NEURAL INFORMATION PROCESSING SYSTEMS

Arizona State University

NEURIPS 2026 · MAIN TRACK

Self-Supervised Keyframe Discovery for Horizon-Invariant Behavior Cloning

Prabin Kumar Rath · Omkar Patil · Nakul Gopalan

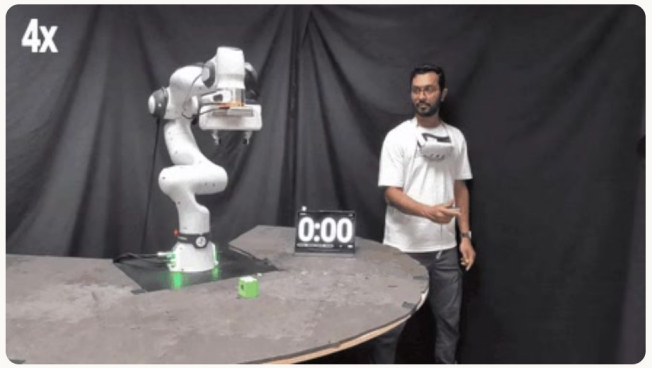
Arizona State University

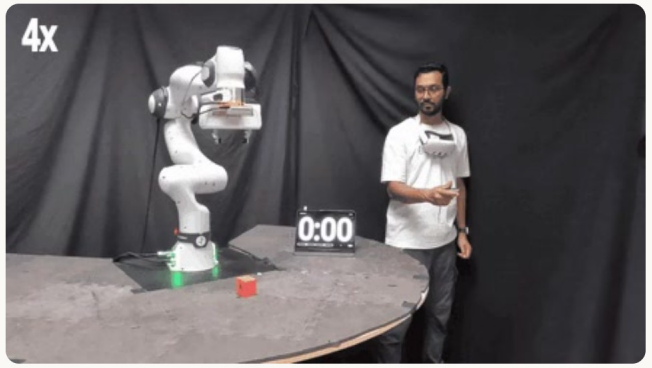
keyframe-mnemonics.github.io

SCAN-ME

THE PROBLEM

What tasks are still hard to learn from demonstrations?





TRAIN Δt = 3–5 s

Task: Remember Color 2

1. The robot is shown a **red** or **green** cube.
2. The cue is removed, and after a delay Δt two cubes appear 30 cm apart.
3. The robot must reach for the right color when the options are shown *even at longer delays*.

> Can a policy learn this behavior and generalize to **longer timescales**?

WHY EXISTING POLICIES FAIL

Markovian policies fail due to lack of memory

45% **Diffusion Policy**
success rate



Fail: chooses wrong color



Fail: moves early

Partial observability is still difficult

Real world is **non-Markovian**

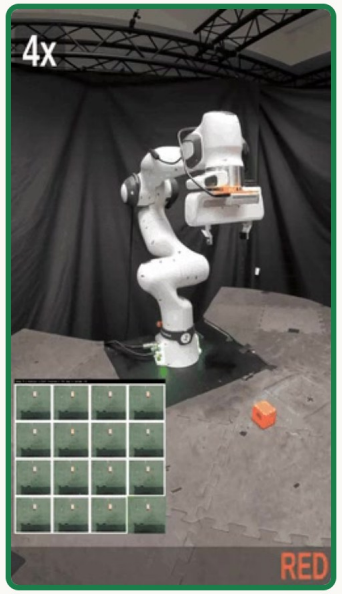
Policies have a few seconds of working memory which is **not sufficient**

Chi et al., Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, RSS, 2023

WHY EXISTING POLICIES FAIL

Transformer and recurrent policies fail at longer horizons

TRANSFORMER POLICIES HAVE LIMITED CONTEXT

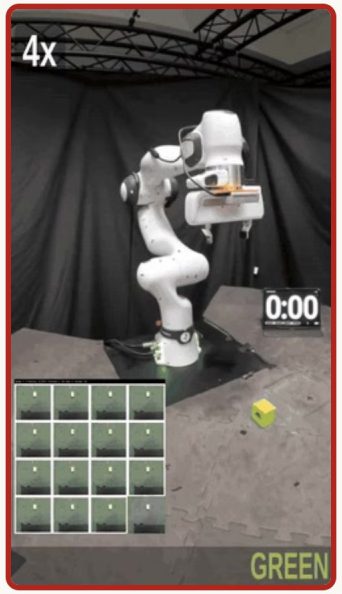


TRAIN

Δt = 3–5 s

PTP 80% SR

More context helps



OOD

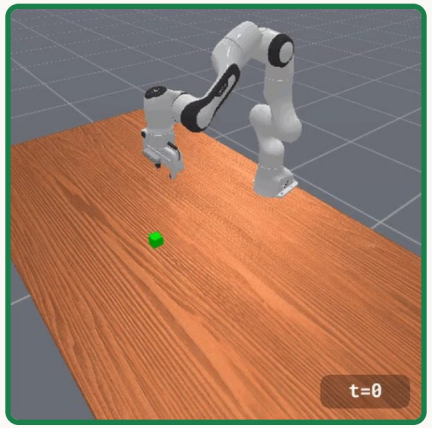
Δt = 100–120 s

PTP 0% SR

Fails at OOD test horizon

> Memory lost after **information drops** from context buffer

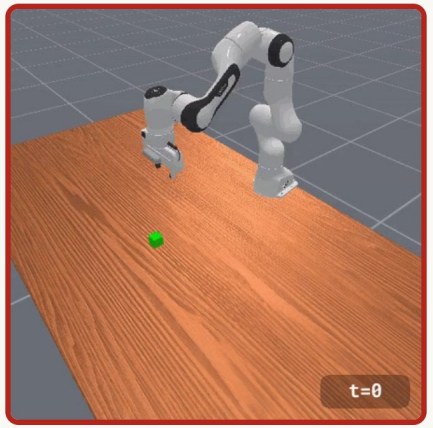
RECURRENT POLICIES SUFFER FROM STATE COLLAPSE



TRAIN Δt = 15 steps

ELMUR

External memory



OOD Δt = 95 steps

ELMUR 0% SR

Fails at OOD test horizon

> Memory lost after **repeated state updates**

Trone, Tang and Liu et al., Learning Long-Context Diffusion Policies via Past-Token Prediction, CoRL, 2025
Cherepanov et al., ELMUR: External Layer Memory with Update/Rewrite for Long-Horizon RL Problems, ICLR, 2026

KEYFRAMES

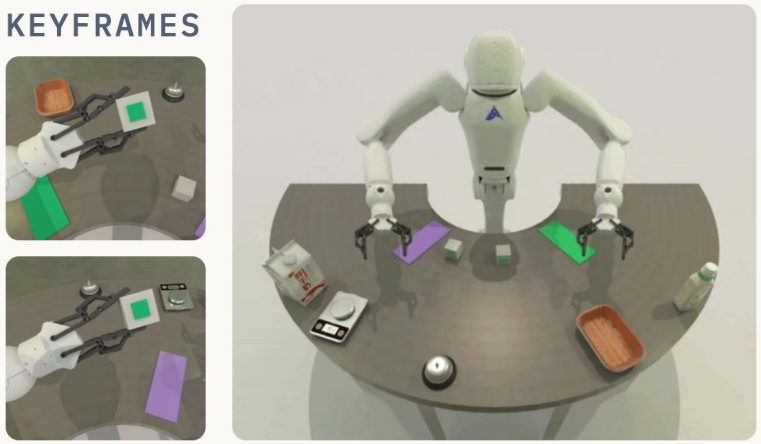
Keyframes help, but have limitations

Keyframes are powerful but

needs annotation → difficult to scale
Sridhar et al., MemER · Mark et al., BPP

heuristic keyframes ≠ best for memory
Kou et al., KISA '24

Which keyframes are worth remembering?



> How can we train **keyframe policies** from pure demonstrations that generalize to longer timescales without performance degradation?

METHOD · ASSUMPTION

Task structure assumption

Let’s assume the expert’s behavior depends on a few keyframes

Keyframes $\mathcal{X}_{\text{key}}$

observations that affect behavior at decision steps

Background frames $\mathcal{X}_{\text{bg}}$

all observations other than keyframes

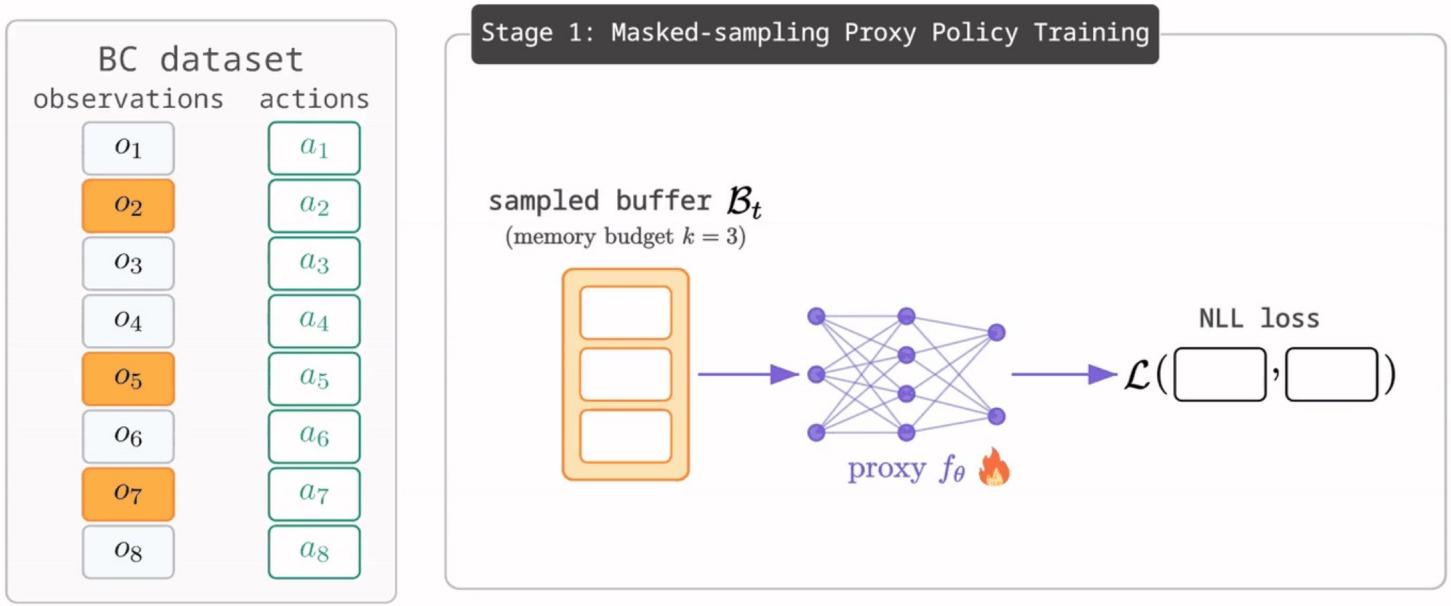
MUTUAL INFORMATION $$I(a_t;\ \mathcal{X}_{\text{key},\,t}) > I(a_t;\ \mathcal{X}_{\text{bg},\,t})$$ where a_t is expert action



METHOD · STEP 1 OF 3

Self-supervised keyframe discovery

1 We first train a **proxy policy** that learns low complexity invariant mappings between past observations and future actions.



⬆ **Higher** mutual information

IMPLIES

⬇ **Low** mapping complexity

Simplicity bias: proxy learns the mapping conditioned on

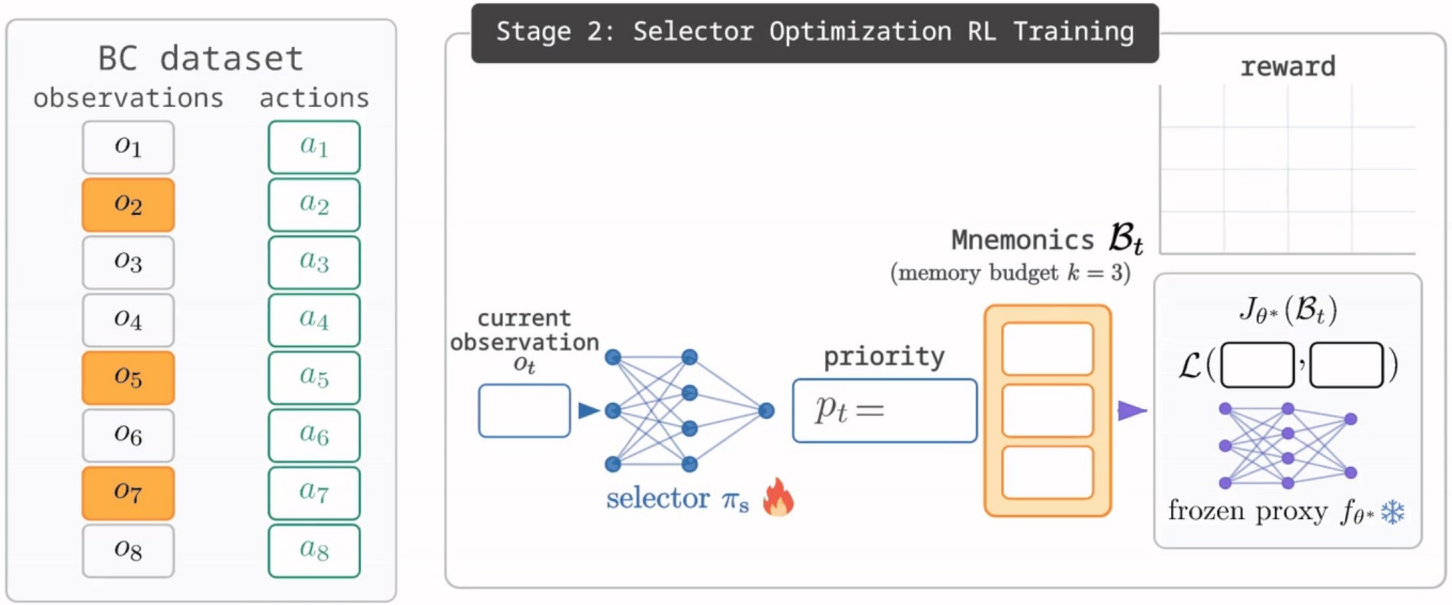
$\mathcal{X}_{\text{key}}$

Grünwald et al., Algorithmic information theory. Handbook of the Philosophy of Information, pages 281–320, 2008

METHOD · STEP 2 OF 3

Identifying decision-relevant keyframes

2 We then use the action error of the proxy policy as a reward to train a **selector policy** that scores keyframes higher than background frames.



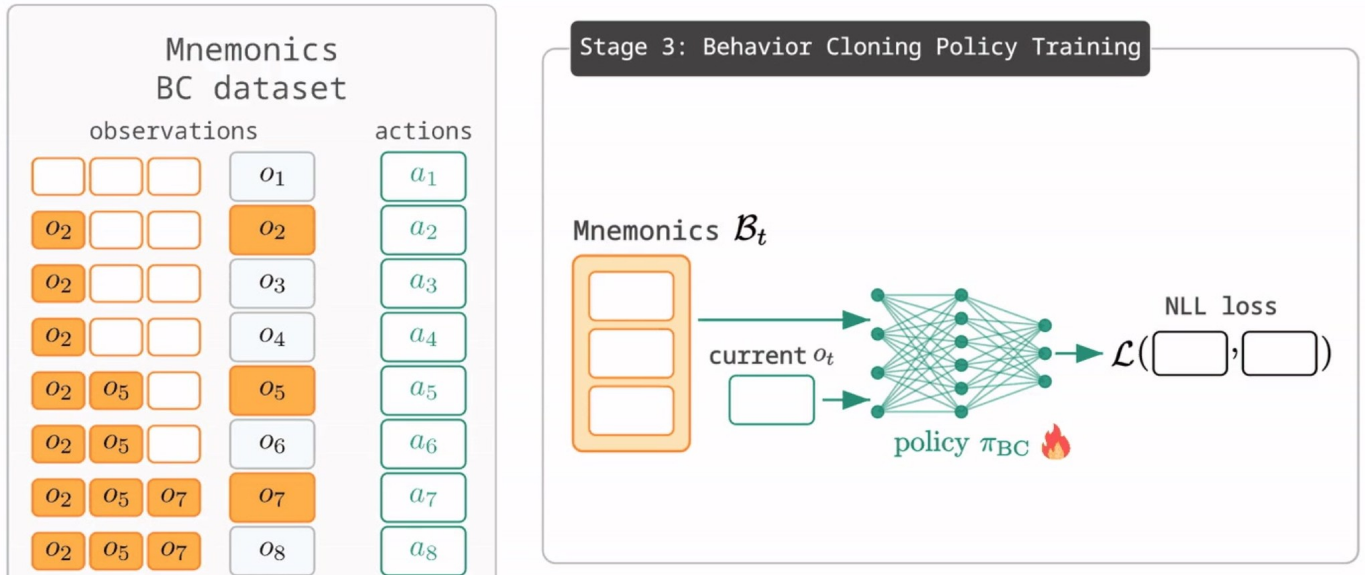
True keyframes result in **lowest action error**

Mnemonics: a set of keyframes that assists with memory

METHOD · STEP 3 OF 3

Training mnemonics-conditioned BC policy

3 Finally we combine discovered keyframes with the current observation to train a **BC policy** that models action distribution.

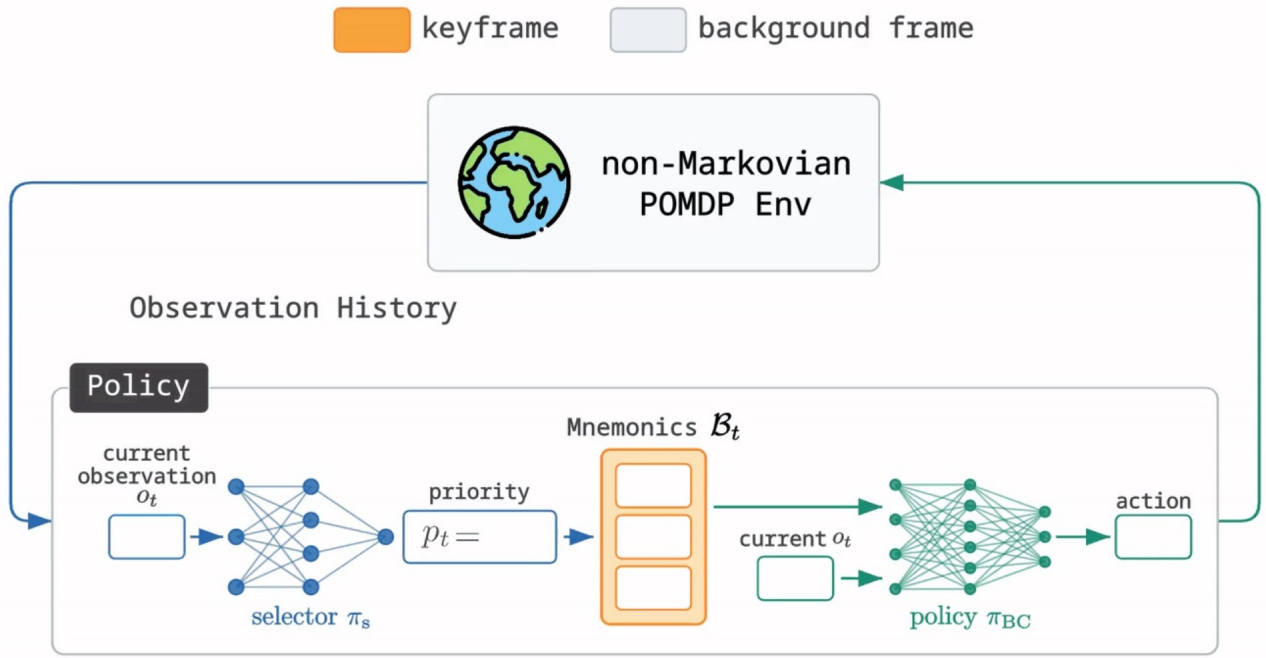


Mnemonics resolve action ambiguity at decision-critical timesteps

METHOD · DEPLOYMENT

Deployment of mnemonics-conditioned BC policy

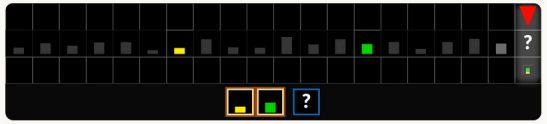
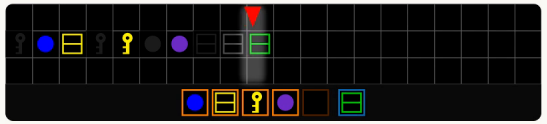
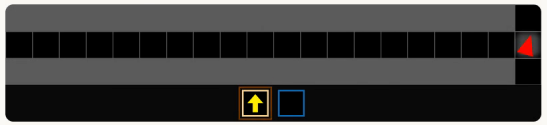
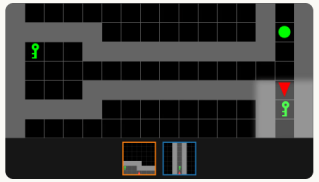
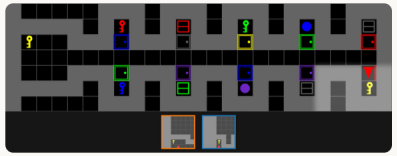
Keyframe Mnemonics (KM) policy works with a high-fidelity working memory



> **Horizon-invariance** is the ability of a policy to generalize at horizons much longer than the training distribution.

RESULTS

Horizon-invariance in synthetic and grid domains

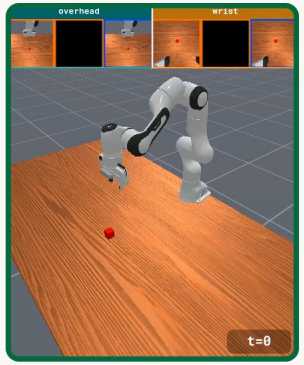
Environment		Train horizon	Best baseline	KM at max horizon
Add Synthetic		**30** steps	**0.218** DLSTM · at 10K steps	**1.6×10^{-5}** MSE at 1M steps
Scattered-Copy Synthetic		**20** steps	**4%** LSTM(T) · at 10K steps	**100%** success at 1M steps
T-Maze Synthetic		**30** steps	**84%** LSTM(B) · at 10K steps	**100%** success at 1M steps
MiniGrid-Memory Grid world		**17** environment size	**50%** DLSTM · at size 1001	**100%** success at size 1001
LTMB-Hallway Grid world		**8** corridor length	**0%** All baselines · at length 1000	**93%** success at length 1000

K = 1,000 · M = 1,000,000

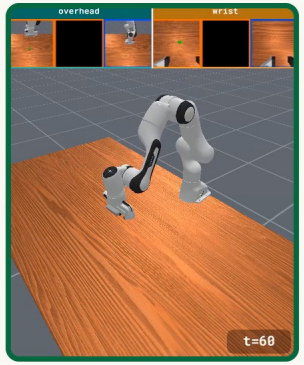
RESULTS

Horizon-invariance in robot manipulation and real world

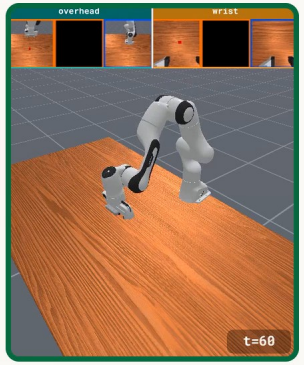
SIMULATION



TRAIN Δt = 15 steps

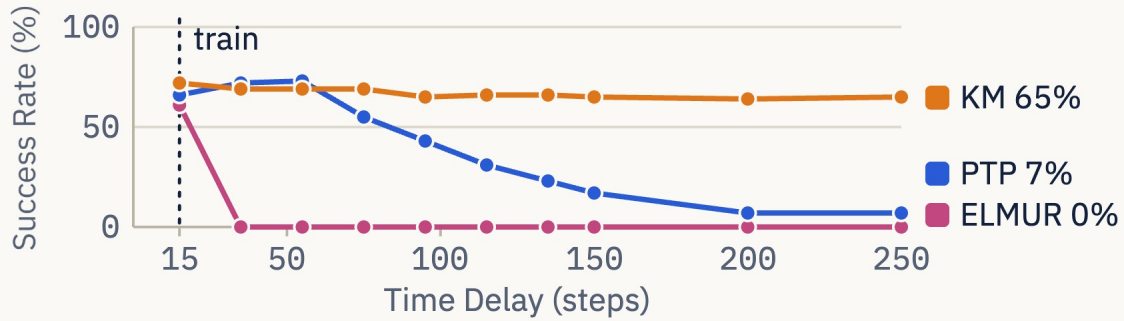


OOD Δt = 95 steps

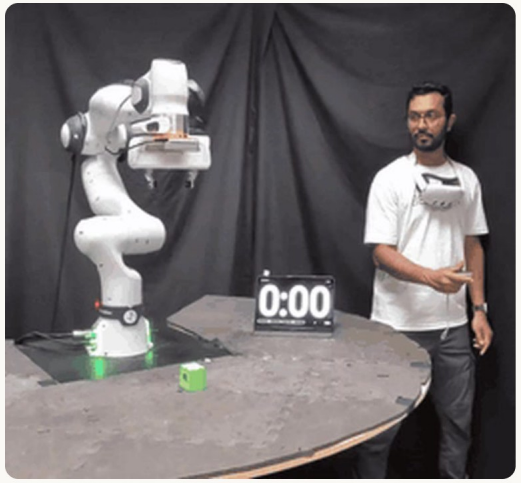


OOD Δt = 250 steps

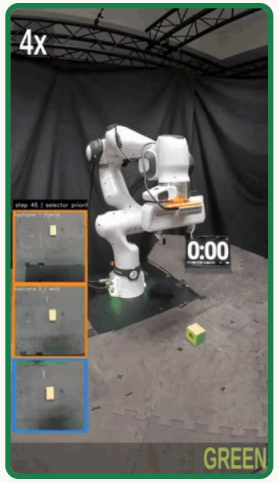
Performance in RememberColor3-ΔT



REAL ROBOT

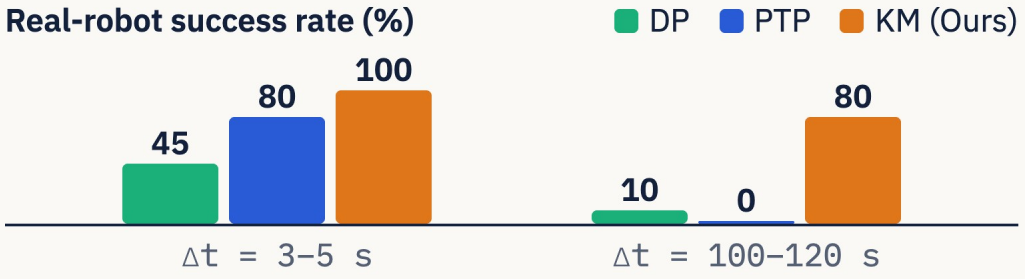


TRAIN Δt = 3–5 s



OOD Δt = 100–120 s

Real-robot success rate (%)



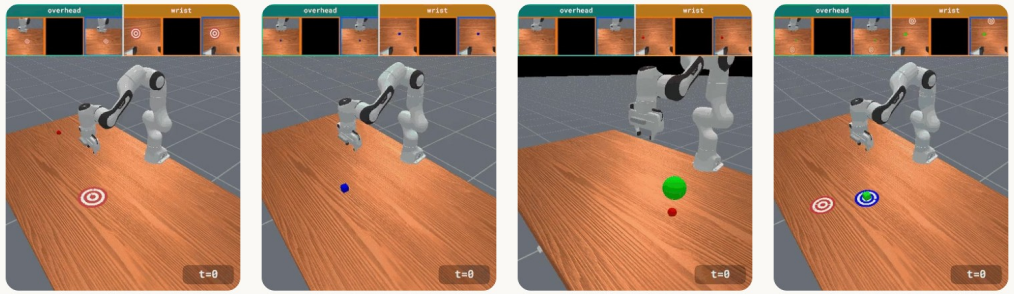
RESULTS

Performance on memory-intensive robot manipulation

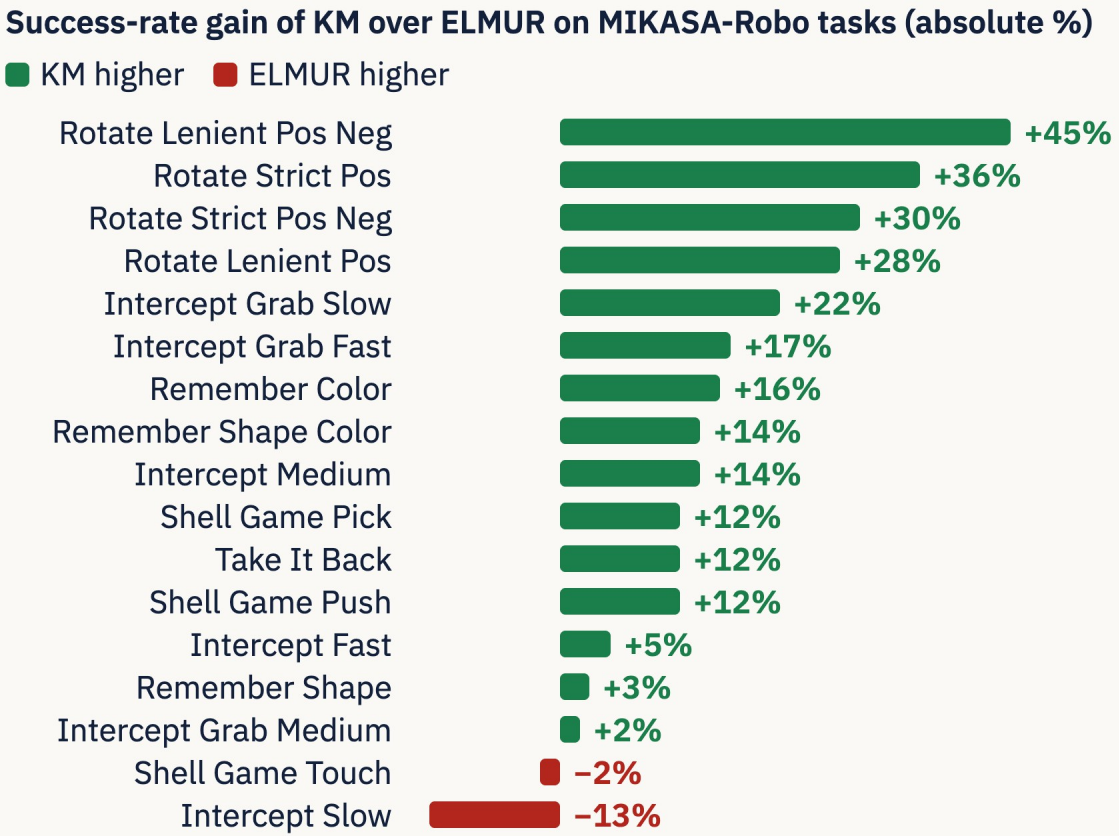
+13.9%

over SoTA on MIKASA-Robo

avg. absolute success rate, 23 tasks



TEMPORAL OBJECT SPATIAL SPATIAL



RESULTS

Robust to distractors, no regression on Markovian tasks

VISUAL DISTRIBUTION SHIFT



Δt = 3–5 s

80% **Robustness to distractors**
KM success rate

Unseen objects are dropped into the workspace mid-rollout.

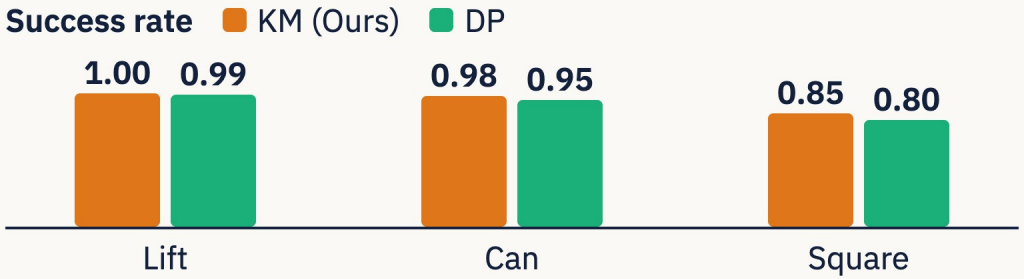
Selector and BC policy both stay robust.

MARKOVIAN TASKS · ROBOMIMIC



Lift Can Square

Mnemonics-conditioned policies remain competitive in Markovian tasks.

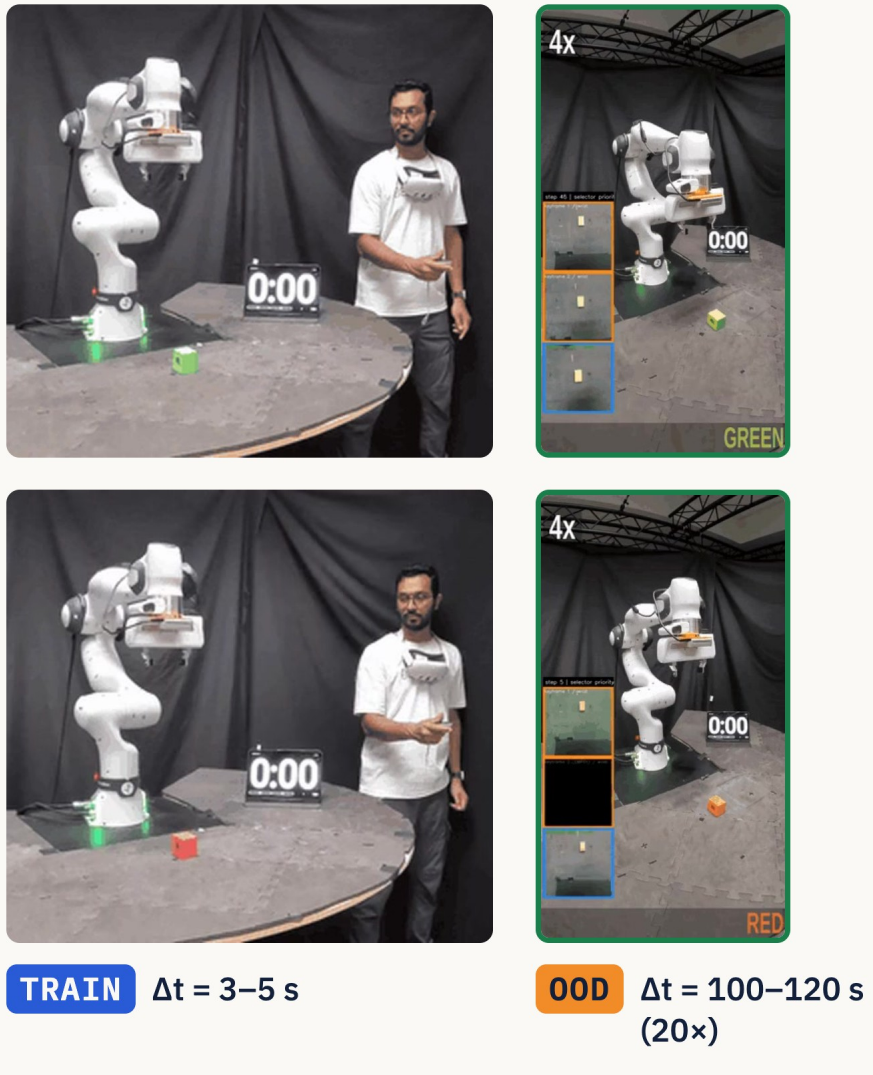


KEYFRAME MNEMONICS

Conclusion

1. We can effectively find the most informative keyframes (*mnemonics*) from expert demonstrations using self-supervised learning on BC datasets without expert annotations.

2. Mnemonics-conditioned BC policies maintain strong performance on test horizons that are significantly longer than those seen during training.



TRAIN Δt = 3–5 s

OOD Δt = 100–120 s (20×)

NEURAL INFORMATION PROCESSING SYSTEMS

Arizona State University

keyframe-mnemonics.github.io

SCAN-ME