Memory

How it is encoded, how it is retrieved, how to improve current systems

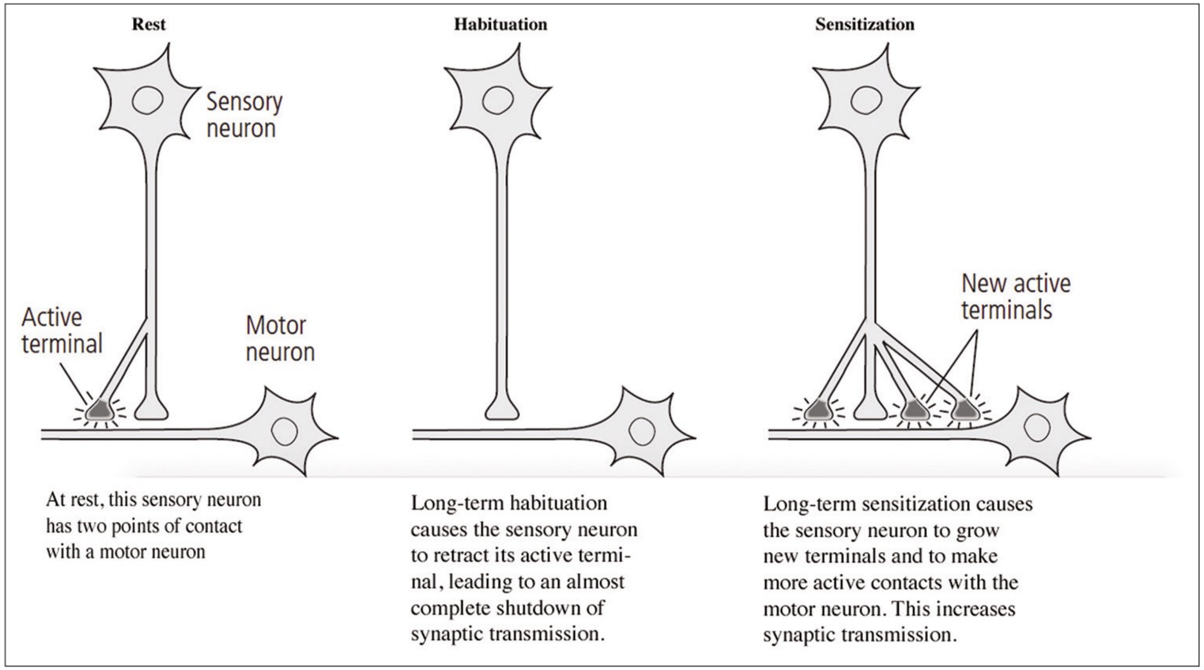
Presented by:
John Tan Chong Min

Memory

How can we store and learn from memory?

In search of memory

- Eric Kandel: Reductionist view
- Long-term potentiation/depression
- Too narrowly focused on just synaptic connection



At rest, this sensory neuron has two points of contact with a motor neuron

Long-term habituation causes the sensory neuron to retract its active terminal, leading to an almost complete shutdown of synaptic transmission.

Long-term sensitization causes the sensory neuron to grow new terminals and to make more active contacts with the motor neuron. This increases synaptic transmission.

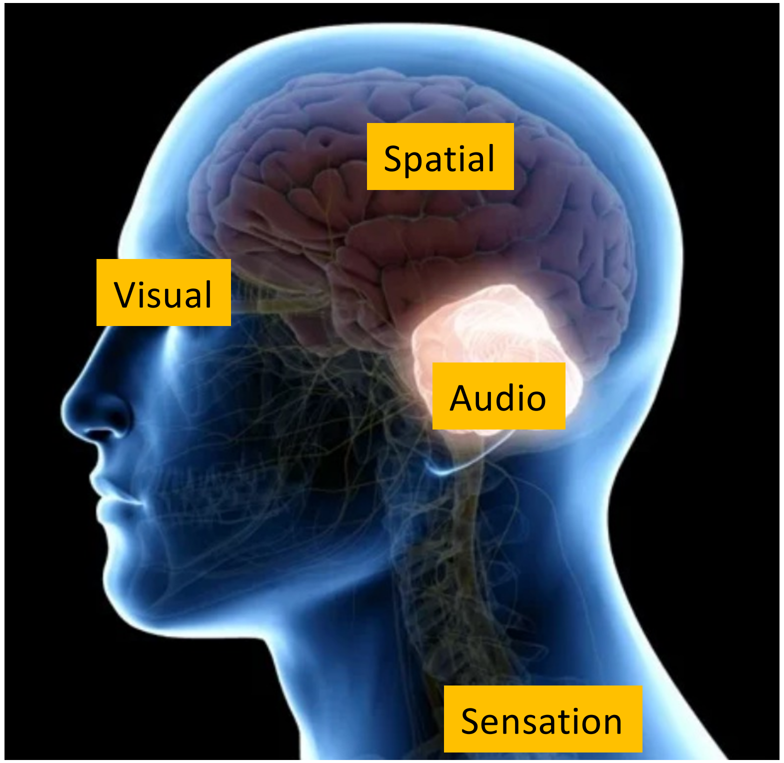
https://surgicalneurologyint.com/surgicalint-articles/the-neurobiology-of-learning-and-memory-as-related-in-the-memoirs-of-eric-r-kandel/

Distributed Memory Storage

- Look at systems of synapses

- The brain takes in multiple inputs from various sensory domains
  - Visual
  - Audio
  - Sensation (e.g. Touch, Pain)
  - Spatial (e.g. Location)

- Need to store these various inputs into a compact form for memory

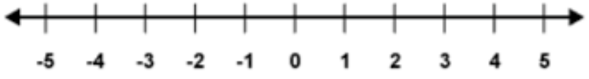


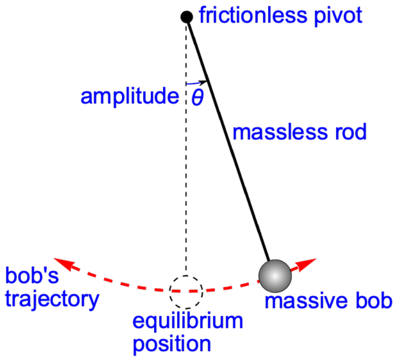
Vector Representation of Memory (my view)

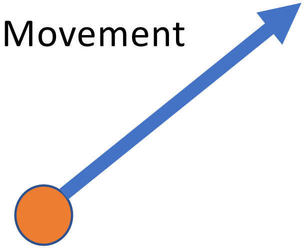
- Initial Reference + Movement

- Naturally represented by vectors

- Fits nicely with concepts
  - **Science:** Initial position/state of object + movement
  - **Movement:** Starting position + action
  - **Geography:** Start from one country and move to another
  - **History:** Starting time and moving up or down timeline
  - **Mathematics:** Starting number and moving up or down number line





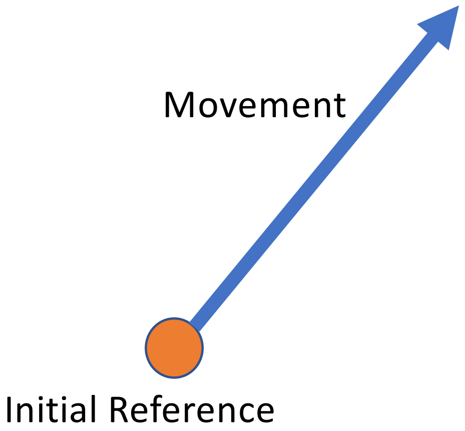


Movement

Initial Reference

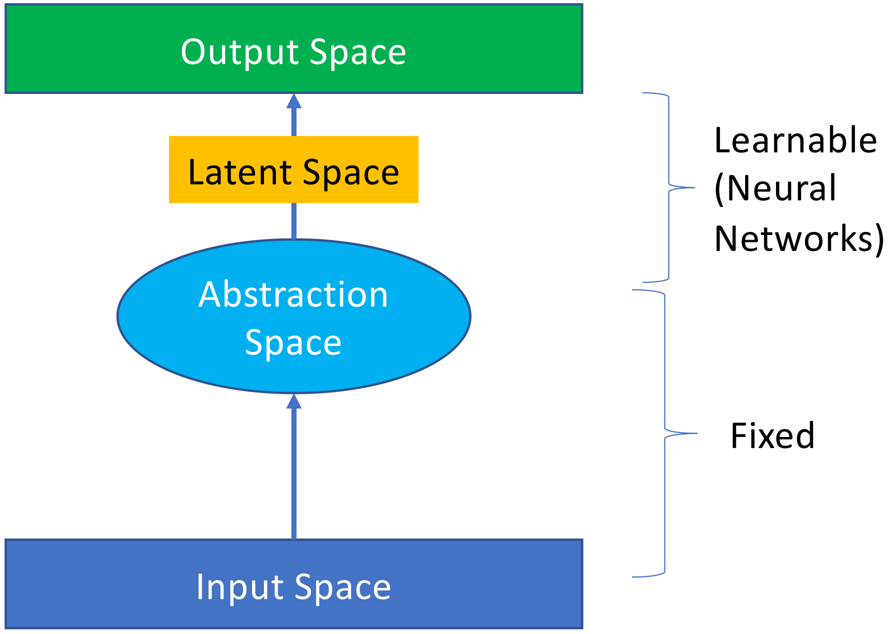
How we generalize from memory (my view)

- When we retrieve memories, the initial reference can be adjusted to suit the situation
- That way, the movement obtained can be applied to the new situation to get the desired outcome
- We can probably simulate various outcomes by retrieving from memory, applying the movement to current situation
- Current vectors don't disentangle between reference and movement – may be required for better memory referencing?



Abstraction (my view)

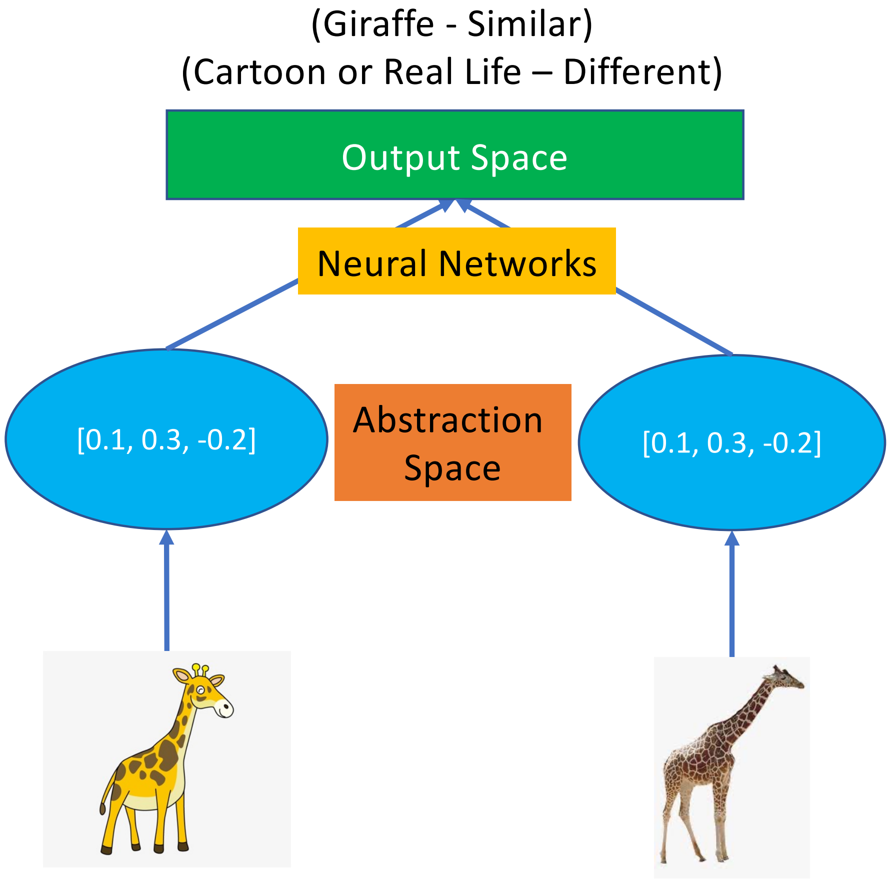
- Memories are stored in an abstract form
- The same abstraction can then referenced across contexts (generalization)
- This enables reuse of learnt memories in different contexts
- Fixed representation can be learnt over an initial training set of data



Abstraction (my view)

- **Hypothesis: Abstraction process is fixed and unlearnable, which stores new memories without the need of updating ALL previously stored memories the moment the abstraction mapping changes**

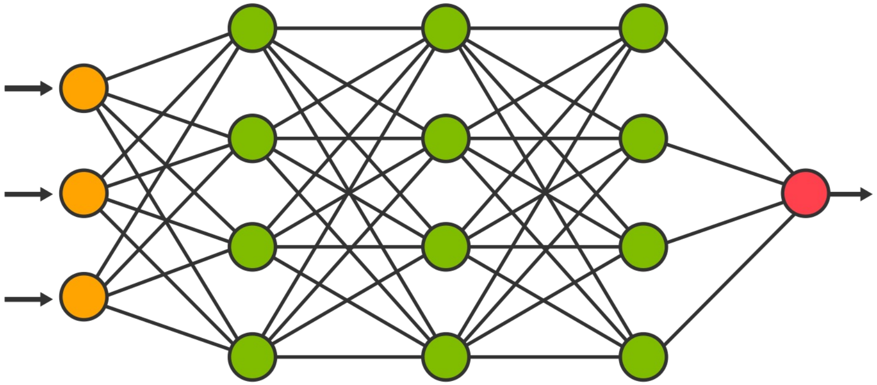
- With the abstraction space as input, neural networks can learn associations / dissociations between them for various tasks



Types of Memory

Biological and non-biological

Types of memory

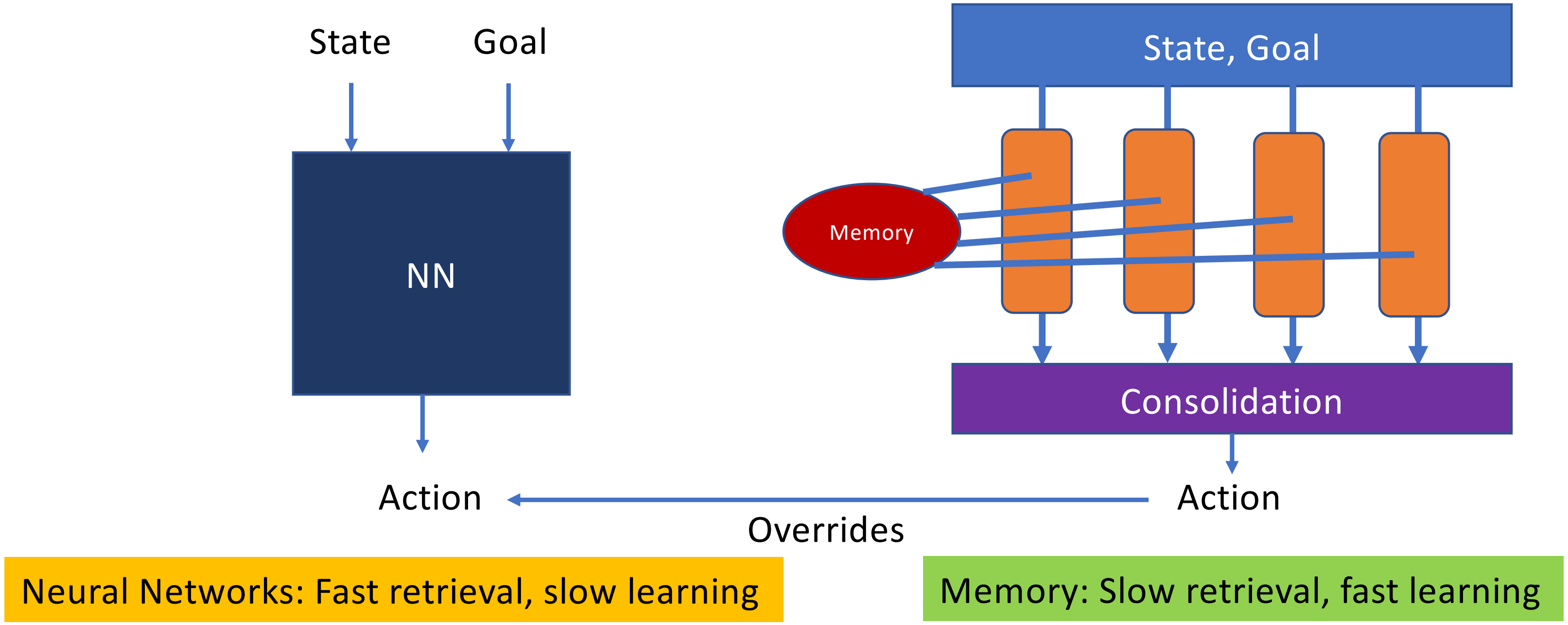


- Fast inference (one pass)
- Slow to learn (requires gradient descent)

- Slow inference unless using approximate techniques
- Can be quick to learn from experience (not done in this paper)

Link to my work (Learning, Fast & Slow)

- Learning is done in both fast and slow systems (slow more important at first)
- Memory (slow system) fades over time so knowledge is retained in fast system



State

Goal

NN

Action

State, Goal

Memory

Consolidation

Action

Overrides

Neural Networks: Fast retrieval, slow learning

Memory: Slow retrieval, fast learning

Learning, Fast and Slow: A Goal-Directed Memory-Based Approach for Dynamic Environments. 2023. John and Motani.

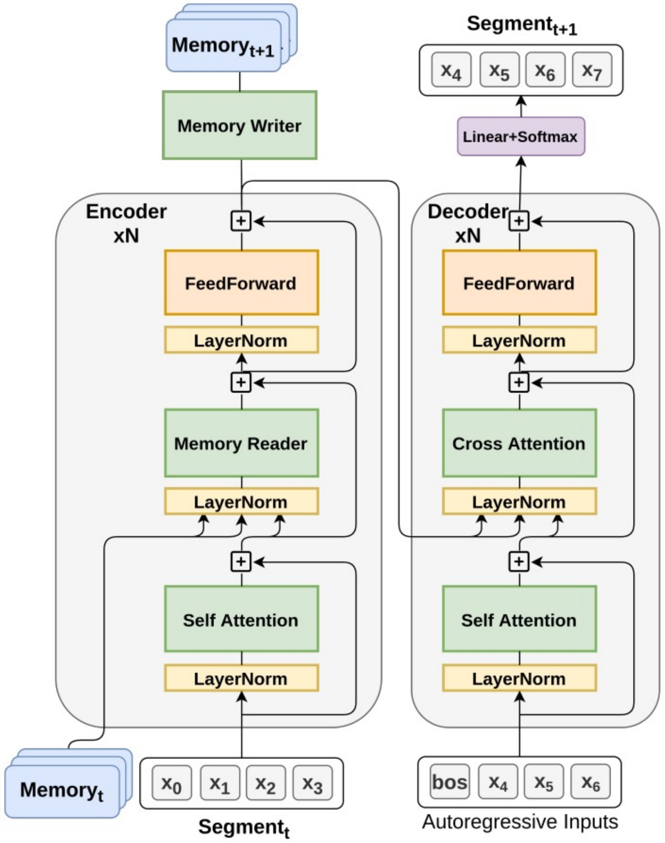
Memory in vector embeddings

Memory in Encoder: Memformer

- Dynamic memory fed into the Encoder part of Transformers

- Memory can be written over and reused in subsequent iterations

- Memory used is of fixed length, can be hard to fit in arbitrary length input prompts



Memformer: A Memory-Augmented Transformer for Sequence Modeling. Wu et al. 2022

Recurrent Memory Transformer

- Memory appended as a prefix to each segment
- Next segment's memory generated by previous Transformer in recurrent fashion
- May be slow in processing and incur compression loss
  - But attention mechanism scales sub-quadratically due to splitting of segment

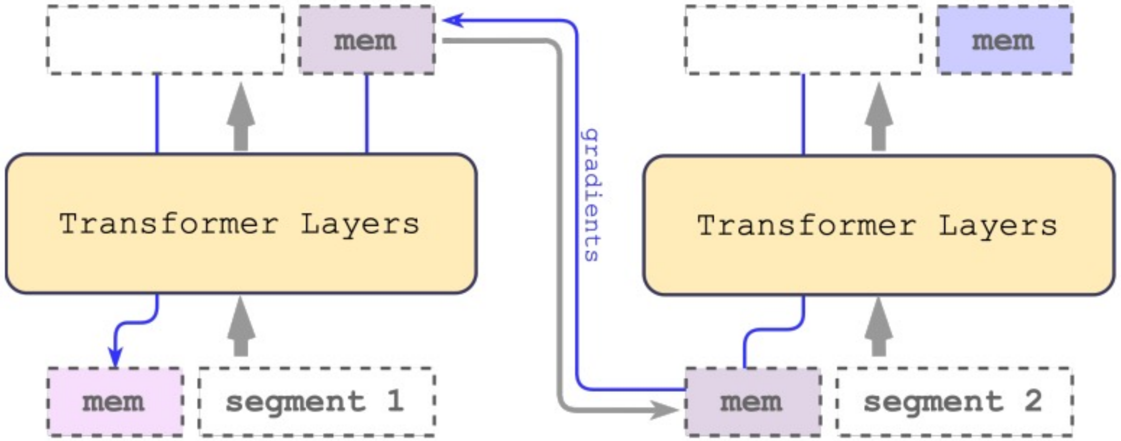


Figure 2: **Recurrent memory mechanism.** Memory is passed to Transformer along input sequence embeddings, and memory output is passed to the next segment. During training gradients flow from the current segment through memory to the previous segment.

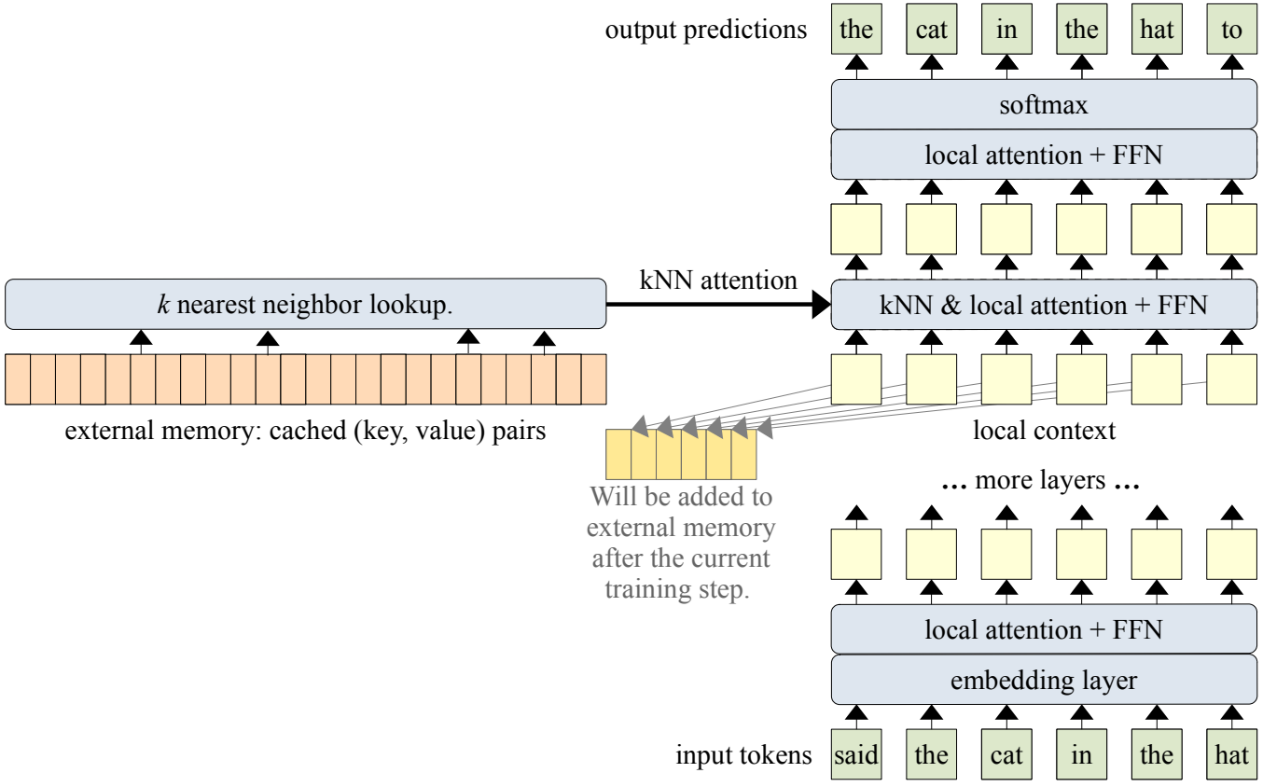
Recurrent Memory Transformer. Bulatov et al. 2022.
Scaling Transformer to 1M tokens and beyond with RMT. Bulatov et al. 2022.

Unlimited context length: kNN-based key-value augmentation

- Can theoretically attend to unlimited numbers of neighbours

- Memory is stored as embedding space of the layer before softmax
  - Only works for tokens of the same sequence, since attention is only for previous tokens

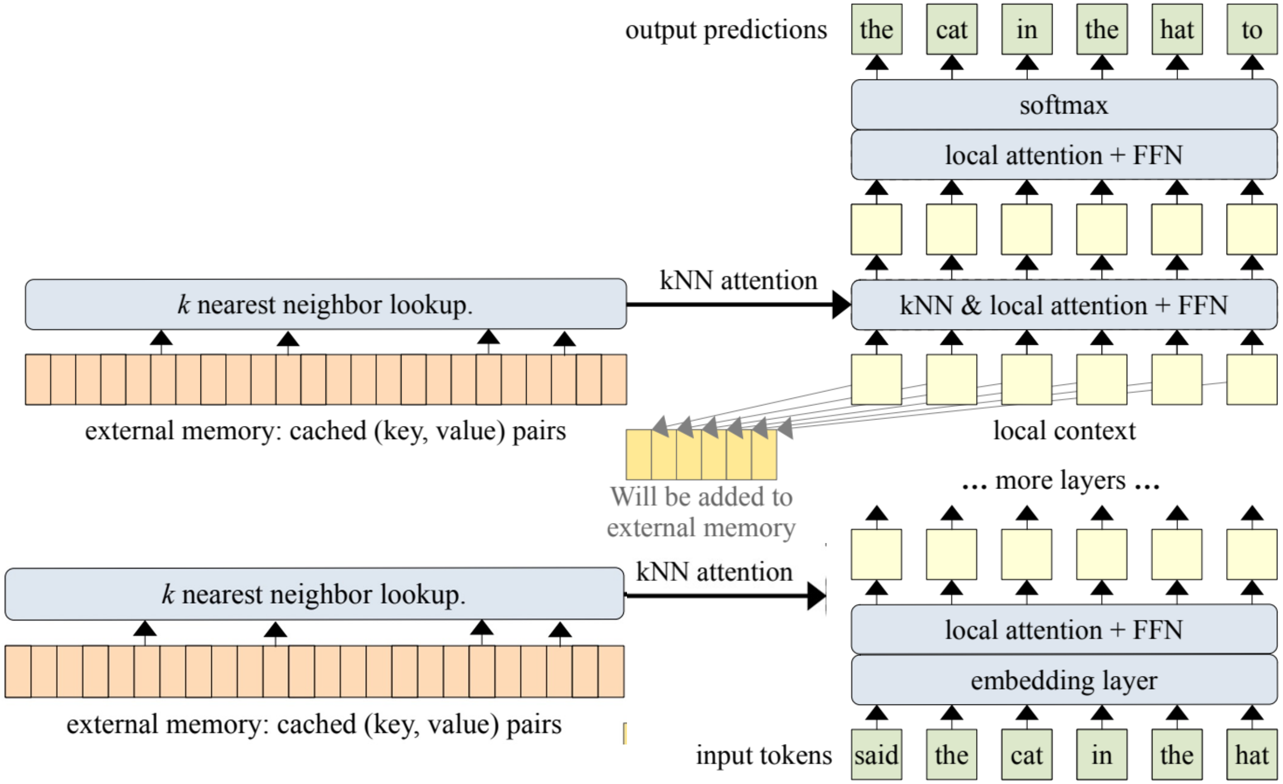
- Can do approximate kNN like ScaNN or Faiss



Memorizing Transformers. Wu et al. 2022.

Hierarchical Memory Referencing in Embeddings (my idea)

- Memory referencing should be done at multiple layers of the hierarchy
- Think about how we can go up and down scales of memory:
    - Tying a shoe (Base)
    - How do you tie a shoe? (One level lower)
    - Why do you tie a shoe? (One level higher)
- Accessing memory at various levels should lead to more accurate retrieval

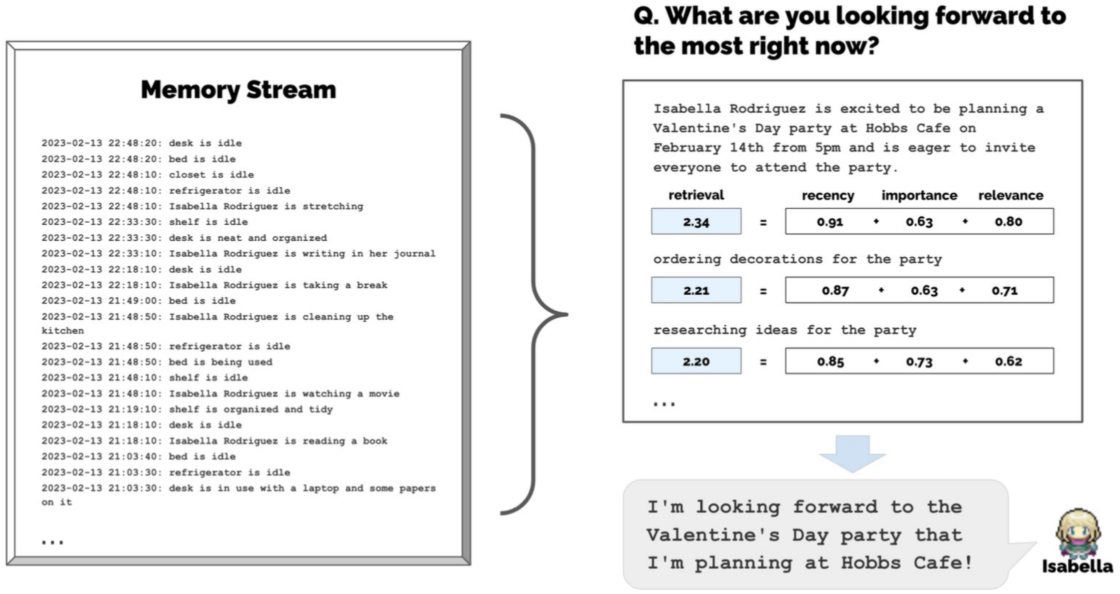

output predictions
the
cat
in
the
hat
to
softmax
local attention + FFN
k nearest neighbor lookup.
kNN attention
kNN & local attention + FFN
external memory: cached (key, value) pairs
local context
Will be added to external memory
... more layers ...
k nearest neighbor lookup.
kNN attention
local attention + FFN
embedding layer
external memory: cached (key, value) pairs
input tokens
said
the
cat
in
the
hat

Modified from: Memorizing Transformers. Wu et al. 2022.

Memory in text

Text-based memory

- Memory stored as text
- Retrieved by recency, importance, relevance
- Hierarchical prompting from most broad action to specific actions



Memory Stream

```
2023-02-13 22:48:20: desk is idle
2023-02-13 22:48:20: bed is idle
2023-02-13 22:48:10: closet is idle
2023-02-13 22:48:10: refrigerator is idle
2023-02-13 22:48:10: Isabella Rodriguez is stretching
2023-02-13 22:33:30: shelf is idle
2023-02-13 22:33:30: desk is neat and organized
2023-02-13 22:33:10: Isabella Rodriguez is writing in her journal
2023-02-13 22:18:10: desk is idle
2023-02-13 22:18:10: Isabella Rodriguez is taking a break
2023-02-13 21:49:00: bed is idle
2023-02-13 21:48:50: Isabella Rodriguez is cleaning up the
kitchen
2023-02-13 21:48:50: refrigerator is idle
2023-02-13 21:48:50: bed is being used
2023-02-13 21:48:10: shelf is idle
2023-02-13 21:48:10: Isabella Rodriguez is watching a movie
2023-02-13 21:19:10: shelf is organized and tidy
2023-02-13 21:18:10: desk is idle
2023-02-13 21:18:10: Isabella Rodriguez is reading a book
2023-02-13 21:03:40: bed is idle
2023-02-13 21:03:30: refrigerator is idle
2023-02-13 21:03:30: desk is in use with a laptop and some papers
on it

...
```

Q. What are you looking forward to the most right now?

```
Isabella Rodriguez is excited to be planning a
Valentine's Day party at Hobbs Cafe on
February 14th from 5pm and is eager to invite
everyone to attend the party.
```

retrieval		recency		importance		relevance
2.34	=	0.91	+	0.63	+	0.80

```
ordering decorations for the party
```

retrieval		recency		importance		relevance
2.21	=	0.87	+	0.63	+	0.71

```
researching ideas for the party
```

retrieval		recency		importance		relevance
2.20	=	0.85	+	0.73	+	0.62

...

```
I'm looking forward to the
Valentine's Day party that
I'm planning at Hobbs Cafe!
```

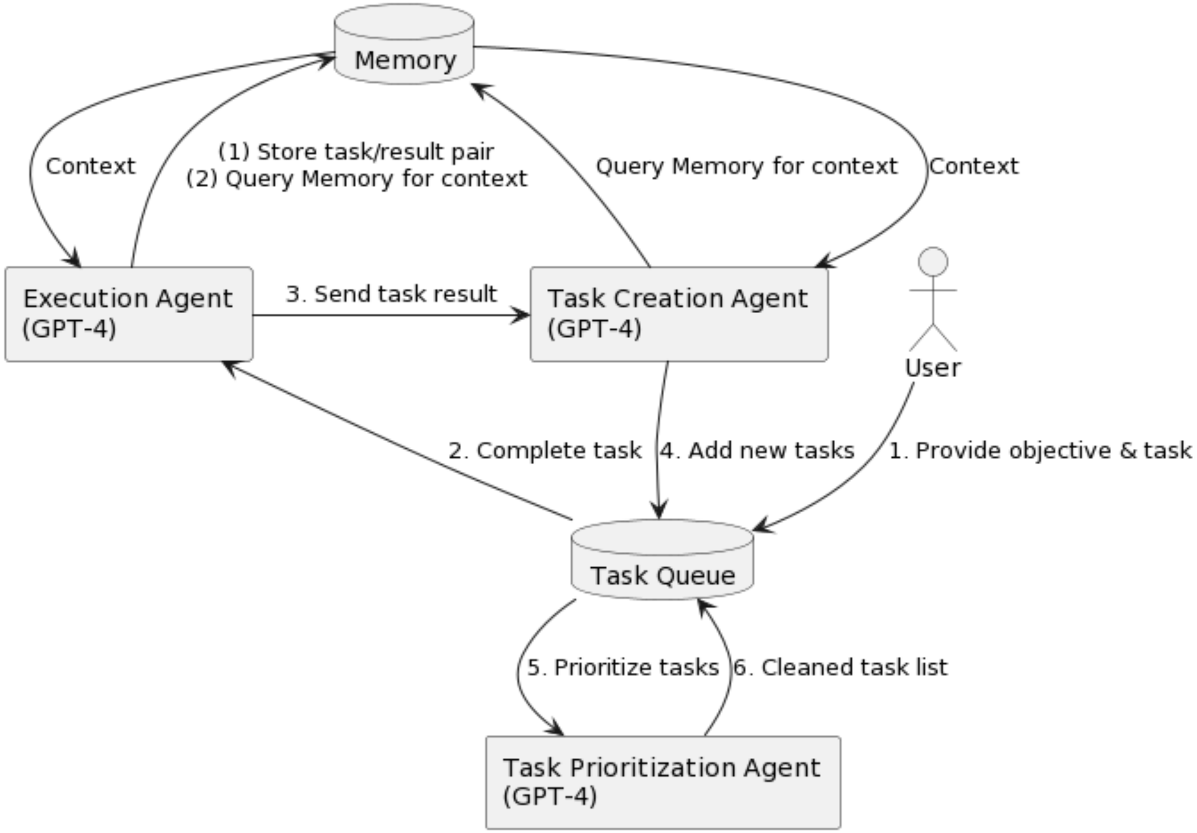
Figure 6: The memory stream comprises a large number of observations that are relevant and irrelevant to the agent's current situation. Retrieval identifies a subset of these observations that should be passed to the language model to condition its response to the situation.

Generative Agents: Interactive Simulacra of Human Behavior. Joon et al. 2022

External Task-based Memory: Task-driven autonomous agent utilizing GPT-4, Pinecone and Langchain for Diverse Applications

- Uses GPT4 to in a larger ecosystem to:
  - Create Tasks
  - Execute Tasks
  - Prioritize Tasks

- Uses memory to store and retrieve task/result pairs
  - Stored by vector embeddings

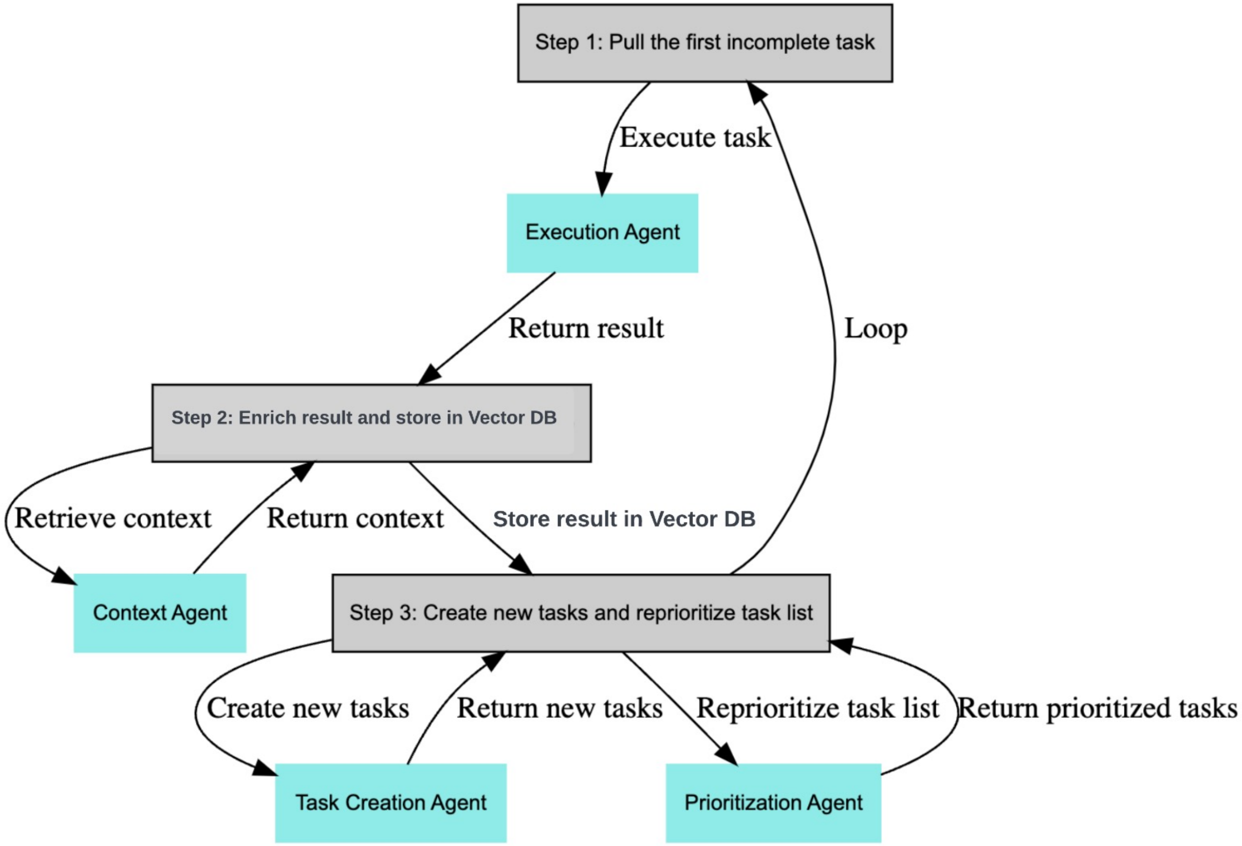


https://yoheinakajima.com/task-driven-autonomous-agent-utilizing-gpt-4-pinecone-and-langchain-for-diverse-applications/

BabyAGI

- New Context Agent to **enrich context** in vector database

- My thoughts:
  - This could help shape the memory to be more relevant for future retrieval
  - Why not also do a context enrichment before executing the task?

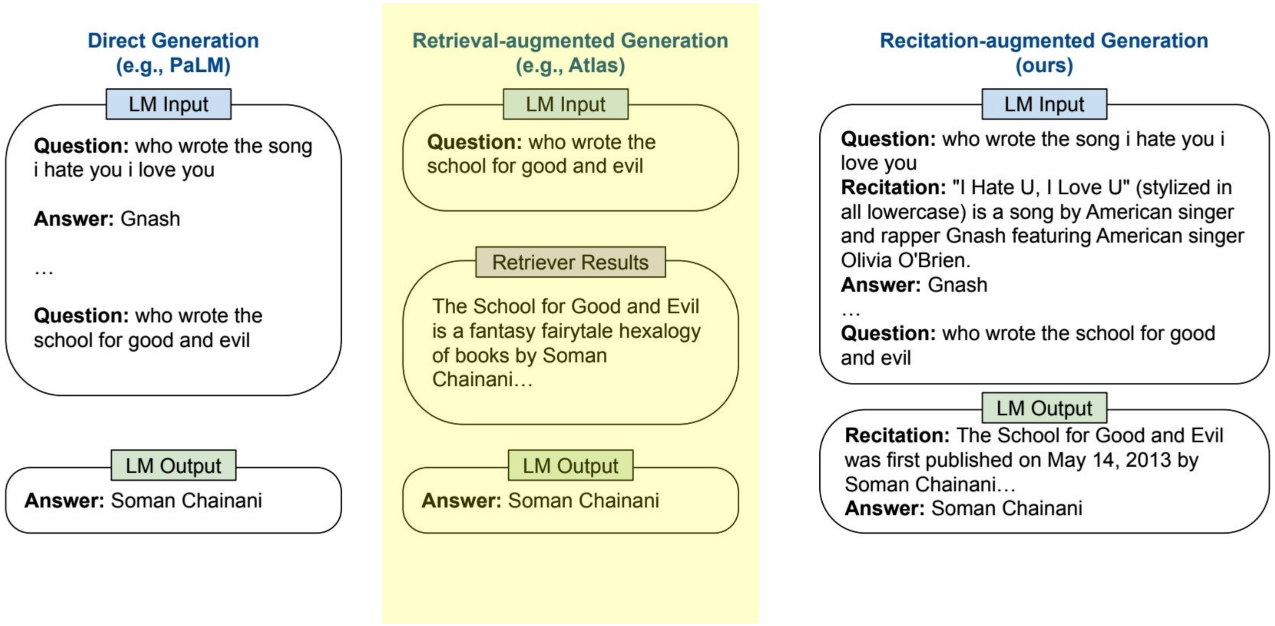


https://github.com/yoheinakajima/babyagi

Retrieval-Augmented Generation

- Past memories (extracted using relevance criteria like cosine similarity) used to ground query

- More consistent and accurate generation



Direct Generation (e.g., PaLM)

LM Input

Question: who wrote the song i hate you i love you

Answer: Gnash

…

Question: who wrote the school for good and evil

LM Output

Answer: Soman Chainani

Retrieval-augmented Generation (e.g., Atlas)

LM Input

Question: who wrote the school for good and evil

Retriever Results

The School for Good and Evil is a fantasy fairytale hexalogy of books by Soman Chainani…

LM Output

Answer: Soman Chainani

Recitation-augmented Generation (ours)

LM Input

Question: who wrote the song i hate you i love you
Recitation: "I Hate U, I Love U" (stylized in all lowercase) is a song by American singer and rapper Gnash featuring American singer Olivia O'Brien.
Answer: Gnash
…
Question: who wrote the school for good and evil

LM Output

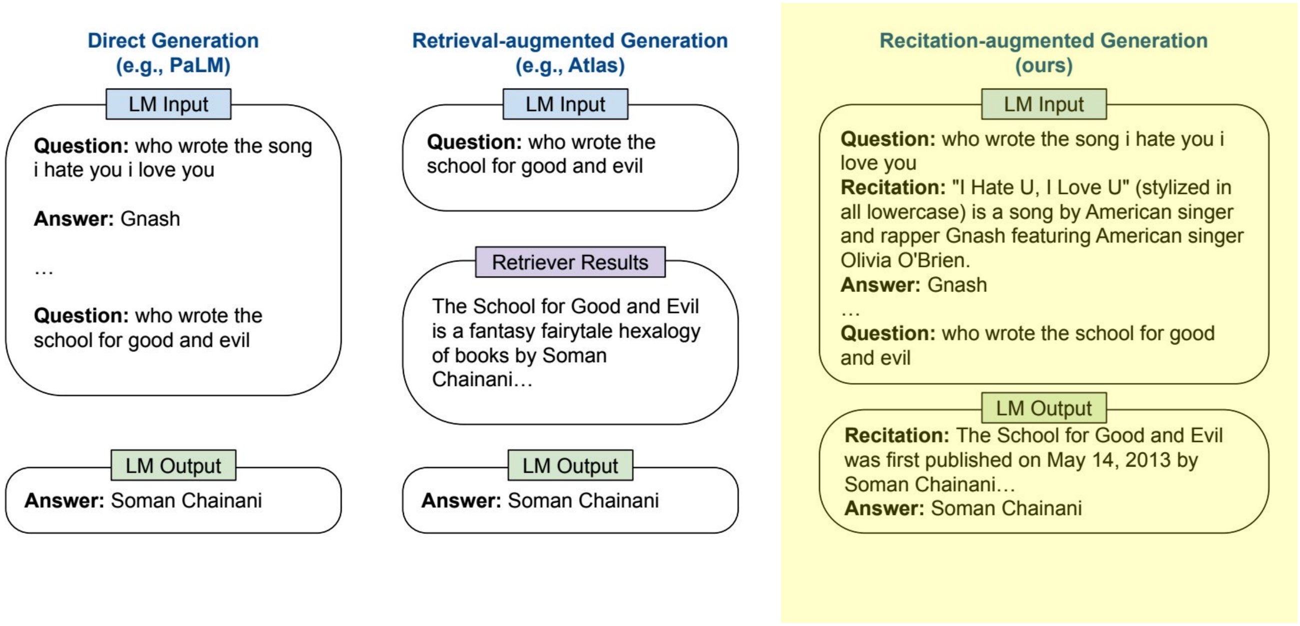
Recitation: The School for Good and Evil was first published on May 14, 2013 by Soman Chainani…
Answer: Soman Chainani

Recitation-Augmented Language Models. Sun et al. 2023.

Recitation-Augmented Generation

- Original Paper: Internal memories (without external reference) can be retrieved to fit multiple contexts (recitation) and improve generation

- My View:
  - You could even alter memories from external sources to fit the context
  - Modification of memory may be akin to modifying the initial reference frame of memory


Direct Generation (e.g., PaLM)
LM Input
LM Output
Retrieval-augmented Generation (e.g., Atlas)
LM Input
Retriever Results
LM Output
Recitation-augmented Generation (ours)
LM Input
LM Output

Question: who wrote the song i hate you i love you

Answer: Gnash

…

Question: who wrote the school for good and evil

Answer: Soman Chainani

Question: who wrote the school for good and evil

The School for Good and Evil is a fantasy fairytale hexalogy of books by Soman Chainani…

Answer: Soman Chainani

Question: who wrote the song i hate you i love you
Recitation: "I Hate U, I Love U" (stylized in all lowercase) is a song by American singer and rapper Gnash featuring American singer Olivia O'Brien.
Answer: Gnash
…
Question: who wrote the school for good and evil

Recitation: The School for Good and Evil was first published on May 14, 2013 by Soman Chainani…
Answer: Soman Chainani

Recitation-Augmented Language Models. Sun et al. 2023.

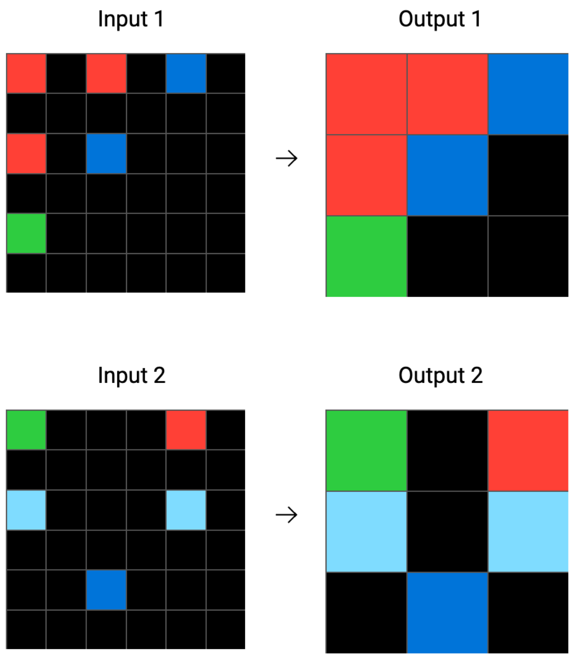
Hierarchical Memory Referencing in Text (my idea)

- Abstraction and Reasoning Corpus (ARC) Challenge

- **Broad Intent:** Reduce the input grid to a smaller size
  - Can reference/recite similar broad intents from memory to refine broad intent

- **Detailed Steps (conditioned on Broad Intent):** Remove every other square from the row and columns of the grid
  - Can reference/recite similar detailed steps from memory to refine detailed steps

- Execution: Perform the detailed steps on the test input to get the answer



Questions to ponder

- Should we encode memory as vector embeddings, or text, or both?
- We can perform recitation to alter memories to be more suitable for the current context. How can we do the same for embeddings?
- Should we store embeddings as two components, initial reference and movement? Is the initial reference part similar to the Key, and the movement the Value? Should we then add up the Value to the refined context (Query)?
- In general, having more retrieval from memory to ground context helps with output accuracy. How can we best improve context length to cater for this?