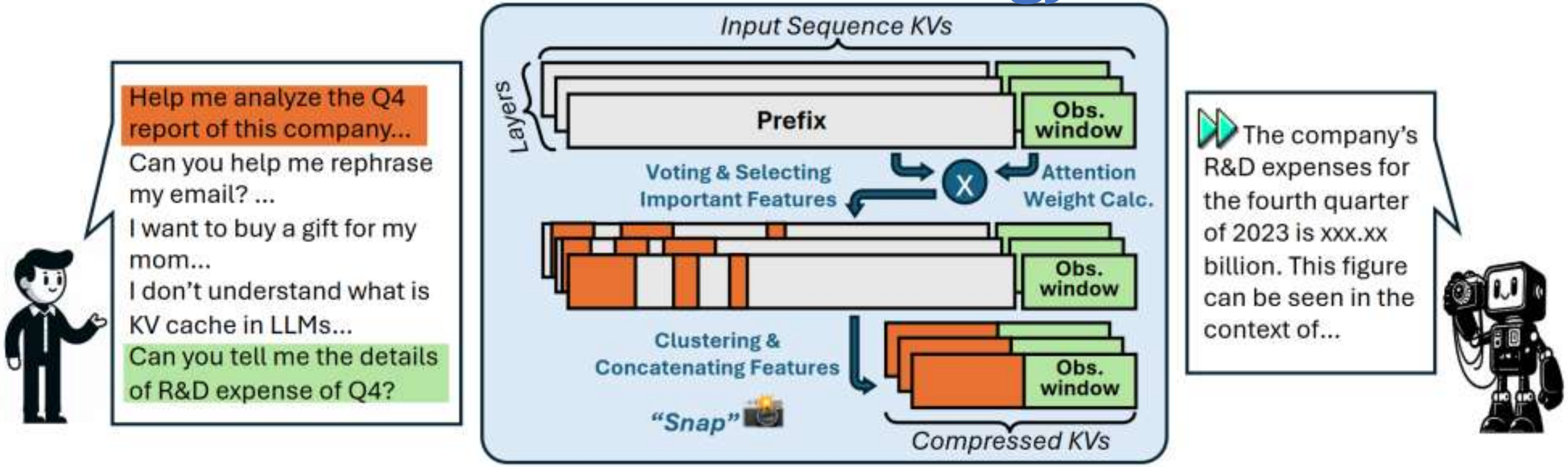
From input_len → 64/128 (Hyper-parameter)

Maybe 100000

Other Methods

Q: How to choose KV cache?

Selection Strategy


Help me analyze the Q4 report of this company...
Can you help me rephrase my email? ...
I want to buy a gift for my mom...
I don't understand what is KV cache in LLMs...
Can you tell me the details of R&D expense of Q4?
Input Sequence KVs
Layers
Prefix
Obs. window
Voting & Selecting Important Features
X
Attention Weight Calc.
Obs. window
Clustering & Concatenating Features
Obs. window
"Snap"
Compressed KVs
The company's R&D expenses for the fourth quarter of 2023 is xxx.xx billion. This figure can be seen in the context of...

Other KV caches will be dropped.

K + V: 2 * (batch_size, **seq_len**, head_dim) → K + V: 2 * (batch_size, **128**, head_dim)

SnapKV: LLM Knows What You are Looking for Before Generation

Other Methods

Problems

1. Layer-level allocation.
   → Treated equally for heads in the same layer

2. Can we do head-level allocation?

↓

How to obtain important heads?

Heads that attend to important/relevant information are more important

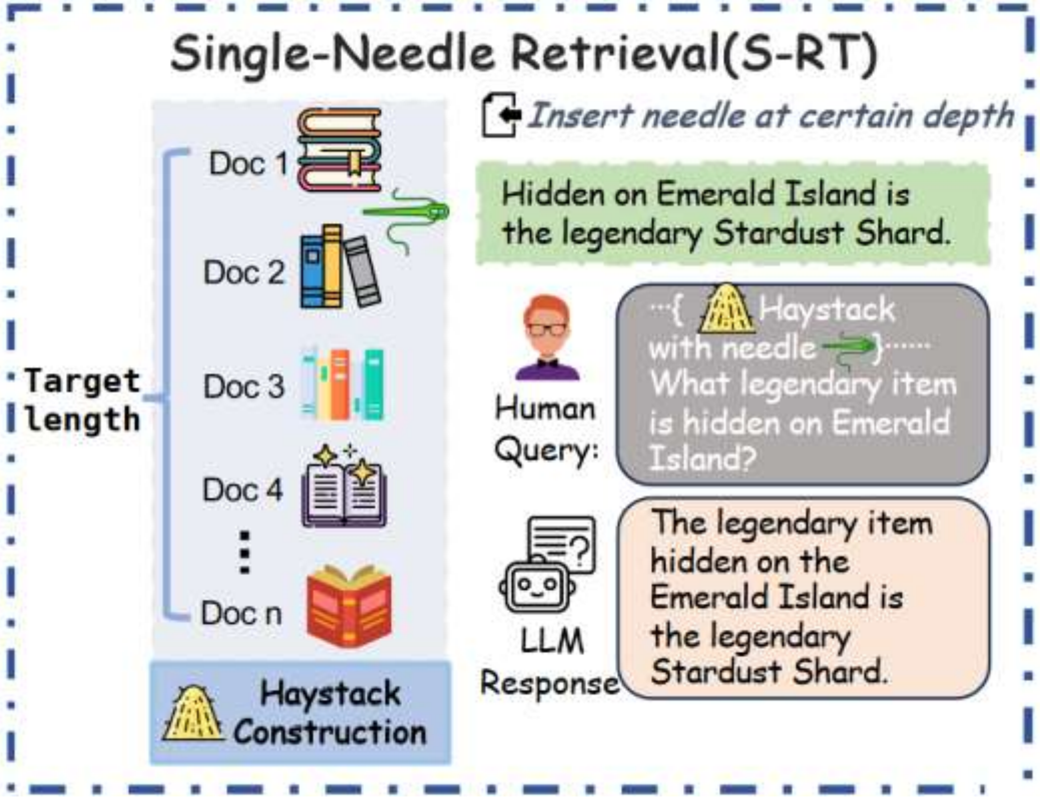
Base Method

Identify Head

Needle-in-the-haystack

1. Haystack (Long-Context)
2. Needle
3. Question

Special Question


Single-Needle Retrieval(S-RT)
Insert needle at certain depth
Hidden on Emerald Island is the legendary Stardust Shard.
Target length
Doc 1
Doc 2
Doc 3
Doc 4
Doc n
Haystack Construction
Human Query:
…{ Haystack with needle }……
What legendary item is hidden on Emerald Island?
LLM Response
The legendary item hidden on the Emerald Island is the legendary Stardust Shard.

Base Method

Special Question

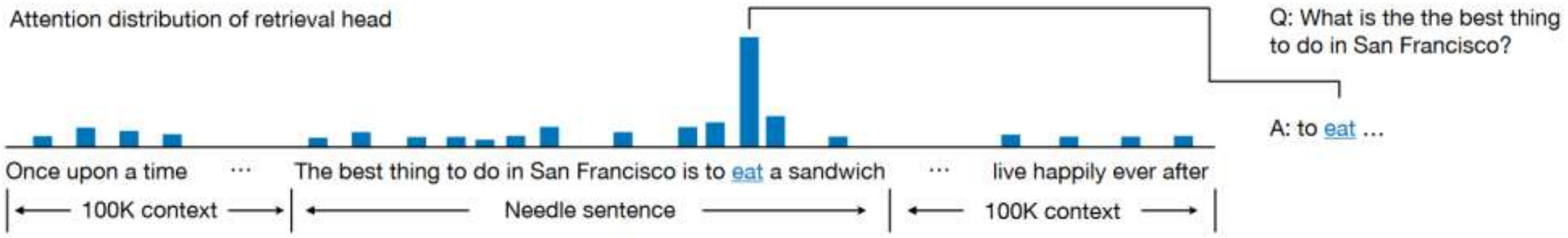
Can only answered based on the inserted Needle

```
"question": "What does a new report from WMO shows ?",
"needle": "A new report from the WMO shows that records were once again
broken, and in some cases smashed, for greenhouse gas levels, surface
temperatures, ocean heat and acidification."
```

Base Method

1. Max attention
2. In the Needle
3. Exact match

Obtain Head Score


Attention distribution of retrieval head
Q: What is the the best thing to do in San Francisco?
A: to eat ...
Once upon a time
...
The best thing to do in San Francisco is to eat a sandwich
...
live happily ever after
100K context
Needle sentence
100K context

|-----------------------------Input------------------------------------| -------Output--------

Base Method

Key Point: Just Copy

Reason Problem? → QA / Summ

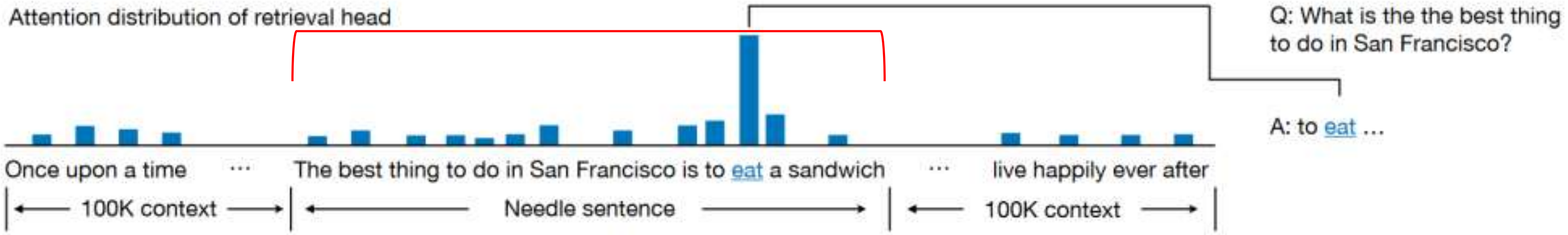
Improve:

1. Improve score function
2. Construct new reason example

Our Method

1. Better Score Function

1. Top-n attention
2. In the Needle


Attention distribution of retrieval head
Once upon a time
···
The best thing to do in San Francisco is to eat a sandwich
···
live happily ever after
100K context
Needle sentence
100K context
Q: What is the the best thing to do in San Francisco?
A: to eat ...

|----------------------------Input------------------------------------| -------Output--------

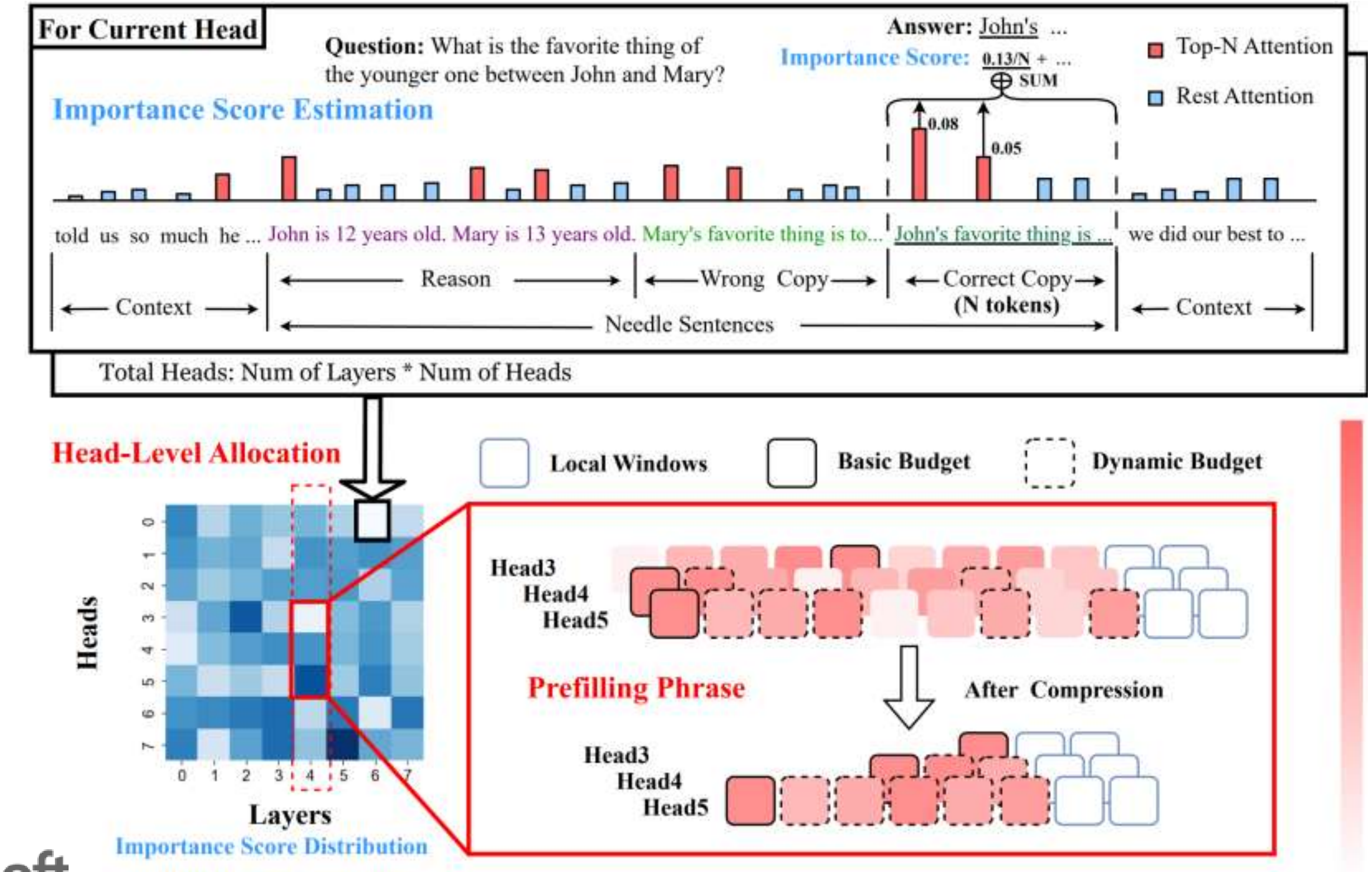
Consider All tokens in the inserted Needle rather than just the exact match token.

Our Method

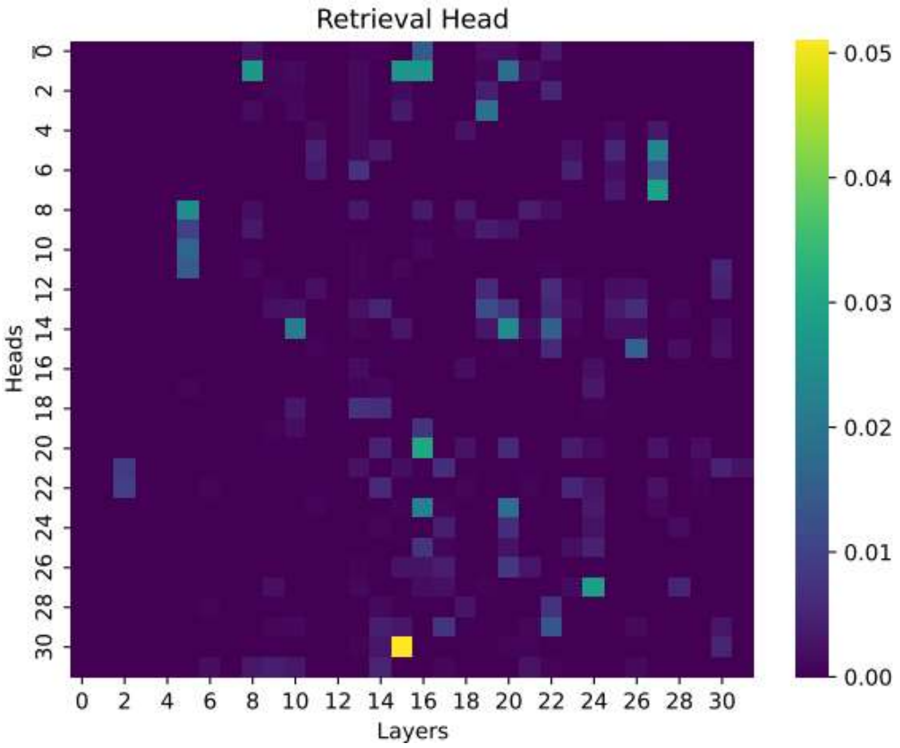
2. Construct reason examples

```
"question": "What is the favorite thing of the younger one between John
and Mary?",
"needle": "John is 12 years old. Mary is 13 years old. John's favorite
thing is to play basketball at the local gym and enjoy a smoothie
afterward. Mary's favorite thing is to take a walk in Chaoyang Park and
have a cup of Espresso in the evening. "
```

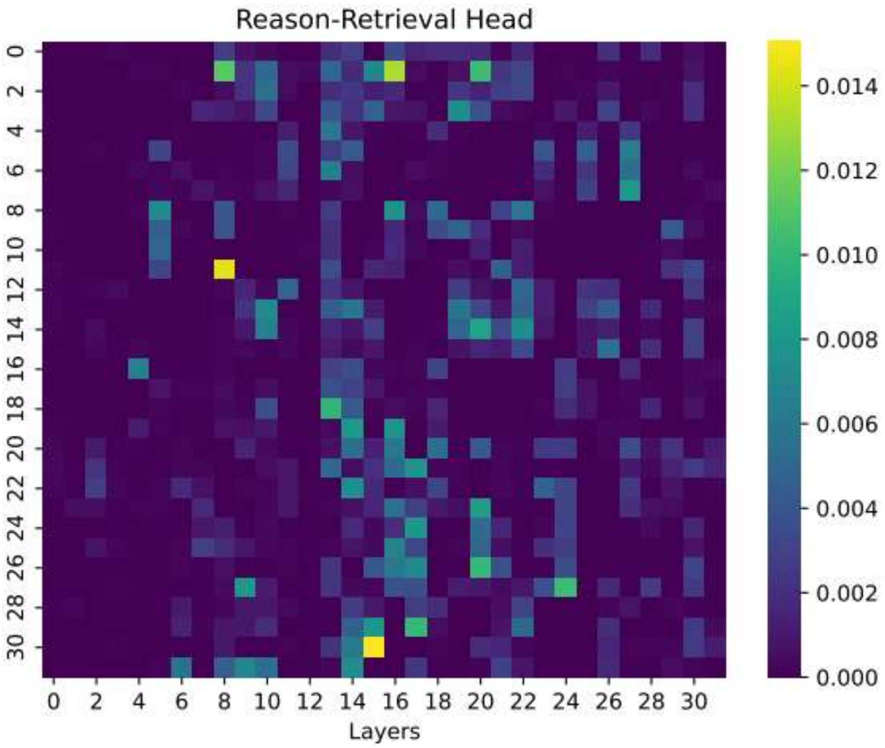
Our Method


For Current Head
Question: What is the favorite thing of the younger one between John and Mary?
Answer: John's ...
Importance Score: 0.13/N + ...
SUM
Top-N Attention
Rest Attention
Importance Score Estimation
0.08
0.05
told us so much he ... John is 12 years old. Mary is 13 years old. Mary's favorite thing is to... John's favorite thing is ... we did our best to ...
Context
Reason
Wrong Copy
Correct Copy (N tokens)
Context
Needle Sentences
Total Heads: Num of Layers * Num of Heads
Head-Level Allocation
Local Windows
Basic Budget
Dynamic Budget
Heads
0
1
2
3
4
5
6
7
Layers
Importance Score Distribution
Head3
Head4
Head5
Prefilling Phrase
After Compression
Head3
Head4
Head5

Head-level Importance Score Distribution


Retrieval Head

Retrieval

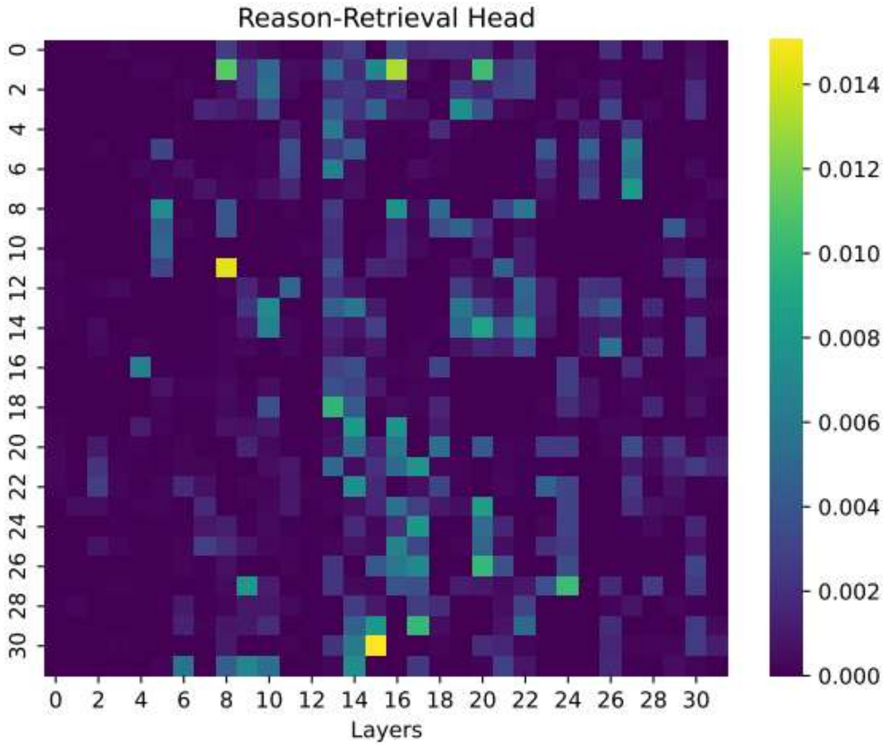

Reason-Retrieval Head

Ours

Ours is much dense

Head-level Allocation

How to use this **head-level** distribution?

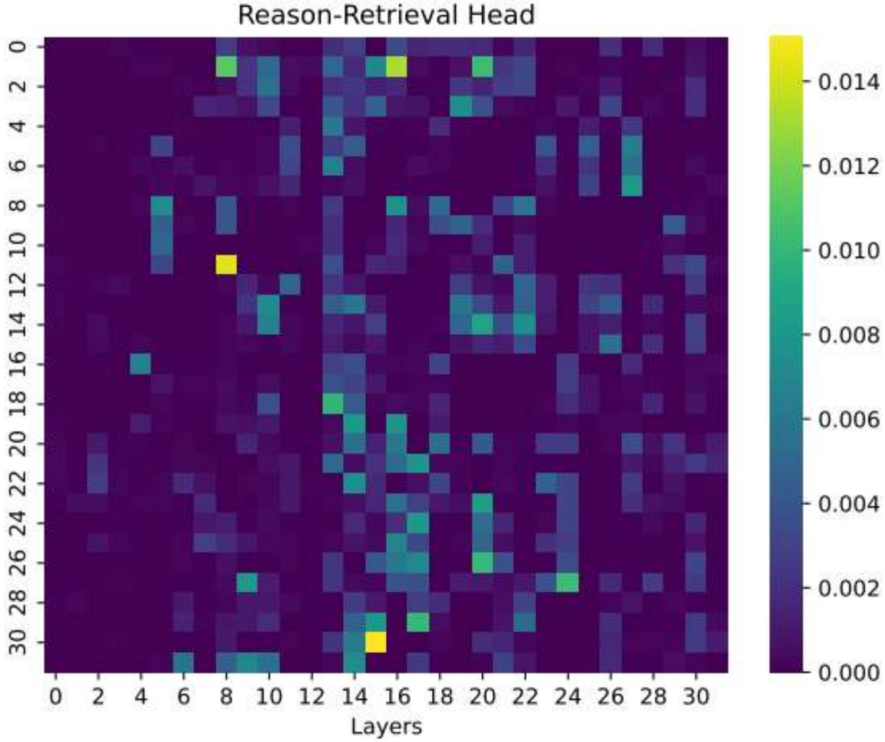


Step1:

each head → N (128) numbers of KV cache

Head-level Allocation

How to use this **head-level** distribution?



Step1:

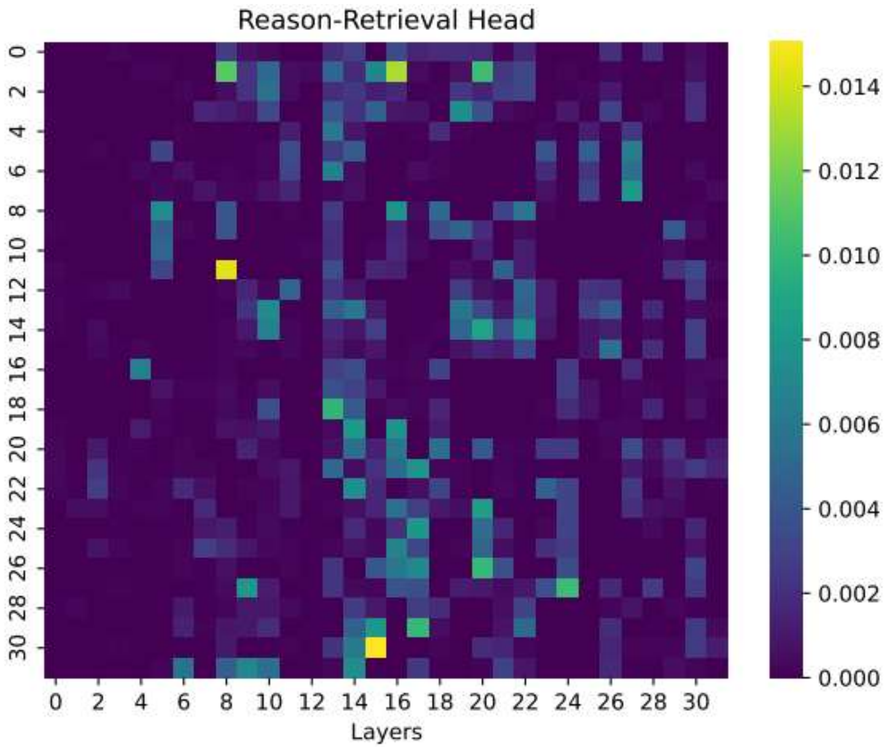
each head → N (128) numbers of KV cache

Step2:

Construct Global Pool → extract M (120) from each head

Head-level Allocation

How to use this **head-level** distribution?



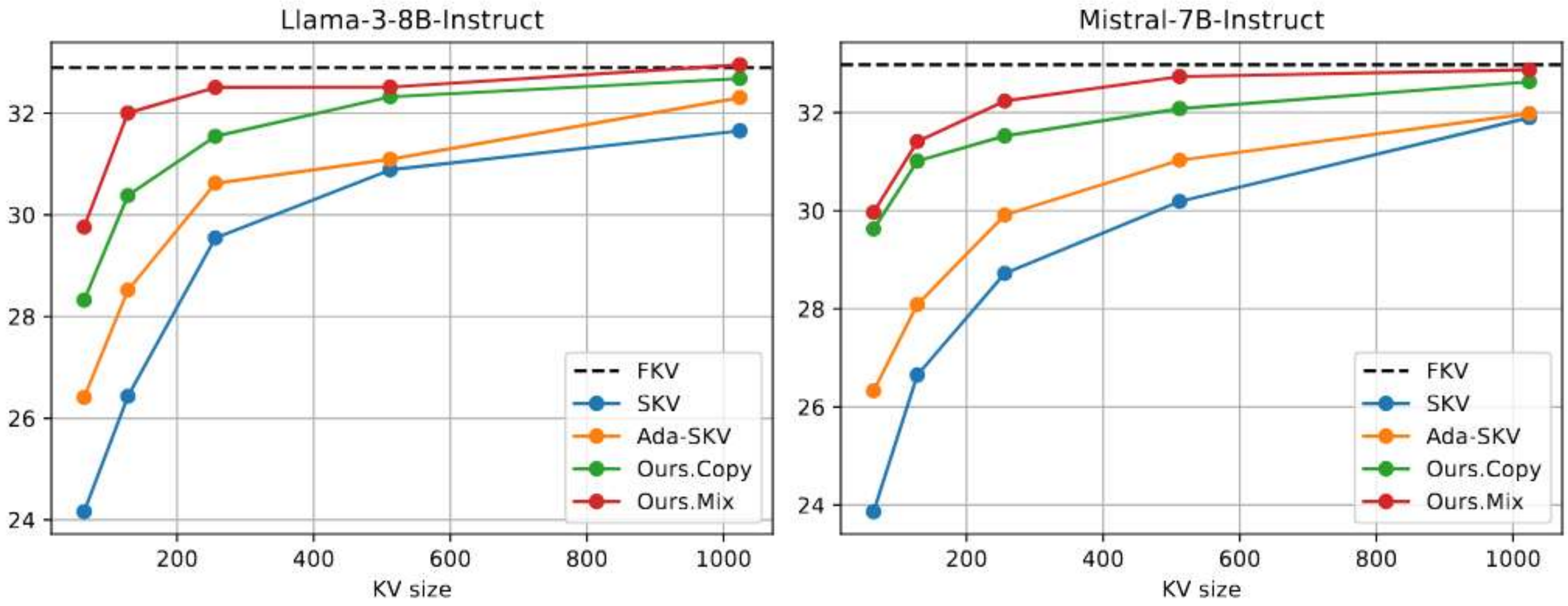
Step1:
each head → N (128) numbers of KV cache

Step2:
Construct Global Pool → extract M (120) from each head

Step3:
Assign dynamic KV cache number based on the score.
S * M * N

Results

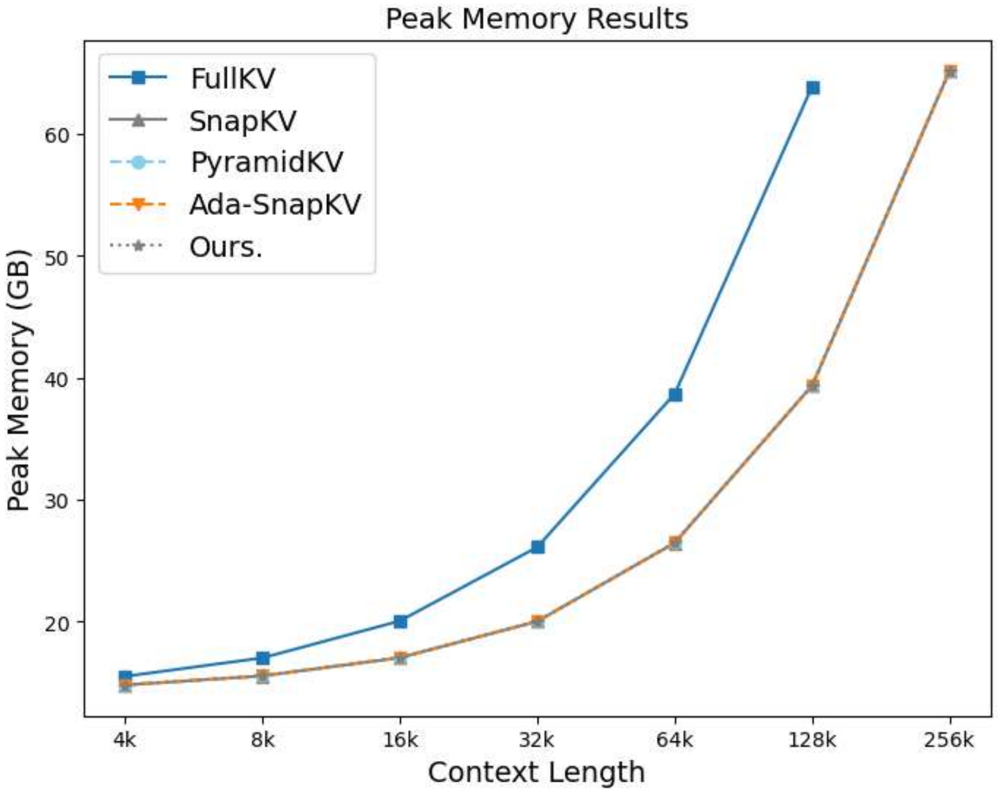
Avg. Accuracy Score Over 6 QA tasks

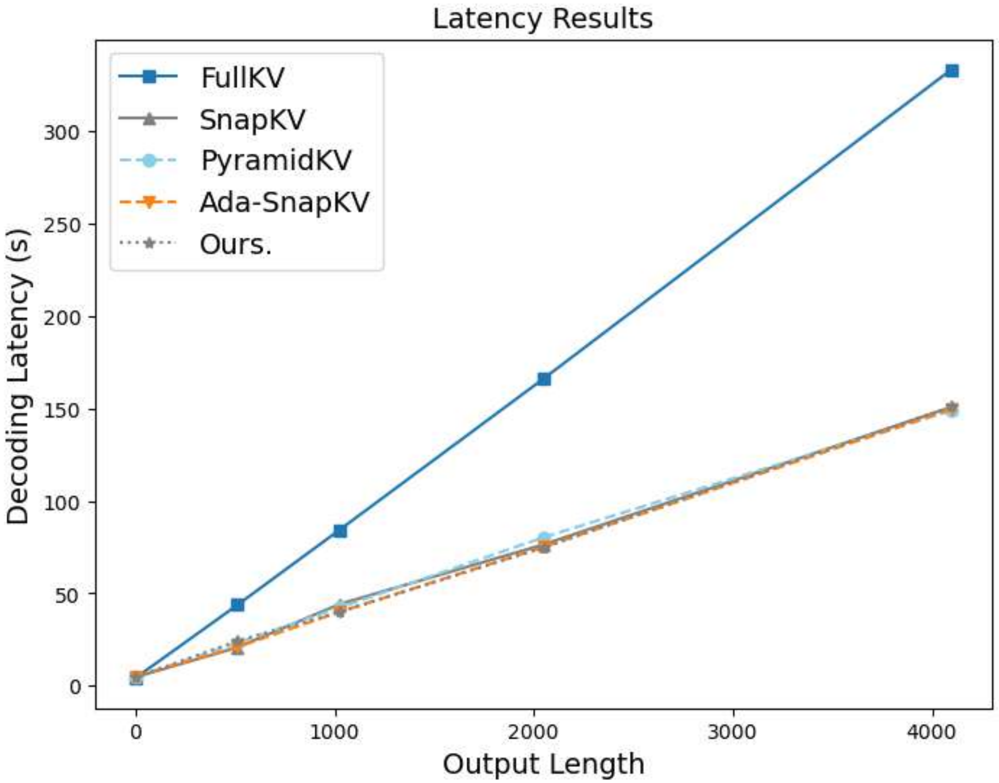


Results across different numbers of KV cache settings.

Ours. Results are significant better than other baselines.

Results





The same latency as the other baselines.

Takeaway

Proposed Head-level KV cache Allocation

FullKV

→ Significant decrease inference latency

→ Maintain the performance

Other KV compression method

→ Outperform all other baselines

→ Keep the same inference speed