# Мультимодальные и поведенческие модели

*Memory, Context, and Long-Horizon Planning*

**MIPT VLA 2026 | 9 ноября 2026**

Алексей Староверов



МФТИ

Center for Cognitive Modeling

## План лекции

About memory

Memory token based memory

VLM based memory

# Why do we need context & memory

**Разумное поведение** предполагает выбор действий на основе имеющейся информации

Информация для принятия решения включает **данные из прошлого**, а не только текущие наблюдения

**Память** - способность накапливать, сохранять и воспроизводить информацию

Без памяти агент может реагировать только на текущие стимулы, не учитывая историю (**реактивное поведение**)

Если мы хотим получить **роботов, обладающих разумным поведением**, требуется разработка механизмов памяти

# Taxonomy of the human memory

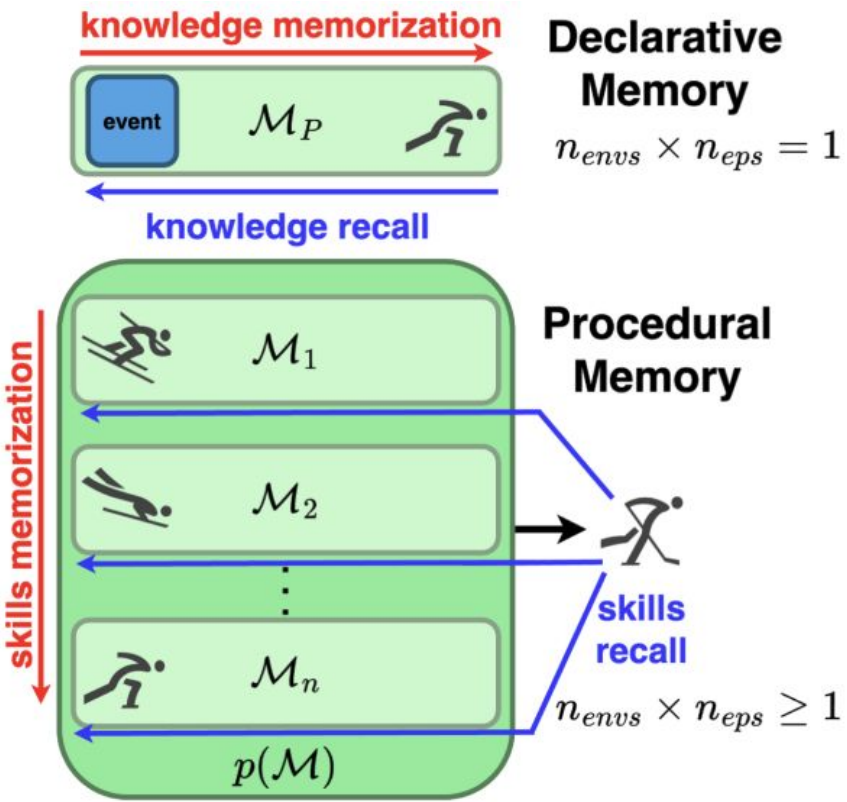
Типы памяти:

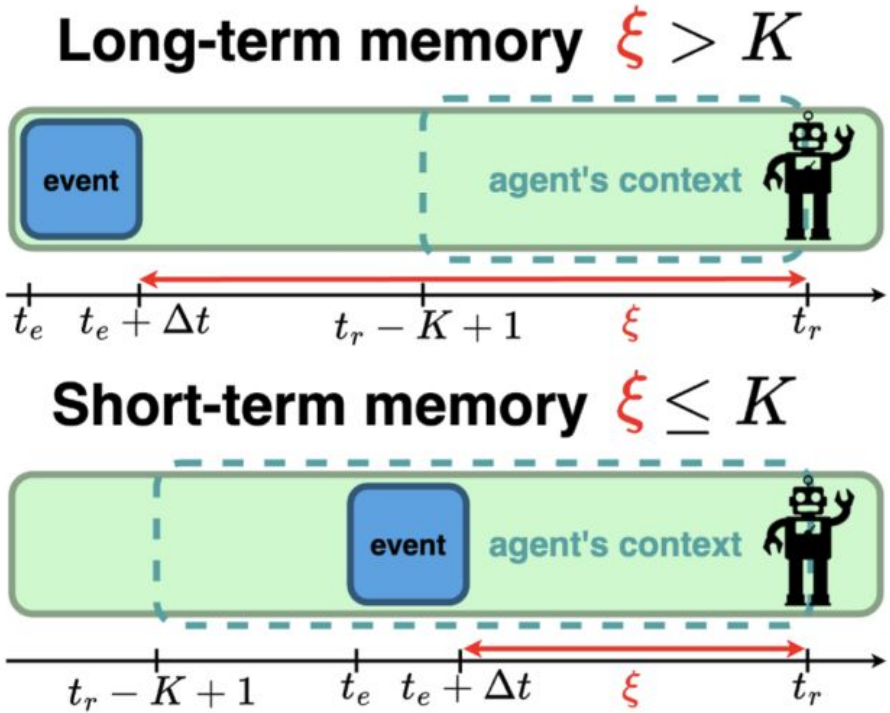
- a. **эпизодическая** - о конкретных событиях и личном опыте с временным и пространственным контекстом
- b. **семантическая** - обобщенные знания о мире, понятиях и фактах без привязки к конкретному событию
- c. **рабочая** - механизм кратковременного удержания и обработки информации, необходимой для текущей задачи.
- d. **процедурная** - память навыков и автоматизированных действий, проявляющаяся через выполнение, а не через воспоминание
- e. **сенсорная** память - кратчайшее удержание сырых сенсорных сигналов для их первичной обработки

Дополнительное деление:

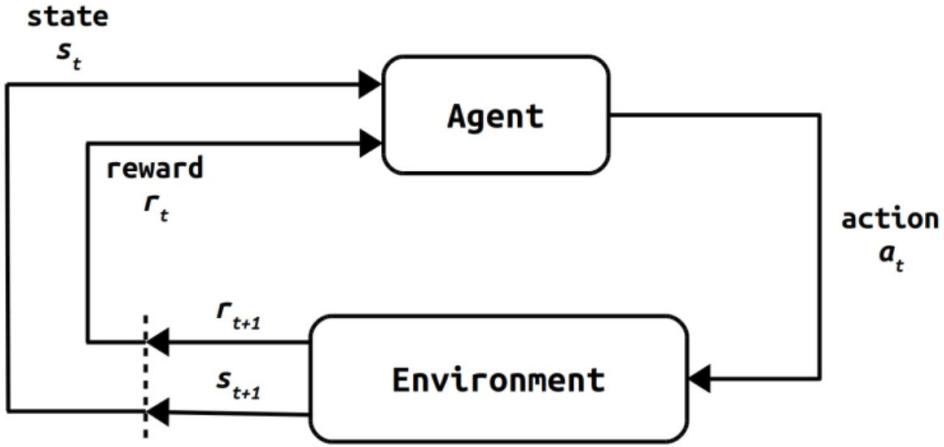
- f. **декларативная** - можно ли осознанно вспомнить и описать событие?
- g. **недекларативная** - проявляется через навыки, реакции и поведение без обязательного осознанного воспоминания
- h. **долгосрочная** / **краткосрочная**

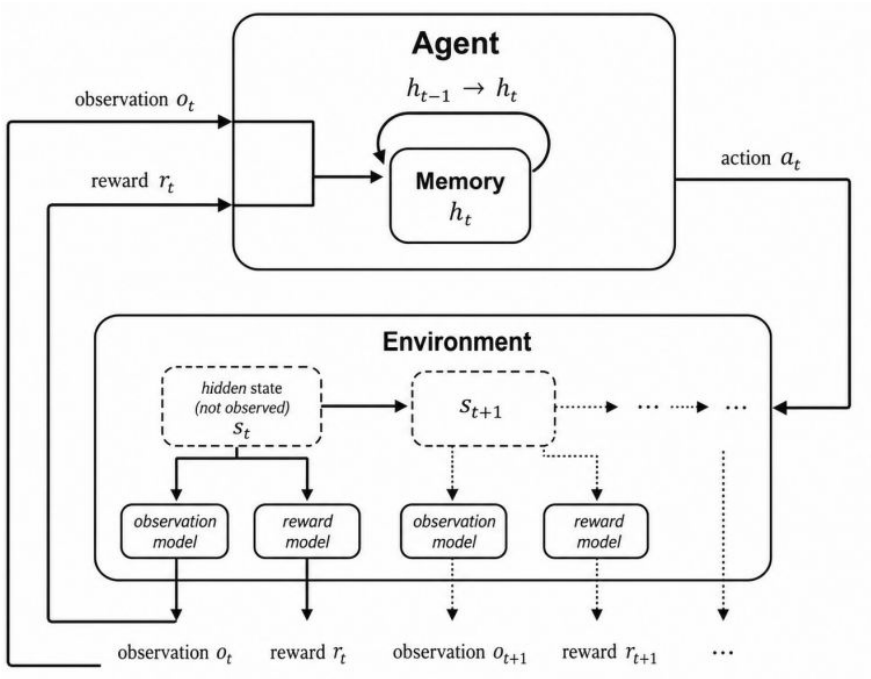
# Taxonomy of the RL agents memory



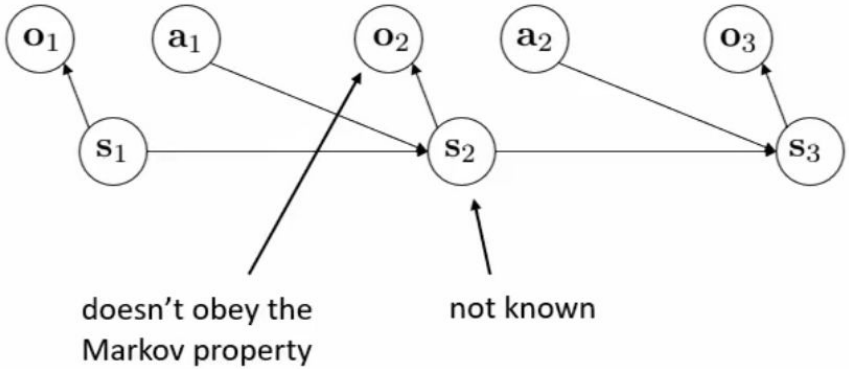



# RL (MPD → POMDP)

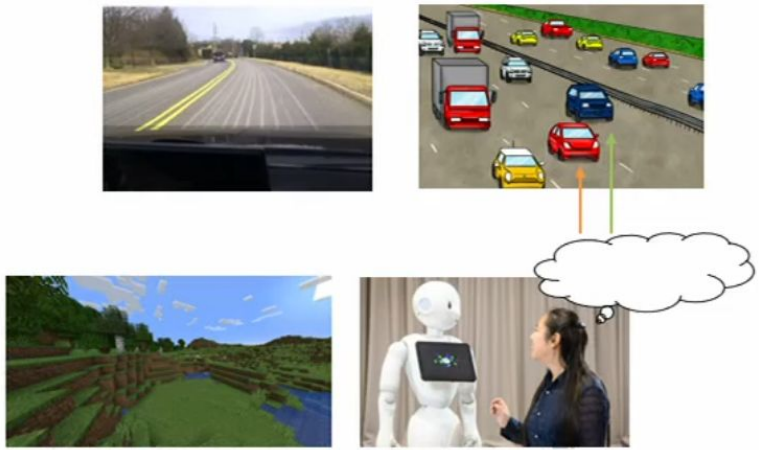

state
$s_t$
Agent
reward
$r_t$
action
$a_t$
$r_{t+1}$
$s_{t+1}$
Environment


Agent
observation $o_t$
$h_{t-1} \rightarrow h_t$
reward $r_t$
Memory
$h_t$
action $a_t$
Environment
hidden state
(not observed)
$s_t$
$s_{t+1}$
observation model
reward model
observation model
reward model
observation $o_t$
reward $r_t$
observation $o_{t+1}$
reward $r_{t+1}$

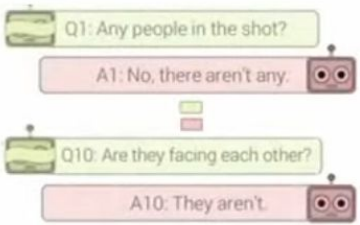
# Beyond MDPs



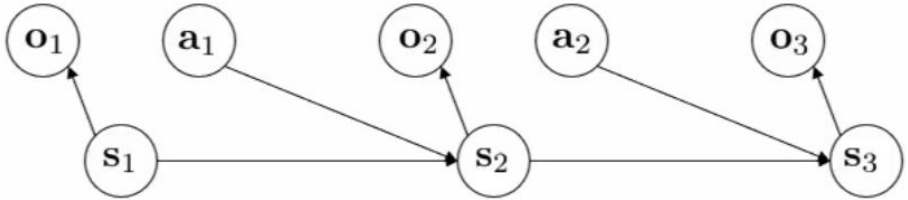

**most** real-world problems are like this!





# Partially observed MDPs can be weird

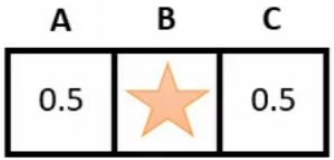



**Example 1:** information-gathering actions





**Example 2:** stochastic optimal policies



# Which methods handle partial observability?

Policy gradients

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})A(\mathbf{s}_{i,t},\mathbf{a}_{i,t}) \quad \overset{??}{\Rightarrow} \quad \nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{o}_{i,t})A(\mathbf{o}_{i,t},\mathbf{a}_{i,t})$$

Value-based methods

$$Q(\mathbf{s},\mathbf{a}) \leftarrow r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') \quad \overset{??}{\Rightarrow} \quad Q(\mathbf{o},\mathbf{a}) \leftarrow r(\mathbf{o},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{o}',\mathbf{a}')$$

# Which methods handle partial observability?

Policy gradients

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})A(\mathbf{s}_{i,t},\mathbf{a}_{i,t}) \quad ?? \quad \nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{o}_{i,t})A(\mathbf{o}_{i,t},\mathbf{a}_{i,t})$$

this is OK (no Markov property!)

this takes some care

**Key point**: advantage is a function of the state $\mathbf{s}_t$

it does *not* depend on $\mathbf{s}_{t-1}$

that's why it's OK to use $r_t + \gamma\hat{V}(\mathbf{s}_{t+1}) - \hat{V}(\mathbf{s}_t)$

every time we see this state, we expect to get this value, regardless of past states

it is *not* OK to train $\hat{V}(\mathbf{o}_t)$

past observations **do** matter for this value

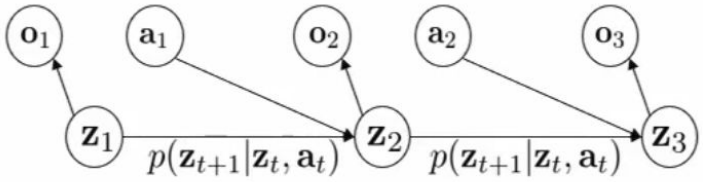
$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{o}_{i,t})\right)\left(\sum_{t=1}^{H} r(\mathbf{o}_{i,t},\mathbf{a}_{i,t})\right)$$



$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{o}_{i,t})(r_{i,t} + \gamma\hat{V}(\mathbf{o}_{i,t+1}) - \hat{V}(\mathbf{o}_{i,t}))$$



# History states

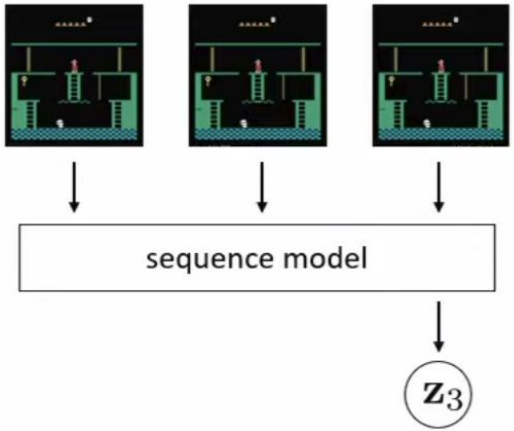



$$q_\phi(\mathbf{z}|\mathbf{o}) = \prod_t q_\phi(\mathbf{z}_t|\mathbf{o}_{1:t})$$

state is inferred from a *history*

state is a *function* of history

can we just give the history to our value function?




$$\mathbf{s}_t = (\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t)$$

Does that work?

Does that obey the Markov property?

$$\mathbf{s}_{t+1} \perp \mathbf{s}_{t-1} | \mathbf{s}_t$$

$(\mathbf{o}_1, ..., \mathbf{o}_{t-1})$ $(\mathbf{o}_1, ..., \mathbf{o}_t)$

# Learned memory

**MDP: current state is sufficient**

$$a_t = \pi(s_t)$$

**POMDP: policy over the belief state**

$$b_t = P(s_t \mid o_{0:t}, a_{0:t-1}, r_{0:t-1}), \qquad a_t = \pi(b_t)$$

**POMDP: policy over the full interaction history**

$$a_t = \pi(o_0, a_0, r_0, o_1, a_1, r_1, \ldots, o_t)$$

Equivalently,

$$\tau_t = (o_{0:t}, a_{0:t-1}, r_{0:t-1}), \qquad a_t = \pi(\tau_t)$$

**POMDP with learned memory**

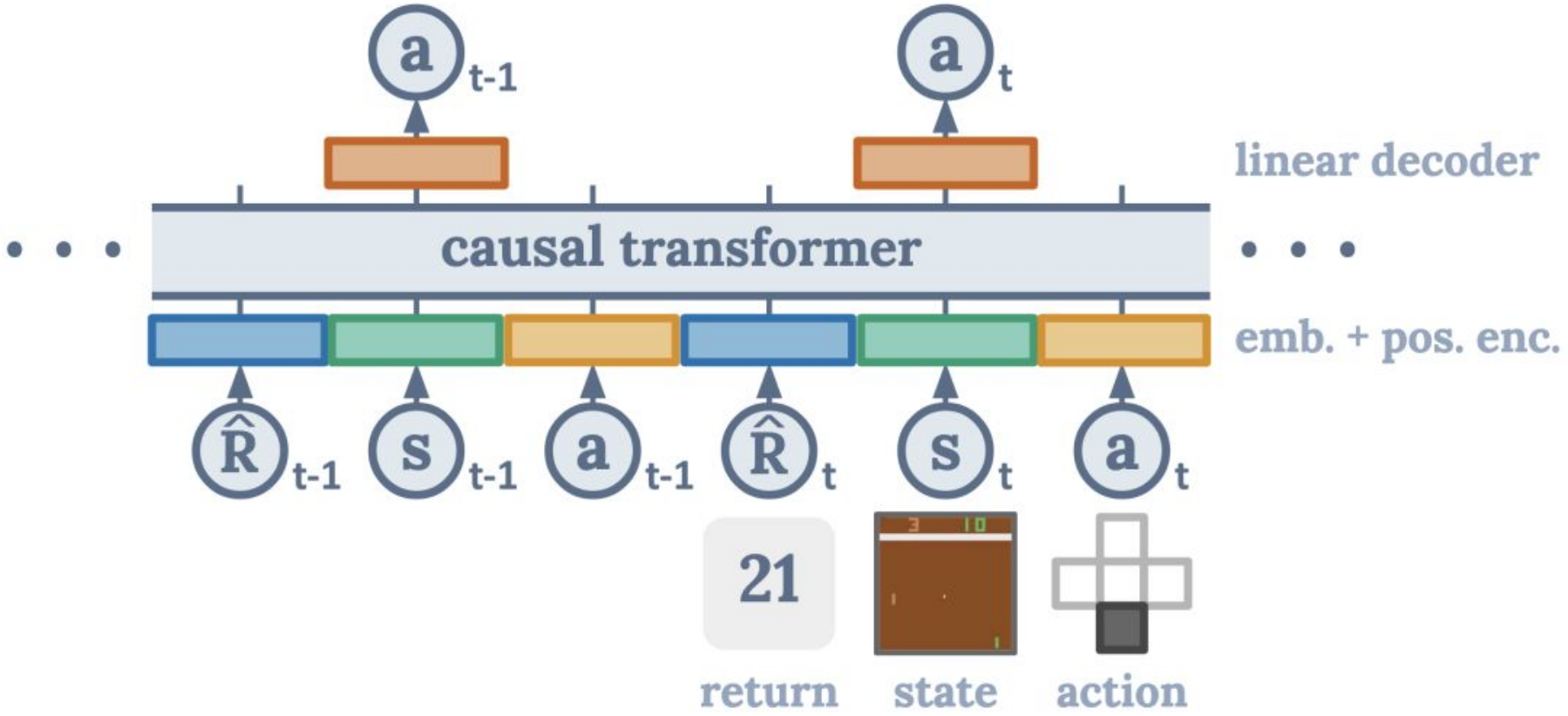
$$m_t = f(m_{t-1}, o_t, a_{t-1}, r_{t-1}), \qquad a_t = \pi(o_t, m_t)$$

$$\underbrace{\pi(o_{0:t}, a_{0:t-1}, r_{0:t-1})}_{\text{policy over the full history}} \approx \underbrace{\pi(o_t, m_t)}_{\text{policy with compressed memory}}$$

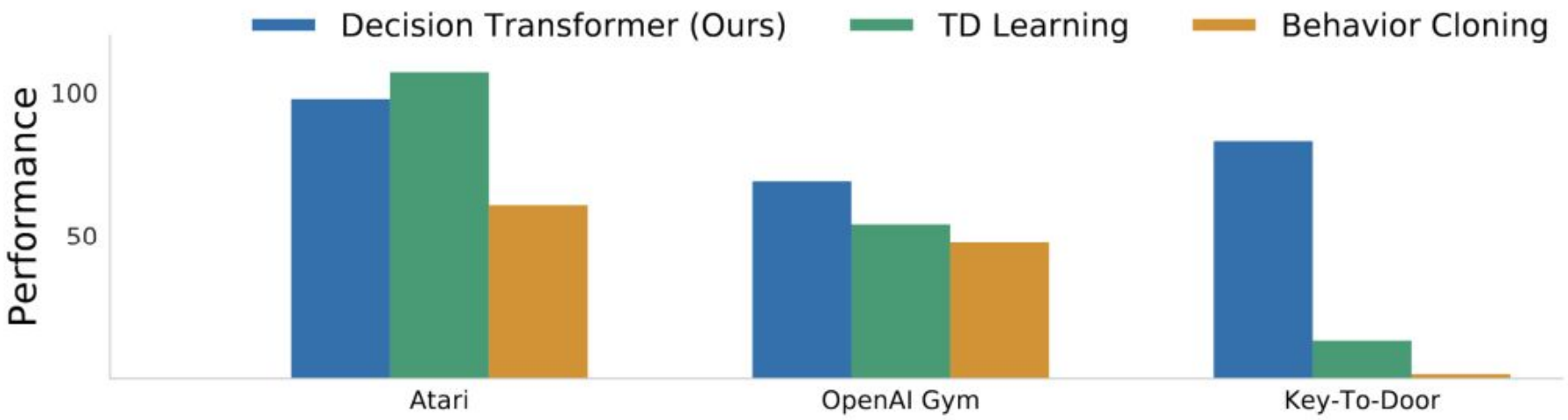
Neural Map: Structured memory for deep RL

$$\begin{aligned}
r_t &= read(M_t) \\
c_t &= context(M_t, s_t, r_t) \\
w_{t+1}^{(x_t,y_t)} &= write(s_t, r_t, c_t, M_t^{(x_t,y_t)}) \\
M_{t+1} &= update(M_t, w_{t+1}^{(x_t,y_t)}) \\
o_t &= [r_t, c_t, w_{t+1}^{(x_t,y_t)}] \\
\pi_t(a|s) &= \mathrm{Softmax}(f(o_t)),
\end{aligned}$$

# Decision Transformer: Reinforcement Learning via Sequence Modeling



# Decision Transformer: Reinforcement Learning via Sequence Modeling

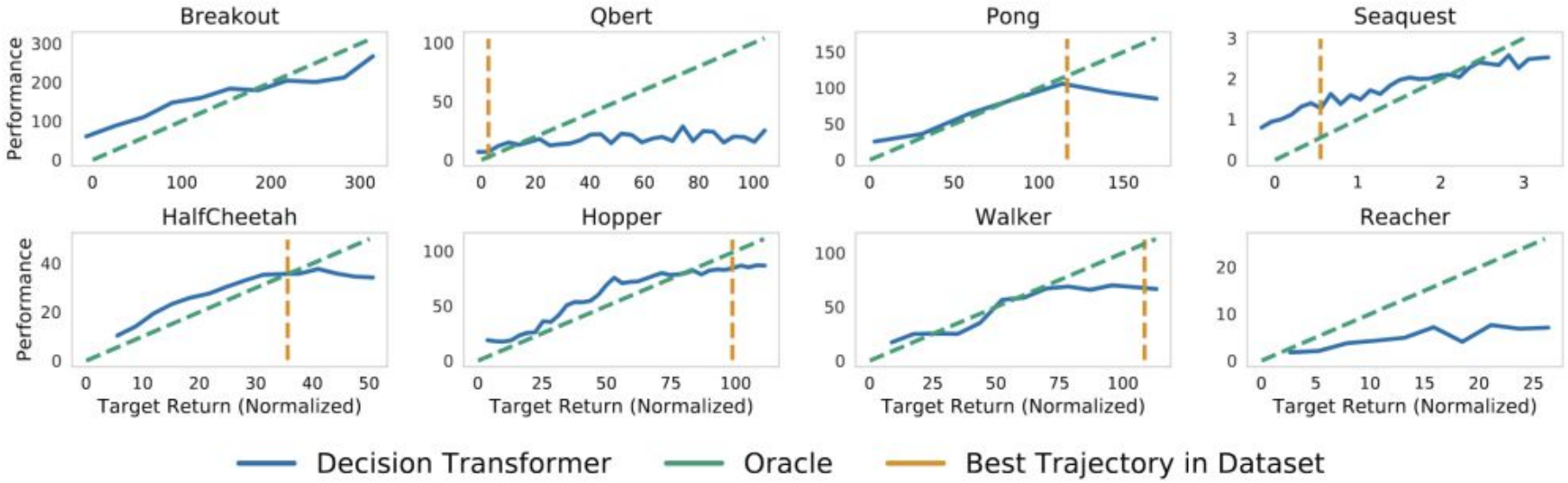


## Decision Transformer: Reinforcement Learning via Sequence Modeling

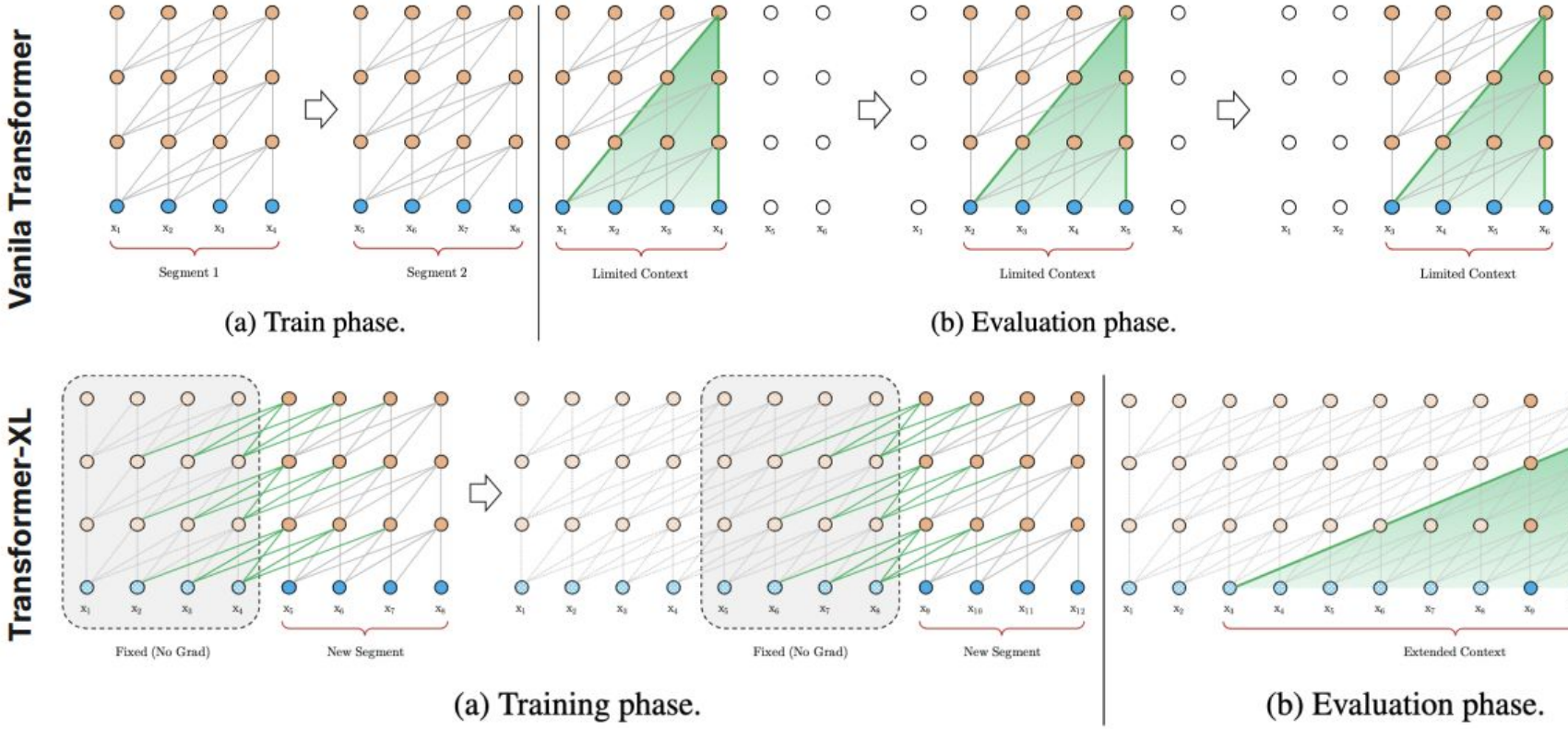
| Game | DT (Ours) | CQL | QR-DQN | REM | BC |
|---|---|---|---|---|---|
| Breakout | **267.5 ± 97.5** | 211.1 | 17.1 | 8.9 | 138.9 ± 61.7 |
| Qbert | 15.4 ± 11.4 | **104.2** | 0.0 | 0.0 | 17.3 ± 14.7 |
| Pong | 106.1 ± 8.1 | **111.9** | 18.0 | 0.5 | 85.2 ± 20.0 |
| Seaquest | **2.5 ± 0.4** | 1.7 | 0.4 | 0.7 | 2.1 ± 0.3 |

| Game | DT (Ours) | DT with no context ($K = 1$) |
|---|---|---|
| Breakout | **267.5 ± 97.5** | 73.9 ± 10 |
| Qbert | **15.1 ± 11.4** | 13.6 ± 11.3 |
| Pong | **106.1 ± 8.1** | 2.5 ± 0.2 |
| Seaquest | **2.5 ± 0.4** | 0.6 ± 0.1 |

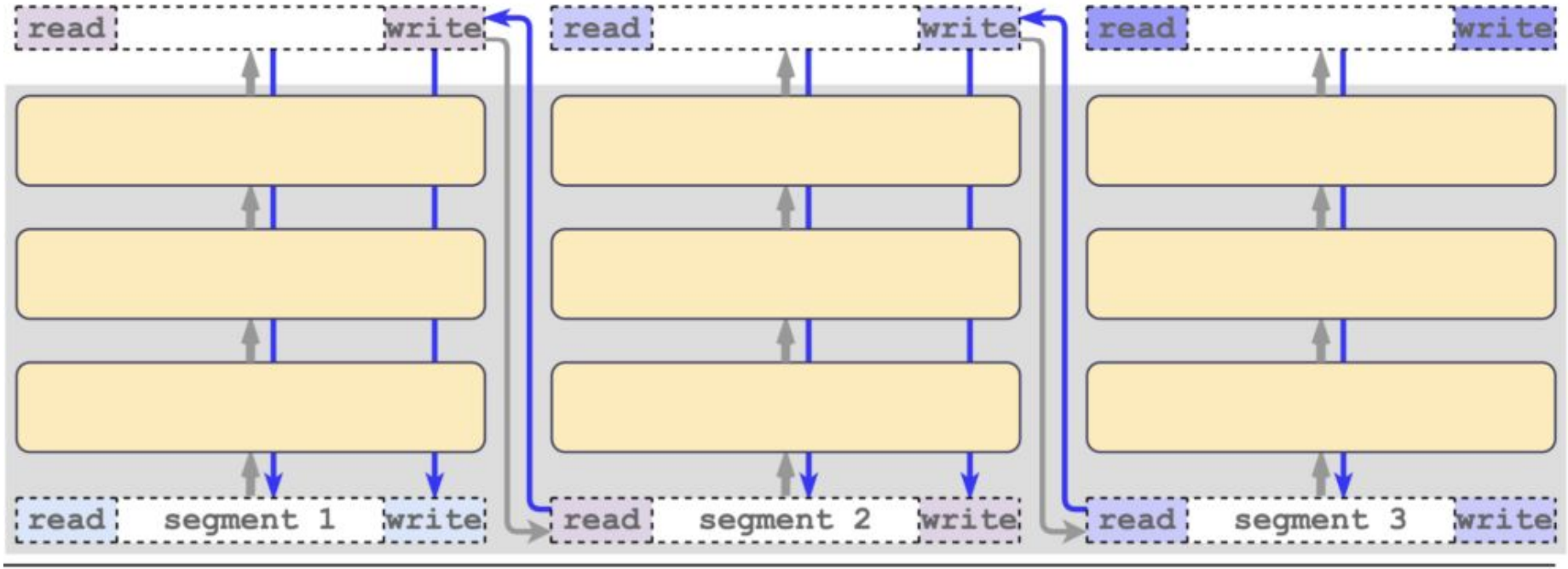
# Decision Transformer: Reinforcement Learning via Sequence Modeling



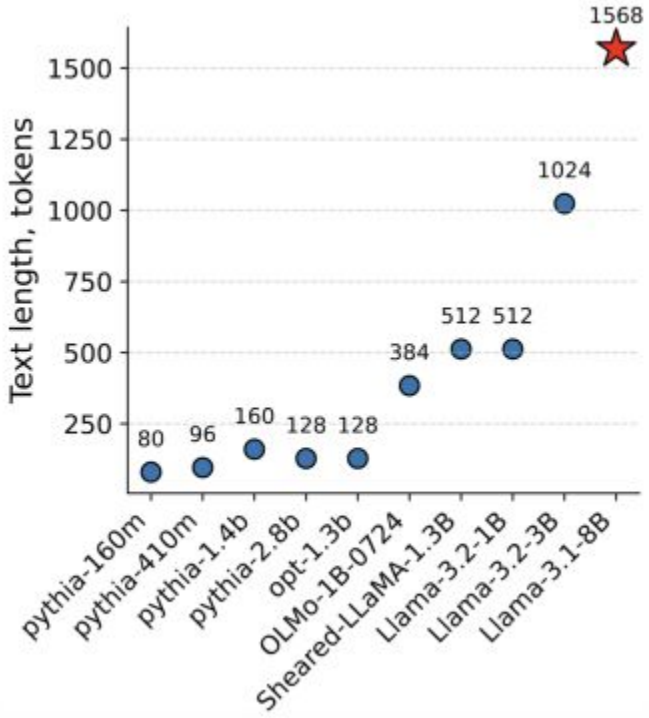
# Transformer-XL: Attentive Language Models Beyond a Fixed Length Context



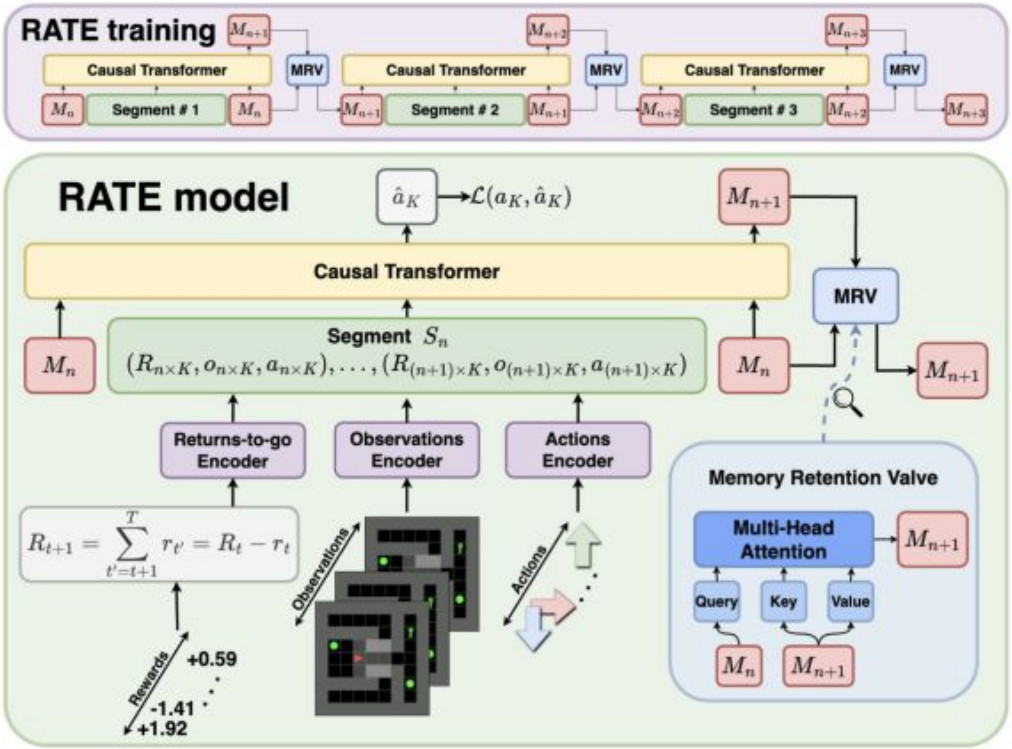
# Recurrent Memory Transformer



# Cramming 1568 Tokens into a Single Vector and Back Again: Exploring the Limits of Embedding Space Capacity



# Recurrent Action Transformer with Memory




**Algorithm 1** RATE

**Require:** $R \in \mathbb{R}^T, o \in \mathbb{R}^{d_o \times T}, a \in \mathbb{R}^T$

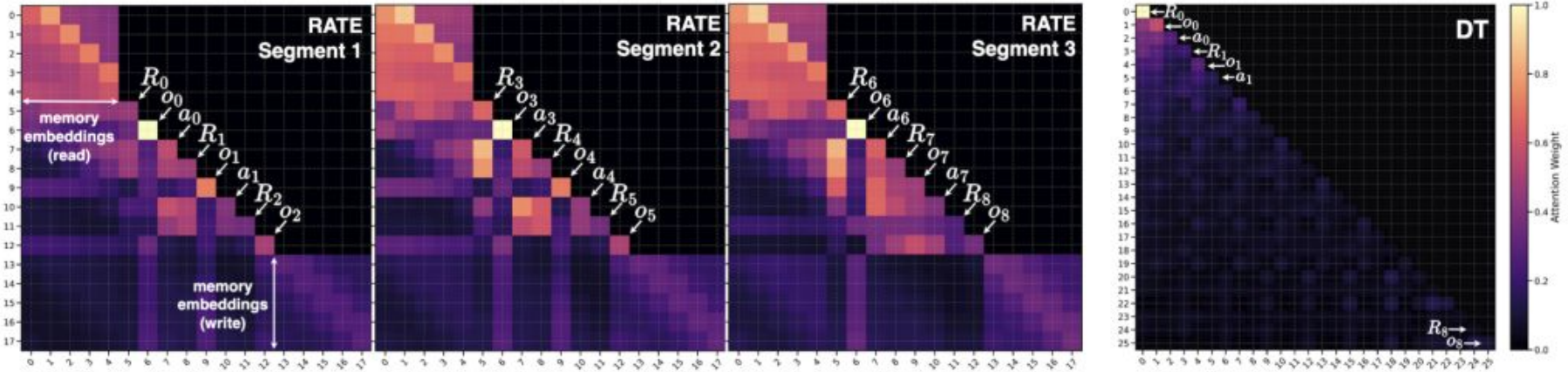
$\tilde{R} \leftarrow \texttt{Encoder}_R(R)$
$\tilde{o} \leftarrow \texttt{Encoder}_o(o)$
$\tilde{a} \leftarrow \texttt{Encoder}_a(a)$
$\tau_{0:T-1} \leftarrow \{(\tilde{R}_t, \tilde{o}_t, \tilde{a}_t)\}_{t=0}^{T-1}$
$M_n \leftarrow M_0 \sim \mathcal{N}(0, 1)$
**for** $n$ in $[0, T//K - 1]$ **do**
  $S_n \leftarrow \tau_{nK:(n+1)K}$
  $\tilde{S}_n \leftarrow \texttt{concat}(M_n, S_n, M_n)$
  $\hat{a}_n, M_{n+1} \leftarrow \texttt{Transformer}(\tilde{S}_n)$
  $M_{n+1} \leftarrow \texttt{MRV}(M_n, M_{n+1})$
  $\hat{a}_n \rightarrow \mathcal{L}(a_n, \hat{a}_n),\ M_{n+1}$
**end for**

**Algorithm 2** Memory Retention Valve

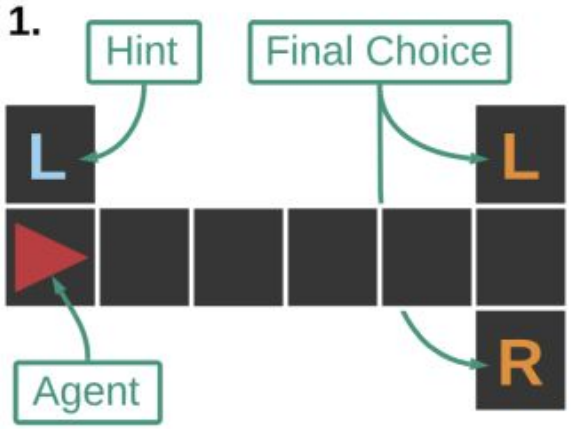
**Require:** $M_n, M_{n+1} \in \mathbb{R}^{m \times d}$

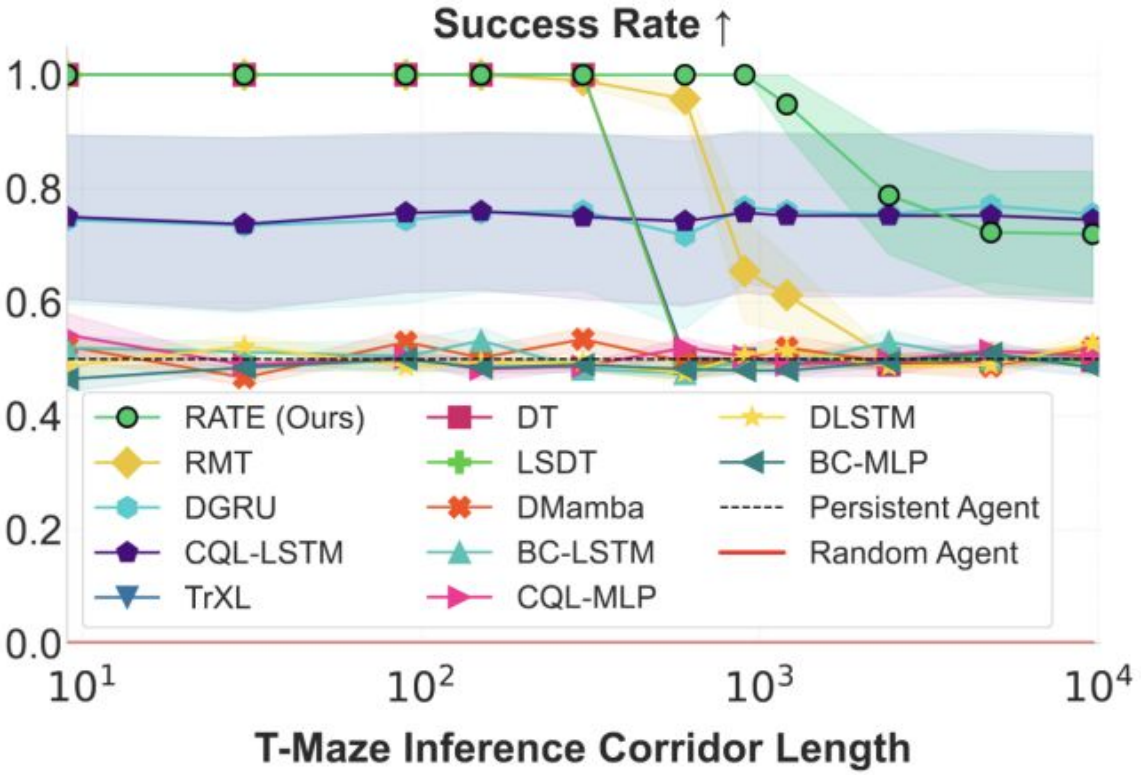
$\mathbf{Q}_h \leftarrow M_n \mathbf{W}_Q^{h\,T}$
$\mathbf{K}_h \leftarrow M_{n+1} \mathbf{W}_K^{h\,T}$
$\mathbf{V}_h \leftarrow M_{n+1} \mathbf{W}_V^{h\,T}$
$M_{n+1}^h \leftarrow \texttt{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^T}{\sqrt{d}}\right) \mathbf{V}_h$
$M_{n+1} \leftarrow \texttt{concat}(M_{n+1}^0, \ldots, M_{n+1}^h)$
$M_{n+1} \leftarrow M_{n+1} \mathbf{W}_M^T$
**Output:** $M_{n+1}$

# Recurrent Action Transformer with Memory

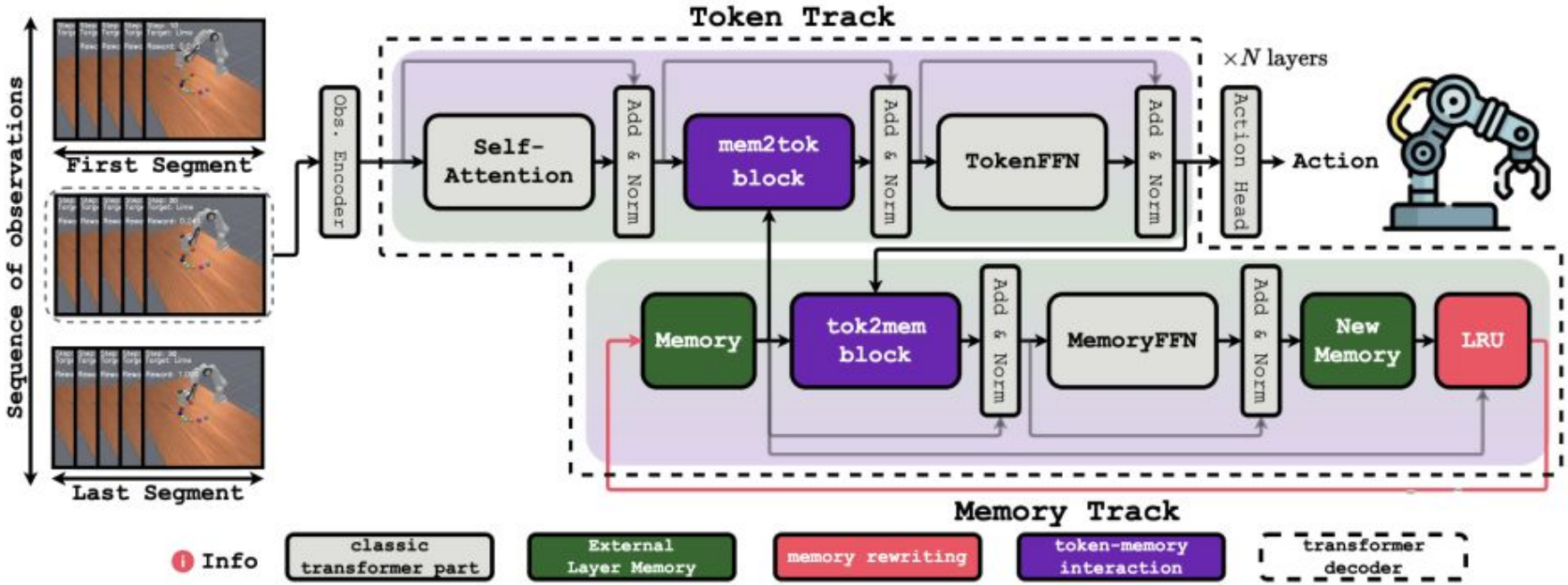


# Recurrent Action Transformer with Memory

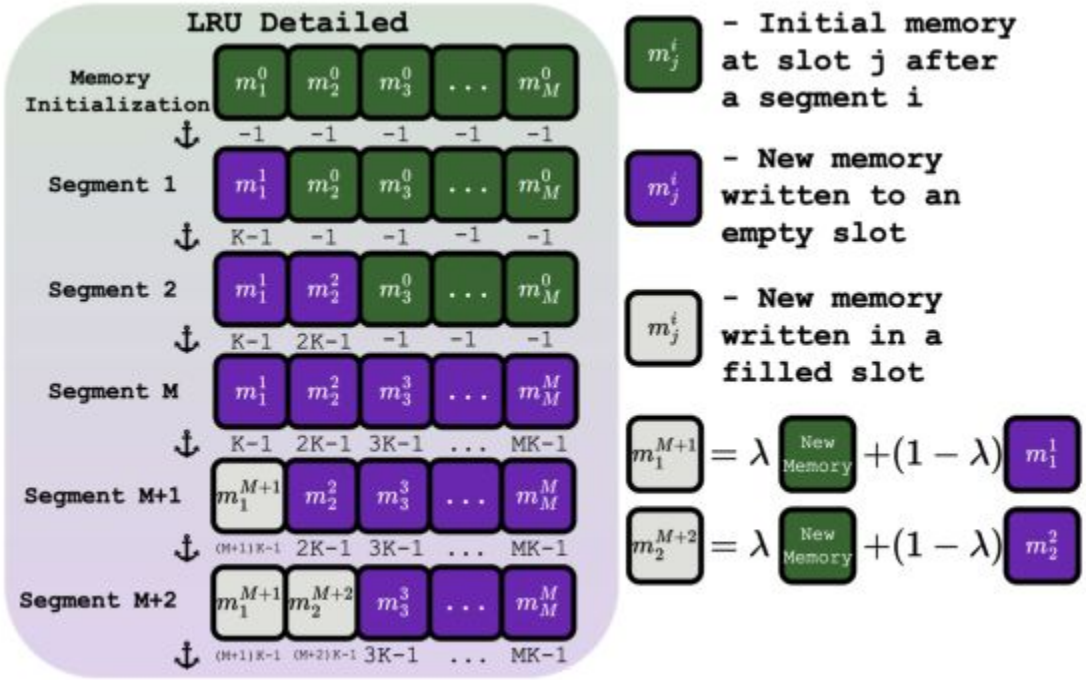


# ELMUR: External Layer Memory with Update/Rewrite for Long-Horizon RL Problems



# ELMUR: External Layer Memory with Update/Rewrite for Long-Horizon RL Problems



# ELMUR: External Layer Memory with Update/Rewrite for Long-Horizon RL Problems

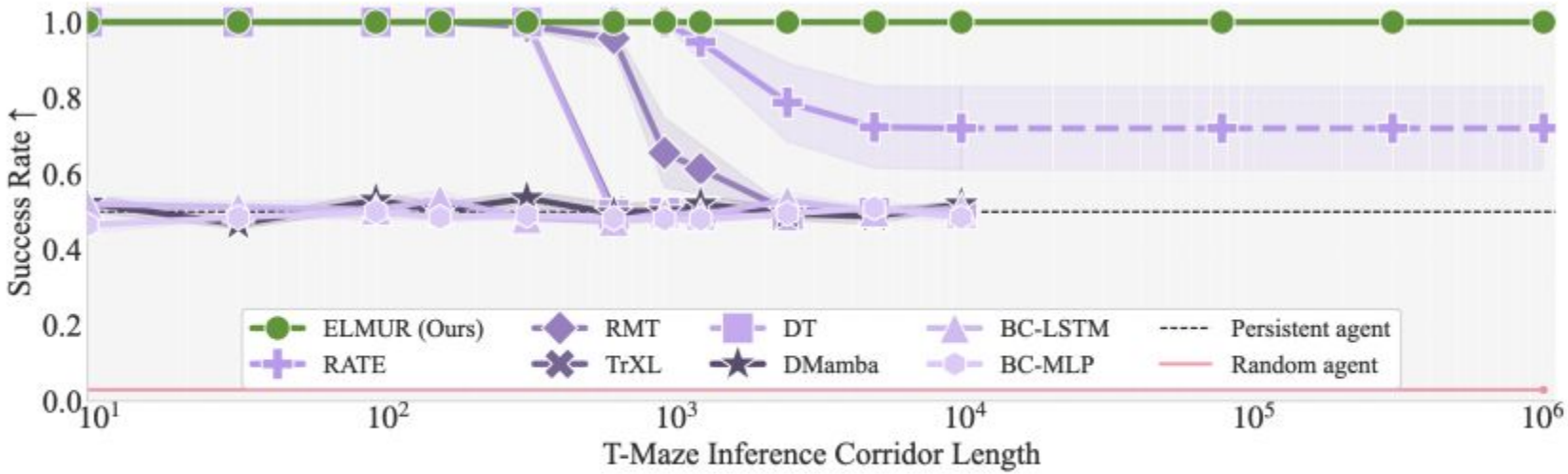
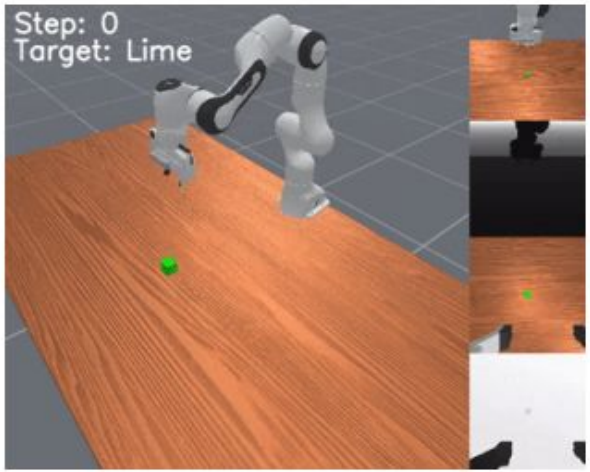
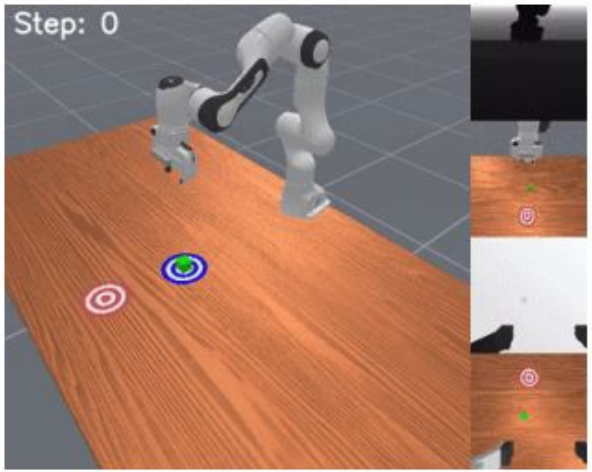



Figure 3: Success rate on the T-Maze task as a function of inference corridor length. ELMUR achieves a **100% success rate** up to corridor lengths of **one million steps**. In this figure, the context length is $L = 10$ with $S = 3$ segments; thus **ELMUR carries information across horizons 100,000 times longer than its context window**.

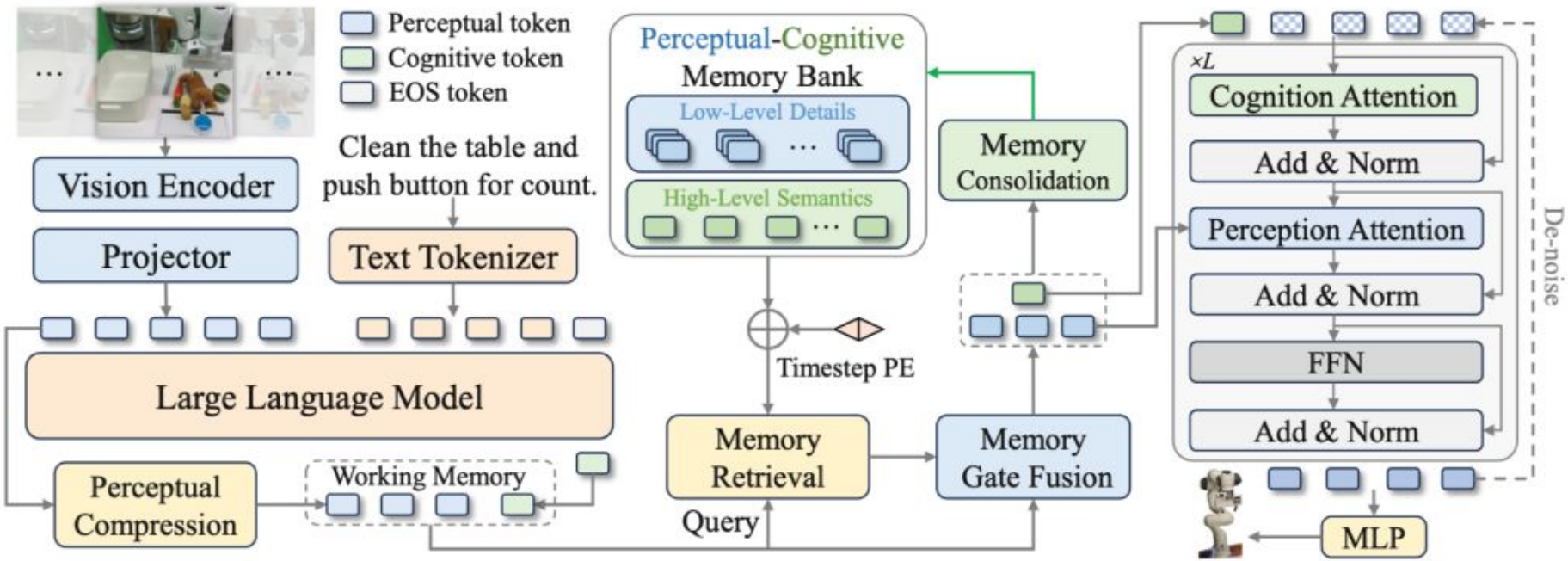
# ELMUR: External Layer Memory with Update/Rewrite for Long-Horizon RL Problems

| Task | RATE | DT | BC-MLP | CQL-MLP | DP | ELMUR (ours) |
|---|---|---|---|---|---|---|
| RememberColor3-v0 | 0.65±0.04 | 0.01±0.01 | 0.27±0.03 | 0.29±0.01 | 0.32±0.01 | **0.89±0.07** |
| RememberColor5-v0 | 0.13±0.03 | 0.07±0.05 | 0.12±0.01 | 0.15±0.02 | 0.10±0.02 | **0.19±0.03** |
| RememberColor9-v0 | 0.09±0.02 | 0.01±0.01 | 0.12±0.02 | 0.15±0.01 | 0.17±0.01 | **0.23±0.02** |
| TakeItBack-v0 | 0.42±0.24 | 0.08±0.04 | 0.33±0.10 | 0.04±0.01 | 0.05±0.02 | **0.78±0.03** |

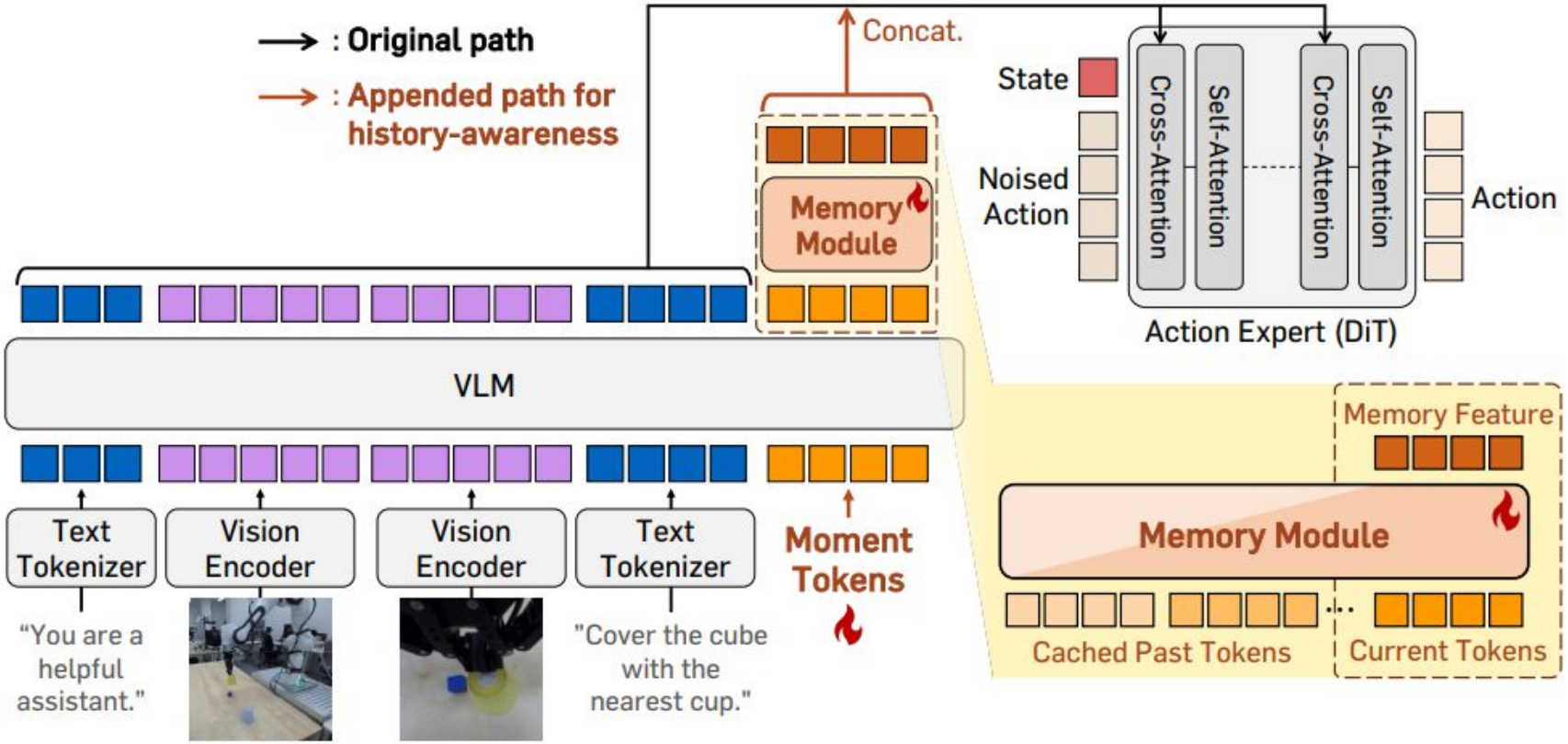


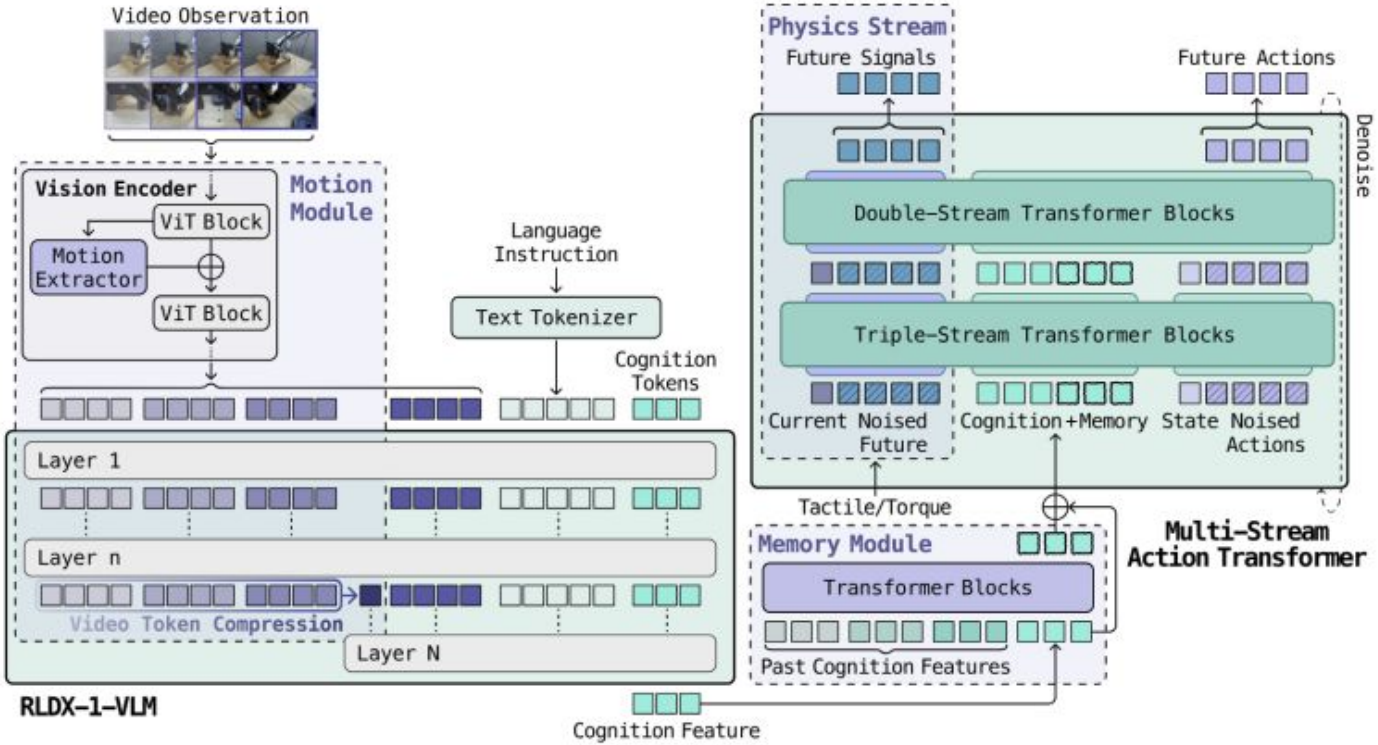
# MemoryVLA


Perceptual token
Cognitive token
EOS token
Clean the table and push button for count.
Vision Encoder
Projector
Text Tokenizer
Large Language Model
Perceptual Compression
Working Memory
Perceptual-Cognitive Memory Bank
Low-Level Details
High-Level Semantics
Memory Consolidation
Timestep PE
Memory Retrieval
Memory Gate Fusion
Query
×L
Cognition Attention
Add & Norm
Perception Attention
Add & Norm
FFN
Add & Norm
De-noise
MLP

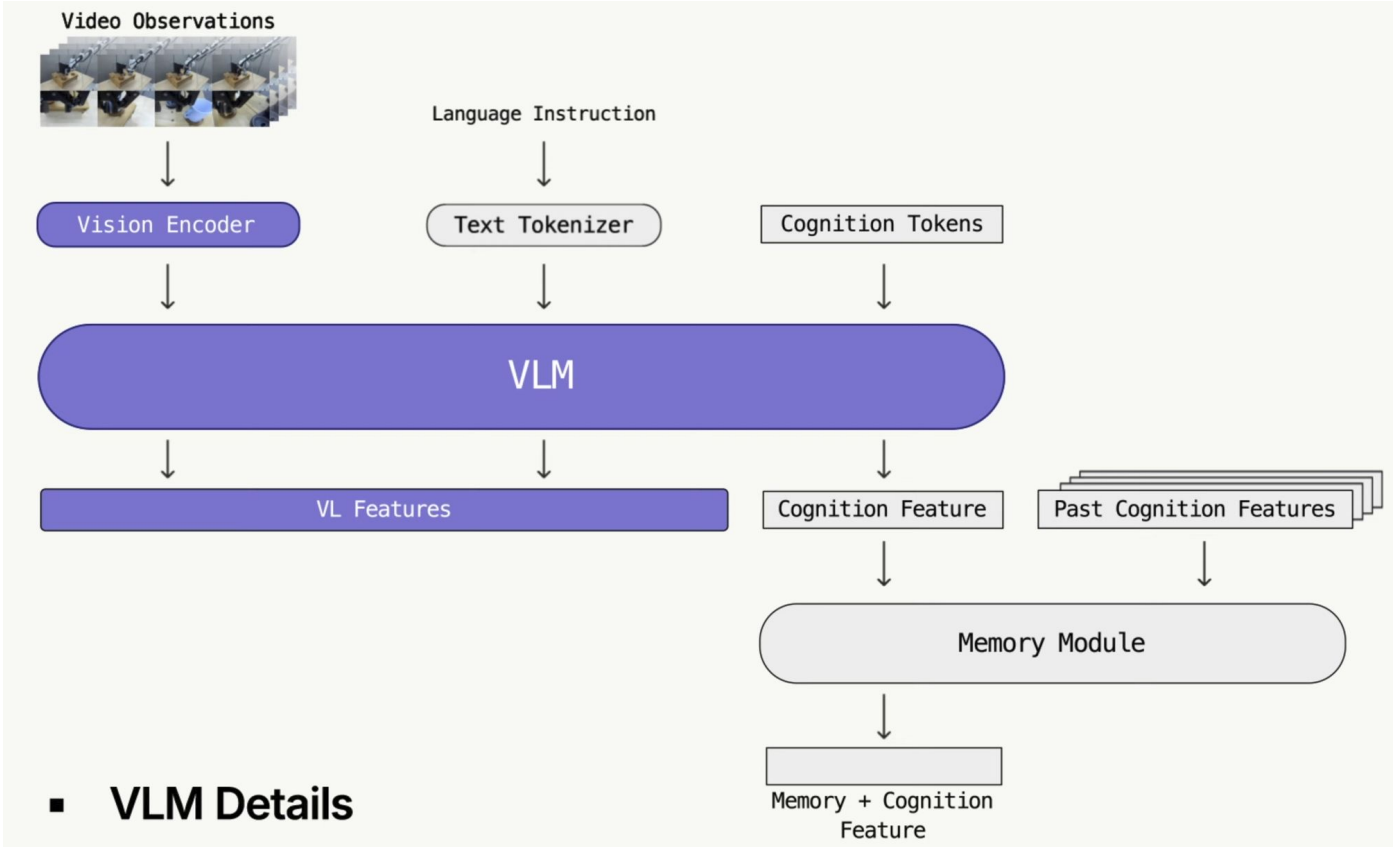
# HAMLET: SWITCH YOUR VISION-LANGUAGE ACTION MODEL INTO A HISTORY-AWARE POLICY



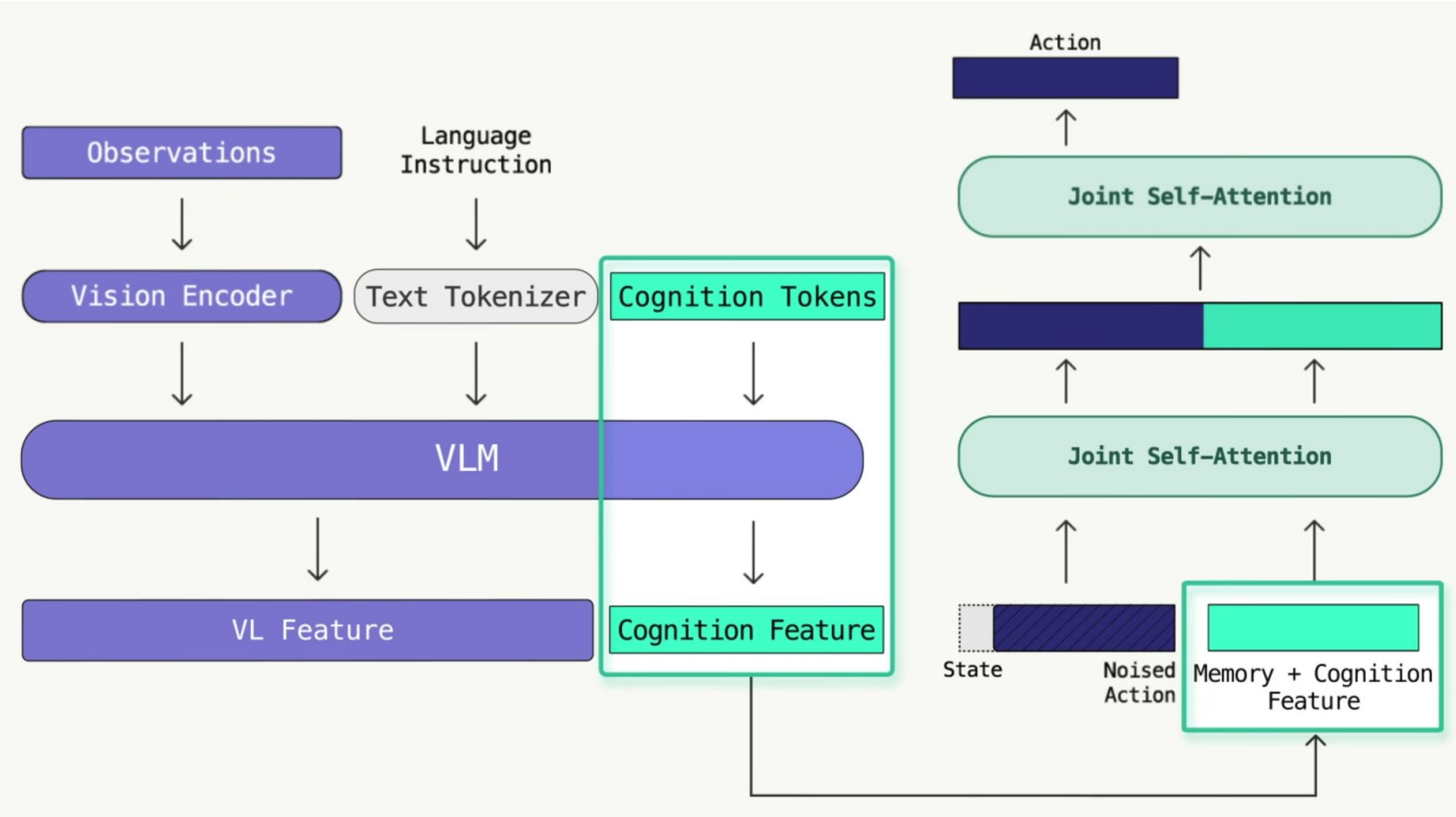
# RLDX-1


Video Observation
Vision Encoder
ViT Block
Motion Extractor
ViT Block
Motion Module
Language Instruction
Text Tokenizer
Cognition Tokens
Layer 1
Layer n
Video Token Compression
Layer N
RLDX-1-VLM
Physics Stream
Future Signals
Future Actions
Denoise
Double-Stream Transformer Blocks
Triple-Stream Transformer Blocks
Current Noised Future
Cognition+Memory
State Noised Actions
Tactile/Torque
Multi-Stream Action Transformer
Memory Module
Transformer Blocks
Past Cognition Features
Cognition Feature

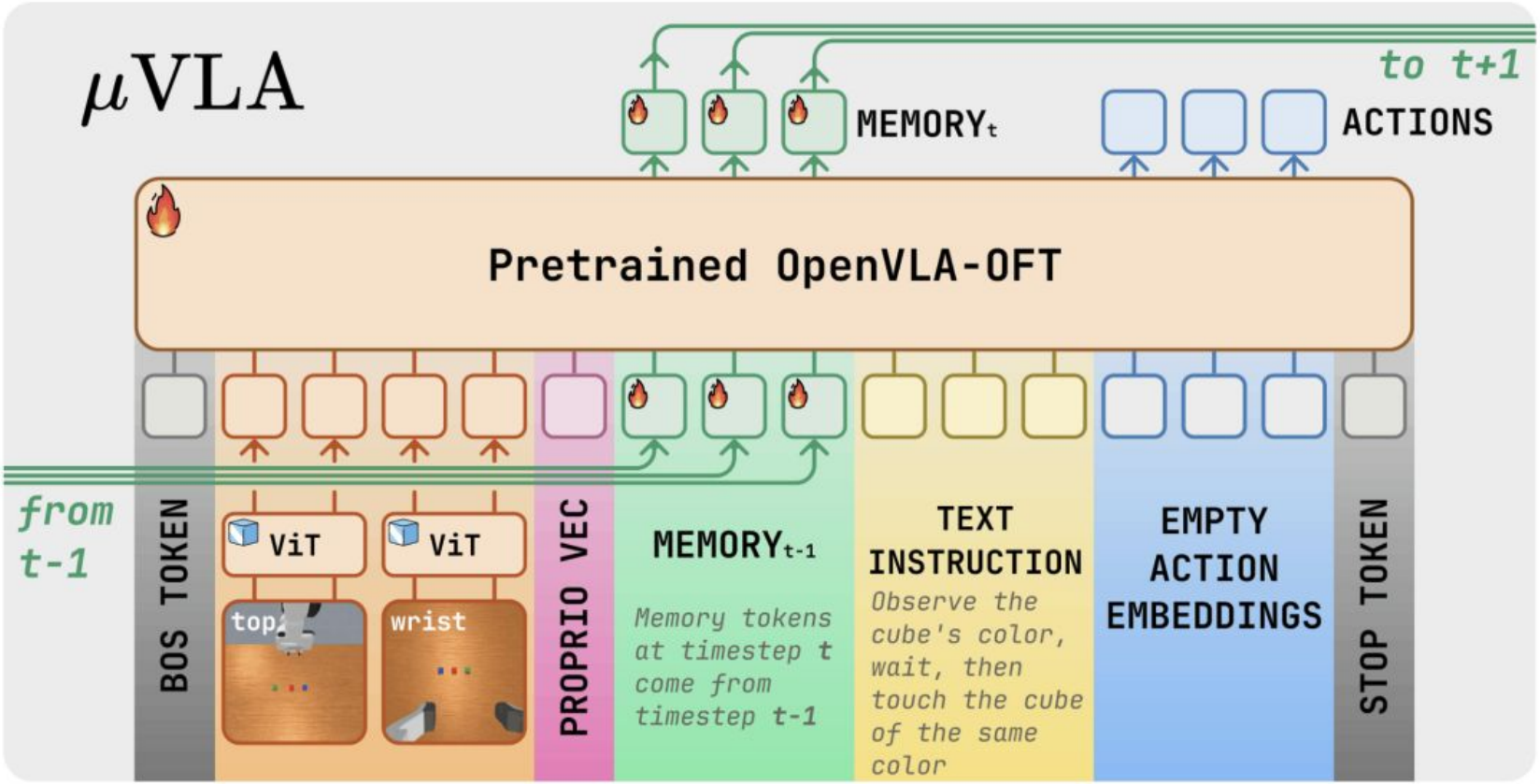
# RLDX-1


Video Observations
Language Instruction
Vision Encoder
Text Tokenizer
Cognition Tokens
VLM
VL Features
Cognition Feature
Past Cognition Features
Memory Module
Memory + Cognition Feature
▪ VLM Details

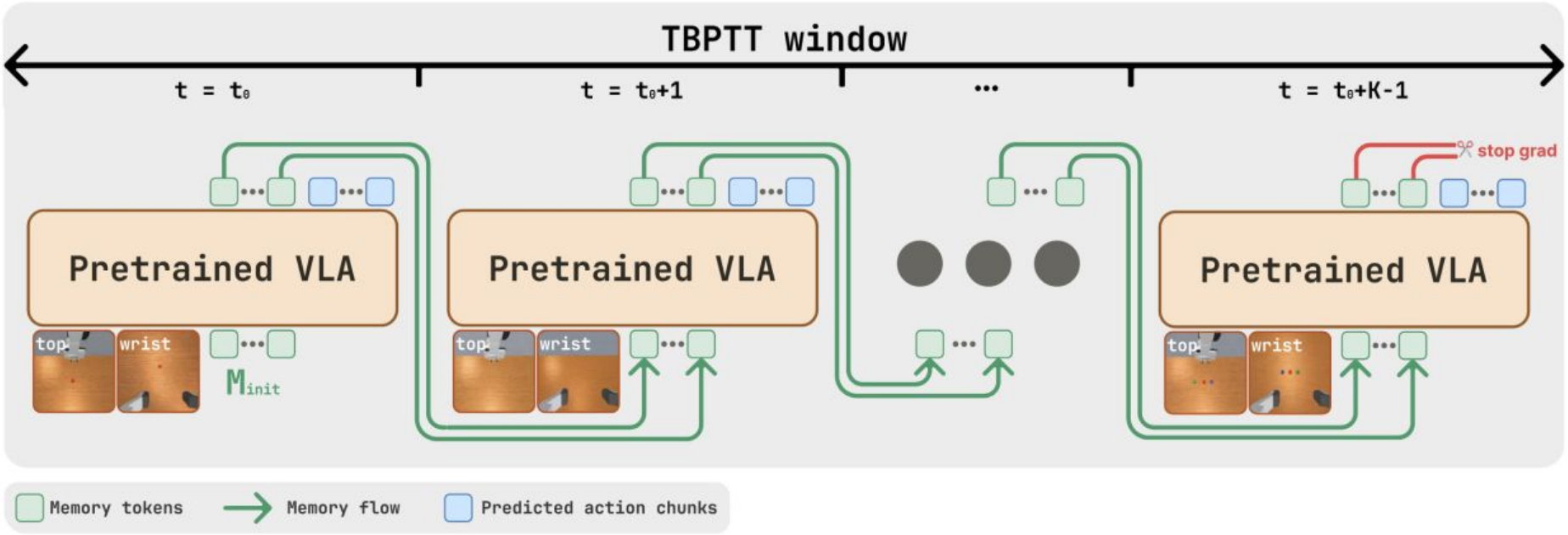
# RLDX-1


Observations
Language Instruction
Vision Encoder
Text Tokenizer
Cognition Tokens
VLM
VL Feature
Cognition Feature
Action
Joint Self-Attention
Joint Self-Attention
State
Noised Action
Memory + Cognition Feature

# μVLA


μVLA
to t+1
MEMORY$_t$
ACTIONS
Pretrained OpenVLA-OFT
from t-1
BOS TOKEN
ViT
ViT
top
wrist
PROPRIO VEC
MEMORY$_{t-1}$
Memory tokens at timestep t come from timestep t-1
TEXT INSTRUCTION
Observe the cube's color, wait, then touch the cube of the same color
EMPTY ACTION EMBEDDINGS
STOP TOKEN

# μVLA


TBPTT window
t = t₀
t = t₀+1
...
t = t₀+K-1
stop grad
Pretrained VLA
Pretrained VLA
Pretrained VLA
top
wrist
top
wrist
top
wrist
M init
Memory tokens
Memory flow
Predicted action chunks

# MEMER: SCALING UP MEMORY FOR ROBOT CONTROL VIA EXPERIENCE RETRIEVAL

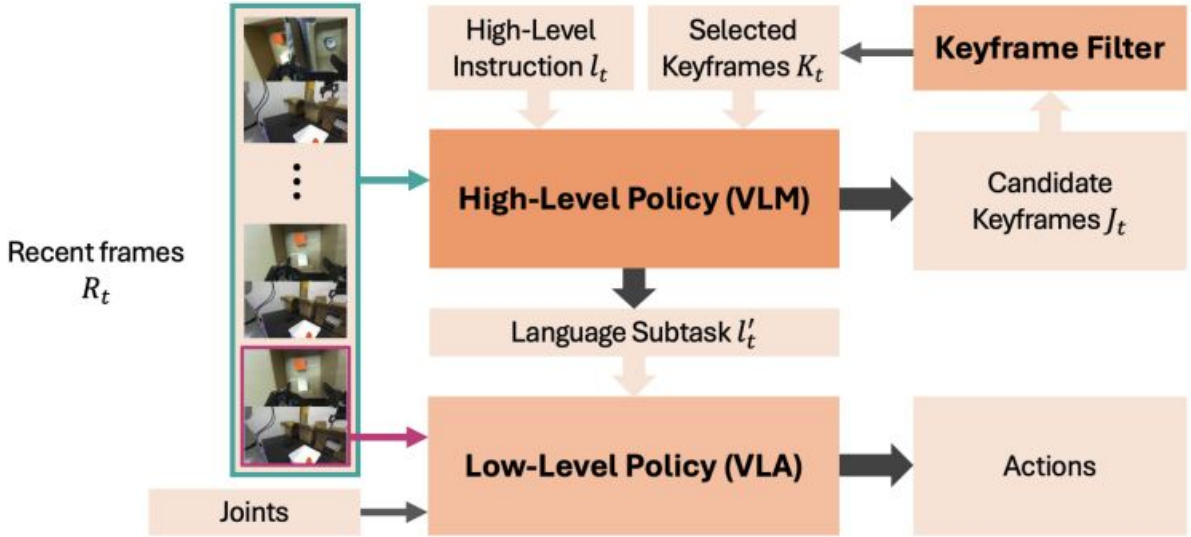
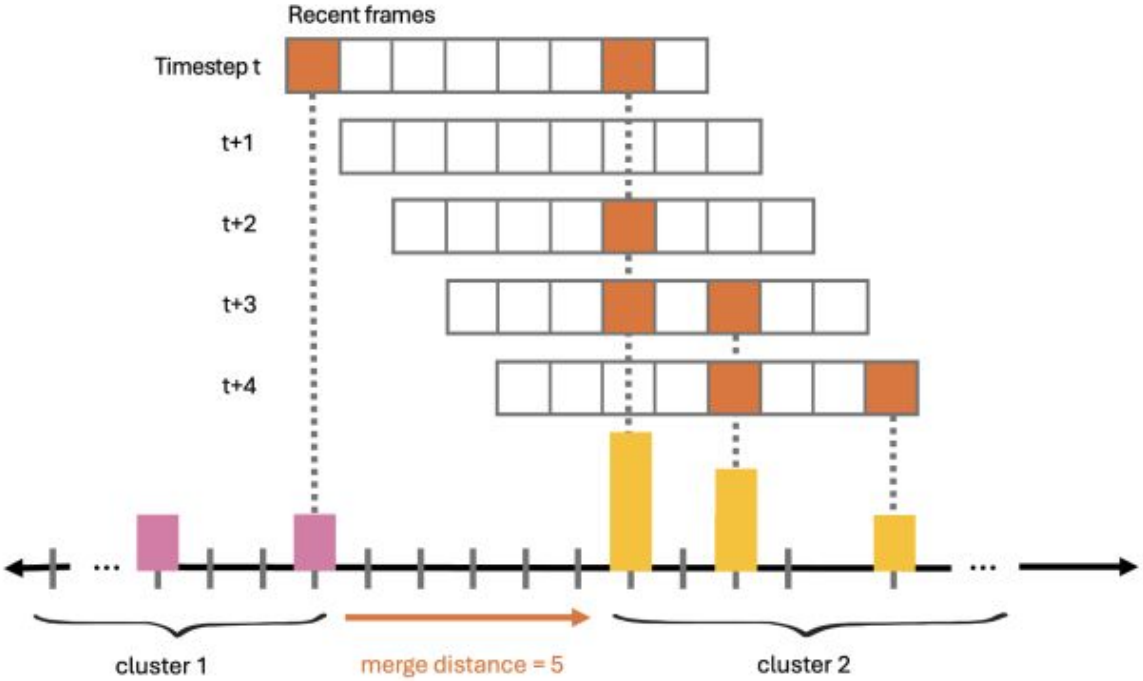
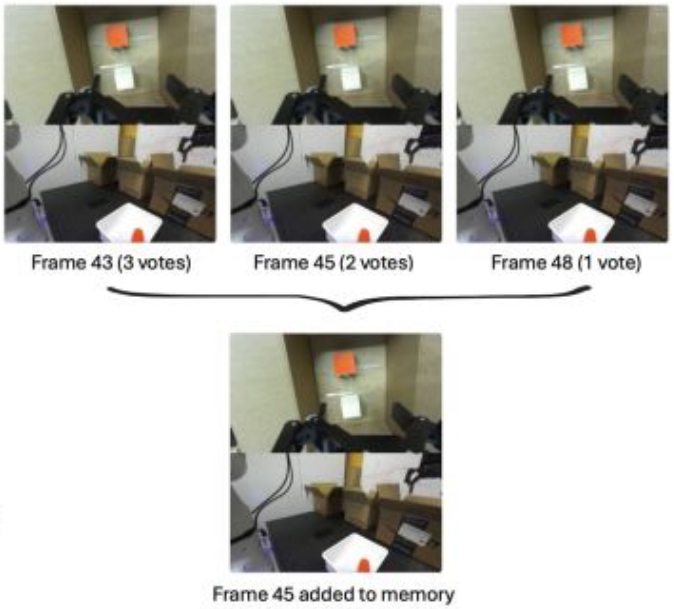


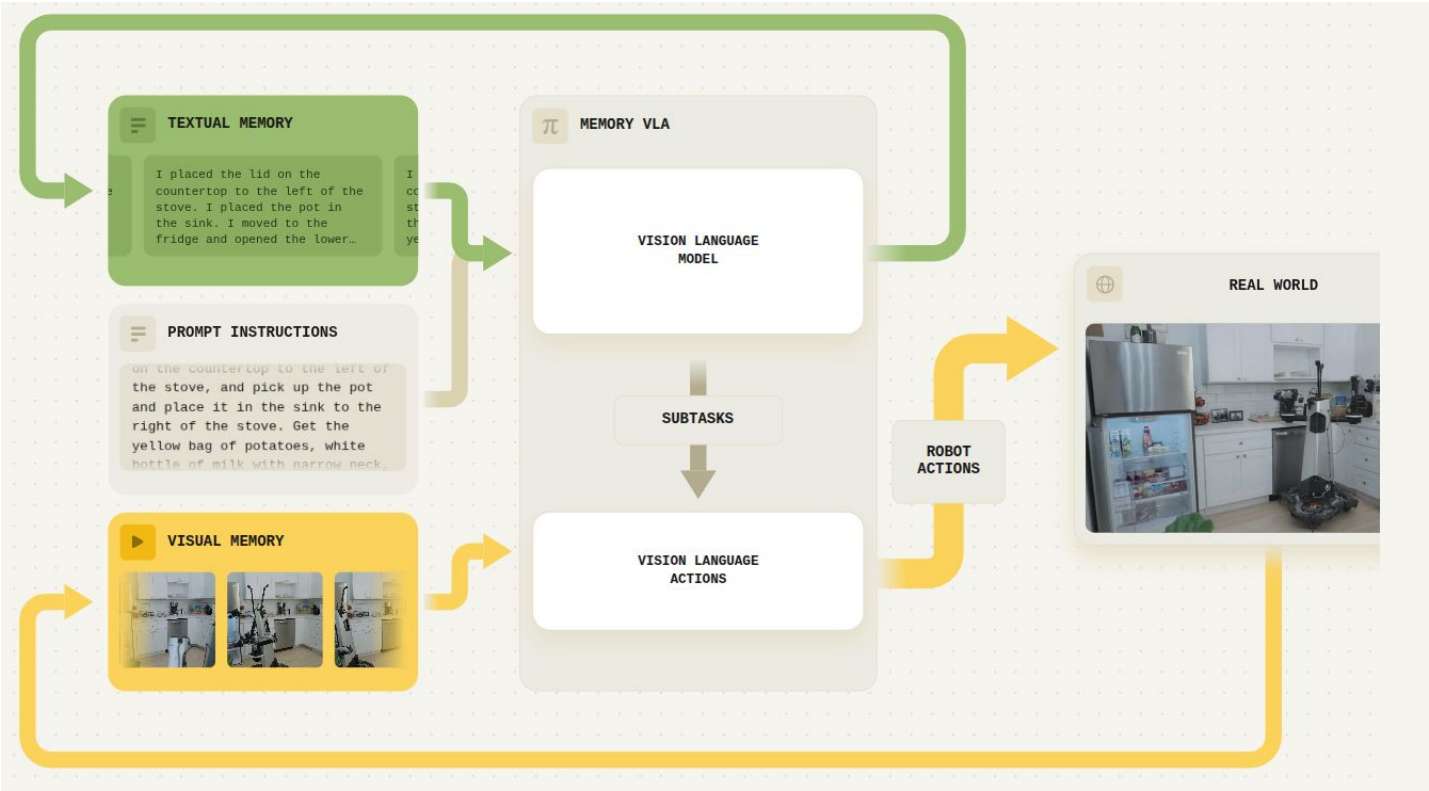

Figure 2: **Architecture of MemER.** The high-level policy processes task instructions, selected keyframes (if any), and recent images from base and wrist-mounted cameras to generate low-level language subtasks and candidate keyframes (if any). The low-level policy uses the subtask, current image, and robot joint states to produce actions. The candidate keyframe(s) are processed by the keyframe filter to obtain the selected keyframes for input during the next step of inference.

# MEMER: SCALING UP MEMORY FOR ROBOT CONTROL VIA EXPERIENCE RETRIEVAL

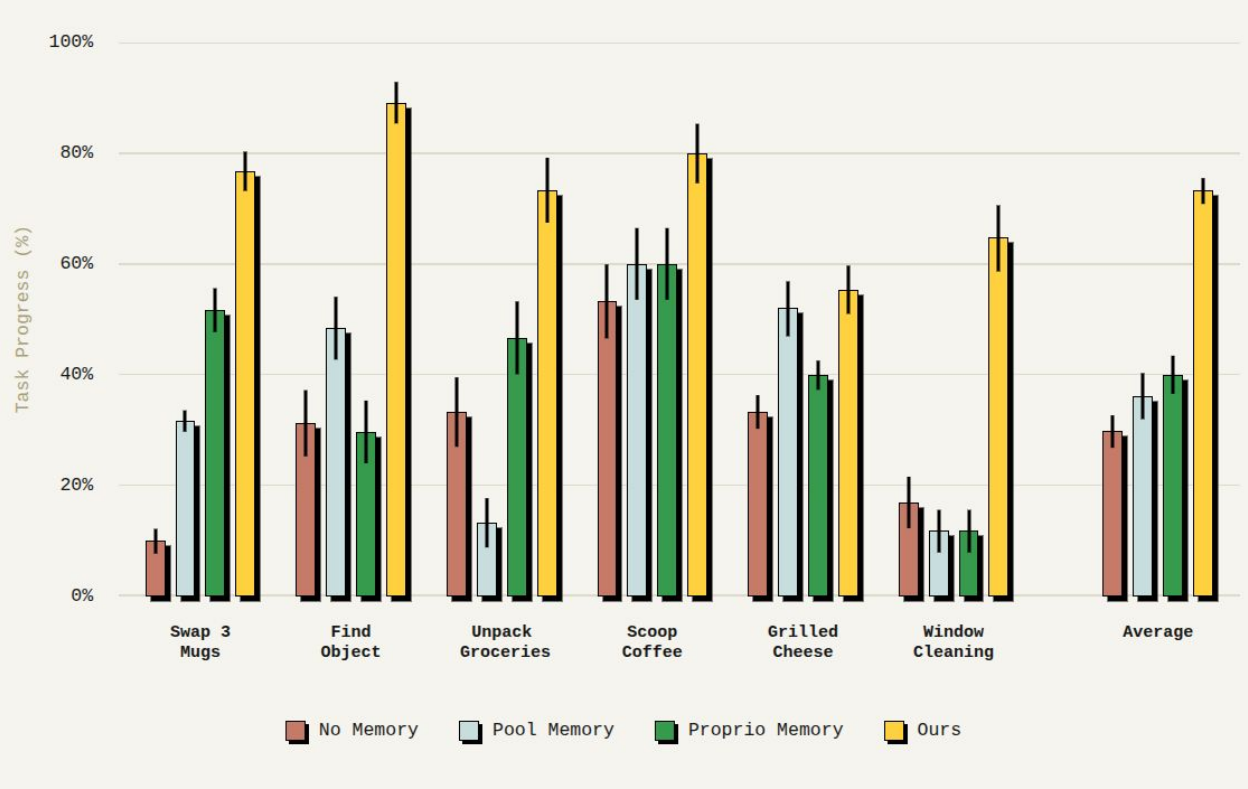


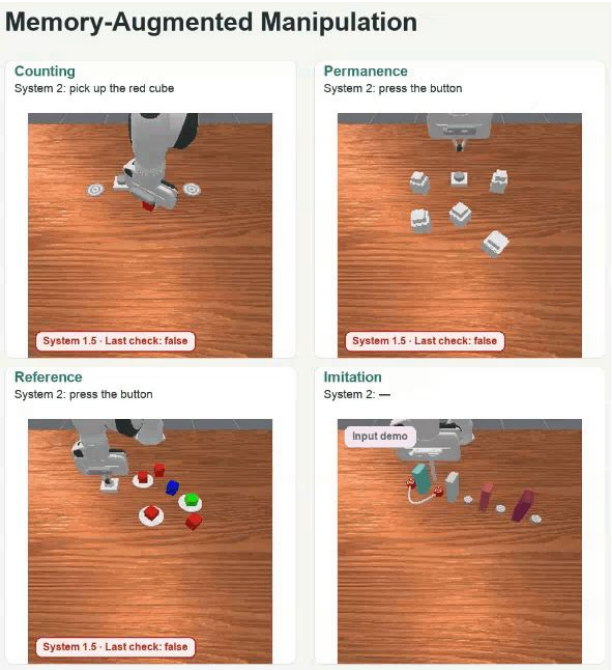
# MEM: Multi-Scale Embodied Memory for Vision Language Action Models

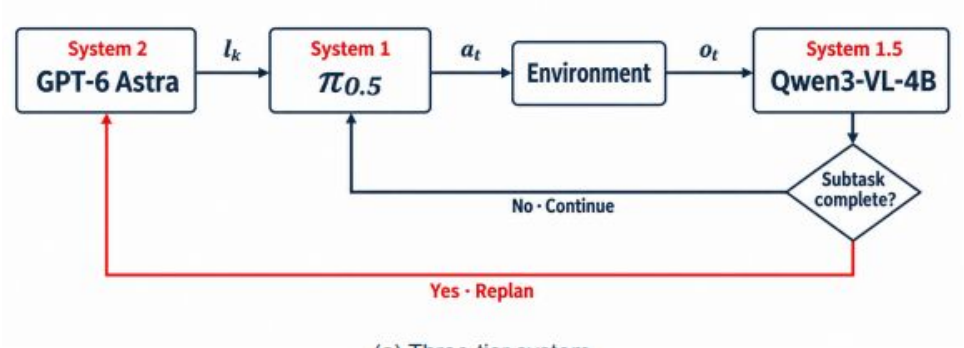


# MEM: Multi-Scale Embodied Memory for Vision Language Action Models

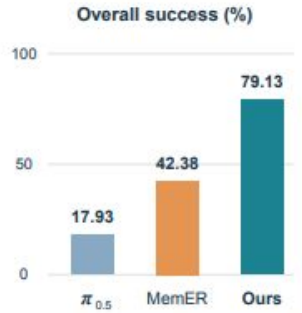


# Can Astra Solve RoboMME without Breaking the Bank?




(a) Three-tier system




(b) Results on RoboMME

| Method | Counting | Permanence | Reference | Imitation | Overall |
|---|---|---|---|---|---|
| $\pi_{0.5}$ [7] | 28.78 | 17.00 | 17.16 | 8.78 | 17.93 |
| FrameSamp+Modul [1] | 65.22 | 25.11 | 36.33 | 51.39 | 44.51 |
| SAM2Act+ [3] | 35.33 | 26.00 | 16.83 | 7.33 | 21.37 |
| MemER [4] | 48.83 | 53.17 | 38.00 | 29.50 | 42.38 |
| GroundSG+QwenVL [1] | 38.00 | 39.34 | 31.56 | 21.89 | 32.70 |
| GroundSG+Gemini [1] | 12.75 | 15.75 | 11.25 | 6.50 | 11.56 |
| **Three-tier Astra (ours)** | **69.50** | **94.00** | **92.00** | **61.00** | **79.13** |
| GroundSG+Oracle [1] | 83.86 | 93.31 | 95.17 | 63.98 | 84.08 |