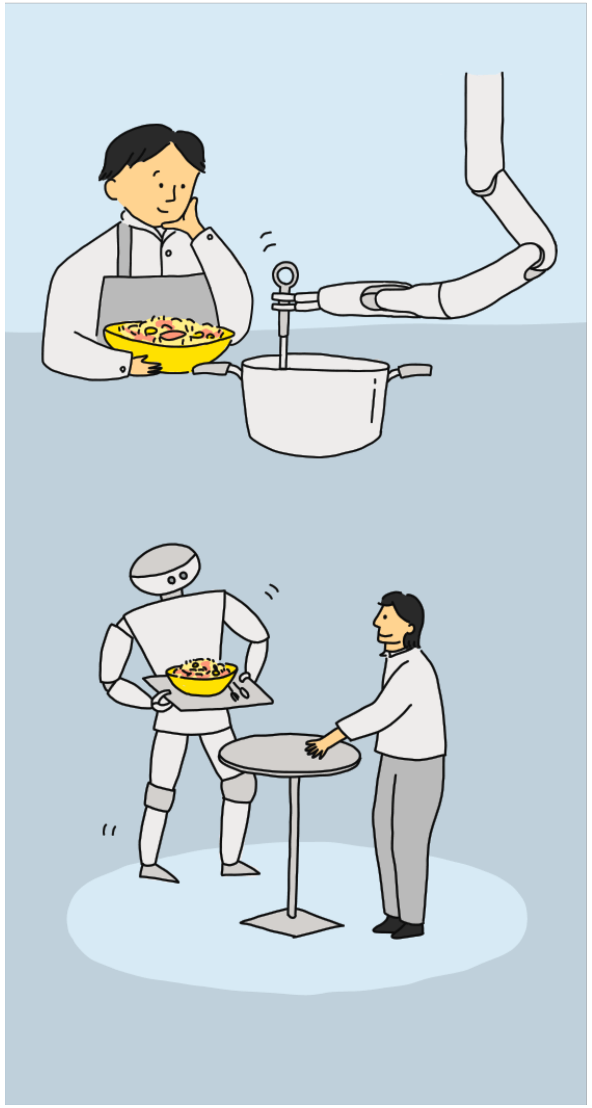
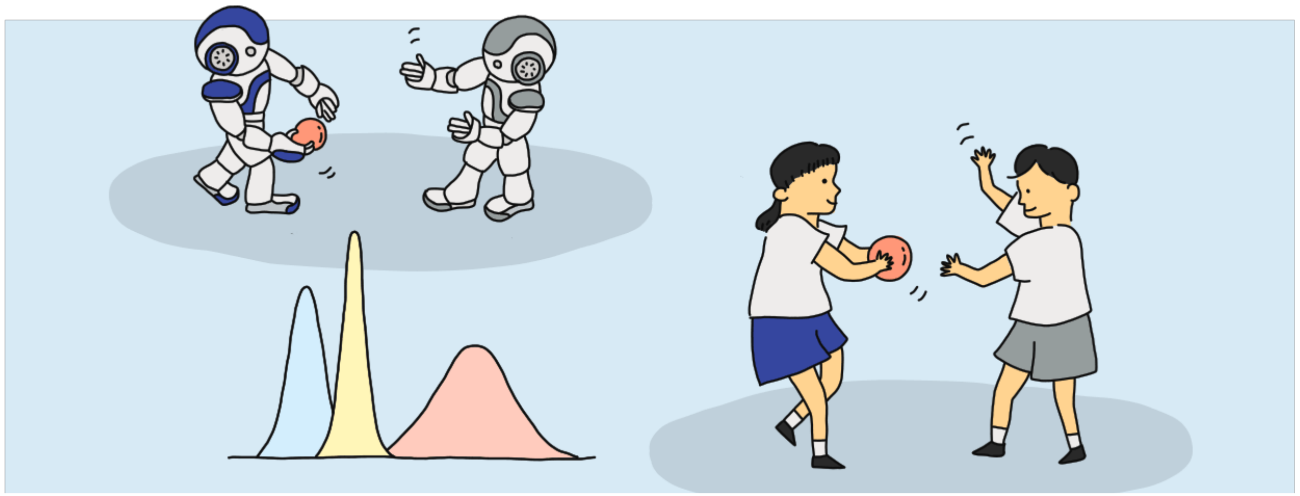
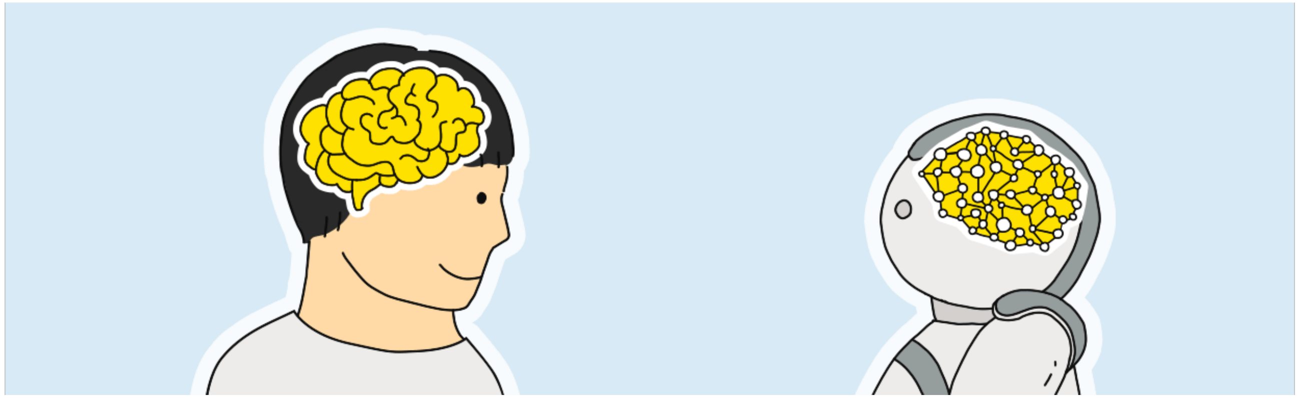
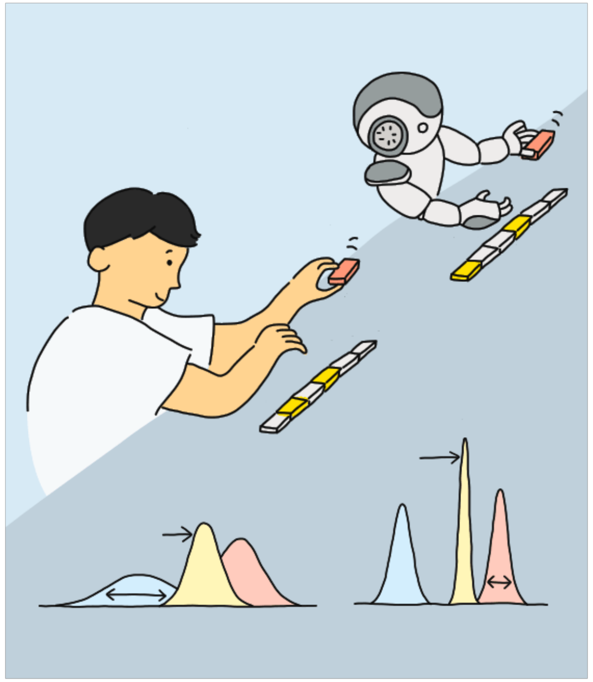
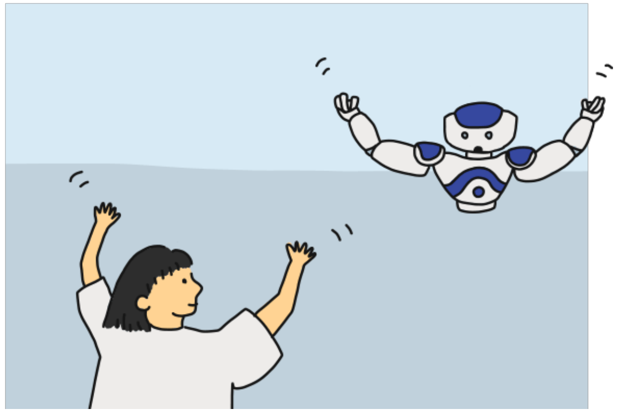




Deep Active Inference for Real-World Robotic Systems







Shingo Murata

Associate Professor @Dept. of EEE, Keio Univ.

https://murata-lab.jp

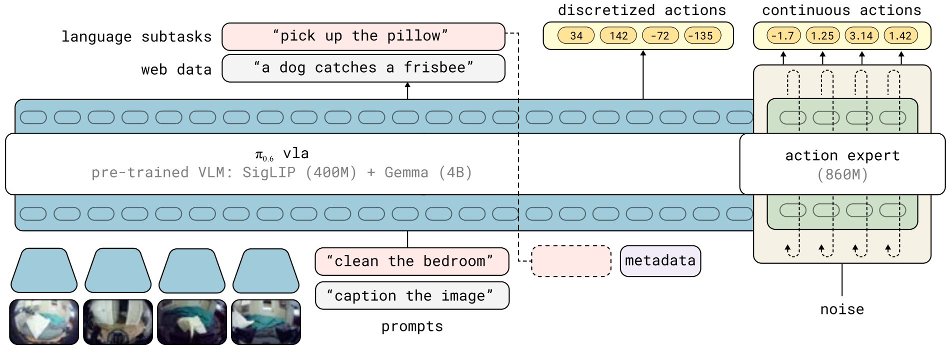
murata@elec.keio.ac.jp

@keio_crl

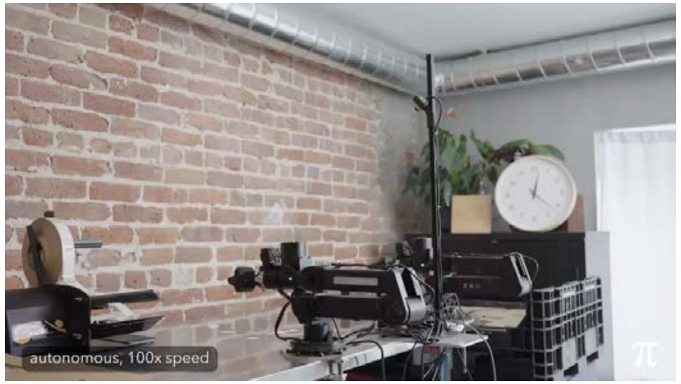
Recent progress

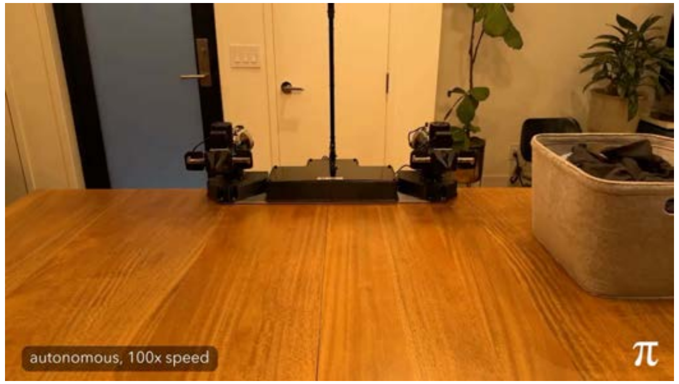
Vision-language-action (VLA) model

- RT-X (Google DeepMind)
- Large behavior model (TRI)
- π series (Physical Intelligence)



$\pi_{0.6}$





Limitation

The robot does not know *how to explore*.

- Same action repeated ☹ / No adaptive exploration ☹ / no belief update ☹

Exploration and goal-directed behavior

- Robots should not only imitate demonstrations
- They must explore when uncertain
- And act toward goals when confident

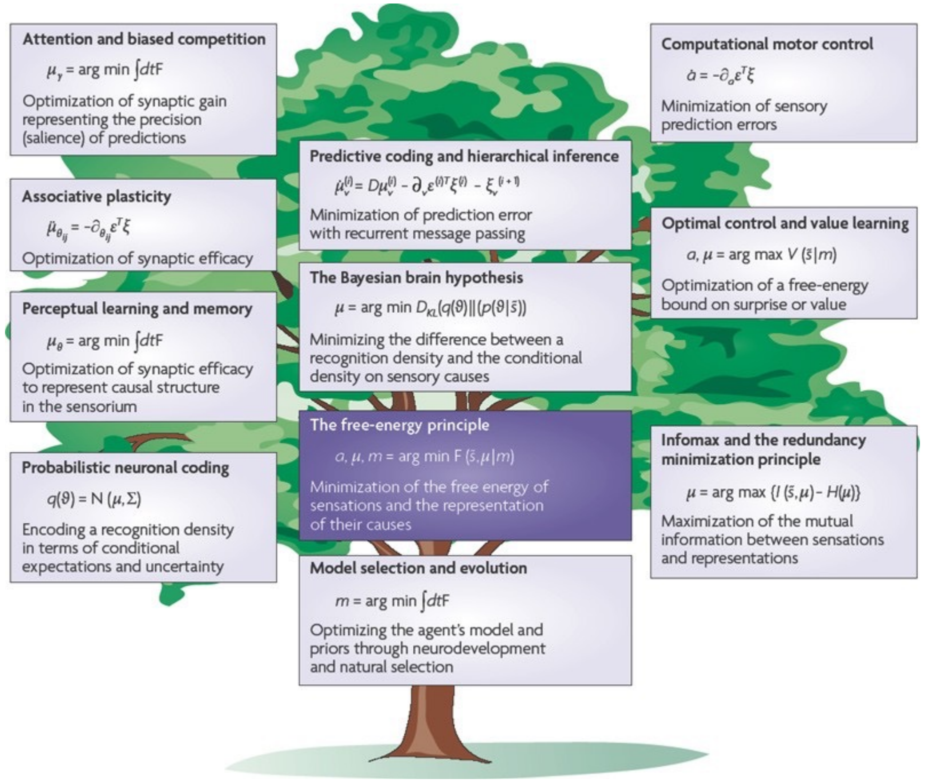
What computational principle governs such behavior?

Free-energy principle

[Friston, *Nat. Rev. Neurosci.* 2010]

Variational free energy (VFE) as a tractable proxy

- Surprise $-\log p(o_t)$ is intractable
- VFE F provides an upper bound on surprise
  - $$\begin{aligned} F &= \mathbb{E}_{q(z_t)}[\ln q(z_t) - \ln p(o_t, z_t)] \\ &= D_{\mathrm{KL}}[q(z_t)||p(z_t|o_t)] - \log p(o_t) \\ &\geq -\log p(o_t) \end{aligned}$$
- Minimizing VFE approximates surprise minimization



[Friston+, *Cognit. Neurosci.* 2015]

Variational free energy and expected free energy

Cognitive functions for minimizing free energies

- Perception & learning: Minimization of VFE F at past and present
  - Upper bound on surprise $-\log p(o_t)$ $\quad F = D_{\mathrm{KL}}[q(z_t)||p(z_t|o_t)] - \log p(o_t) \geq -\log p(o_t)$
  - Equivalent to the negative of ELBO $\quad F = D_{\mathrm{KL}}[q(z_t)||p(z_t)] - \mathbb{E}_{q(z_t)}[\log p(o_t|z_t)]$
- Action: Minimization of expected free energy (EFE) G at future $\tau > t$
  - Policy π represents action sequence $a_{1:T}$
  - Preference distribution $\tilde{p}(o_\tau)$

$$G(\pi) = \underbrace{-\mathbb{E}_{q(o_\tau|\pi)}\left[D_{\mathrm{KL}}[q(z_\tau|o_\tau,\pi)||q(z_\tau|\pi)]\right]}_{\textbf{Epistemic value}} \underbrace{- \mathbb{E}_{q(o_\tau|\pi)}[\log \tilde{p}(o_\tau)]}_{\textbf{Extrinsic value}}$$

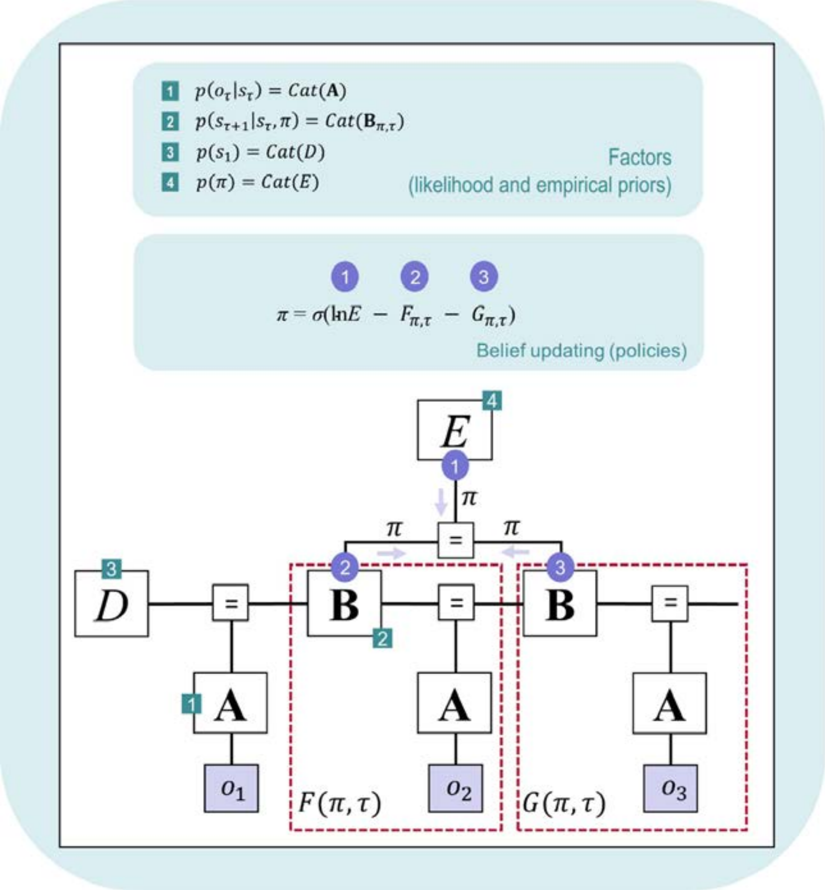
[Smith+, *J. Math. Psychol.* 2015]

Classical implementation of active inference

Matrix representation of each distribution

- A matrix for likelihood mapping
- B matrix for state transition
- C matrix for preference
- D matrix for initial state prior
- E matrix for policy distribution

Limited to toy problems (e.g., T-maze)



[Heins+, *J. Open Source Software* 2022]

Classical implementation of active inference

Matrix representation of each distribution

- A matrix for likelihood mapping
- B matrix for state transition
- C matrix for preference
- D matrix for initial state prior
- E matrix for policy distribution

Limited to toy problems (e.g., T-maze)

```python
import numpy as np

import pymdp
from pymdp import utils, maths
from pymdp.agent import Agent

# create a simple model with one hidden state factor, and one observation modality

n_obs = 3
n_states = 3

A = utils.obj_array(1)
A[0] = np.array([[1.0, 0.0, 0.0],
                 [0.0, 1.0, 0.0],
                 [0.0, 0.0, 1.0]])

# introduce uncertainty into one of the hidden states
inv_temperature = 0.5
A[0][:,2] = maths.softmax(inv_temperature* A[0][:,2])

# create a simple transition model with two possible actions

B = utils.obj_array(1)
B[0] = np.zeros((3, 3, 2))

# first action leads to first two states with uncertainty
B[0][:,:,0] = np.array([[0.5, 0.5, 0.5],
                        [0.5, 0.5, 0.5],
                        [0.0, 0.0, 0.0]])

# second action leads to last state with certainty
B[0][:,:,1] = np.array([[0.0, 0.0, 0.0],
                        [0.0, 0.0, 0.0],
                        [1.0, 1.0, 1.0]])

# specify prior preferences (C vector)
C = utils.obj_array_uniform([n_obs])

# specify prior over hidden states (D vector)
D = utils.obj_array(1)
D[0] = utils.onehot(1, n_states)

# instantiate your agent with a call to the ‘Agent()‘ constructor
my_agent = Agent(A=A, B=B, C=C, D=D)

# write a simple environment class, where state depends on the action probabilistically, and observation is deterministic function of the state except for state 2, where it’s randomly sampled
```

DNN implementation of active inference

Scaling up active inference to real-world problems

[Ueltzhöffer, *Biol. Cybern.* 2018; Mazzaglia+, *Entropy* 2022]

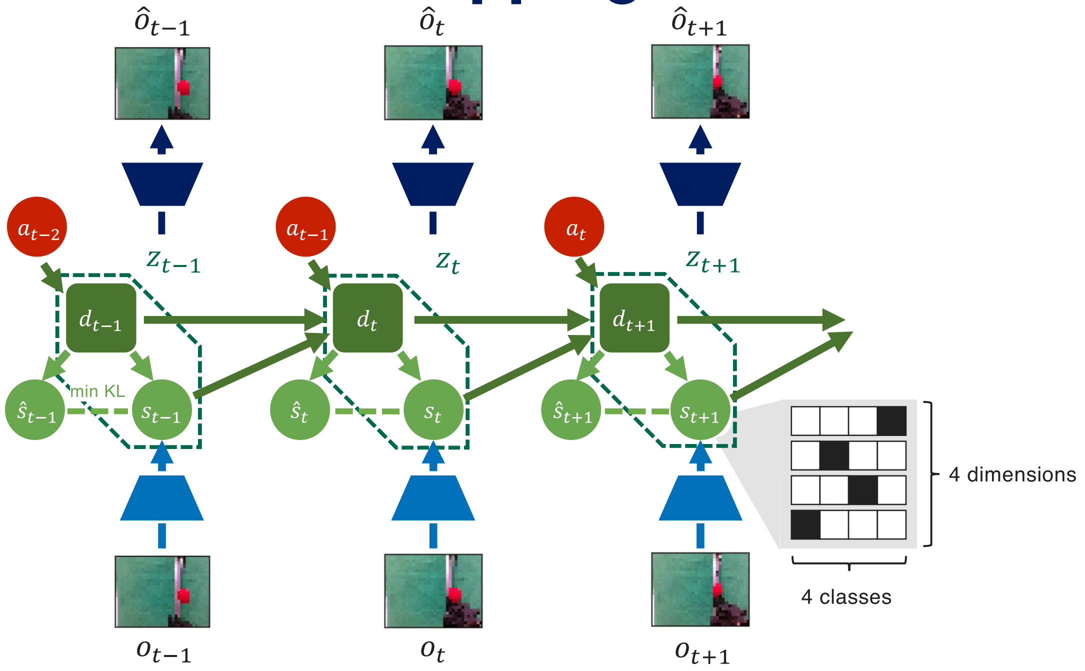
- World modeling
  - Recurrent state-space model (RSSM) [Hafner+, *ICML* 2019]
  - Predictive coding-inspired variational recurrent neural network (PV-RNN) [Ahmadi+, *Neural Comput.* 2019]
- Policy modeling
  - Diffusion/flow-matching-based policies (diffusion policy, streaming flow policy, …)
  - Transformer-based policies (ACT, BeT, VQ-BeT, …)

Generative world models

RSSM representing state transition and likelihood mapping

- Latent State $z_t = \{d_t, s_t\}$
  - $d_t \sim p_\theta(d_t | z_{t-1}, a_{t-1})$

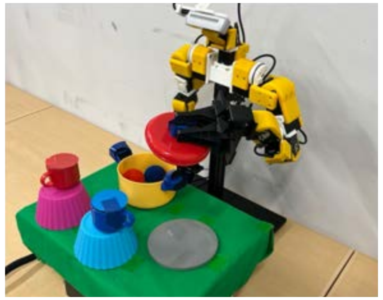
    → $d_t = \mathrm{GRU}(z_{t-1}, a_{t-1})$
- Prior $s_t \sim p_\theta(s_t | d_t)$
- Posterior $s_t \sim q_\phi(s_t | d_t, o_t)$
  - Observation o_t is replaced with image feature from CNN
- Likelihood $p_\theta(o_t | z_t)$



$$F_t = D_{\mathrm{KL}}[q_\phi(s_t | d_t, o_t) || p_\theta(s_t | d_t)] - E_{q_\phi(s_t | d_t, o_t)}[\log p_\theta(o_t | z_t)]$$

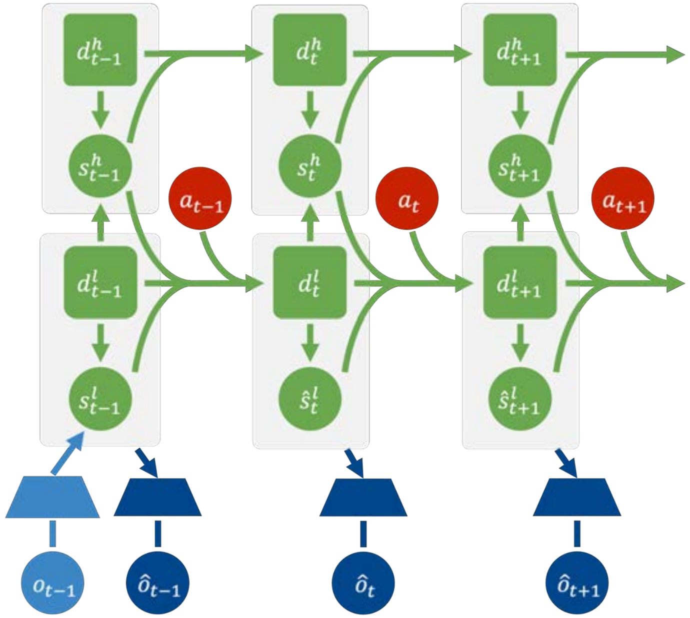
Multiple Timescale RSSM (MTRSSM)

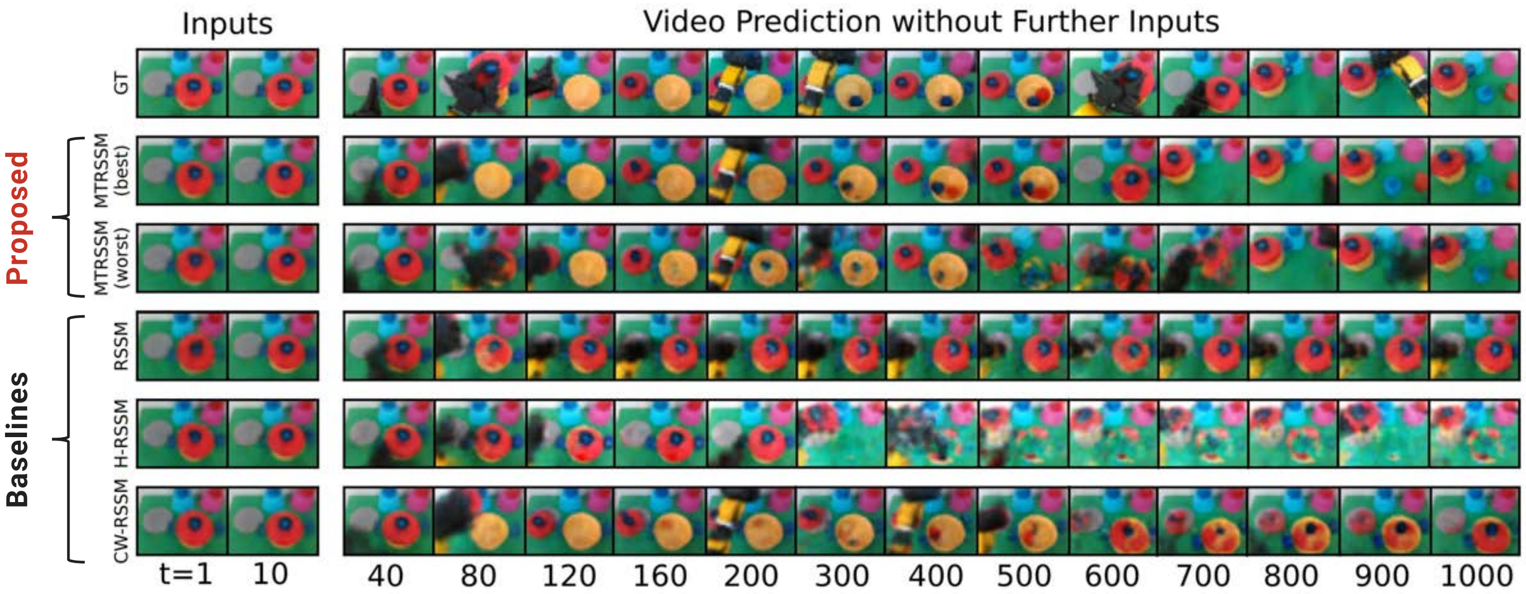
[Fujii & Murata, *ICDL* 2023]



Introduction of multiple timescales

- Trained to minimize VFE
  - Long-horizon action-conditioned video predictions





[Yokozawa, Fujii, Nomura & Murata, *arXiv* 2025]

Multiple Timescale RSSM (MTRSSM)

Metrics: PSNR / SSIM / LPIPS

- MTRSSM
  - Outperforms all baselines on all metrics
- RSSM [Hafner et al., *ICLR* 2020]
  - Unable to learn long-term temporal dependencies
- H-RSSM
  - Worse than RSSM
- CW-RSSM [Saxena et al., *NeurIPS* 2021]
  - Temporal hierarchy improves over RSSM

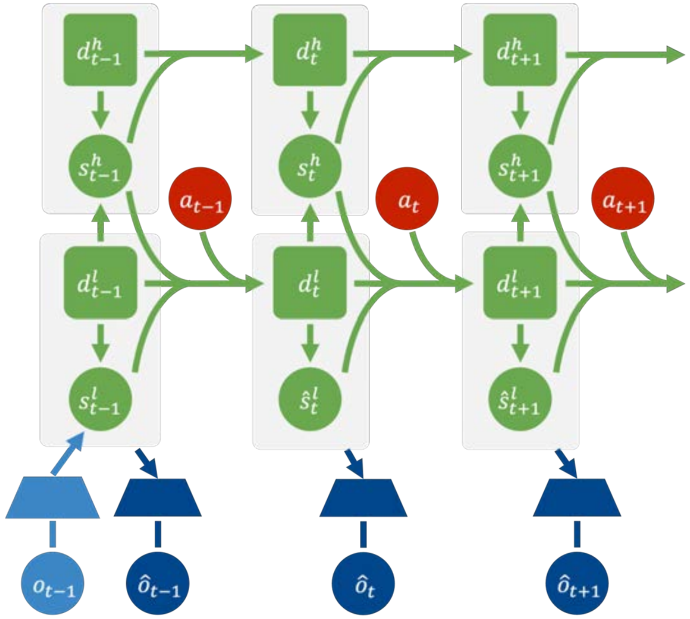
Model	SSIM ⇧	PSNR ⇧	LPIPS ⇩
MTRSSM	**0.458**	**14.321**	**0.138**
RSSM	0.371	13.119	0.182
H-RSSM	0.334	12.058	0.205
CW-RSSM (interval : 8)	0.403	13.509	0.165
CW-RSSM (interval : 16)	0.387	13.403	0.184
CW-RSSM (interval : 32)	0.303	12.469	0.232
CW-RSSM (interval : 64)	0.396	13.477	0.174
CW-RSSM (interval : 128)	0.360	13.122	0.219

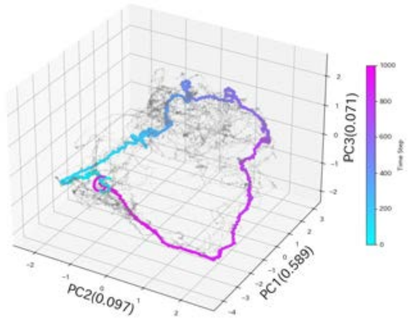
[Yokozawa, Fujii, Nomura & Murata, *arXiv* 2025]

Multiple Timescale RSSM (MTRSSM)

Introduction of multiple timescales

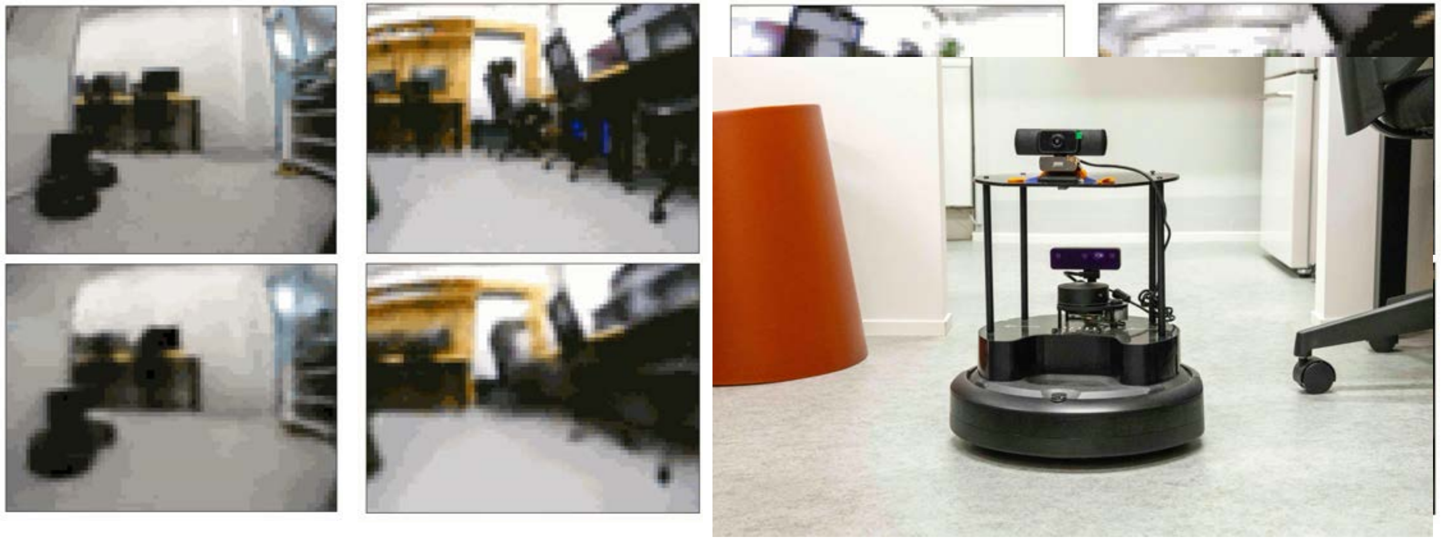
- Trained to minimize VFE
  - Long-horizon action-conditioned video predictions





Target

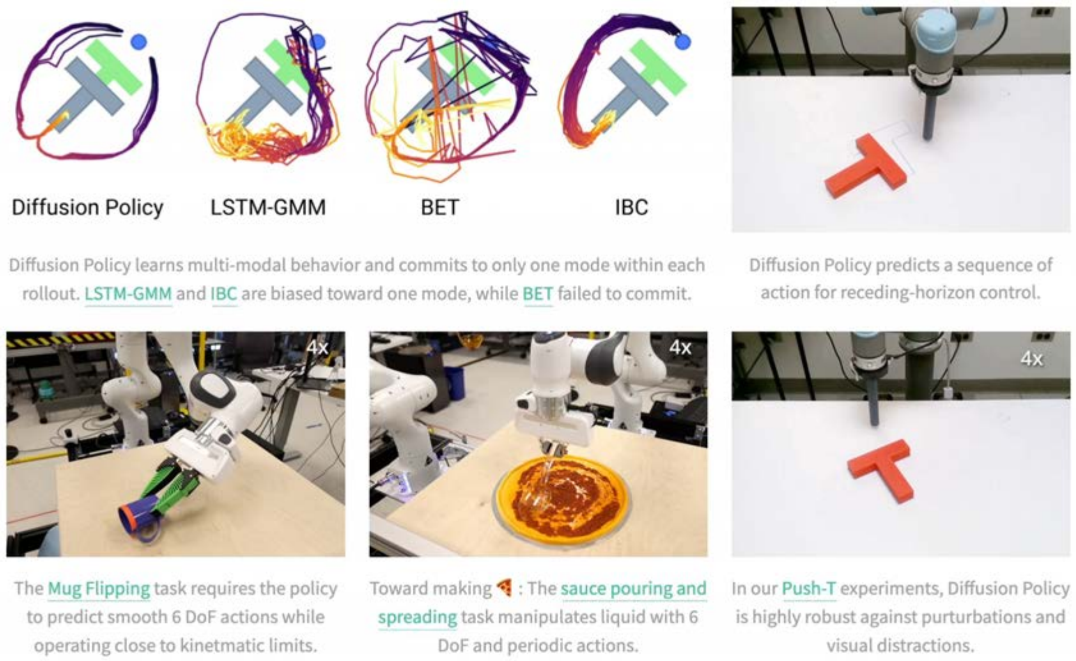
Imagination



Generative policies

Sampling multiple candidate action sequences

- e.g. Diffusion policy [Chi+, *RSS* 2023]
  - $\boldsymbol{a}_{t,k-1} = \frac{1}{\sqrt{\alpha_k}}\left(\boldsymbol{a}_{t,k} - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_\theta(\boldsymbol{o}_t, \boldsymbol{a}_{t,k}, k)\right) + \epsilon_k$
  - $\begin{cases} \boldsymbol{a}_t = a_{t-1:t+T_\mathrm{F}} \\ \boldsymbol{o}_t = o_{t-1:t} \end{cases}$
  - $\epsilon_k \sim \mathcal{N}(0, \sigma_k^2 I)$



Approximation of expected free energy

Approximating expectation with Monte Carlo sampling

- Expected free energy

$$\mathcal{G}_\tau(\pi) = \mathbb{E}_{q(o_\tau, s_\tau|\pi)}[\log q(s_\tau|\pi) - \log p(o_\tau, s_\tau|\pi)] \approx -\mathbb{E}_{q(o_\tau|\pi)}[D_{\mathrm{KL}}[q(s_\tau|o_\tau, \pi)||q(s_\tau|\pi)]] - \mathbb{E}_{q(o_\tau|\pi)}[\log p(o_\tau|C)]$$

- Approximate expected free energy

$$\mathcal{G}_\tau(\boldsymbol{a}_t) \approx \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left\{-D_{\mathrm{KL}}\left[q_\phi\left(s_\tau\middle|d_\tau, \hat{o}_\tau^{i,j}\right)||p_\theta(s_\tau|d_\tau)\right] - \log p(\hat{o}_\tau^{i,j}|C)\right\}$$

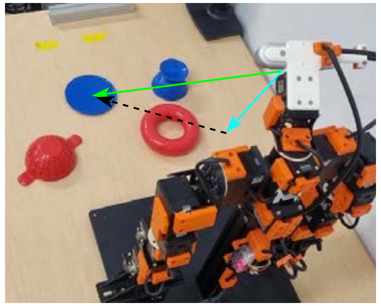
  - Sample $s_t^i \sim q_\phi(s_t|d_t, o_t)$ from the posterior given the current observation (M samples).
  - From each s_t^i, generate a sequence of priors by latent imagination: $p_\theta^i(s_{t+1}|d_{t+1}), \dots, p_\theta^i\left(s_{t+T_\mathrm{a}}\middle|d_{t+T_\mathrm{a}}\right)$.
  - At each time step, sample$s_\tau^{i,j} \sim p_\theta^i(s_\tau|d_\tau)$ (N samples).
  - From $z_\tau^{i,j} = \left\{d_\tau, s_\tau^{i,j}\right\}$, generate predicted images $\hat{o}_\tau^{i,j}$ using the likelihood model and compute the posterior.

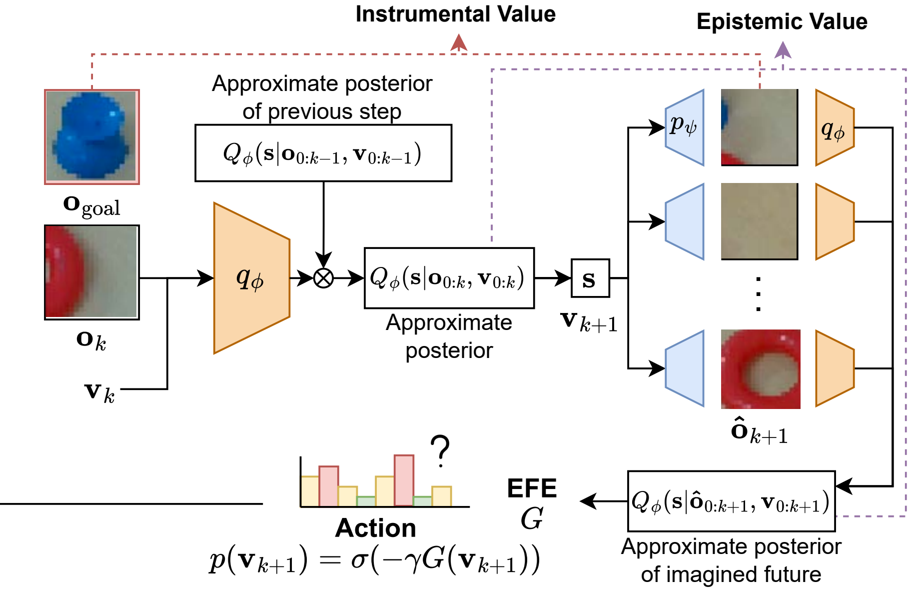
VAE-based framework

[Haddon-Hill & Murata, *ICANN* 2024]

Determine the next viewpoint v_{k+1} from past observations $o_{0:k}$

- Represent the environment state as a scene s
  - How are the objects arranged in the scene?
  - Need to integrate multiple observations





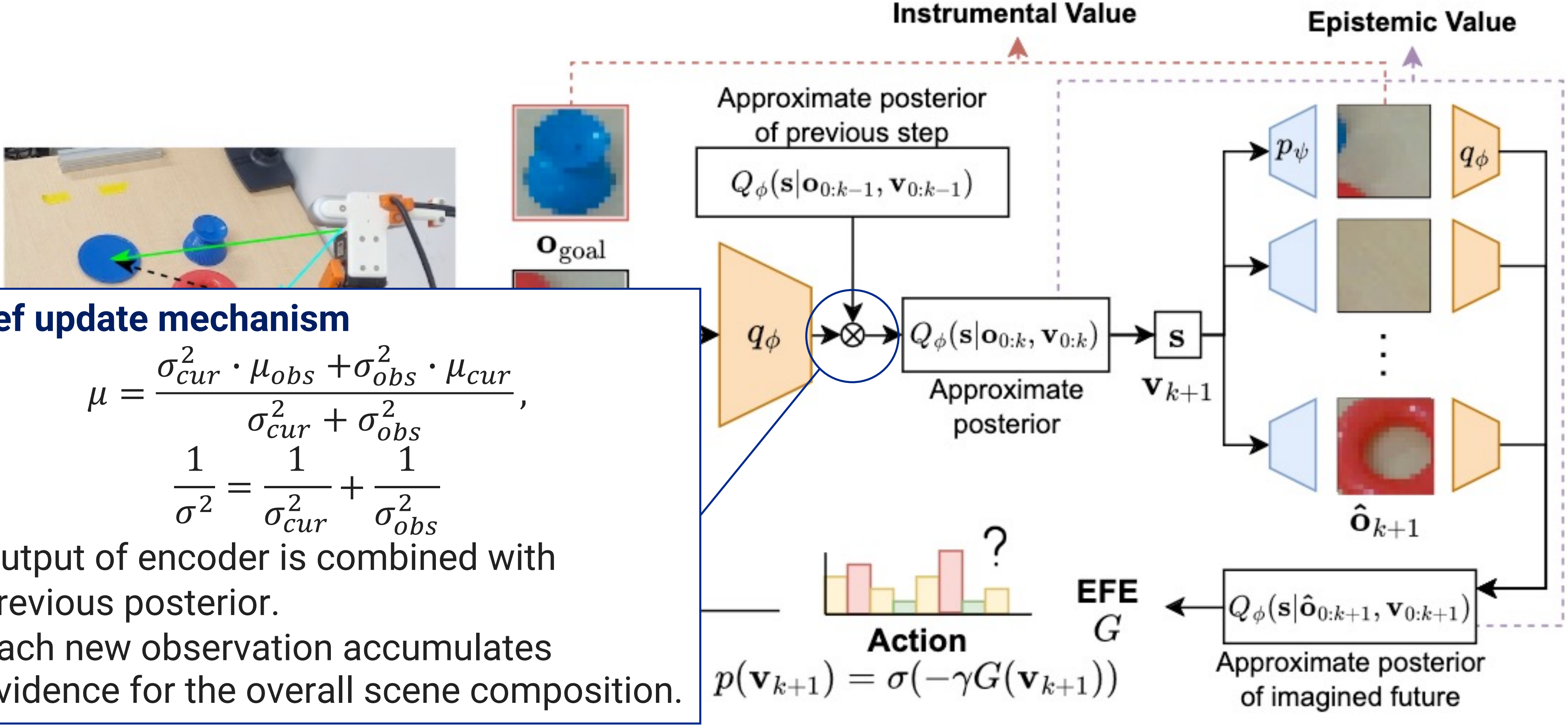
- Replace the policy π in EFE with the next viewpoint v_{k+1}

$$G(v_{k+1}) = \underbrace{-\mathbb{E}_{\tilde{Q}_\phi}\left[D_{\mathrm{KL}}[Q_\phi(s|o_{0:k+1}, v_{0:k+1})||Q_\phi(s|o_{0:k}, v_{0:k})]\right]}_{\textbf{Epistemic value}} - \underbrace{\mathbb{E}_{\tilde{Q}_\phi}[\log \tilde{P}(o)]}_{\textbf{Extrinsic value}}$$

where, $\tilde{Q}_\phi \coloneqq Q_\phi(o_{0:k+1}|v_{0:k+1})$

[Haddon-Hill & Murata, *ICANN* 2024]

Belief update mechanism



Belief update mechanism

$$\mu = \frac{\sigma_{cur}^2 \cdot \mu_{obs} + \sigma_{obs}^2 \cdot \mu_{cur}}{\sigma_{cur}^2 + \sigma_{obs}^2},$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_{cur}^2} + \frac{1}{\sigma_{obs}^2}$$

- Output of encoder is combined with previous posterior.
- Each new observation accumulates evidence for the overall scene composition.

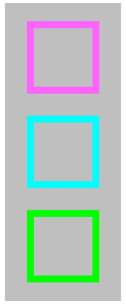
Fig: Overview of approach

[Haddon-Hill & Murata, *ICANN* 2024]

Analysis of reconstructions and epistemic values

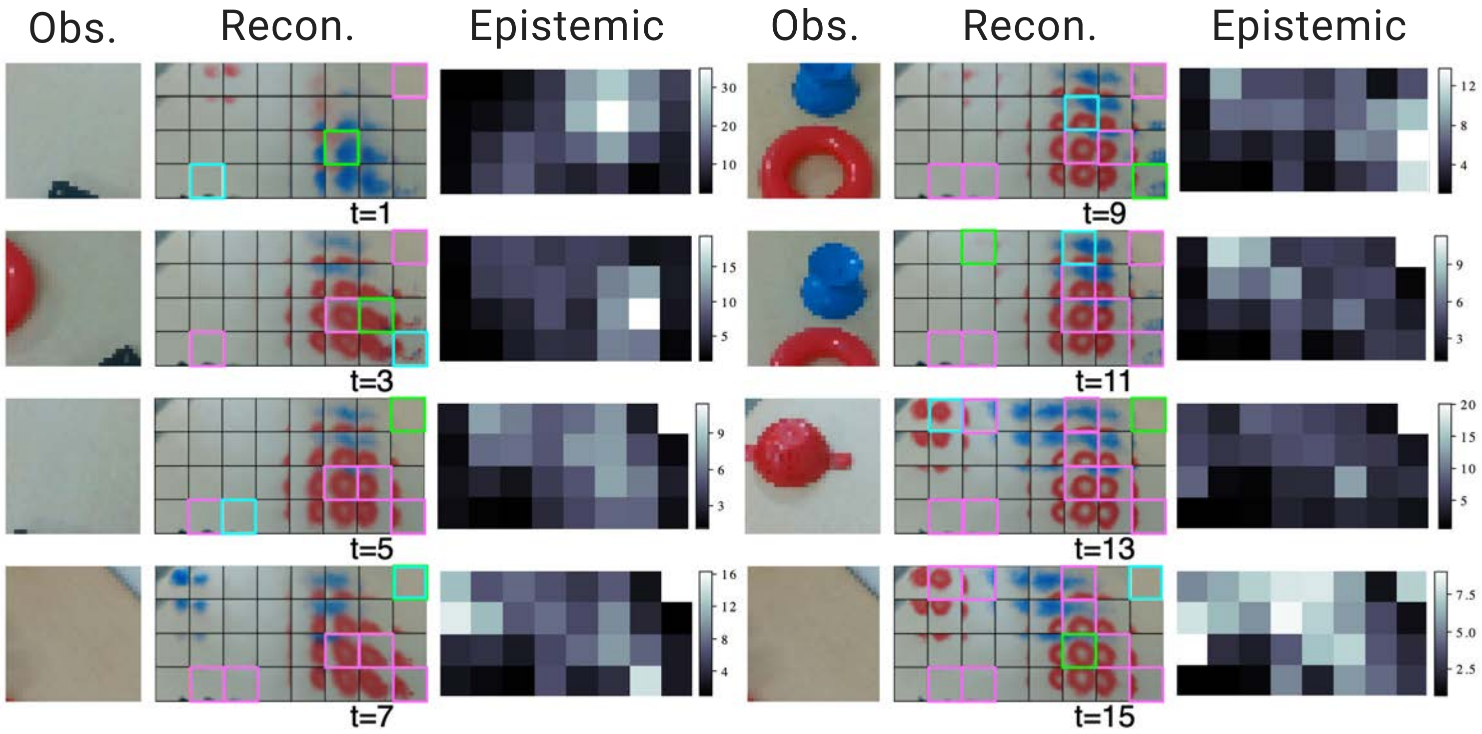
Epistemic only

- Viewpoints
  - Previously selected
  - Current
  - Future



- Collect observations to infer the scene representation $Q_\phi(s|\cdot)$.

$$G(v_{k+1}) = -\underbrace{\mathbb{E}_{\tilde{Q}_\phi}\left[D_{\mathrm{KL}}[Q_\phi(s|o_{0:k+1}, v_{0:k+1})||Q_\phi(s|o_{0:k}, v_{0:k})]\right]}_{\textbf{Epistemic}} - \underbrace{\mathbb{E}_{\tilde{Q}_\phi}[\log \tilde{P}(o)]}_{\textbf{Extrinsic}}$$

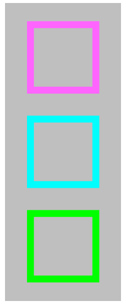


[Haddon-Hill & Murata, *ICANN* 2024]

Analysis of reconstructions and epistemic/extrinsic values

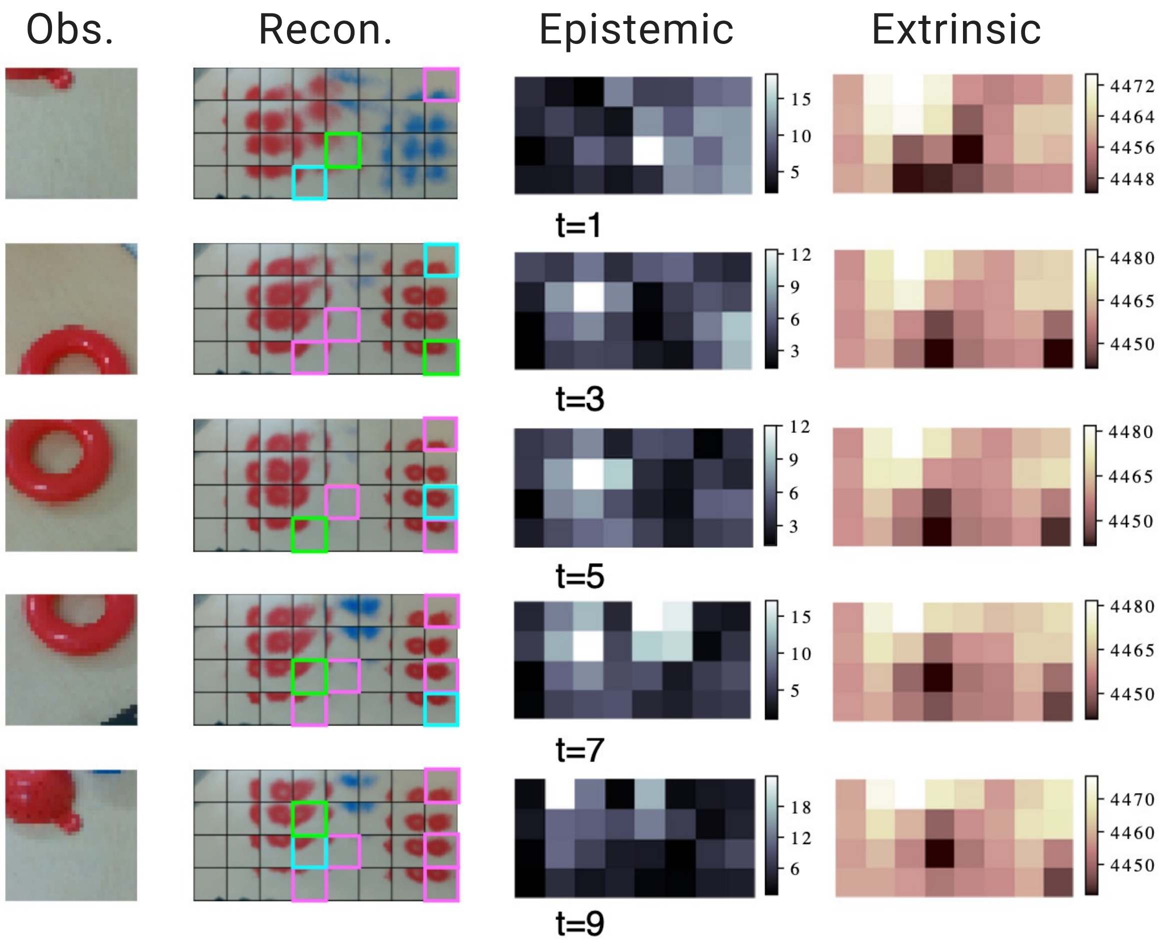
Full EFE (epistemic + extrinsic)

- Viewpoints
  - Previously selected
  - Current
  - Future



Goal $\mathbf{o}_{\text{goal}}$

- Collect observations to infer the scene representation $Q_\phi(s|\cdot)$, while reaching the target image.

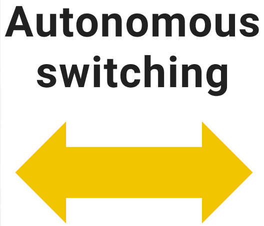


[Yokozawa, Fujii, Nomura & Murata, *arXiv* 2025]

Switching or balancing between exploration and goal-directed behavior

Exploration

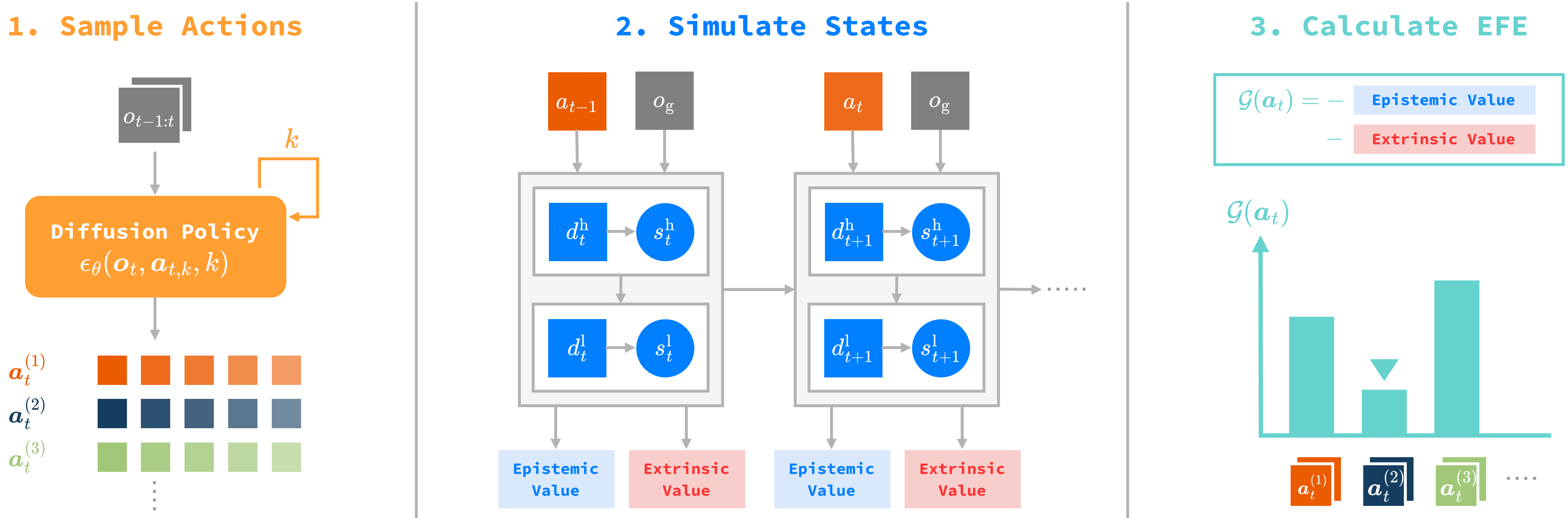
Goal-directed behavior



[Yokozawa, Fujii, Nomura & Murata, *arXiv* 2025]

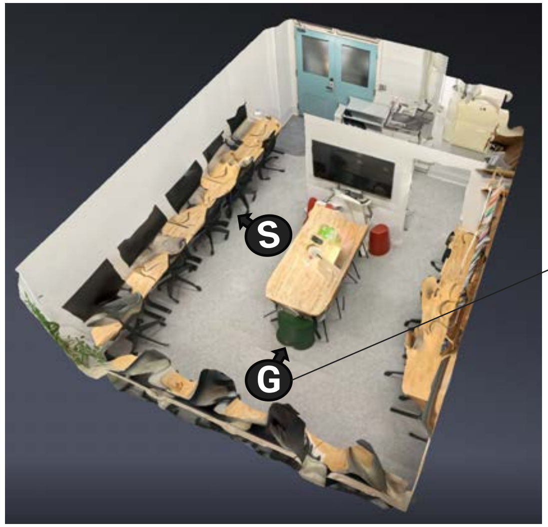
Deep generative mode-based framework

Deep active inference with diffusion policy and MTRSSM



Video

Autonomous goal reaching

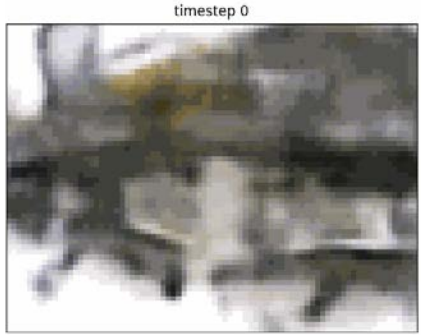


Goal image

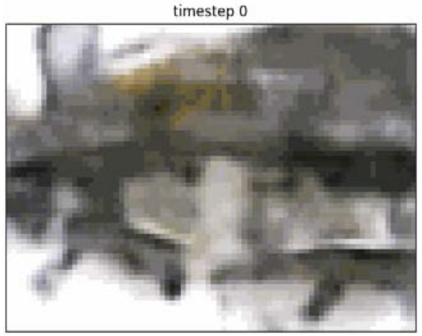
[Yokozawa, Fujii, Nomura & Murata, *arXiv* 2025]

Action evaluation based on EFE

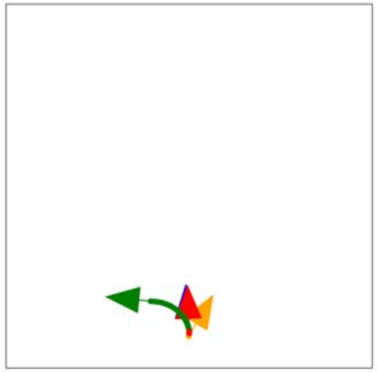
Early in the action



Candidate action1

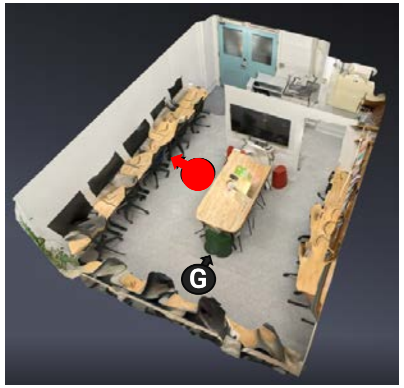


Candidate action 2

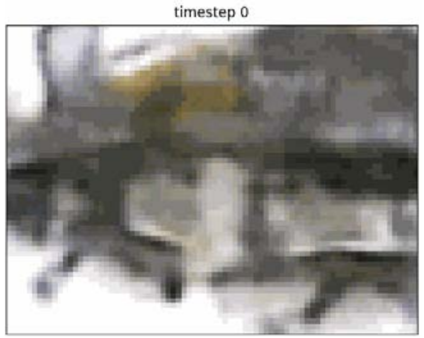


Candidate actions

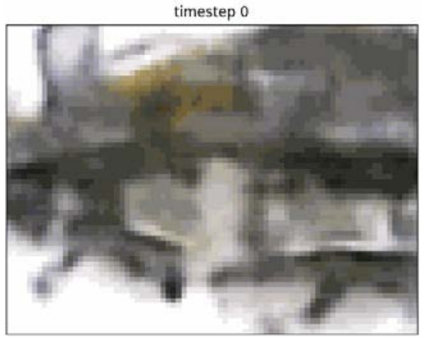
Current observation



Current position

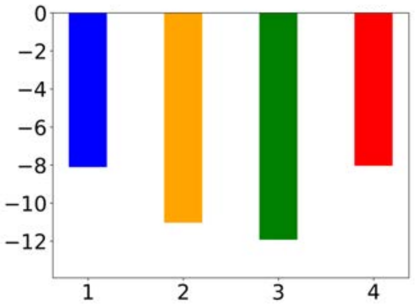


Candidate action 3

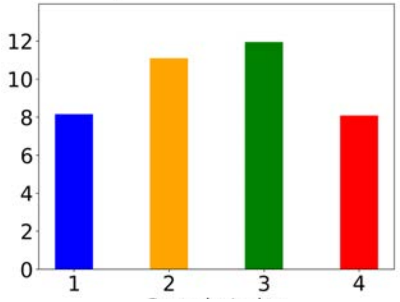


Candidate action 4

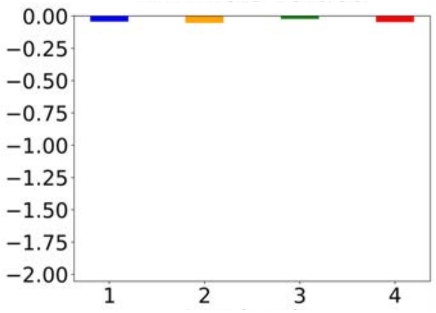
Mental simulation inside the world model



EFE



Epistemic value

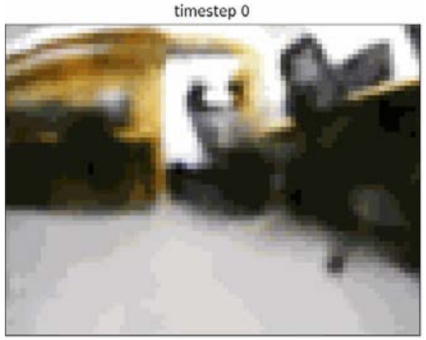


Extrinsic value

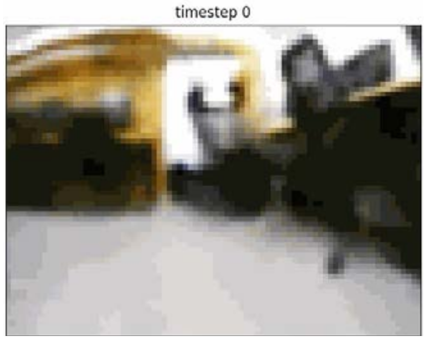
[Yokozawa, Fujii, Nomura & Murata, *arXiv* 2025]

Action evaluation based on EFE

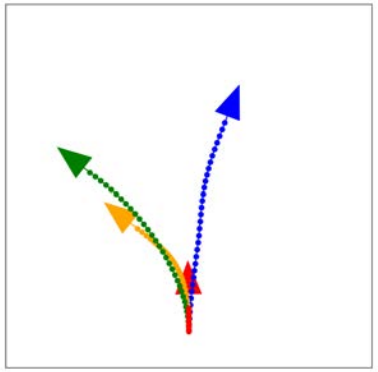
Later in the action



Candidate action1

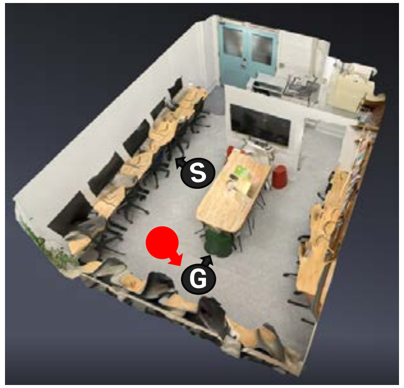


Candidate action 2

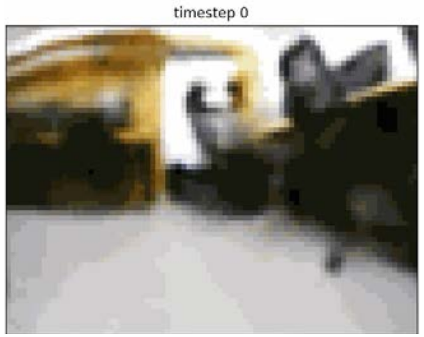


Candidate actions

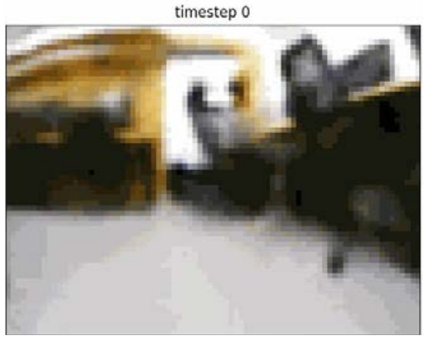
Current observation



Current position

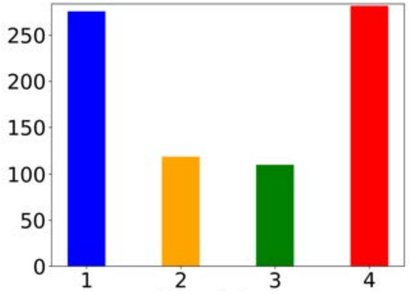


🏆 **Candidate action 3**

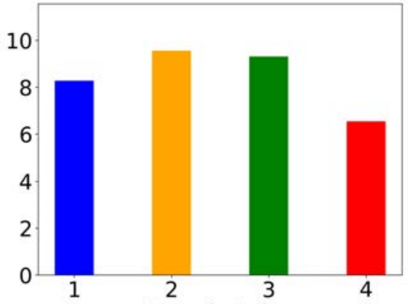


Candidate action 4

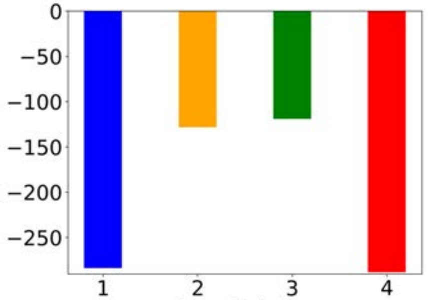
Mental simulation inside the world model



EFE



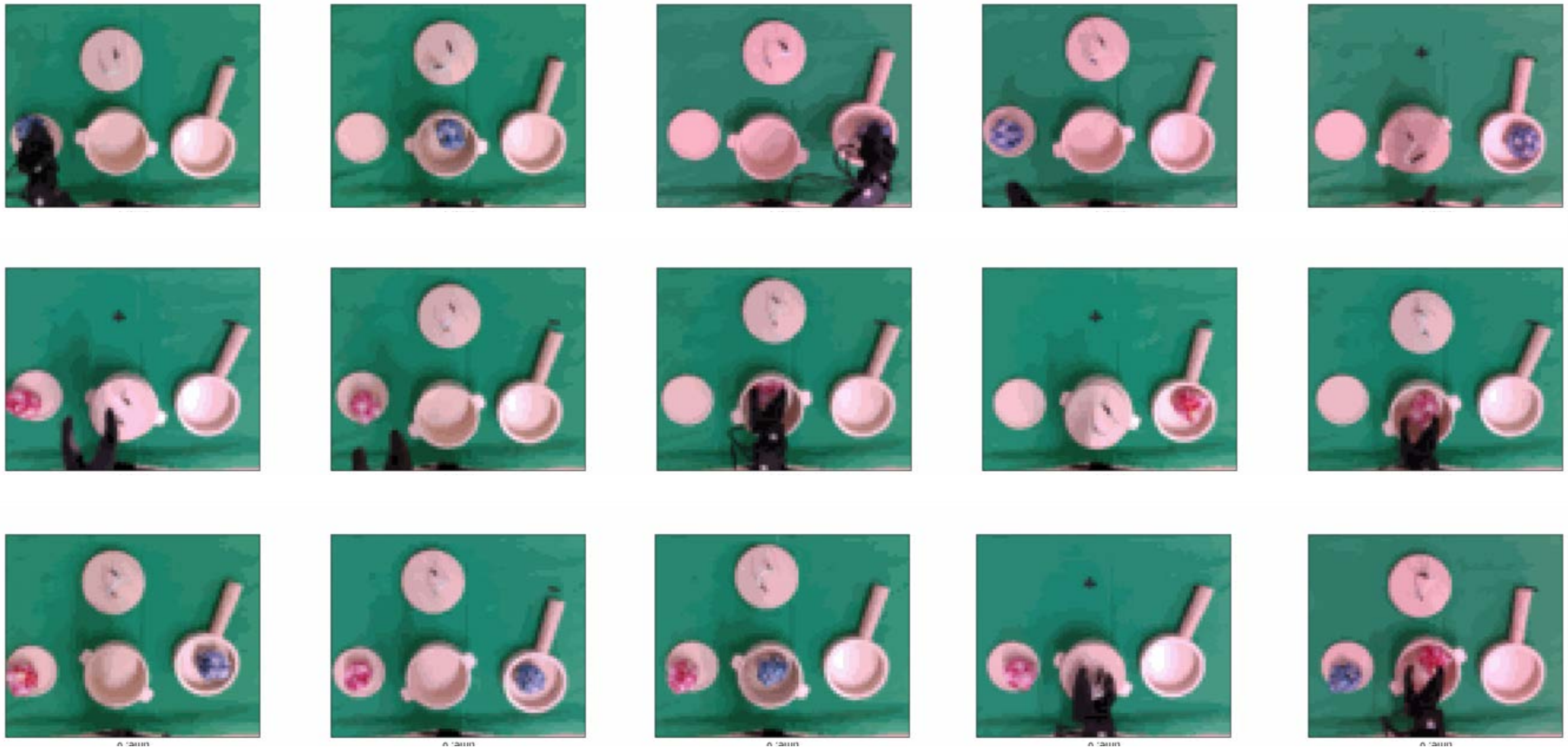
Epistemic value



Extrinsic value

Problem setting

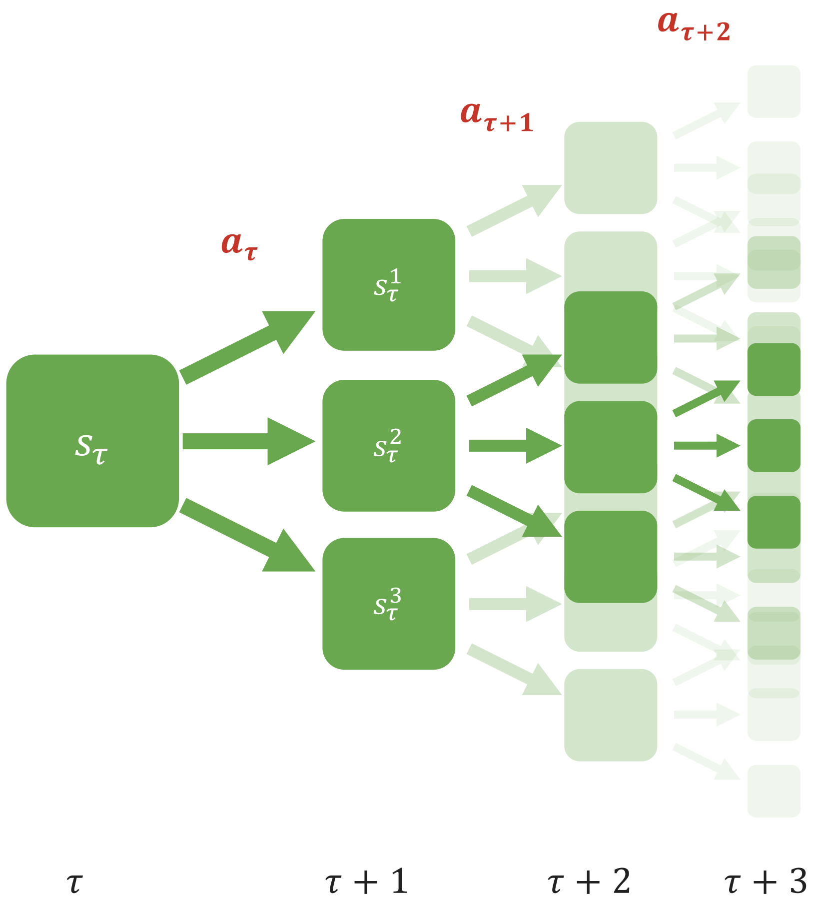
An environment with diverse state transitions



Challenges in real-world robots

Combinatorial explosion of state transitions

- Many possible actions at each time step
- Diverse state transitions generated by those actions
- EFE must be evaluated over a huge number of trajectories



Hierarchical solution

[Fujii & Murata, *RA-L* 2026]

World model (WM)

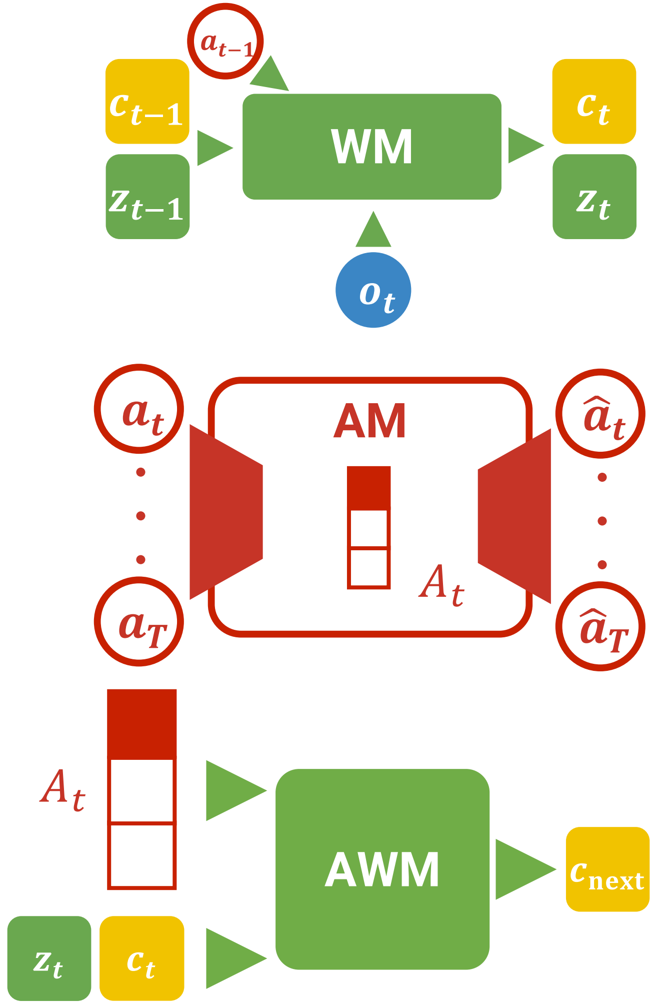
- State inference and prediction
  - Coarse dynamics state c_t
  - Fast dynamics state z_t

Action model (AM)

- Learn abstract actions A_t

Abstract world model (AWM)

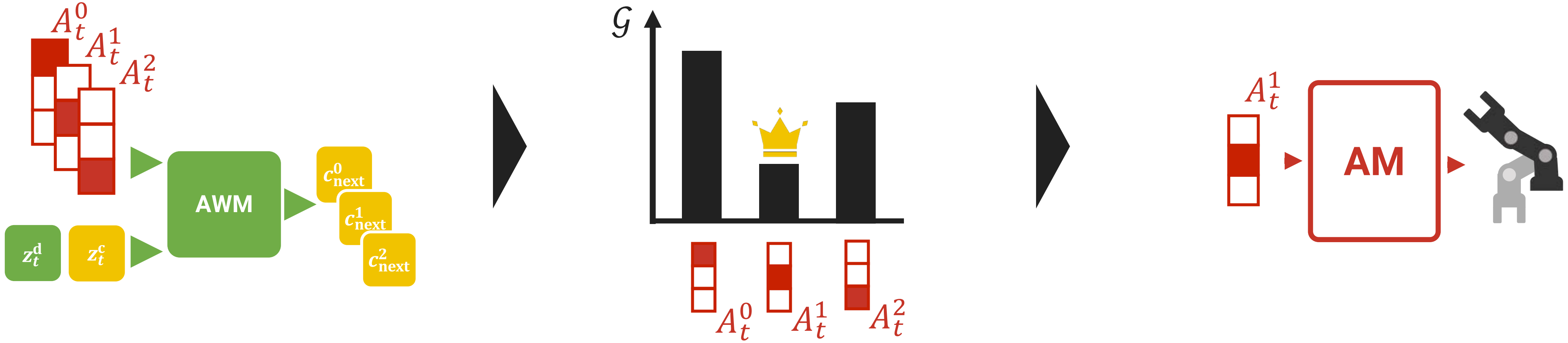
- Predict the next slow-timescale state c_{next} from the current states c_t, z_t and abstract action A_t



[Fujii & Murata, *RA-L* 2026]

Action generation based on EFE minimization using abstract actions

- Predict next states $\{c^i_{\text{next}}\}_{i=0}^N$ from each abstract action $\{A^i_t\}_{i=0}^N$
- Compute the EFE $\{\mathcal{G}_i\}_{i=0}^N$ for each slow-timescale state $\{c^i_{\text{next}}\}_{i=0}^N$
- Select the abstract action $A_t^{\arg\min \mathcal{G}_i}$ with the minimum EFE
- Generate the actual action using the action model



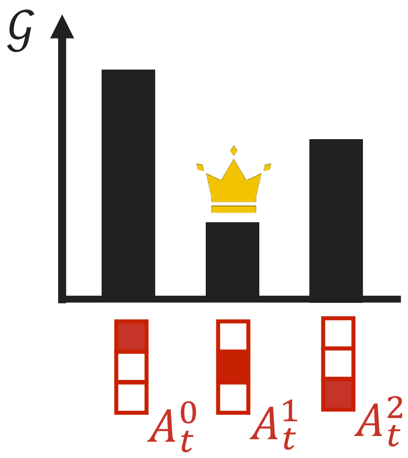
[Fujii & Murata, *RA-L* 2026]

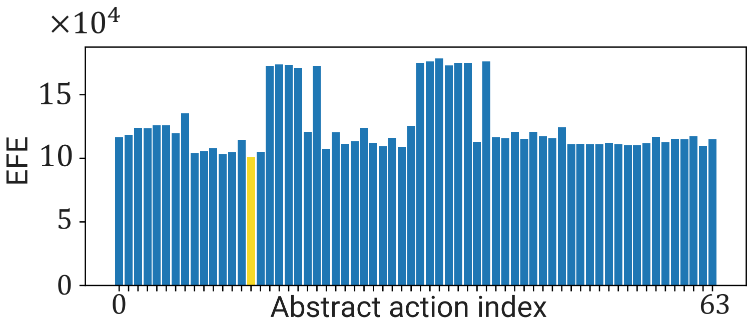
Action execution by the real robot

Actual action generation to reach the goal

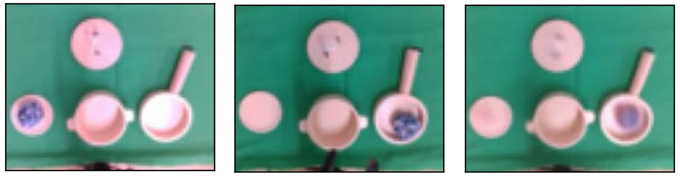
- Choose the abstract action with the minimum EFE
  → Decoded to the actual action and executed on the robot

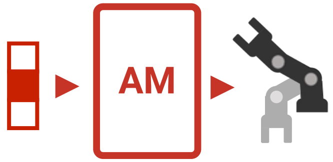
97% reduction in computation time
Planning w/ AWM: 2.4 ms
Planning w/ WM: 71.8 ms





Initial | Goal | Prediction with minimum EFE





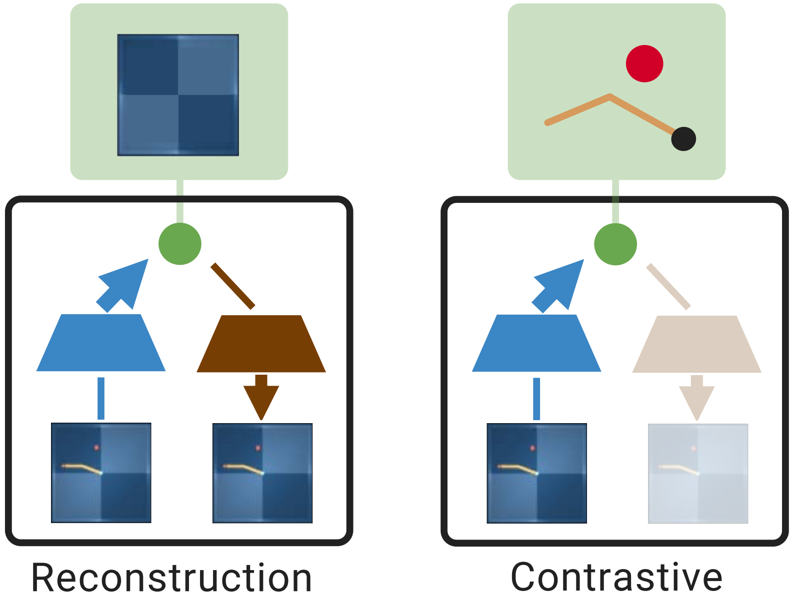
Introducing contrastive learning

[Fujii, Isomura & Murata, *IWAI* 2024]

Key idea

- Features that are easy to extract
  - Reconstruction learning → large structures (e.g., background)
  - Contrastive learning → small objects

→ **The agent's attention may shift**



Combining reconstruction and contrastive learning

- Propose a new upper bound on surprise
  - Original $F = D_{\mathrm{KL}}[q(z_t)||p(z_t|o_t)] - \log p(o_t) \geq -\log p(o_t)$

[Fujii, Isomura & Murata, *IWAI* 2024]

Novel upper bound on surprise

Free energies controlled by α

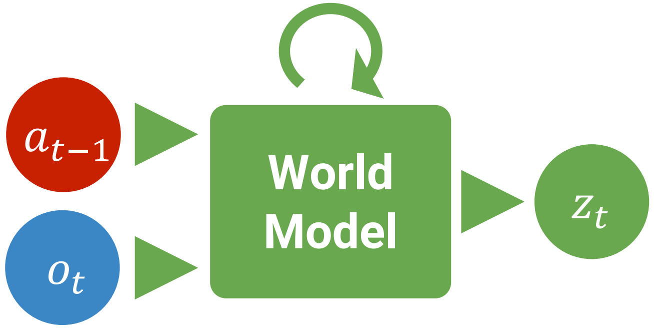
- Upper bound on surprise (and on VFE) at time t: F_α
  - $F_\alpha \triangleq \alpha \cdot D_{\mathrm{KL}}[q(z_t)||p(z_t)] - \mathbb{E}_{q(z_t)}[\log p(o_t|z_t)] - (\alpha - 1) \cdot I_{\mathrm{NCE}}(Z_t; O_t)$
    $\geq \alpha \cdot D_{\mathrm{KL}}[q(z_t)||p(z_t|o_t)] - \log p(o_t) \geq -\log p(o_t)$
- Extension to future time τ: G_α
  - $G_\alpha \triangleq -\mathbb{E}_{q(o_\tau)}\left[D_{\mathrm{KL}}[q(z_\tau|o_\tau,)||q(z_\tau)]\right] - \mathbb{E}_{q(o_\tau)}[\log \tilde{p}(o_\tau)] - (\alpha - 1) \cdot \tilde{I}_{\mathrm{NCE}}(Z_\tau; O_\tau)$
- $\alpha = 1$ → Reconstruction learning only (standard free energy)
- $\alpha \gg 1$→ Contrastive learning only (Contrastive free energy [Mazzaglia+, *NeurIPS* 2022])

Model components

[Fujii, Isomura & Murata, *IWAI* 2024]

World model

- Infer latent state z_t from observation o_t
- Train to minimize F_α
  - $F_\alpha = \alpha \cdot D_{\mathrm{KL}}[q(z_t)||p(z_t)] - \mathbb{E}_{q(z_t)}[\log p(o_t|z_t)] - (\alpha - 1) \cdot I_{\mathrm{NCE}}(Z_t; O_t)$



Action model

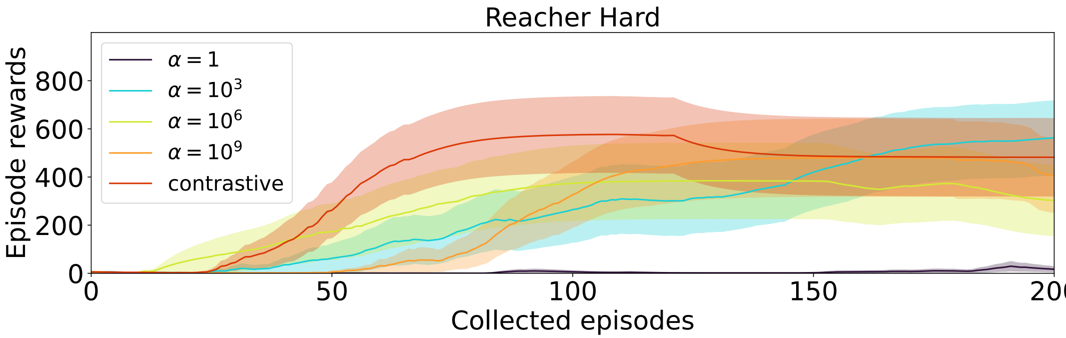
- Generate action a_t from latent state z_t
- Train to minimize G_α
  - $G_\alpha = -\mathbb{E}_{q(o_\tau)}\left[D_{\mathrm{KL}}[q(z_\tau|o_\tau)||q(z_\tau)]\right] - \mathbb{E}_{q(o_\tau)}[\log \tilde{p}(o_\tau)] - (\alpha - 1) \cdot \tilde{I}_{\mathrm{NCE}}(Z_\tau; O_\tau)$

Simulation experiment

[Fujii, Isomura & Murata, *IWAI* 2024]

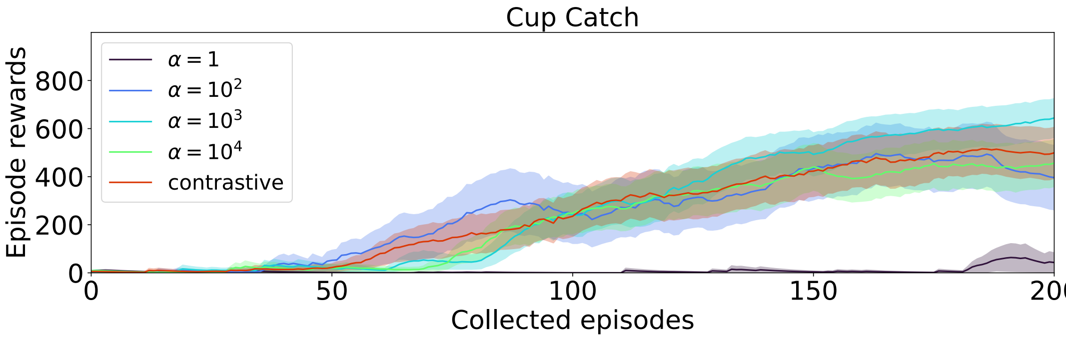
Reacher-hard task

- "Contrastive" achieves the best episode reward the fastest
- Best performance with $\alpha = 10^3$



Ball-in-cup task

- Performance improves in a similar manner
- Best performance with $\alpha = 10^3$

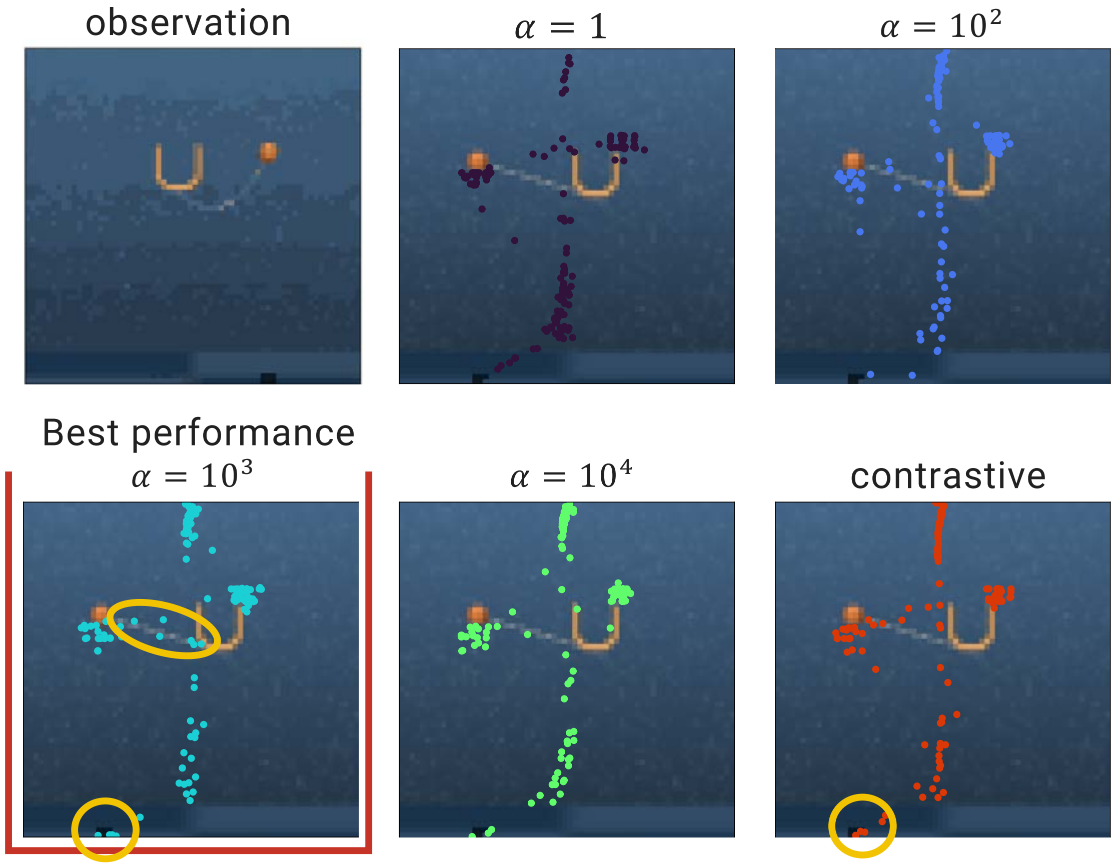


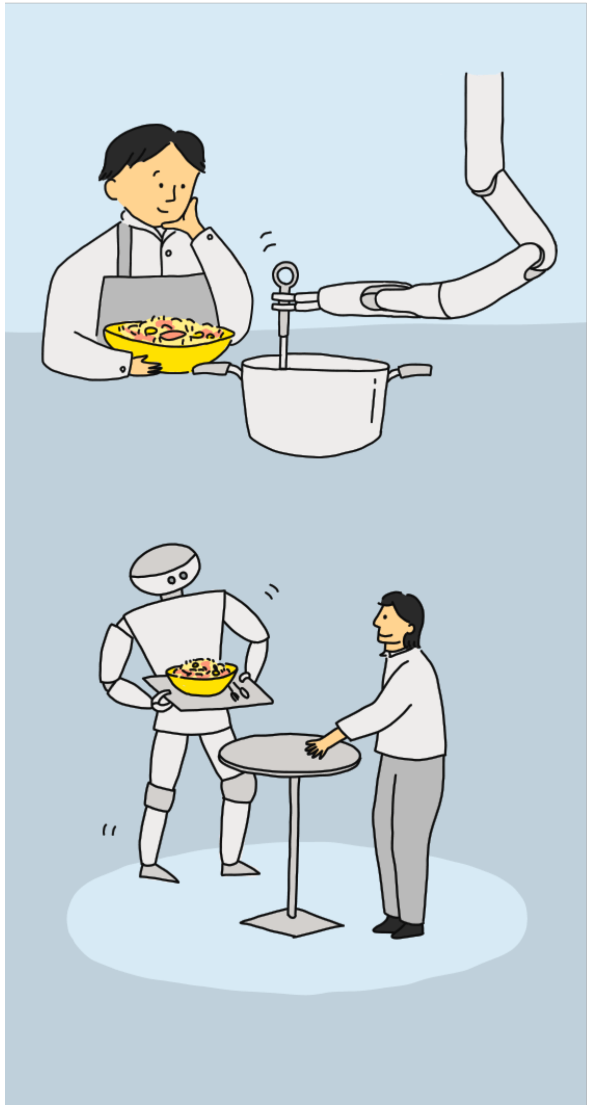
Simulation experiment

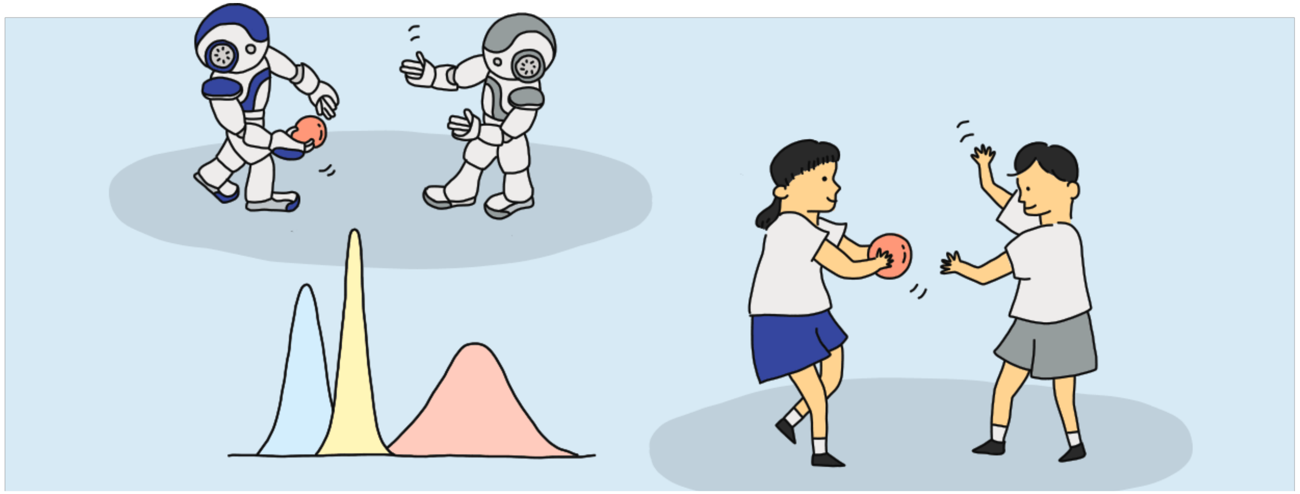
[Fujii, Isomura & Murata, *IWAI* 2024]

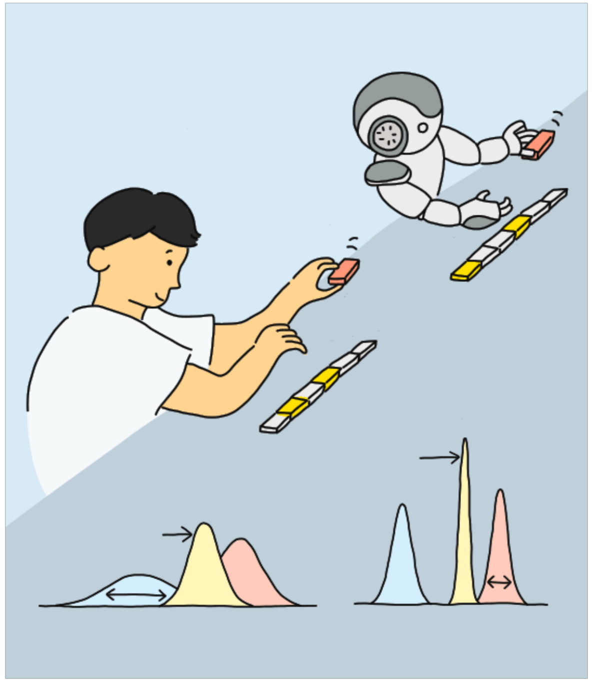
Agent attention in the ball-in-cup task

- $\alpha = 1$ (reconstruction only)
  - Upper and lower parts of the cup and the ball
- $\boldsymbol{\alpha = 10^3}$
  - Upper and lower parts of the cup and the ball
  - Cup, shadow, and **string**
- Contrastive (contrastive only)
  - Upper and lower parts of the cup and the ball
  - Cup and shadow

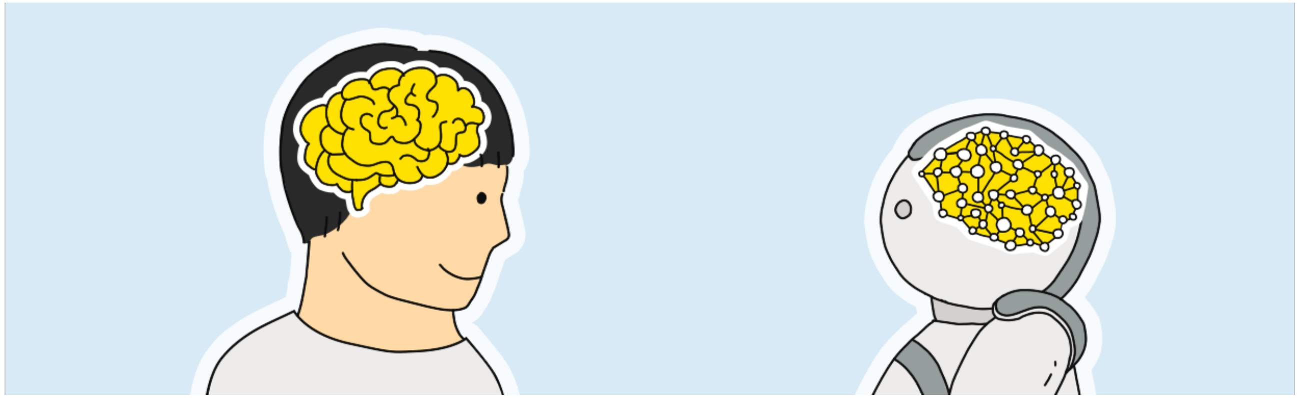


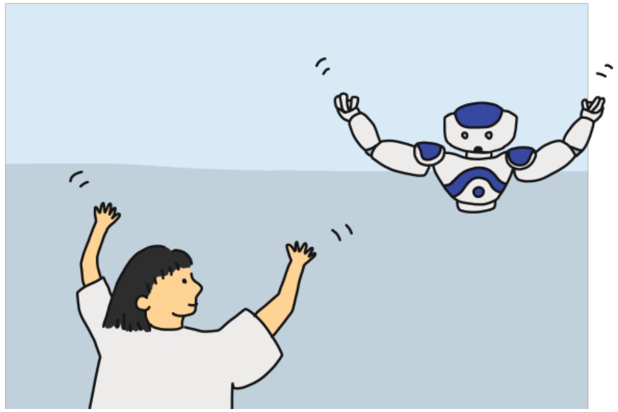






Deep Active Inference for Real-World Robotic Systems





Shingo Murata

Associate Professor @Dept. of EEE, Keio Univ.

https://murata-lab.jp

murata@elec.keio.ac.jp

@keio_crl