Learning Human-Like RL Agents Through Trajectory Optimization With Action Quantization

Jian-Ting Guo[1], Yu-Cheng Chen[1], Ping-Chun Hsieh[1], Kuo-Hao Ho[1], Po-Wei Huang[1], Ti-Rong Wu[2], I-Chen Wu[1,2]

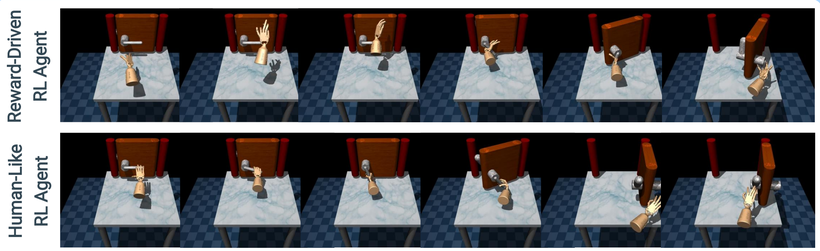
[1]National Yang Ming Chiao Tung University, [2]Academia Sinica

Project page

NEURAL INFORMATION PROCESSING SYSTEMS

We propose MAQ, a human-like RL framework that improves human-likeness while achieving strong performance, and can be easily integrated into various RL algorithms.

Human-Like Reinforcement Learning



- **Reward-driven** agents can successfully open the door, but the **behavior is unnatural and not human-like**.
- In contrast, **Human-Like Reinforcement Learning focuses on both *human-like behavior* and *optimal performance***, but remains underexplored in the RL community.

Human-Like Trajectory Optimization

Trajectory Optimization

The canonical trajectory optimization problem is to find an action sequence that can maximize the total discounted return under episode length T.

$$a^*_{0:T-1} \coloneqq \text{argmax}_{a_{0:T-1} \in \mathcal{A}^T} \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t) \middle| s_0 \sim p_0; a_{0:T-1}\right]$$

Human-like Receding-horizon Control (HRC)

The full-horizon optimization can be intractable for large T, we resort to the receding-horizon control problem, at each step t, finding an action sequence of length H.

$$\text{argmax}_{a_{t:t+H-1}} \mathbb{E}\left[\sum_{i=t}^{t+H-1} \gamma^{i-t} R(s_i, a_i) \middle| s_t \sim p_t; a_{t:t+H-1}\right]$$

To achieve human-likeness, we introduce the *human-like sequence constraint*, the search space of the action sequence is constrained to a human dataset $\mathcal{H}$.

$$a^*_{t:t+H-1} = \text{argmax}_{a_{t:t+H-1} \in \mathcal{H}} \mathbb{E}\left[\sum_{i=t}^{t+H-1} \gamma^{i-t} R(s_i, a_i) \middle| s_t \sim p_t; a_{t:t+H-1}\right]$$

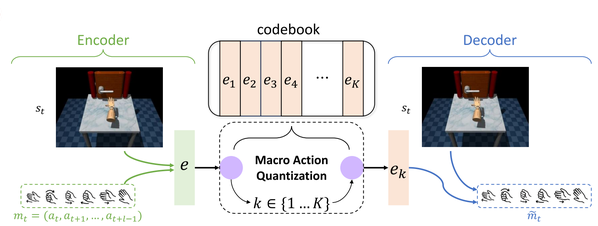
Action-segment Quantization for Efficient HRC

- Searching for all $|\mathcal{H}|$ segments at every step is prohibitively slow.
- Reusing segments requires expensive alignment on matching state s_t exactly.

We propose **segment-level planning, replanning H-step sequences on each state by adopting Macro Action Quantization strategy.**

Macro Action Quantization

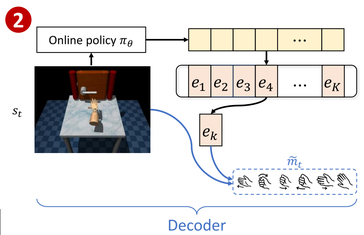
We propose **Macro Action Quantization** (MAQ) that includes two training stages.



❶ Human behavior distillation

- We train a Conditional-VQVAE to distill macro action $m_t = (a_t, a_{t+1}, \dots, a_{t+H-1})$ from human demonstration to learn a discrete codebook.
- **The embeddings represent different human-like macro actions.**

❷ RL with macro actions



- We train an online policy (π_θ) that acts in the learned discrete code space by selecting codebook indices.
- **The action space is transformed from primitive actions into macro actions.**
- To optimize π_θ, maximize the expected reward: $m^*_t = \text{argmax}_{m_t \in \mathcal{H}} \mathbb{E}\left[R(s_t, m_t) \middle| s_t; m_t\right]$

Trajectory Similarity Evaluation

We train three RL algorithms (w/ and w/o MAQ) on four Adroit tasks.

- MAQ-based agents **outperform all RL agents w/o MAQ in human-likeness**.
- MAQ-based agents also **achieve a comparable success rate**.

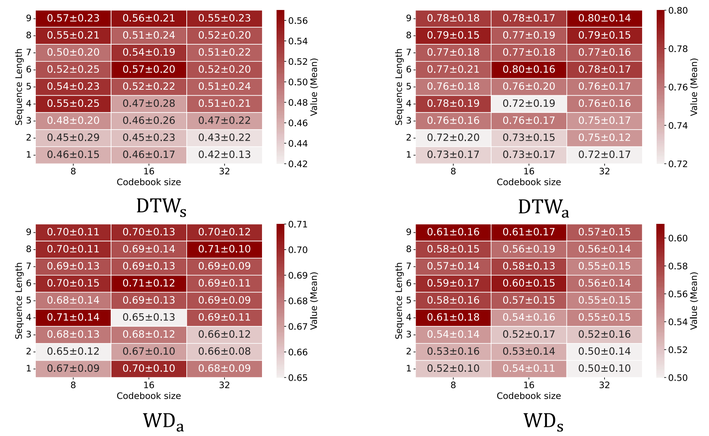
Tasks		BC	IQL	MAQ+IQL	SAC	MAQ+SAC	RLPD	MAQ+RLPD
Door	DTW_s(↑)	0.18 ± 0.09	0.43 ± 0.06	**0.84 ± 0.06**	-0.39 ± 0.10	**0.80 ± 0.08**	-0.06 ± 0.04	**0.76 ± 0.04**
	DTW_a(↑)	0.42 ± 0.13	0.61 ± 0.04	**0.95 ± 0.01**	-0.25 ± 0.04	**0.91 ± 0.03**	0.28 ± 0.08	**0.91 ± 0.05**
	WD_s(↑)	0.32 ± 0.05	0.48 ± 0.05	**0.75 ± 0.05**	-0.28 ± 0.02	**0.71 ± 0.08**	-0.14 ± 0.04	**0.71 ± 0.03**
	WD_a(↑)	0.41 ± 0.08	0.50 ± 0.02	**0.81 ± 0.03**	-0.15 ± 0.02	**0.77 ± 0.07**	0.10 ± 0.02	**0.76 ± 0.03**
	Success(↑)	0.02 ± 0.01	0.16 ± 0.06	**0.93 ± 0.04**	0.43 ± 0.23	**0.56 ± 0.50**	**0.96 ± 0.07**	0.93 ± 0.05
Hammer	DTW_s(↑)	-0.16 ± 0.07	-0.14 ± 0.34	**0.64 ± 0.17**	-1.10 ± 0.35	**0.61 ± 0.21**	-0.03 ± 0.14	**0.68 ± 0.17**
	DTW_a(↑)	0.47 ± 0.02	0.45 ± 0.20	**0.92 ± 0.06**	-0.33 ± 0.11	**0.91 ± 0.10**	0.37 ± 0.08	**0.94 ± 0.07**
	WD_s(↑)	0.11 ± 0.06	0.12 ± 0.13	**0.75 ± 0.03**	-0.44 ± 0.09	**0.64 ± 0.12**	-0.03 ± 0.08	**0.76 ± 0.04**
	WD_a(↑)	0.30 ± 0.03	0.30 ± 0.10	**0.84 ± 0.02**	-0.19 ± 0.04	**0.78 ± 0.11**	0.20 ± 0.03	**0.85 ± 0.03**
	Success(↑)	0.00 ± 0.00	**0.01 ± 0.01**	0.00 ± 0.00	**0.01 ± 0.01**	0.00 ± 0.00	**1.00 ± 0.00**	0.56 ± 0.37
Pen	DTW_s(↑)	0.53 ± 0.13	0.34 ± 0.09	**0.55 ± 0.17**	0.06 ± 0.20	**0.58 ± 0.17**	0.48 ± 0.24	**0.54 ± 0.18**
	DTW_a(↑)	0.58 ± 0.05	0.51 ± 0.05	**0.58 ± 0.09**	-0.34 ± 0.16	**0.58 ± 0.11**	0.40 ± 0.17	**0.59 ± 0.13**
	WD_s(↑)	0.59 ± 0.12	0.54 ± 0.11	**0.59 ± 0.13**	0.29 ± 0.10	**0.61 ± 0.12**	0.49 ± 0.08	**0.59 ± 0.12**
	WD_a(↑)	0.65 ± 0.14	0.63 ± 0.12	**0.66 ± 0.15**	0.22 ± 0.15	**0.67 ± 0.14**	0.44 ± 0.12	**0.66 ± 0.14**
	Success(↑)	0.40 ± 0.03	0.40 ± 0.05	**0.42 ± 0.07**	0.32 ± 0.09	**0.41 ± 0.01**	**0.62 ± 0.09**	0.42 ± 0.05
Relocate	DTW_s(↑)	0.09 ± 0.14	0.20 ± 0.20	**0.52 ± 0.06**	-0.55 ± 0.20	**0.25 ± 0.19**	0.03 ± 0.13	**0.27 ± 0.14**
	DTW_a(↑)	0.47 ± 0.15	0.51 ± 0.11	**0.82 ± 0.01**	-0.10 ± 0.16	**0.66 ± 0.09**	0.32 ± 0.13	**0.69 ± 0.10**
	WD_s(↑)	0.27 ± 0.15	0.36 ± 0.06	**0.47 ± 0.07**	-0.22 ± 0.06	**0.40 ± 0.07**	0.02 ± 0.07	**0.38 ± 0.09**
	WD_a(↑)	0.45 ± 0.12	0.50 ± 0.04	**0.65 ± 0.03**	-0.05 ± 0.03	**0.61 ± 0.03**	0.20 ± 0.03	**0.55 ± 0.08**
	Success(↑)	0.01 ± 0.02	0.00 ± 0.00	**0.20 ± 0.10**	0.00 ± 0.00	**0.14 ± 0.07**	0.14 ± 0.03	**0.17 ± 0.10**
Average	DTW_s(↑)	0.16 ± 0.29	0.21 ± 0.25	**0.63 ± 0.14**	-0.49 ± 0.48	**0.56 ± 0.23**	0.10 ± 0.25	**0.56 ± 0.21**
	DTW_a(↑)	0.49 ± 0.07	0.52 ± 0.06	**0.82 ± 0.17**	-0.26 ± 0.11	**0.76 ± 0.17**	0.34 ± 0.05	**0.78 ± 0.17**
	WD_s(↑)	0.32 ± 0.20	0.38 ± 0.18	**0.64 ± 0.14**	-0.17 ± 0.32	**0.59 ± 0.14**	0.08 ± 0.28	**0.61 ± 0.17**
	WD_a(↑)	0.45 ± 0.15	0.48 ± 0.14	**0.74 ± 0.10**	-0.04 ± 0.18	**0.71 ± 0.08**	0.24 ± 0.15	**0.70 ± 0.13**
	Success(↑)	0.11 ± 0.19	0.14 ± 0.19	**0.39 ± 0.40**	0.19 ± 0.22	**0.28 ± 0.25**	**0.68 ± 0.40**	0.52 ± 0.32

* DTW: Dynamic Time Warping; WD: Wasserstein Distance; $_s$: state distance; $_a$: action distance

Impact of Macro Action Length and Codebook Size

We analyze how macro action length and codebook size affect scores on MAQ+RLPD.

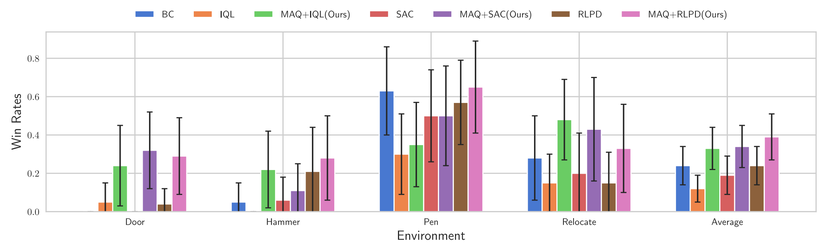
- **Macro action length positively correlates with trajectory similarity scores.**
- **Codebook size shows no effect.**



Human Evaluation Study

Turing Test: Evaluators choose between two videos (human-agent).

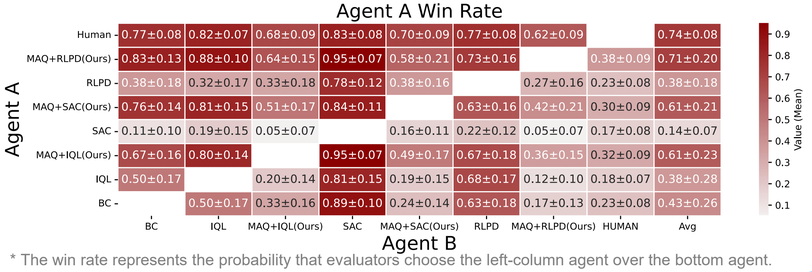
- MAQ+RLPD achieves the highest win rate.
- Reward-driven RL agents, such as RLPD and SAC, achieve nearly 0% win rates.



* The win rate represents the probability that evaluators choose the corresponding agent.

Ranking Test: Evaluators choose between two videos (human-agent or agent-agent).

- MAQ+RLPD achieves similar win rates (71% vs. 74%) to the human demonstration, suggesting that **it can fool evaluators into believing its behavior is human-like**.
- Reward-driven agents are easily recognized as computers by evaluators.



* The win rate represents the probability that evaluators choose the left-column agent over the bottom agent.