virtualcharacters.github.io/links/ALA2021

Data-Driven Reinforcement Learning for Virtual Character Animation Control

Vihanga Gamage, Cathy Ennis, Robert Ross

School of Computer Science, Technological University Dublin, Dublin, Ireland

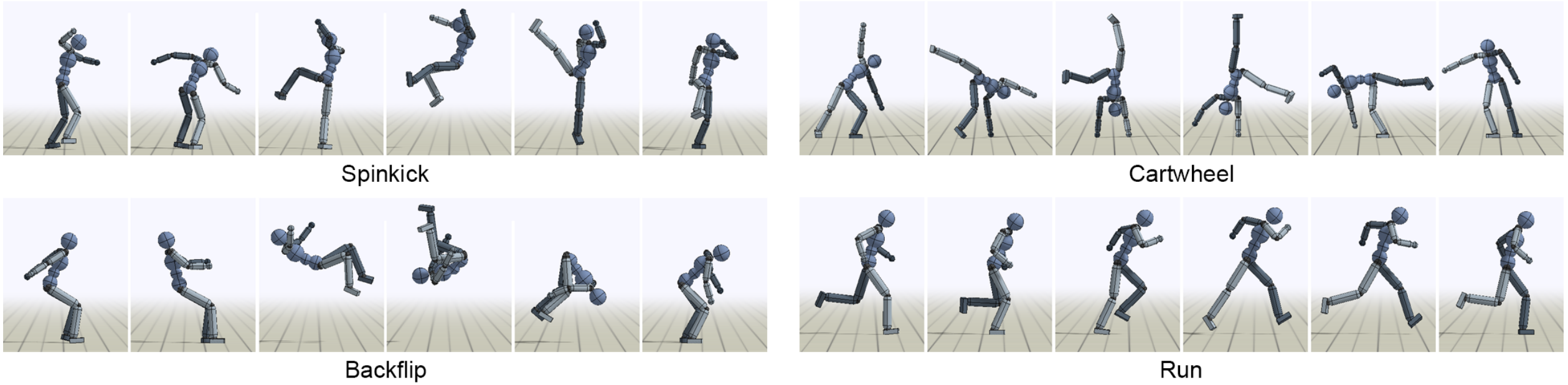
Data-driven character animation

Neural-network based approaches have been a popular choice for data-driven character animation. While supervised learning can be used for training agents to generate varied animation portraying within a range of specified behaviours, due to their heavy reliance on the training dataset, these approaches offer limited flexibility, and can lead to unpredictable outputs in conditions different to the training set.

Alex Klein, Zerrin Yumak, Arjen Beij, and A. Frank van der Stappen. 2019. Data-Driven Gaze Animation Using Recurrent Neural Networks. In Proceedings of the 12th Annual International Conference on Motion, Interaction, and Games (Newcastle upon Tyne, United Kingdom) (MIG '19). Association for Computing Machinery, New York, NY, USA, Article 4, 11 pages. https://doi.org/10.1145/3359566.3360054

Data-driven character animation

Recent work explore the use of reinforcement learning, with physics-based simulation being leveraged to implement character animation agents, such as DeepMimic. These methods rely heavily on interaction with physical surfaces and objects, as the feedback signals from the physics engine are required for agents to learn and function. As a result, they can not be applied for social, interactive gestures.



Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions on Graphics (TOG) 37, 4 (2018), 1–14.

Our Contributions

We introduce RLAnimate, a novel data-driven deep reinforcement learning approach for character animation, that allows for trained agents to portray human-like behaviour.

Novel modelling framework for character animation

Latent dynamics model for human-like animation

Training algorithm for RL animation agents informed by motion data

Mathematical Framework for Character Animation

RL problems are typically described as a Markov Decision Process (MDP).

set of states : $s \in S$ set of actions : $a \in A$ reward term : $R_a(s, s')$

state transition probability function: $P(s_{t+1} = s' | s_t = s, a_t = a)$

policy : $\pi(s) \rightarrow a$

Markovian property : $P(s_{t+1} | s_t) = P(s_{t+1} | s_1, \ldots, s_t)$

Mathematical Framework for Character Animation

However, when we consider an animation agent portraying human-like behaviour:

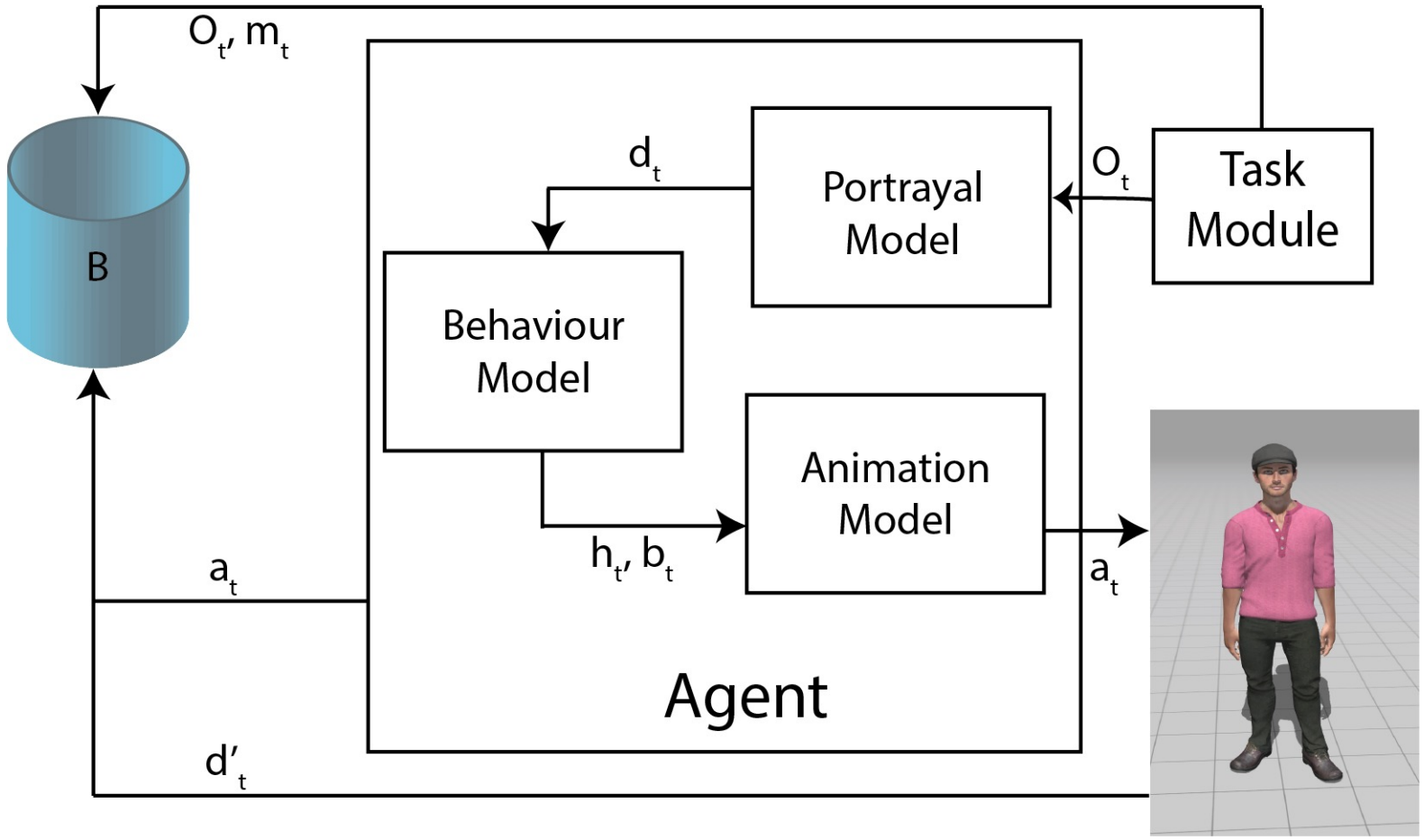
- All actions in the animation trajectory need to be optimal
- Observation and action spaces are complex
- A straightforward reward function can not be mathematically articulated

We propose the use of a model-based RL:

- The state observation is split into objective and description signals.
- Agents are trained to maximize the idealness I of the animation at each time step.

Output animation : $\{a_t\}_{t=0}^{n}$ State observation : $s_t = o_t \cdot d_t$ Overall training objective: $E\left[\sum_{t=0}^{n} I_t\left(a_0, ..., a_t, o_t, d_t\right)\right]$

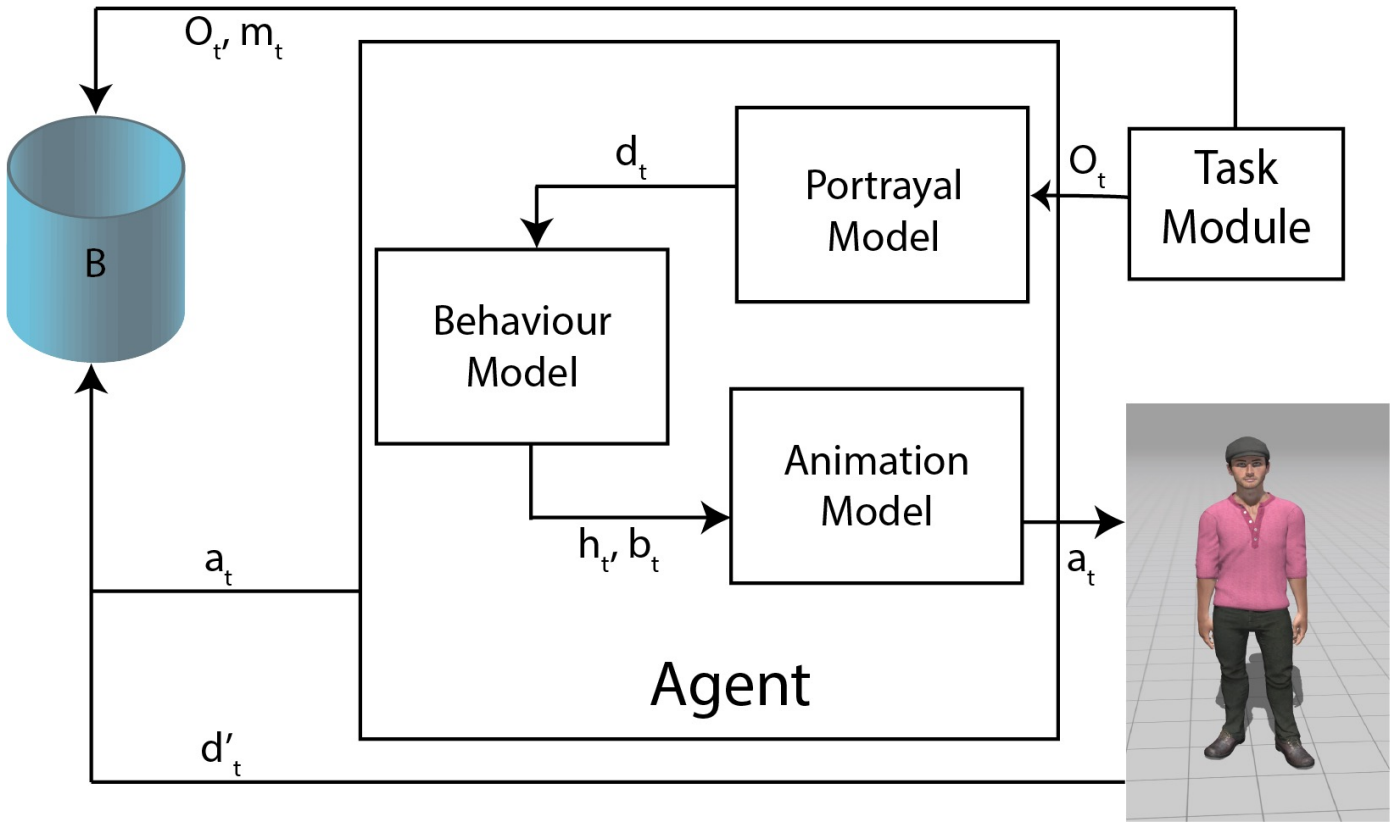
RLAnimate agents - overview



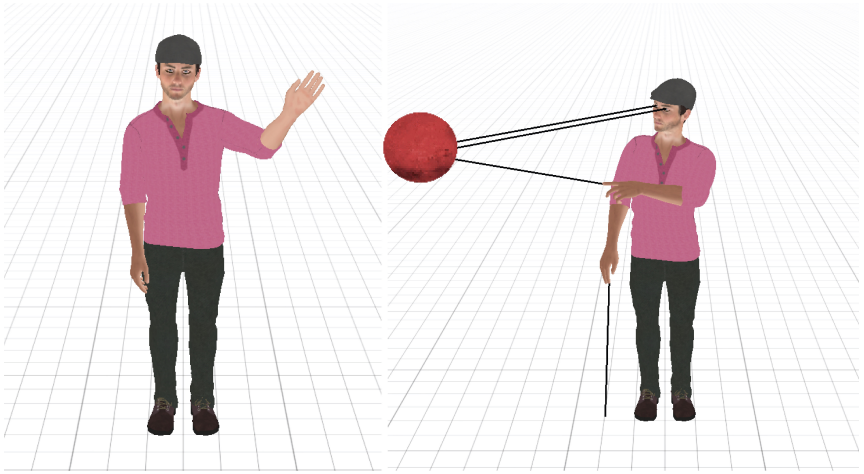
- Agents are trained by generating episode rollouts imitating motion capture clips
- The environment provides input to the agent via objective signals generated by the task module.
- The agent is used to generate an animation which is applied to the character.
- The objective, animation, real description and ideal motion per motion clip are saved to the sample buffer.
- Samples are drawn at random from the buffer to update the model parameters.

RLAnimate agents – Tasks and Objectives

Agent Overview



Task Overview

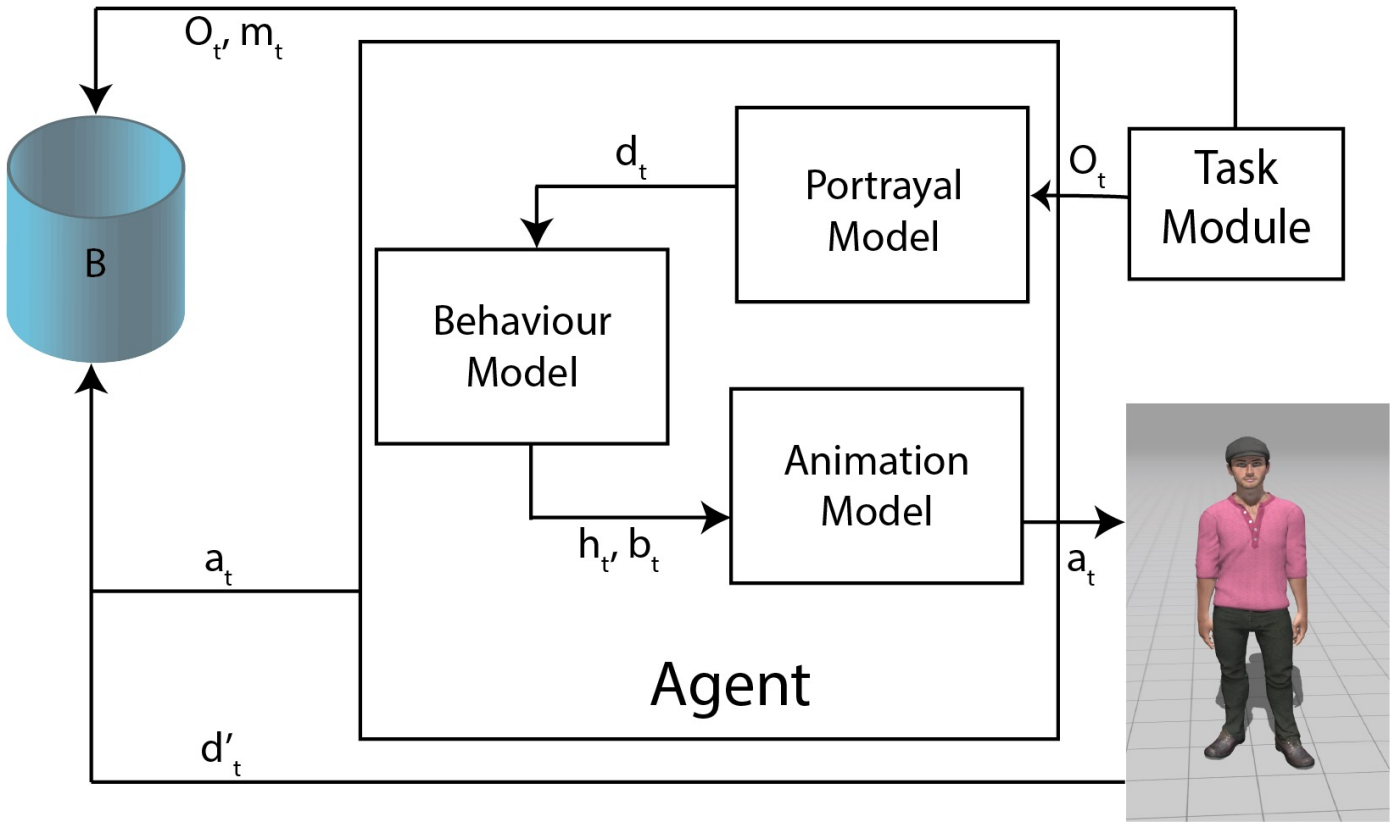


Objective Space

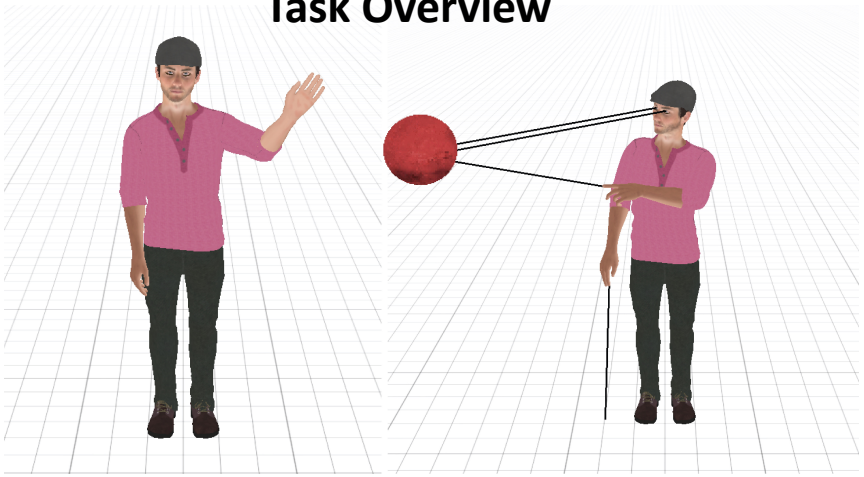
	Type	Arm		Attributes			Time
Point	1	Left	Right	Unit vector to target			t/N
		0	1	x	y	z	
Wave	2	Left	Right	Exaggeration			t/N
		0	1	0 - 1	0 - 1	0 - 1	

RLAnimate agents – Descriptions

Agent Overview



Task Overview

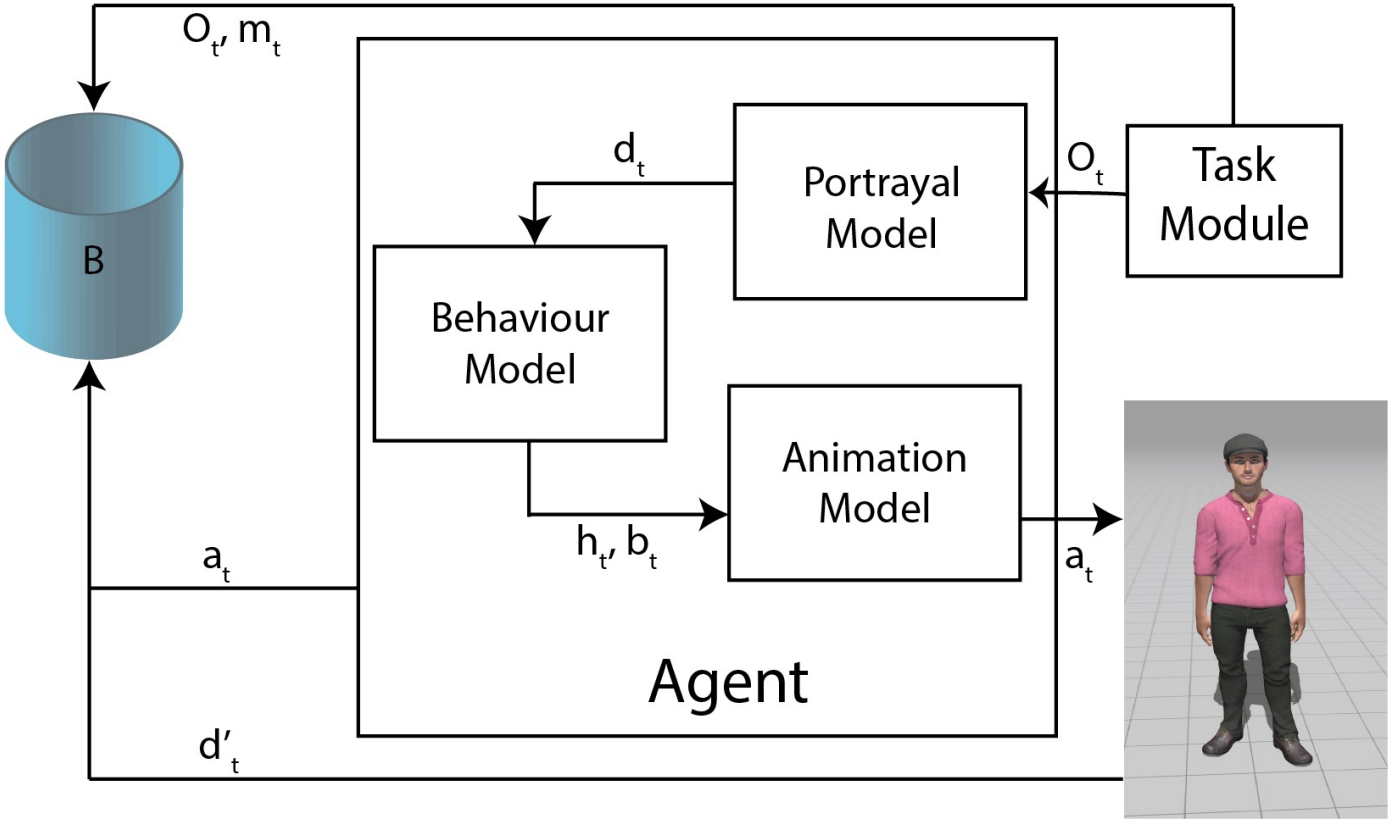


Description Space

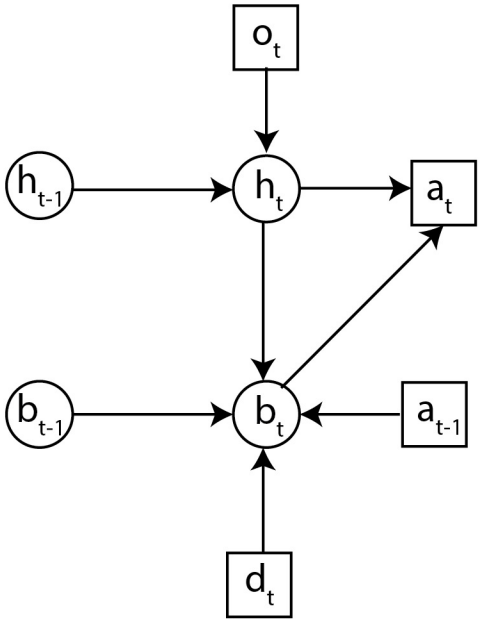
Attribute	Articles	Size
Effector Vector	left & right eyes, and index fingers.	12
Joint Position	left & right collars, shoulders, elbows, wrists, and index finger bases.	30

RLAnimate agents – Dynamics learnt via the behaviour model

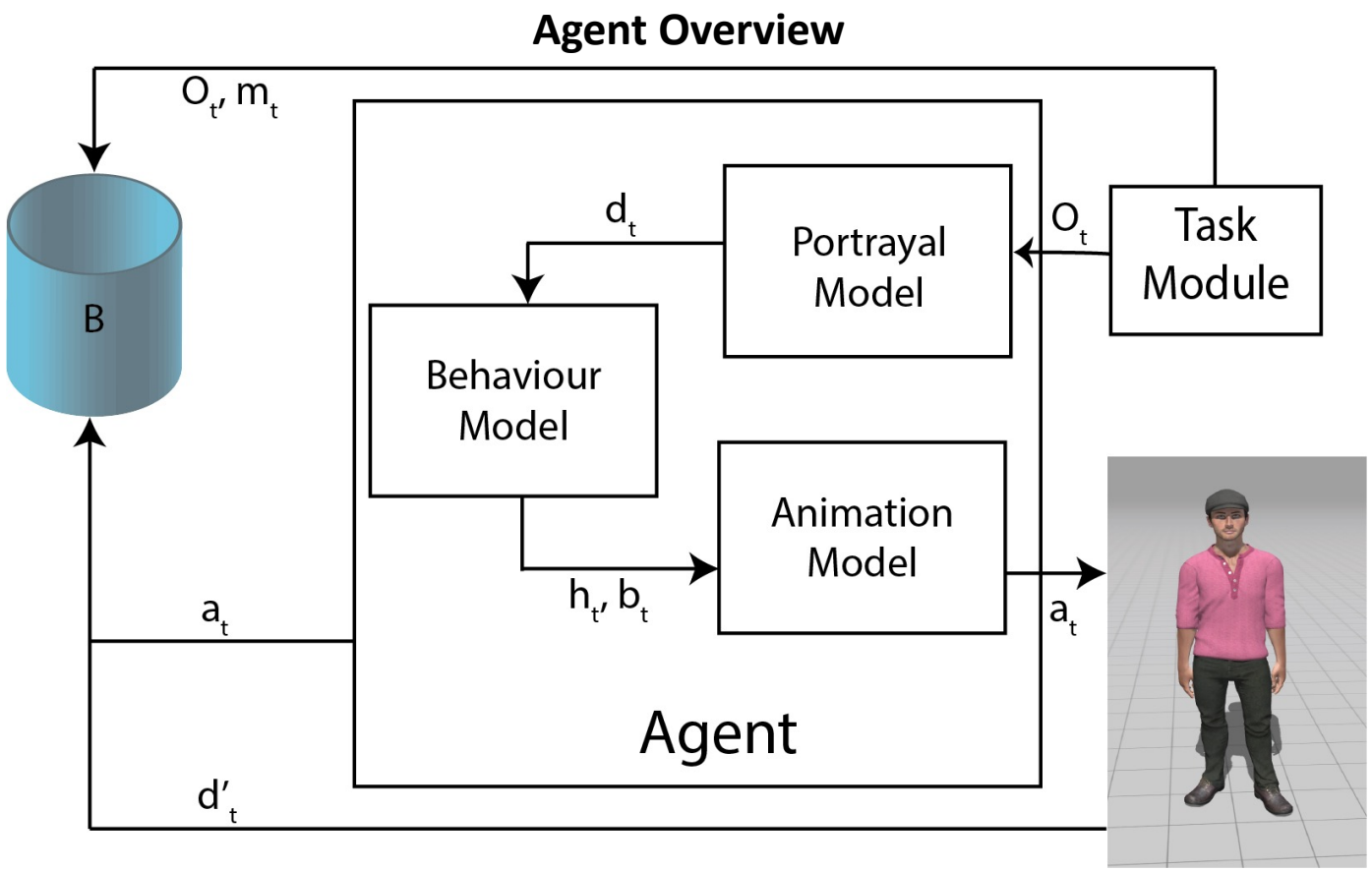
Agent Overview



Learned Latent Dynamics



RLAnimate agents – Animation output and episode rollouts



- Animation model – $p(a_t|h_t, b_t)$
- Animation space is parameterized as a beta distribution
- After an animation is applied, the environment records the real description
- The objective, animation, real description and ideal animation per motion clip is saved to the sample buffer

$$B \rightarrow B \cup \left\{ (o_t, a_t, d'_t, m_t)_{t=1}^{F} \right\}$$

Training Agents

Overall training objective :

$$L(\theta) \triangleq E\left[\sum_{t=0}^{n} I_t\left(a_0, \ldots, a_t, o_t, d_t\right)\right] \triangleq L1(\theta) + L2(\theta) + L3(\theta)$$

Description construction loss:

$$L1(\theta) = mse\left[d_t' - E\left[p\left(d_t | o_t\right)\right]\right]$$

KL divergence loss:

$$L2(\theta) = KL\left[p\left(b_t | h_{t-1}, o_t, b_{t-1}, a_{t-1}\right) \parallel q\left(b_t | d_t, a_{t-1}\right)\right]$$

Animation loss:

$$L3(\theta) = \begin{cases} \frac{1}{2}\left(M_t - f\left(h_t, b_t\right)^2\right) & \text{for } \left|M_t - f\left(h_t, b_t\right)\right| \leq \delta \\ \delta\left|M_t - f\left(h_t, b_t\right)\right| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Experiments

- We obtained 50 motion clips each for pointing and waving behaviours.
- The pointing clips covered the use of each arm, a variety of target positions, as well as variations to timing.
- The waving clips included those portraying different degrees of exaggeration for each arm.
- To evaluate agents, we held back from training a test of 10 clips.

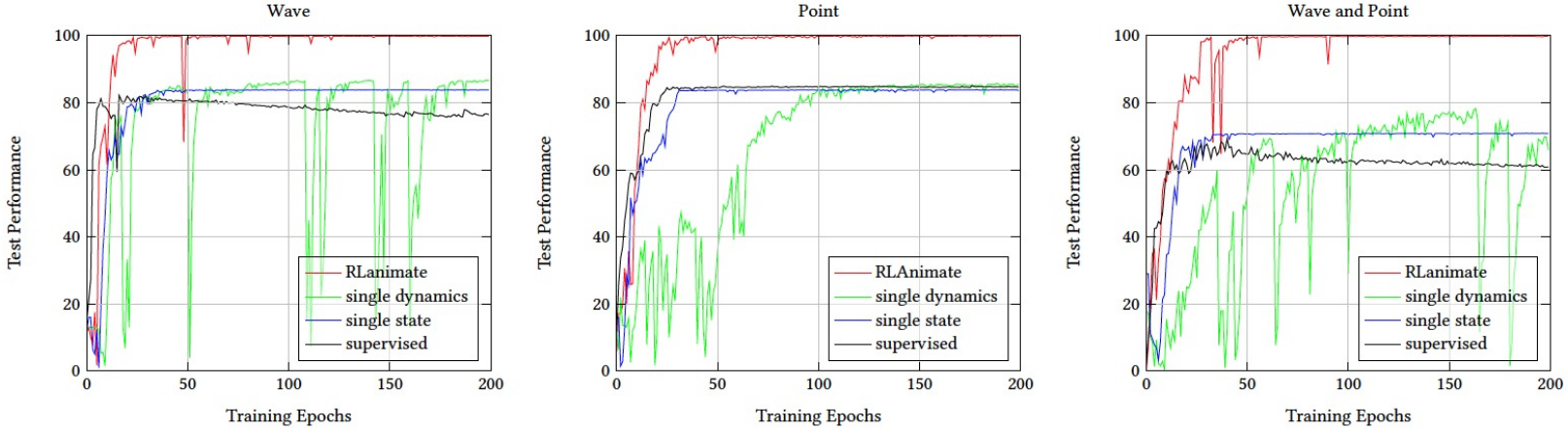
- Evaluation score out of 100 obtained by:

$$100 - \frac{\text{total error}}{\text{frames}}$$

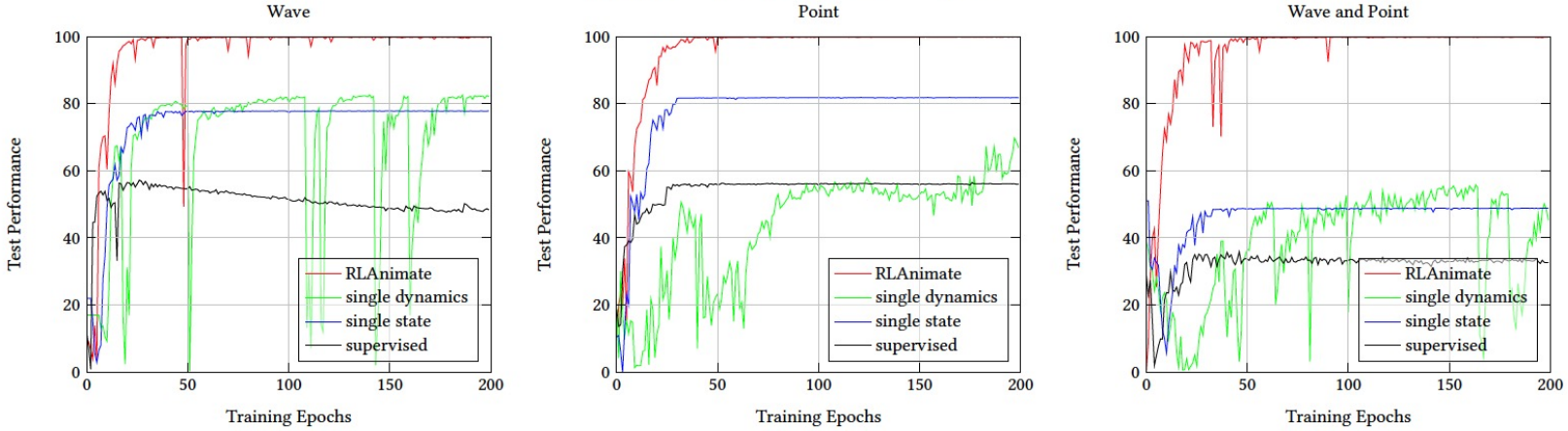
$$error_t = \sum_{j=0}^{J} \left(\left| p_x^j - p_x^{j\prime} \right| + \left| p_y^j - p_y^{j\prime} \right| + \left| p_z^j - p_z^{j\prime} \right| \right)$$

$$\text{total error} = \sum_{t=0}^{T} error_t + max\left\{0, \log_{1.01} error\right\}$$

Results – Performance of agents measured over training epochs



(a) Imitating clips from training set



(b) Imitating clips excluded from training set

Output Sequences

To view output sequences demonstrating our work, and comparing the differences in performance and output quality of the agents and controls evaluated, please visit:

virtualcharacters.github.io/links/ALA2021

Summary

- We present RLAnimate, an approach for model-based animation control capable of portraying human-like behaviours.

- RLAnimate agents learn a model to self-generate descriptions from the objective signal, and learn an advanced dynamics model to that maintains latent representations that can be used to obtain an animation sequence portraying natural human behaviour.

- Our evaluation shows that RLAnimate agents are able to learn to portray different behaviours, using 0.5M x fewer sample episodes generated relative to physics-based model-free RL methods.

- In our future work, we plan to ascertain the impact of small imperfections that can affect human perception of output animation, and explore how agents can be influenced to avoid those pitfalls.

- Also, we plan to examine how RLAnimate can be applied to generate animation portraying a wider range of behaviours, in particular more complex portrayal of beat gestures and other complex behaviours required to interact with a user.