Enhancing Tactile-based Reinforcement Learning for Robotic Control

Elle Miller[1], Trevor McInroe[1], David Abel[1,2], Oisin Mac Aodha[1], and Sethu Vijayakumar[1]

***Abstract*— Effectively combining tactile sensing and reinforcement learning (RL) creates powerful new pathways for sophisticated robot manipulation. However, tactile information is not always fully exploited by neural network-based approaches in deep RL due to its unique characteristics (e.g. sparsity). Departing from conventional reliance on idealised state representations, we present a new approach to strengthen the performance of sensory-driven agents for complex manipulation tasks. We provide a novel application and analysis of tailored reconstruction and multi-step dynamics objectives that help the agent more effectively leverage its tactile observations. We find that dynamics-based objectives unlock higher-performing agents that are able to predict future contacts with high precision. Experimental results show the efficacy of our approach through a simulated robotic agent on three complex control tasks with touch and proprioception alone. Project page with videos: elle-miller.github.io/tactile_rl**

I. INTRODUCTION

Imagine a humanoid robot gently lifting your grandmother out of her bed and into her wheelchair with the same delicacy as a human caregiver, or a person with severe motor impairments using a re-enabling robotic arm to brush their teeth. For this future to safely materialise, we posit that robots must evolve beyond only *seeing* their environment - they must possess some capacity to *feel* it. Reinforcement learning (RL) is a primary candidate for enabling tactile-based manipulation, allowing robots to learn complex and optimal motions directly from rich sensory data. However, despite interest in combining tactile sensing with RL over the last decade [41, 22, 27, 43, 47, 36], key questions regarding optimal sensor type, sensor placement, and information representation remain unanswered. Most importantly, the very necessity of tactile sensing for manipulation is ambiguous. For instance, [37, 32] achieve impressive in-hand manipulation using proprioceptive history alone, with [31] claiming that binary contacts do not add value because this information is implicitly contained in proprioceptive history.

Despite human proficiency in "blind" manipulation (requiring no visual or privileged information at any point), we have yet to see on-par capabilities emerge in RL agents. To our knowledge, the most advanced dexterous blind RL demos are object rotation [37, 50] and half-rotation of Baoding balls [46]. [46] is the only work to study fully blind Baoding ball rotation, and learn by explicitly estimating the ball poses. The fastest robot policies in both simulation and real-world experiments achieve 3 complete rotations in 10 seconds [52], whereas humans can obtain ~ 13.

We hypothesise that a capable blind agent hasn't been realised because deep RL struggles to extract a useful representation from raw tactile data for robotic control. This is due to the demanding task of simultaneously learning the observation representation, policy, and value function from a scalar reward. To alleviate this learning burden, self-supervised learning (SSL) can provide an auxiliary signal to help agents convert complex observations into useful representations [17]. SSL objectives have been highly successful in improving the performance of pixel-based RL agents [49, 25, 48], and recent works have attempted to apply these techniques to the tactile modality (e.g. pixel reconstruction [36], augmentation [11]). However, we suggest that these objectives do not encourage the encoding of temporal features that dictate state transitions such as object velocity, mass, or friction, which can be important for complex control. We aim to *develop general-purpose self-supervised learning methodologies that effectively leverage tactile observations for robotic control tasks in RL.* We desire the following characteristics: (a) no global scene or visual information (b) no privileged information (c) works across a diverse range of contact dynamics (d) low sim2real gap. Inspired by the low-cost setups in [50, 52], we study learning with binary tactile activations to avoid transfer difficulties that can come with continuous measurements [50]. Our main contributions are:

- **New findings on the need for tactile sensing:** We show that sparse binary contacts offer performance gains over control errors and are sufficient for superhuman performance in simulation.
- **New SSL losses:** We propose and analyse four SSL objectives (tactile reconstruction, full reconstruction, tactile dynamics, full dynamics) that enable tactile-based robotic agents to outperform policy-gradient methods, finding dynamics-based losses to be most useful.
- **Super-human blind agents:** Compared to RL-only agents, our best self-supervised agents on average find an object 36% faster (1.4 vs 1.9 seconds), bounce a ball 8 more times in 10 seconds (79 vs 71), and complete 17 in hand ball rotations compared to 5 in 10 seconds.
- **Tactile manipulation benchmark:** We release our three Isaac Lab environments as a benchmark called `RoTO: Robot Tactile Olympiad` to inspire progress in tactile-based manipulation.

elle.miller@ed.ac.uk
[1]University of Edinburgh, UK
[2]Google DeepMind

*This work was accepted at NeurIPS 2025, and is presented in 4-page format here for the Dexterous Humanoid Manipulation workshop at the 2025 Humanoids conference.

II. Method

A. Problem setting

We study a partially-observable Markov decision process (POMDP) [18] parameterized by a tuple $\langle \mathscr{S}, \mathbb{O}, \mathbb{F}, \mathscr{A}, \mathscr{T}, R, \gamma \rangle$, where $\mathscr{S}$ is the state space, $\mathbb{O} = \{\mathscr{O}^i\}_{i=1}^N$ is a set of N independent observation spaces, $\mathbb{F} = \{(f_i, \mathscr{O}_i)\}_{i=1}^N$ is a set of N unknown function-codomain pairs where each function $f_i : \mathscr{S} \to \mathscr{O}_i$ maps elements of the state space to elements of its paired observation space, $\mathscr{A}$ is the action space, $\mathscr{T} : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to [0,1]$ is the transition kernel, $R : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0,1)$ is a discount factor.An agent interacts with the POMDP via a policy $\pi : \mathbb{O} \to \Delta(\mathscr{A})$. The agent's goal is to find the optimal policy that maximises discounted returns $\pi^\star = \arg\max_\pi \sum_{i=1}^{\infty} \gamma^i R(s_t, a_t)$. We consider the class of problems where an agent only has access to the set of $N = 2$ observation spaces $\mathbb{O} = \{\mathscr{O}^{prop}, \mathscr{O}^{tact}\}$ (proprioceptive and tactile). Note that we include the last taken robot action in our definition of proprioception, from which control errors can be deduced. Our RL setup (Figure 1, top) is as follows. At each timestep t the environment returns proprioception and tactile measurements, which are concatenated to form the observation o_t. This is inserted into a circular buffer, which stores the last k observations to form the timestep's state s_t. The state is passed to the agent, which is comprised of an encoder e, policy π and value function v. The encoder is a large MLP that learns the state representation $z_t = e(s_t)$, which the shallower policy and value function are conditioned on. The self-supervised loss is added to the policy, value, and entropy loss and backpropagated in a single pass. The auxiliary loss optimises the encoder e and its own task-specific networks.

B. Reconstruction-based SSL objectives

We hypothesise that a tactile-based RL agent learning with gradient-based optimisation may prematurely converge to features that provide immediate reward correlations, such as proprioceptive histories. Thus, we propose to tailor the typical input reconstruction objective to instead only decode the tactile state $\hat{s}_t^{tact}$ from the multimodal representation z_t (Figure 1, middle).. We formulate **tactile reconstruction (TR)** as a binary classification problem using a binary cross-entropy with logits loss. A positive weighting p_c is used to penalise instances where there are positive elements in s_t^{tact} that the prediction $\hat{s}_t^{tact}$ misses, computed per minibatch.

$$\mathscr{L}_{\text{TR}}(\hat{s}_t^{tact}, s_t^{tact}) = -(p_c s_t^{tact} \cdot \log(\hat{s}_t^{tact}) + (1 - s_t^{tact}) \cdot \log(1 - \hat{s}_t^{tact})) \quad (1)$$

To compare the effect of only decoding the tactile state, we also analyse **full reconstruction (FR)** with an MSE loss for the proprioceptive state.

$$\mathscr{L}_{\text{FR}} = \mathscr{L}_{\text{TR}} + MSE(\hat{s}_t^{prop}, s_t^{prop}) \quad (2)$$

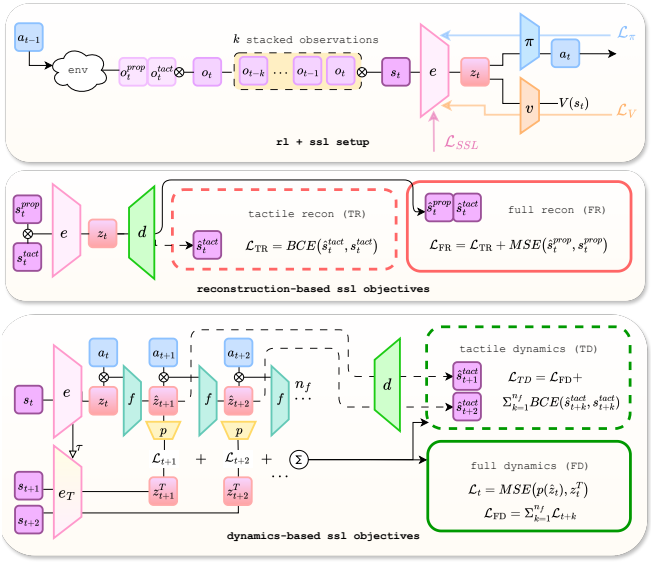


Fig. 1: **Top:** Our RL with self-supervision setup. The SSL loss optimises the encoder e to help learn the state representation z_t which the policy and value function are conditioned on. **Middle:** The reconstruction objectives encourage the encoder e to preserve the tactile state in the representation by implementing reconstruction as binary classification. **Bottom**: The multi-step forward dynamics objectives optimise the encoder e to extract information from the state s_t that will aid in predicting representations n_f timesteps into the future.

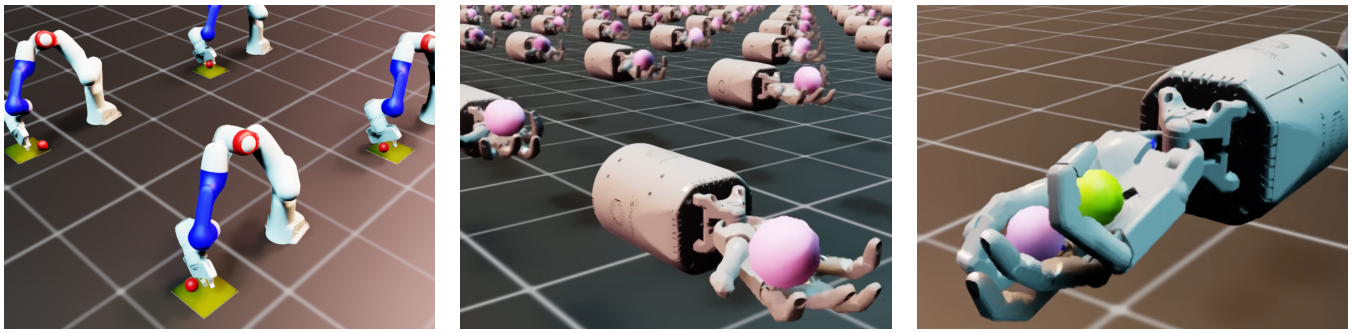
C. Dynamics-based SSL objectives

We propose that a multi-step forward dynamics objective could distill tactile features into underlying components that dictate state transitions, by encouraging the encoder to extract information from the state s_t that will aid in accurately predicting n_f timesteps into the future. We implement this as follows: a trajectory $\tau = (s_t, a_t, ...s_{t+n_f}, a_{t+n_f})$ of $n_f + 1$ state-action pairs are sampled from a memory, and any sequence containing episode-terminating transitions are filtered out. Given the first latent state $z_t = e(s_t)$ and action a_t, the forward model f predicts the next latent state $\hat{z}_{t+1} = f(z_t, a_t)$, and this prediction is used as the input latent state for the next prediction in an autoregressive fashion (Figure 1, bottom). The dynamics loss $\mathscr{L}_{\text{dyn}}$ is the sum of the MSE between a nonlinear projection of the predicted latent state $p(\hat{z}_{t+i})$ and target state z_{t+i}^T. The target state at a given timestep $t+i$ is produced by a target encoder e_T, an exponential moving average of the online encoder e, by embedding the actual state at that timestep $z_{t+i}^T = e_T(s_{t+i})$. The loss for the **full dynamics (FD)** prediction is given by:

$$\mathscr{L}_{\text{FD}}(\tau) = \sum_{i=1}^{n_f} MSE\left(p(\hat{z}_{t+i}), z_{t+i}^T\right) \quad (3)$$

Finally, we combine the ideas above into a novel objective called **tactile dynamics (TD)** that optimises the encoder to learn a state representation z_t that can both predict future latent states *and* reconstruct the future tactile states.

$$\mathscr{L}_{\text{TD}} = \mathscr{L}_{\text{FD}} + \sum_{i=1}^{n_f} \mathscr{L}_{\text{TR}}(\hat{s}_{t+i}^{tact}, s_{t+i}^{tact}). \quad (4)$$



(a) **Find** (b) **Bounce** (c) **Baoding**

Fig. 2: Our `Robot Tactile Olympiad (RoTO)` environments. (a) find a randomised object (T=5s), (b) bounce a ball (T=10s), and (c) rotate Baoding balls (T=10s) – aiming for maximum efficiency in each. The observation space for all agents is a history of proprioception and binary tactile activations.

III. Experimental Setup

RL. We use a customised version of Proximal Policy Optimisation (PPO) [33] from SKRL [35] for RL training, using 4096 parallelised environments with 100 reserved for continuous agent evaluation.

Hyperparameters. We run an individual sweep per environment and method combination. While highly compute expensive, it is necessary because the self-supervision's impact on the state representation fundamentally changes the learning problem. Please consult the Appendix for the procedure and values.

Environments. We evaluate our method on three custom robotic manipulation tasks in Isaac Lab 2.0 [28] (Figure 2). The environments were designed to cover a wide range of tactile interactions (sparse, intermittent, and sustained) to evaluate the generalisability of our method. We release our baselines and environments as a benchmark called `RoTO: Robot Tactile Olympiad` to inspire progress in tactile-based robot manipulation. Please see the Appendix for full POMDP and simulation details.

- **Find:** Find the location of a fixed sphere with randomised initial position on a 20 cm $\times$ 20 cm plate as quickly as possible in 5 seconds.
- **Bounce:** Bounce a ball as many times as possible in 10 seconds. The ball is modelled off a typical office stress ball (70mm diameter, 30g). The (human) Guinness World Record for this task is 353 bounces in 1 minute, which corresponds with 58.8 bounces in 10 seconds.
- **Baoding:** Originating a millenia ago in China, Baoding balls are used by rotating two or more balls repeatedly in-hand. The task is to rotate two balls around each other as many times as possible in 10 seconds. The fastest human demonstration we could find online achieves 13 rotations in 10 seconds.

IV. Results

RL-only. To evaluate the ability of RL-only agents to extract effective tactile representations, we test proprioceptive-tactile agents (`prop-tactile`) and proprioceptive agents (`prop`). To isolate the contribution of the last action, we also test removing this from the proprioception (`prop(no last action)`). All results in Figure 3 depict the mean evaluation return across at least 5 seeds in bold and $\pm$ 1 standard deviation as shaded. In *Find*, the agent with access to tactile information is slightly more sample efficient, but ultimately converges to the performance of the proprioceptive agent. It is evident that the last action is key to success of the proprioceptive agent, allowing contact inference with the object. In *Bounce*, the agent with tactile information is more sample efficient and reaches higher returns. The proprioceptive agent without the last action still attains high returns by performing a bounce motion with an outstretched hand that appears agnostic to the ball state. In *Baoding*, the additional tactile information drives the agent from complete failure to success with high variance (5 successes, 6 failures across 11 seeds). These results demonstrate the importance of tactile information in the context of our tasks.

RL+SSL. We evaluate our four proposed SSL objectives: tactile reconstruction, full reconstruction, forward dynamics, and tactile forward dynamics. In the Appendix, these plots are shown as physical quantities and for improved clarity alternative figures of self-supervision in the *Baoding* task are provided. As illustrated in Figure 4, agents trained with tactile reconstruction and full dynamics objectives outperform RL-only agents across all environments. Compared against each other, the dynamics-optimised agent produces higher mean returns in *Find* and *Bounce*. In *Baoding*, while the dynamics-optimised agent achieves a much higher upper bound on performance, the tactile reconstruction agent achieves a higher mean return because of its tight performance distribution. The performance of reconstruction and tactile dynamics was more sensitive to environment, with no clear trend.

V. Discussion

Q1: Do binary contacts offer benefits beyond proprioceptive history for RL-only agents? Our results indicate that that binary contacts *do* offer benefits beyond proprioceptive histories and RL-only agents are capable of extracting useful tactile representations, but not always *reliably* (Figure 3, *Baoding*). The degree of usefulness also varies significantly across tasks, suggesting the relevance of context. We hypothesise

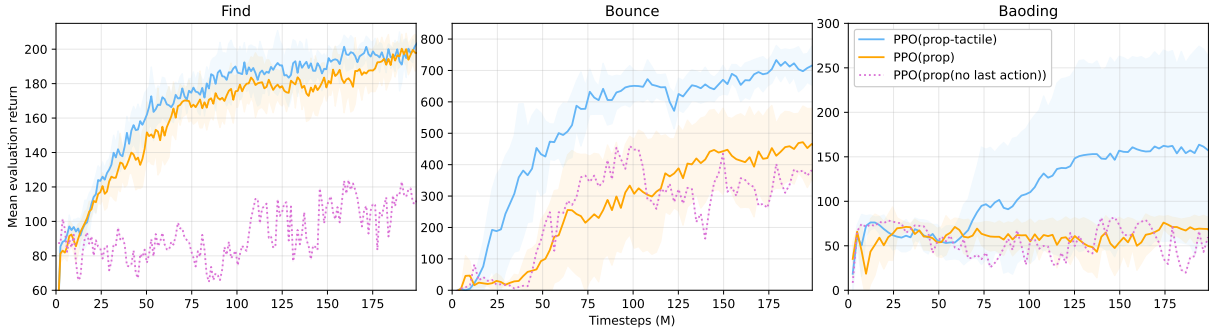


Fig. 3: **RL-only.** Mean evaluation returns of proprioceptive-tactile vs proprioceptive agents. To ascertain the importance of the last taken action for the proprioceptive agents, we run one seed with this quantity removed from the observation.

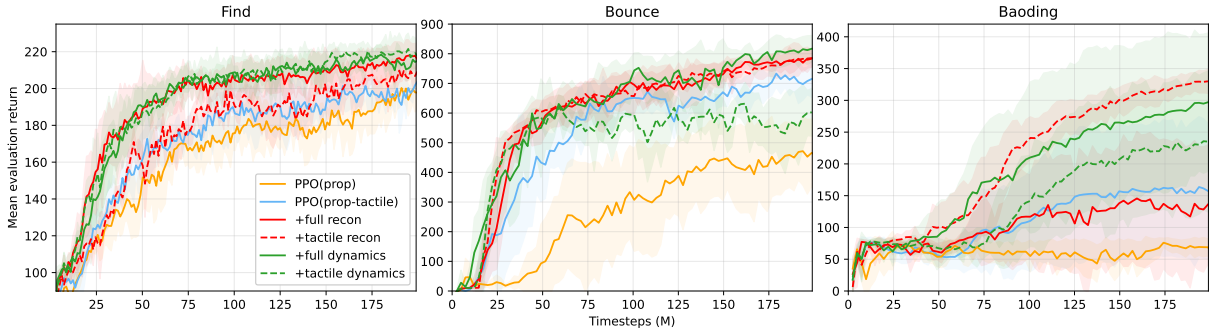


Fig. 4: **RL+SSL.** Mean evaluation returns of the self-supervised agents.

that since binary activations can reveal new contacts one timestep faster than proprioceptive histories, this information is more useful in dynamic tasks such as *Bounce* and *Baoding* compared to *Find* which is static. We also hypothesise the benefit in *Bounce* could also stem from providing contact information about the ball (30g) that is perhaps too lightweight to register in the control errors. Similarly, in *Baoding* the balls' motion is mostly perpendicular to the joint motion, leading to negligible control errors.

Q2: How do the results correspond with real-world metrics? The performance using real-world metrics is shown in the Appendix Figure 5, with the bold bars depicting the maximum value of the mean across seeds, and the shaded bars depicting the maximum value across all seeds. The performance improvements do correspond with meaningful changes in physical behaviours. For example, our best self-supervised agents on average find an object 36% faster (1.4 vs 1.9 seconds), bounce a ball 8 more times in 10 seconds (79 vs 71), and complete 17 Baoding rotations compared to 5 in 10 seconds. We note that the best *Bounce* agent achieves 88 bounces in 10 seconds, beating the (human) Guinness World Record of $\sim$ 58.

Q3: How well can a forward model learn the dynamics of tactile interactions?

Very well. See the Appendix for plots of true positive rate, false negative rate, precision, and recall for up to 10 timesteps into the future for *Bounce* and *Baoding*. There we also also provide a spatio-temporal visualisation of predicted vs actual next tactile states. Fascinatingly, from the no-contact state s_7 in *Bounce*, the decoder correctly anticipates contact in the next state (albeit in the wrong locations). This result possibly suggests that some form of object position information is being encoded.

Q4: How does your research translate to practical recommendations? We compress our findings into two recommendations. Our work has demonstrated that "blind" robotic agents trained jointly with self-supervision outperform RL-only agents across a diverse range of control tasks. Thus, our first recommendation is to train tactile-based RL agents jointly with tactile reconstruction or full dynamics objectives, if you are working in a similar setting and would like to get a higher (and potentially more reliable) distribution of returns. Second, while we acknowledge that increased sensory information is theoretically advantageous, it incurs substantial computational costs as well as being statistically harder the space of functions over Z^n is much larger than $\{0,1\}^n$. In addition, the bandwidth of pixel-based signals directly limits the number of parallel environments that can be executed in Isaac Lab and other simulators. Since our work has revealed unexpected efficacy using binary tactile observations, we recommend initially implementing simpler tactile information formats (binary, discrete, continuous) to maximise the advantages of GPU-accelerated RL, and switching to pixel-based tactile representations if required by research objectives.

References

[1] Rishabh Agarwal et al. "Contrastive Behavioral Similarity Embeddings for Generalization in Reinforcement Learning". In: *ICLR*. 2021.

[2] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[3] Philipp Becker et al. *Joint Representations for Reinforcement Learning with Multiple Sensors*. en. arXiv:2302.05342 [cs]. June 2023. URL: http://arxiv.org/abs/2302.05342 (visited on 06/04/2024).

[4] Kaiqi Chen, Yong Lee, and Harold Soh. *Multi-Modal Mutual Information (MuMMI) Training for Robust Self-Supervised Deep Reinforcement Learning*. arXiv:2107.02339 [cs]. July 2021. DOI: 10.48550/arXiv.2107.02339. URL: http://arxiv.org/abs/2107.02339 (visited on 06/04/2024).

[5] Yizhou Chen et al. *Visuo-Tactile Transformers for Manipulation*. arXiv:2210.00121 [cs]. Sept. 2022. DOI: 10.48550/arXiv.2210.00121. URL: http://arxiv.org/abs/2210.00121 (visited on 06/12/2024).

[6] Vedant Dave, Fotios Lygerakis, and Elmar Rueckert. *Multimodal Visual-Tactile Representation Learning through Self-Supervised Contrastive Pre-Training*. arXiv:2401.12024 [cs]. Jan. 2024. URL: http://arxiv.org/abs/2401.12024 (visited on 01/30/2024).

[7] Mhairi Dunion and Stefano V. Albrecht. "Multi-view Disentanglement for Reinforcement Learning with Multiple Cameras". In: *1st Reinforcement Learning Conference*. 2024.

[8] Mhairi Dunion et al. "Conditional Mutual Information for Disentangled Representations in Reinforcement Learning". In: *Advances in Neural Information Processing Systems* 36 (2024).

[9] Mhairi Dunion et al. "Temporal Disentanglement of Representations for Improved Generalisation in Reinforcement Learning". In: *International Conference on Learning Representations (ICLR)*. 2023.

[10] Carles Gelada et al. "DeepMDP: Learning Continuous Latent Space Models for Representation Learning". In: *ICML*. 2019.

[11] Irmak Guzey et al. *Dexterity from Touch: Self-Supervised Pre-Training of Tactile Representations with Robotic Play*. arXiv:2303.12076 [cs]. Mar. 2023. URL: http://arxiv.org/abs/2303.12076 (visited on 03/21/2024).

[12] Irmak Guzey et al. *See to Touch: Learning Tactile Dexterity through Visual Incentives*. arXiv:2309.12300 [cs]. Sept. 2023. URL: http://arxiv.org/abs/2309.12300 (visited on 01/30/2024).

[13] Danijar Hafner et al. "Dream to Control: Learning Behaviors by Latent Imagination". In: *ICLR*. 2020.

[14] Danijar Hafner et al. "Mastering Atari with Discrete World Models". In: *ICLR*. 2021.

[15] Danijar Hafner et al. "Mastering Diverse Domains through World Models". In: *arXiv preprint arXiv:2301.04104* (2023).

[16] Johanna Hansen et al. "Visuotactile-RL: Learning Multimodal Manipulation Policies with Deep Reinforcement Learning". en. In: *2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 8298–8304. ISBN: 978-1-72819-681-7. DOI: 10.1109/ICRA46639.2022.9812019. URL: https://ieeexplore.ieee.org/document/9812019/ (visited on 01/30/2024).

[17] Max Jaderberg et al. "Reinforcement Learning with Unsupervised Auxiliary Tasks". In: *arXiv preprint: arXiv:1611.05397* (2017).

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore. "Reinforcement Learning: A Survey". en. In: *Journal of Artificial Intelligence Research* 4 (May 1996), pp. 237–285. ISSN: 1076-9757. DOI: 10.1613/jair.301. URL: https://www.jair.org/index.php/jair/article/view/10166 (visited on 11/21/2023).

[19] Sascha Lange and Martin Riedmiller. "Deep auto-encoder neural networks in reinforcement learning". In: *International Joint Conference on Neural Networks (IJCNN)*. 2010.

[20] Sascha Lange, Martin Riedmiller, and Arne Voigtländer. "Autonomous reinforcement learning on raw visual input data in a real world application". In: *International Joint Conference on Neural Networks (IJCNN)*. 2012.

[21] Alex X. Lee et al. "Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 741–752. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/08058bf500242562c0d031ff830ad094-Paper.pdf.

[22] Joonho Lee et al. "Learning Quadrupedal Locomotion over Challenging Terrain". In: *Science Robotics* 5.47 (Oct. 2020). arXiv:2010.11251 [cs, eess], eabc5986. ISSN: 2470-9476. DOI: 10.1126/scirobotics.abc5986. URL: http://arxiv.org/abs/2010.11251 (visited on 12/14/2023).

[23] Kuang-Huei Lee et al. "Predictive Information Accelerates Learning in RL". In: *NeurIPS*. 2020.

[24] Michelle A. Lee et al. "Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks". en. In: *IEEE Transactions on Robotics* 36.3 (June 2020), pp. 582–596. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2019.2959445. URL: https://ieeexplore.ieee.org/document/9043710/ (visited on 12/13/2023).

[25] Trevor McInroe, Lukas Schäfer, and Stefano V. Albrecht. "Multi-Horizon Representations with Hierarchical Forward Models for Reinforcement Learning". In: *Transactions on Machine Learning Research (TMLR)* (2024).

[26] Trevor McInroe et al. "Planning to Go Out-of-Distribution in Offline-to-Online Reinforcement Learning". In: *RLC*. 2024.

[27] Andrew Melnik et al. "Using Tactile Sensing to Improve the Sample Efficiency and Performance of Deep Deterministic Policy Gradients for Simulated In-Hand Manipulation Tasks". In: *Frontiers in Robotics and AI* 8 (2021). ISSN: 2296-9144. URL: https://www.frontiersin.org/articles/10.3389/frobt.2021.538773 (visited on 12/12/2023).

[28] Mayank Mittal et al. "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments". In: *IEEE Robotics and Automation Letters* 8.6 (2023), pp. 3740–3747. DOI: 10.1109/LRA.2023.3270034.

[29] Volodymyr Mnih et al. "Asynchronous Methods for Deep Reinforcement Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. 2016.

[30] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518 (2015), pp. 529–533.

[31] Haozhi Qi et al. "General In-hand Object Rotation with Vision and Touch". en. In: *Proceedings of The 7th Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Dec. 2023, pp. 2549–2564. URL: https://proceedings.mlr.press/v229/qi23a.html (visited on 01/30/2024).

[32] Haozhi Qi et al. *In-Hand Object Rotation via Rapid Motor Adaptation*. arXiv:2210.04887 [cs]. Oct. 2022. URL: http://arxiv.org/abs/2210.04887 (visited on 11/17/2023).

[33] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[34] Max Schwarzer et al. "Data-Efficient Reinforcement Learning with Self-Predictive Representations". In: *ICLR*. 2021.

[35] Antonio Serrano-Muñoz et al. "skrl: Modular and Flexible Library for Reinforcement Learning". In: *Journal of Machine Learning Research* 24.254 (2023), pp. 1–9. URL: http://jmlr.org/papers/v24/23-0112.html.

[36] Carmelo Sferrazza et al. *The Power of the Senses: Generalizable Manipulation from Vision and Touch through Masked Multimodal Learning*. arXiv:2311.00924 [cs]. Nov. 2023. URL: http://arxiv.org/abs/2311.00924 (visited on 11/28/2023).

[37] Leon Sievers, Johannes Pitz, and Berthold Bäuml. "Learning Purely Tactile In-Hand Manipulation with a Torque-Controlled Hand". In: *2022 International Conference on Robotics and Automation (ICRA)*. May 2022, pp. 2745–2751. DOI: 10.1109/ICRA46639.2022.9812093. URL: https://ieeexplore.ieee.org/document/9812093 (visited on 12/14/2023).

[38] Austin Stone et al. "The Distracting Control Suite – A Challenging Benchmark for Reinforcement Learning from Pixels". In: *arXiv preprint arXiv:2101.02722* (2021).

[39] Adam Stooke et al. "Decoupling Representation Learning from Reinforcement Learning". In: *arXiv preprint: arXiv:2009.08319* (2021).

[40] Naftali Tishby, Fernando C. Pereira, and William Bialek. *The information bottleneck method*. en. arXiv:physics/0004057. Apr. 2000. URL: http://arxiv.org/abs/physics/0004057 (visited on 06/24/2024).

[41] Herke Van Hoof et al. "Learning robot in-hand manipulation with tactile features". en. In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. Seoul, South Korea: IEEE, Nov. 2015, pp. 121–127. ISBN: 978-1-4799-6885-5. DOI: 10.1109/HUMANOIDS.2015.7363524. URL: http://ieeexplore.ieee.org/document/7363524/ (visited on 12/06/2023).

[42] Herke Van Hoof et al. "Stable reinforcement learning with autoencoders for tactile and visual data". en. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea: IEEE, Oct. 2016, pp. 3928–3934. ISBN: 978-1-5090-3762-9. DOI: 10.1109/IROS.2016.7759578. URL: http://ieeexplore.ieee.org/document/7759578/ (visited on 12/06/2023).

[43] Nikola Vulin et al. "Improved Learning of Robot Manipulation Tasks Via Tactile Intrinsic Motivation". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021). Conference Name: IEEE Robotics and Automation Letters, pp. 2194–2201. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3061308. URL: https://ieeexplore.ieee.org/abstract/document/9361156 (visited on 11/17/2023).

[44] Niklas Wahlström, Thomas B. Schön, and Marc Peter Deisenroth. "From Pixels to Torques: Policy Learning with Deep Dynamical Models". In: *arXiv preprint: arXiv:1502.02251* (2015).

[45] Manuel Watter et al. "Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images". In: *NeurIPS*. 2015.

[46] Linhan Yang et al. "TacGNN: Learning Tactile-Based In-Hand Manipulation With a Blind Robot Using Hierarchical Graph Neural Network". en. In: *IEEE Robotics and Automation Letters* 8.6 (June 2023), pp. 3605–3612. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2023.3264759. URL: https://ieeexplore.ieee.org/document/10093019/ (visited on 03/18/2024).

[47] Max Yang et al. "Sim-to-Real Model-Based and Model-Free Deep Reinforcement Learning for Tactile Pushing". en. In: *IEEE Robotics and Automation Letters*

8.9 (Sept. 2023), pp. 5480–5487. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2023.3295236. URL: https://ieeexplore.ieee.org/document/10182274/ (visited on 02/02/2024).

[48] Denis Yarats, Ilya Kostrikov, and Rob Fergus. "Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels". In: *International Conference on Learning Representations (ICLR)*. 2021.

[49] Denis Yarats et al. "Improving Sample Efficiency in Model-Free Reinforcement Learning from Images". In: *arXiv preprint: arXiv:1910.01741* (2019).

[50] Zhao-Heng Yin et al. "Rotating without Seeing: Towards In-hand Dexterity through Touch". In: *Robotics: Science and Systems* (2023).

[51] Bang You and Huaping Liu. "Multimodal information bottleneck for deep reinforcement learning with multiple sensors". In: *Neural Networks* 176 (Aug. 2024), p. 106347. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2024.106347. URL: https://www.sciencedirect.com/science/article/pii/S0893608024002715 (visited on 06/04/2024).

[52] Ying Yuan et al. *Robot Synesthesia: In-Hand Manipulation with Visuotactile Sensing*. arXiv:2312.01853 [cs]. Dec. 2023. URL: http://arxiv.org/abs/2312.01853 (visited on 01/30/2024).

APPENDIX

VI. Related work

Pixel representation learning in reinforcement learning (RL). RL agents tend to be poor representation learners when input observations are high dimensional (e.g., pixels). While early works in pixel-based RL did not target this issue [30, 29], most works in this space leverage auxiliary objectives [17] in order to guide the learning of the representation. Reconstruction is a popular objective studied in both the model-free [19, 20, 49] and model-based paradigms [45, 44, 21, 13, 14, 15]. Pixel-perfect reconstruction requires that all information is maintained, yet much of this information may be irrelevant to learning control [38]. Instead, many auxiliary objectives operate within the latent space, ranging widely from forward dynamics [10, 34, 25], disentanglement [8, 9, 7], contrastive approaches [**curl**, 39, 1], to information-theoretic objectives [23, 26]. In this work we examine how these techniques can be applied to proprioceptive and tactile robotic sensory modalities.

Multimodal representation learning in RL. Recent works have aimed to produce a generalisable auxiliary objective that operates over all modalities. [4] propose maximising mutual information (MI) between unimodal and multimodal representations to align the latent spaces. Noting this does not filter irrelevant information, [51] suggest using an information bottleneck [40] that maximises forward dynamics information. [3] propose that self-supervised objectives should be modality-specific, reconstruction for proprioception and contrastive losses for images. We do not propose a generalisable multimodal auxiliary objective, but auxiliary objectives to better leverage the tactile modality.

Tactile representation learning in RL. Few works focus specifically on leveraging self-supervised objectives to improve tactile-based learning. Early work used a variational autoencoder with forward dynamics objective for a stabilisation task [42]. More recently, pixel-based objectives such as masked autoencoding (MAE) [36] and augmentation [16, 12] have been explored. Other approaches use contrastive learning, maximise the similarity of unimodal visual and tactile representations from the same timestep [6]. Others have proposed tactile-specific objectives, such as predicting the presence of contacts [24, 5]. Like [42], we identify dynamics as a promising objective for tactile information and are the first to study multi-step tactile dynamics prediction in RL. Unlike [42, 16, 12, 36, 6], we focus on learning from simple binary contacts.

Tactile-tailored RL. Tactile interactions are both important and sparse. As a result, some works have explored how aspects of the RL setup can be modified to better leverage tactile input. [43] propose to increase the sampling probability of contact-rich episodes in off-policy algorithms. [16] propose tactile gating, in which a tactile encoder is only updated if there is contact. In the space of tactile-based on-policy RL, we are the first work to modify the dataset the self-supervised objective is trained on from the typical on-policy rollout.

VII. Limitations

The primary limitation of our work is the lack of real-world robot experiments which is a natural direction for future work, but hope our choice to study binary tactile activations greatly minimises the potential sim2real gap. We also note that training self-supervised agents increases computation time compared to RL-only. The effect is less noticeable for reconstruction, but becomes more dramatic the higher the value of n_f in forward dynamics. In addition, a limitation of using a separated auxiliary memory is that more memory is required. Regarding generalisation, we would expect to see similar results if our approach is applied to other environment domains (e.g., robotic locomotion).

VIII. Results as real-world metrics

Figure 6 shows the average number of seconds it takes the agent to locate the object within different tolerances. The distance d is measured between the Franka end-effector (imaginary fixed frame in the center of the parallel-jaw gripper) and the object center. While the performance between a proprioceptive and proprioceptive-tactile agent is similar for a $d = 3$ cm threshold, the benefits of tactile data become more pronounced with smaller tolerances. The relative performance between SSL objectives is consistent except for $d = 0.05$ cm, where the tactile dynamics objective is on average the fastest. Figure 7 shows learning through the number of bounces the agent achieves in 10 seconds, and Figure 8 through the number of complete Baoding rotations achieved. Due to the overlapping performance distributions in *Baoding*, we provide alternative figure versions with only a subset of runs and/or

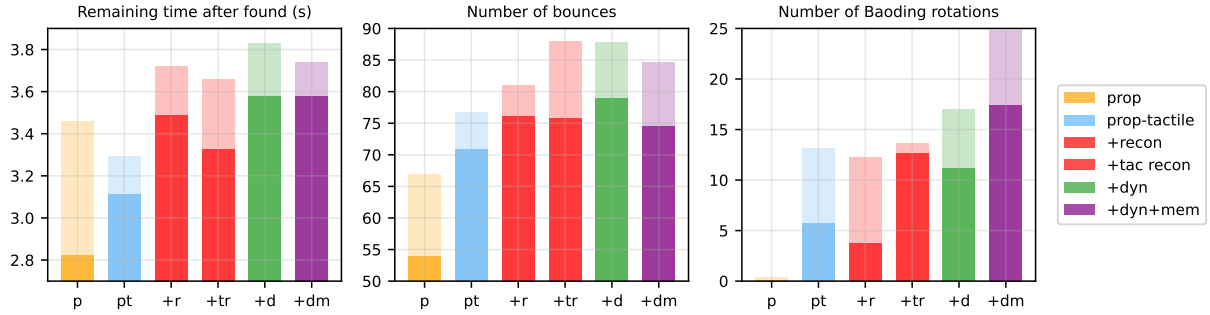


Fig. 5: **Real-world metrics.** Maximum (shaded) and maximum of mean (bold) across seeds.

no bad seeds. Across all seeds, from Figure 8 (bottom left) we can see applying tactile reconstruction or full dynamics self-supervision approximately doubles the number of complete rotations achieved.

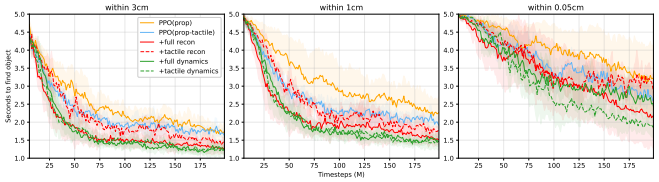


Fig. 6: Number of seconds to find the object within 3, 1, and 0.05 cm in the *Find* environment.

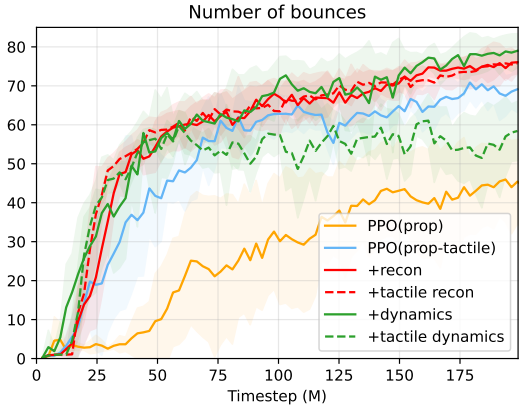


Fig. 7: Number of bounces in 10 seconds.

IX. Future tactile state prediction analysis

To understand how well MLPs can model the forward dynamics of tactile interactions, we provide various classification metrics throughout training (Figure 9) and spatio-temporal rollout visualisations of trained *Bounce* and *Baoding* agents (Figures 10 and 11). We can compute metrics like recall and precision because tactile reconstruction was formulated as classification (rather than regression), but unlike typical supervised learning the performance does not increase monotonically because of the nonstationary data distribution. We perform this analysis on agents trained with the tactile dynamics objective since they were specifically trained for future tactile prediction. This analysis could be done on agents trained with full dynamics, but would require learning a separated tactile decoder. Finally, due to the large proportion of 0s to 1s, we found it necessary to apply a positive weighting in the binary cross entropy loss of $p_c = 10$ across all classes (activation regions) due to tactile data imbalance.

Overall, we were surprised how robust the performance remained n_f timesteps into the future (evidenced by how difficult it is to distinguish between the timesteps in Figure 9). This suggests that the multi-step dynamics objective was very effective at encoding information relevant for future state predictions. Moreover, the rate of missed contacts was $<1\%$ throughout training for both *Bounce* and *Baoding*, which we attribute to the positive weighting applied. Interestingly, despite the agent having access to the 3 last tactile states that would form the first 3 tactile states of the prediction, some of these states were not always perfectly "copied" over (e.g., s_5, s_{11}, and s_{12} in *Bounce*). This highlights that the states are not merely being 'memorised', but being represented in some (imperfect) way. For future work, we believe dynamically updating the positive weighting would be beneficial to reflect the nonstationary training distribution. Additionally, our implementation applied the same weighting to each activation region, but some regions are much more active than others (e.g., compare palm to pinky in Figure 11), and this discrepancy should be accounted for.

For *Bounce*, tactile interactions are increasingly sparse (e.g., compare Figures 10 and 11). Thus the metrics we were most interested in were true positive rate (proportion of contacts caught) and false negative rate (proportion of contacts missed). True positive rate (TPR) decreased from $\sim$99% to 90% throughout training, which we attribute to increased difficulty predicting the landing sites of a bouncing ball. The proportion of contacts that were missed (FNR) was surprisingly low throughout training, converging to $\sim$ 0.2%. Fascinatingly, from the no-contact state s_7, the decoder correctly anticipates contact in the next state (albeit in the wrong locations, however two predictions are just 1 timestep early). This result suggests that some form of object position information is being encoded.

For *Baoding*, the frequency of tactile interactions remains high throughout learning, thus all metrics are of relevance. The proportion of contacts that were correctly detected was

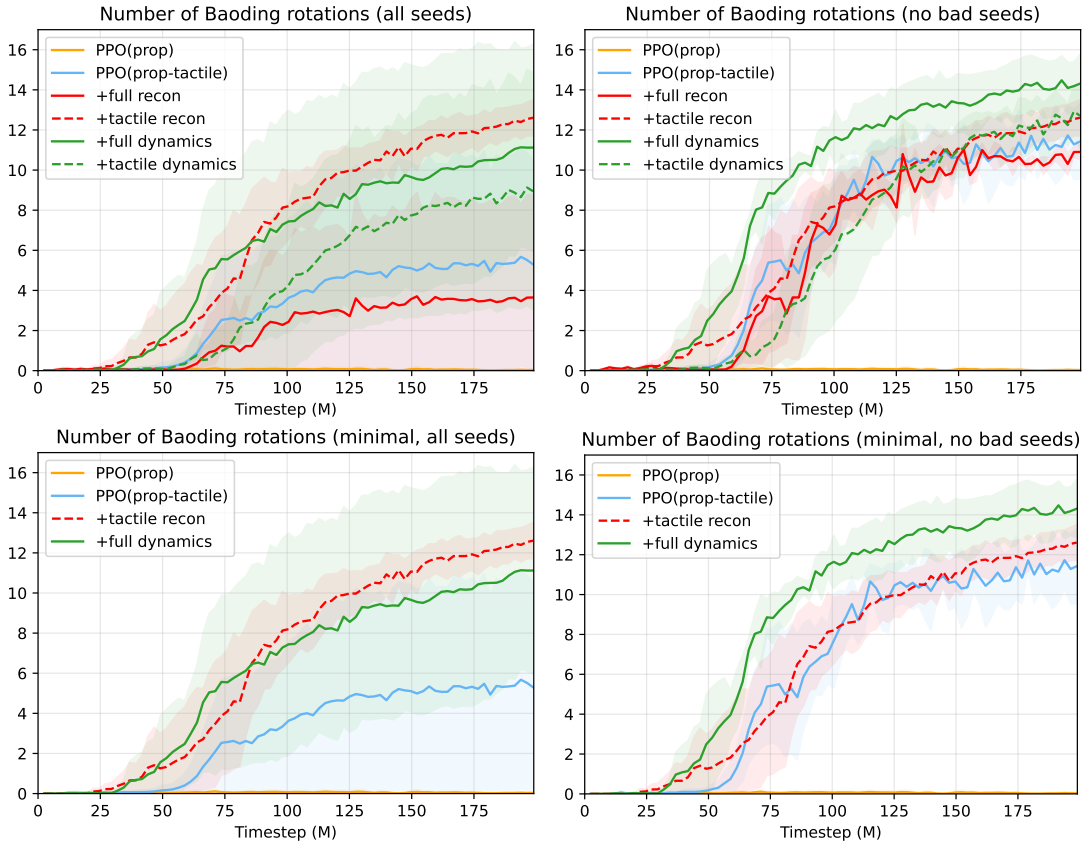


Fig. 8: Number of complete Baoding ball rotations when each ball makes a full rotation in 10 seconds. **Left:** The distribution across all seeds. **Right:** The distribution across seeds that learned at least 1 rotation. **Top:** All experiments. **Bottom:** A subset of experiments.

$\sim$99%, which is very high. Similarly to *Bounce*, very few true contacts were missed (FNR), and the false positive rate was relatively high at $\sim$15%. The accuracy remained at >96% for majority of training. Also similarly to *Bounce*, some of the false positive predictions (e.g., in states s_{10}, s_{11}) are just one-step too early. Finally, across all metrics in Figure 9, we can see negative performance spikes at fixed intervals throughout learning that are not present for *Bounce*. This is discussed in the next section.

X. MDP

We use a physics simulation frequency of 120 Hz across all environments with a control decimation of 2. This means the agent receives observations and computes actions at 60 Hz, because the same action is applied twice in the physics engine. The default static and dynamic friction for robots and objects was set to 1.0.

A. Observations

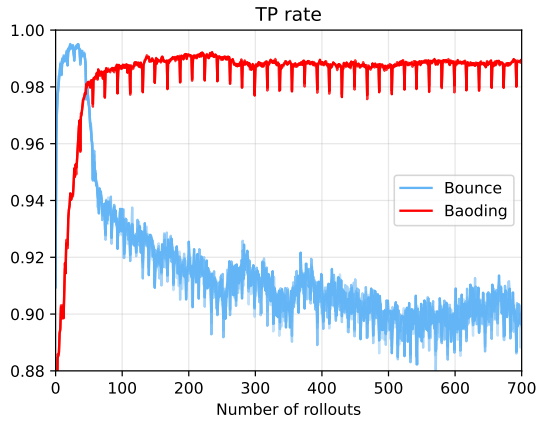
A summary of the different types of observation is shown in Table I. Each environment uses a stack of observations o_t to form the state s_t (16 for the Franka Find environment, 4 for the Shadow environments). We apply input preprocessing as follows: joint angles θ are normalised between [-1,1]. Joint velocities $\dot{theta}$ are scaled down by a scalar (0.33 for Franka, 0.2 for Shadow). The norm of each 3-dim normal force vector is clamped between [0, MAX], and normalised between [0,1]. The value of MAX was chosen to be 20N for Franka and 30N for Shadow.

TABLE I: Observation spaces across environments.

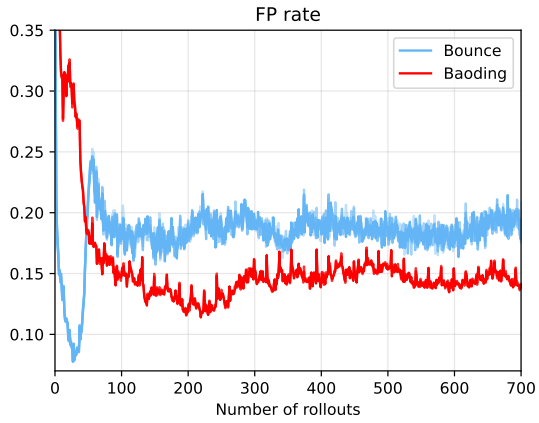
Symbol	Description	*Find*	*Bounce*	*Baoding*
f	normal forces	2	17	17
a	last action	9	20	20
θ	joint angles	9	24	24
$\dot{\theta}$	joint velocities	9	24	24
w	gripper width	1		
x_{ee}	EE position	3		
q_{ee}	EE quaternion	4		
o_t	timestep observation	37	85	85
s_t	stacked state $(S \times o_t)$	560 $(16o_t)$	340 $(4o_t)$	340 $(4o_t)$

We retrieve the forces through Isaac Lab's `ContactSensor` class, which returns the net contact force acting on a given rigid body[1]. To mimic real-world tactile sensing and make the task more challenging, we registered two 'plate-like' bodies to atop the Franka fingers to act as contact sensors (Figure 12a). With this setup, the sensors would only register forces that resulted from collision with these bodies, which is only possible from the object or other

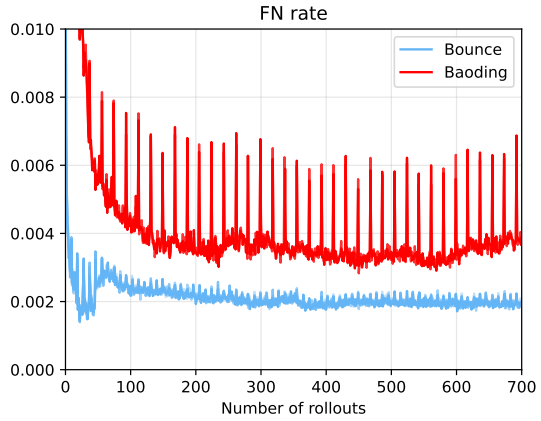
[1]https://isaac-sim.github.io/IsaacLab/main/source/overview/core-concepts/sensors/contact_sensor.html



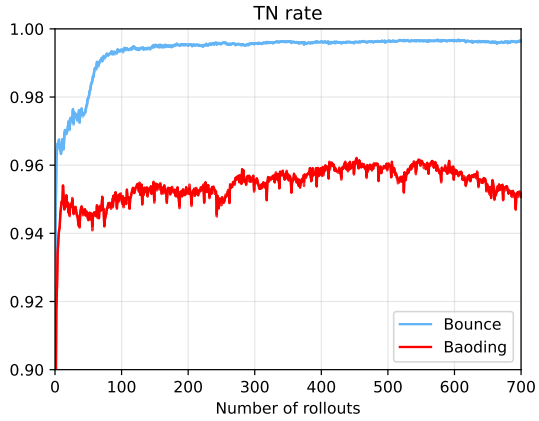
(a) $TPR = \frac{TP}{TP+FN}$ (also known as *recall* or *probability of detection*, proportion of actual contacts that were classified as contacts)



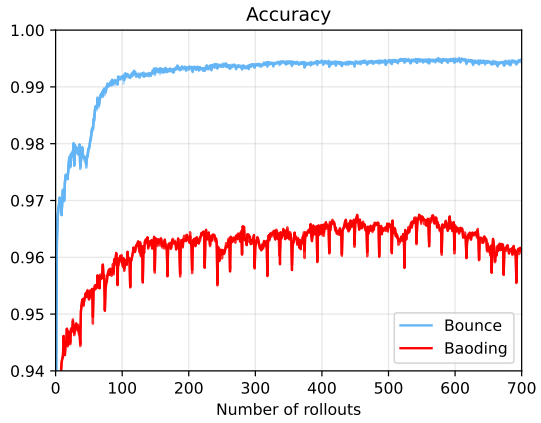
(b) $FPR = \frac{FP}{FP+TN}$ (also known as *probability of false alarm*, proportion of null contacts that were misclassified as contacts)



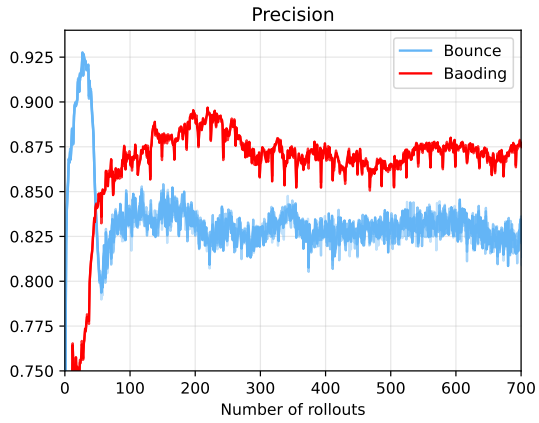
(c) $FNR = \frac{FN}{TP+FN}$ (proportion of actual contacts that were missed)



(d) $TNR = \frac{TN}{FP+TN}$ (proportion of null contacts that were classified as null contacts)



(e) Accuracy $A = \frac{TP+TN}{TP+FP+TN+FN}$



(f) $Precision = \frac{TP}{TP+FP}$

Fig. 9: Classification metrics for future predicted tactile states of *Bounce* and *Baoding* agents trained with tactile dynamics SSL objective for $t = 1, 2, 3, 9$ and $t = 1, 2$ timesteps into the future respectively. The metrics for future timesteps are shown with decreasing opacity but strongly overlap with $t = 1$, making it difficult to distinguish. The metrics were computed on one minibatch after the RL/SSL update.

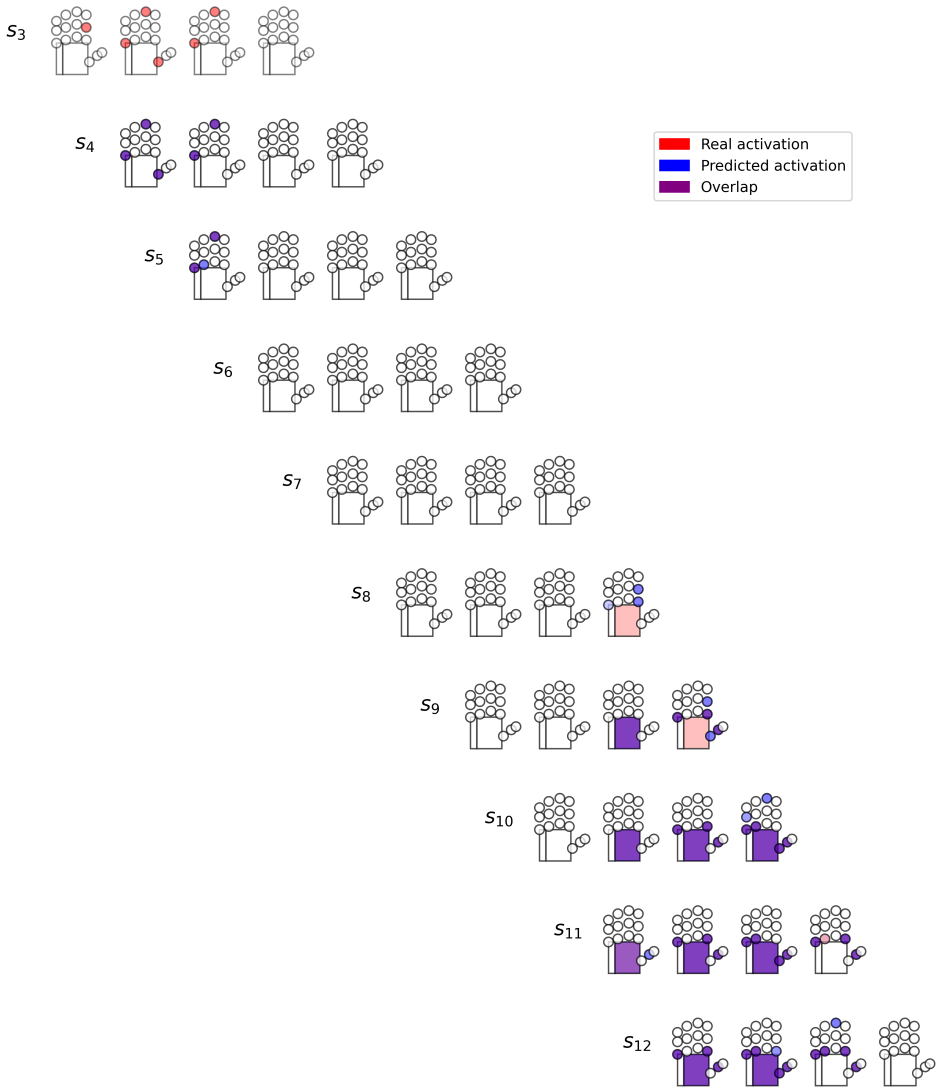


Fig. 10: Spatio-temporal visualisation of 1-step predicted future tactile states of a *Bounce* agent trained with the tactile dynamics SSL objective. The state s is comprised of 4 observations, and each column corresponds to the same observation. The sigmoid activation of the tactile prediction is displayed with varying opacity (e.g., less confident is lighter blue).

finger, and not the ground. Since the Shadow Hand was fixed in midair we let each link become a sensor, resulting in 17 sensors (Figure 12b).

B. Actions

Both robots are joint position controlled, with dimensions 9 and 20 for the Franka and Shadow Hand respectively. The Shadow Hand is underactuacted, with coupled distal and proximal joints like in humans (2 wrist + 5 thumb + 3 index + 3 middle + 3 ring + 4 pinky = 20).

C. Rewards

The rewards for each environment step are given by the sum of the different terms each multiplied by the given scale. For enhanced value function learning, we track a running mean and variance for normalising the returns and values in the PPO update.

For *Find* there is one reward term r_{dist} that grows as distance between the object and end-effector decreases.

For *Bounce*, the reward is given by $r_{air} + r_{bounce} + r_{fall}$. To aid initial exploration, the agent is rewarded proportionally to number of time steps since last contact, r_{air}. A bounce event is defined as there being contact, then no contact, and then contact again, for which the agent is rewarded with a bonus r_{bounce}. The fall penalty is applied if the object is more than 24 cm from a fixed central position.

For *Baoding* the reward is given by $r_{dist_1} + r_{dist_2} + r_{rotation} + r_{fall}$. Our original approach was to maximise the xy angular velocity of the vector connecting the two balls which worked well, but sometimes the agent came up with creative strategies that no longer resembled the original Baoding task. Thus, we reformulated the reward around two fixed target poses (Figure 13). When the centers of both balls were within 1.0 cm of the given target centers, the targets switched and the

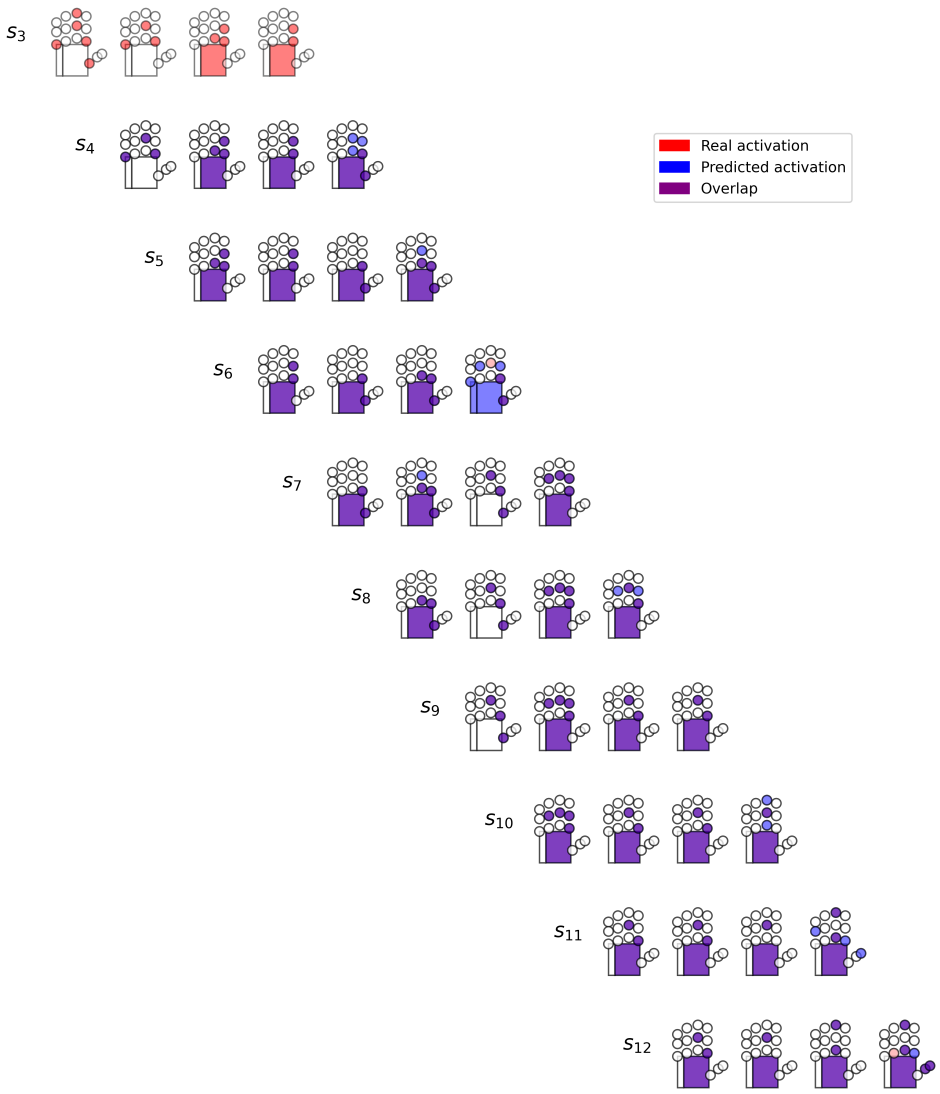
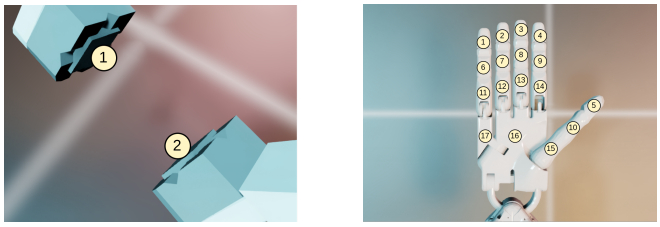


Fig. 11: Spatio-temporal visualisation of 1-step predicted future tactile states of a *Baoding* agent trained with the tactile dynamics SSL objective. The state s is comprised of 4 observations, and each column corresponds to the same observation. The sigmoid activation of the tactile prediction is displayed with varying opacity (e.g., less confident is lighter blue).



(a) Franka (b) Shadow Hand

Fig. 12: Contact sensor configuration across robot embodiments.

agent receives a bonus reward $r_{rotation}$. We also needed to use two dense ball-center-to-target distance rewards r_{dist_1}, r_{dist_2} to aid exploration in the beginning . This approach better constrained the ball positions and stabilised policies for comparing methodologies. The fall penalty is applied if the distance between the balls exceeds 15 cm.

D. Reset

Episodes can be terminated by a failure state, or truncated by a time limit. The *Find* environment episode length is $T = 300$ timesteps (5s), and *Bounce* and *Baoding* environments are $T = 600$ timesteps (10s). The *Bounce* and *Baoding* episodes can be terminated early if the balls fall out of the hand, measured by a distance. At the beginning of each episode, the Franka joint angles are randomised up to $\pm 7°$ and the Shadow joint angles are randomised upto $\pm$ 20%. The ball in *Find* is randomised to any position on the 20cm $\times$ 20cm plate. The *Bounce* ball and *Baoding* balls are randomised by ± 1 cm and ± 0.5cm respectively along the global xyz axis .

TABLE II: Reward components across environments.

Symbol	Description	Equation	Scale	*Find*	*Bounce*	*Baoding*
r_{dist}	distance to target	$1-\tanh d_{target}/0.1$	1, 0.1	✓		✓✓
r_{air}	time without contact	+1	0.01		✓	
r_{bounce}	successful bounce	+1	10		✓	
$r_{rotation}$	successful rotation	+1 if $d_{1\&2} < 1.0$cm	10			✓
r_{fall}	fall penalty	-1 if $d > d_{max}$	10		✓	✓

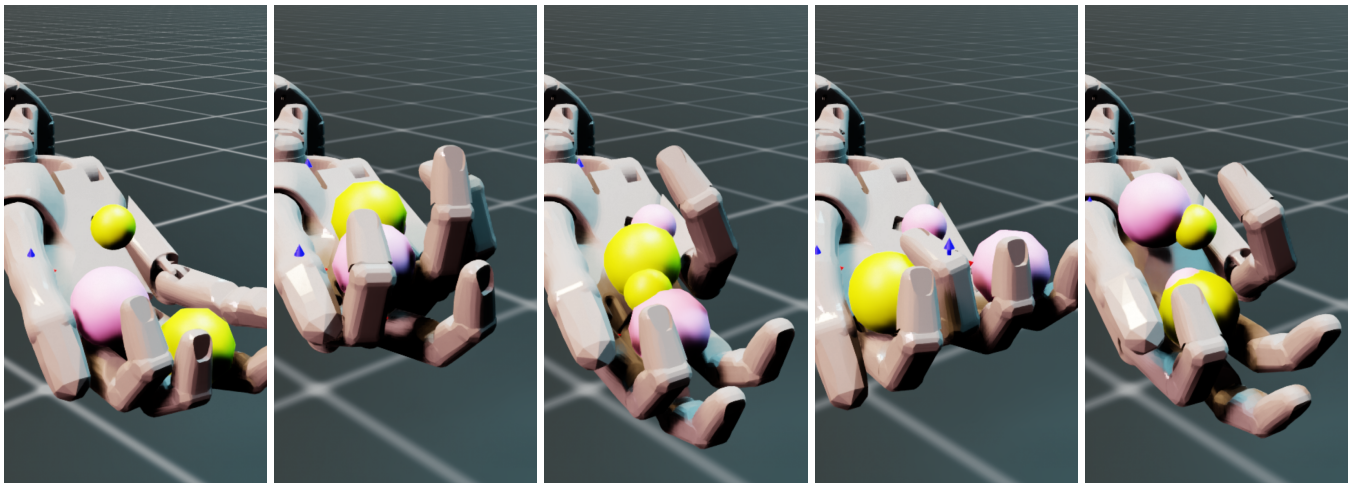


Fig. 13: *Baoding*: When both balls are within 1cm of their virtual targets (shown as smaller balls), the targets switch.

XI. Network Architectures

Encoder. The encoder e is a 3-layer MLP with dimensions $s_t \to 1024 \to 512 \to 256 \to z_t$. Layer normalisation and ELU activations are applied after each layer.

Policy. The policy π is a 3-layer MLP with dimensions $z_t \to 128 \to 64 \to n_{actions} \to a_t$. ELU activations are applied after the first two layers. The output layer activation is tanh for *Find* and identity for *Bounce* and *Baoding*.

Value function. The value function v is a 3-layer MLP with dimensions $z_t \to 128 \to 64 \to 1 \to V_t$. ELU activations are applied after the first two layers. The output layer activation is identity.

Reconstruction. The decoder d is a 3-layer MLP. The full reconstruction decoder has dimensions $z_t \to 512 \to 512 \to \texttt{len}(s_t) \to \hat{s}_t$. The tactile reconstruction decoder has dimensions $z_t \to 512 \to 512 \to \texttt{len}(s_t^{tact}) \to \hat{s}_t^{tact}$. ELU activations are applied after the first two layers. The output layer activation is sigmoid for tactile predictions, and identity for proprioception predictions.

Forward dynamics. The forward model f is a 3-layer MLP with dimensions $(z_t, a_t) \to 512 \to 256 \to 256 \to \hat{z}_{t+1}$, with ELU activations after the first two layers. The projector is a 2-layer MLP with dimensions $\hat{z}_{t+i} \to 256 \to 256 \to \mathscr{L}_i$ with an ELU activation after the first layer. The target encoder e_T is identical to e, but updated according to Equation 5 with $\alpha = 0.01$.

$$\theta_{e_T} \leftarrow (1-\alpha)\theta_{e_T} + \alpha\theta_e \tag{5}$$

XII. Training Details

The combined loss L is the summed policy, value, entropy, and auxiliary loss.

RL. We use separate Adam optimisers for the policy, value, and encoder. The policy, value, and encoder optimisers share a constant learning rate lr. The gradient norms of the policy, value, and encoder networks are clipped at 1.0.

SSL. The same Adam optimiser is used to optimise the encoder and auxiliary-related networks (e.g., decoder d, forward model f, nonlinear projector p) with constant learning rate lr_{aux}. The gradient norms are *not* clipped: this seemed to degrade performance. The formulations of L_{aux} for the different objectives are shown in Section 3.2.

XIII. Human Capabilities

Find. Replicating the environment in real life and testing with one human subject, the average time to find and grasp the object across 10 trials was 2.1 seconds.

Bounce. The most tennis ball touches using the hand in one minute is 353, and was achieved by Manikandan Thirumaniselvam in India on 4 February 2023 [2]. This translates to $353/6 \sim 58$ bounces in 10 seconds. We note the properties of our ball (30g, 70mm diameter) are different to a tennis ball ($\sim$ 58g, 67mm diameter), but would expect to see similar results.

Baoding. The fastest demonstration we could find online achieves 13 rotations in 10 seconds[3]. We believe the properties of our Baoding balls and the ones in the video are identical (55g, 1.5 inch diameter) since we possess the same ones and modelled our simulated ones off them.

[2]**Bounce record**: https://www.guinnessworldrecords.com/world-records/590513-most-tennis-ball-touches-using-the-hand-in-one-minute, **Bounce video**: https://www.youtube.com/watch?v=ORiHY0MwT4A

[3]**Baoding video**: https://www.youtube.com/shorts/x-ns-auc098

XIV. HYPERPARAMETER TUNING

We carefully tuned all our experiments to give each agent the best shot. For fairness, we followed the same hyperparameter tuning recipe for each individual experiment (3 environments $\times$ 7 experiments = 21 sweeps). We use the Optuna library [2] with the TPE sampler (5 startup trials) and no pruner. We wait for each sweep to reach 20 complete trials (some hyperparameter combinations lead to policy/value NaNs which are terminated early). The hyperparameters and possible ranges we tested are provided in Table III, with the optimised values in Table IV. We did not sweep over the following hyperparameters: discount factor $\gamma = 0.99$, value loss scale $c_v = 0.1$, gradient norm clip 1.0, value clip 0.2, ratio clip 0.2.

TABLE III: Tunable hyperparameters and ranges for each experiment.

Hyperparameter	Symbol	Tunable values
Rollout	R	$\{16, 32, 64\}$
Minibatches	mb	$\{4, 8, 16, 32, 64\}$
Learning epochs	le	$\{4, 8, 16, 32\}$
Learning rate	lr	$[10^{-5}, 10^{-3}] \subset \mathbb{R}$
Entropy loss scale	c_{ent}	$\{0, 0.05, 0.1\}$
Auxiliary learning rate	lr_{aux}	$[10^{-5}, 10^{-3}] \subset \mathbb{R}$
Auxiliary loss weight	c_{aux}	$[10^{-3}, 10] \subset \mathbb{R}$
Dynamics sequence length	$n_f + 1$	$\{2, 3, 4, 10\}$
Auxiliary memory size	$N_{rollouts}$	$\{2, 3, 4\}$

TABLE IV: Tuned hyperparameters for each experiment.

Environment	Experiment	R	mb	le	lr	c_{ent}	lr_{aux}	c_{aux}	n_f	$N_{rollouts}$
Find	PPO(prop)	32	16	8	1.06×10^{-5}	0				
	PPO(prop-tactile)	32	16	8	1.06×10^{-5}	0				
	+full recon	64	64	4	7.39×10^{-5}	0	5.91×10^{-5}	0.0023		
	+tactile recon	64	16	8	1.36×10^{-5}	0.1	2.55×10^{-5}	0.004477		
	+full dynamics	64	64	4	1.15×10^{-5}	0.1	1.55×10^{-4}	0.0062	2	
	+tactile dynamics	64	64	4	2.32×10^{-5}	0	1.57×10^{-4}	0.0024563	4	
	+full dynamics+$N_{rollouts}$	64	64	4	1.15×10^{-5}	0.1	3.81×10^{-5}	0.1364	3	3
Bounce	PPO(prop)	32	32	4	5.93×10^{-5}	0				
	PPO(prop-tactile)	16	8	4	3.21×10^{-4}	0				
	+full recon	64	16	16	1.88×10^{-4}	0	2.77×10^{-5}	0.05669		
	+tactile recon	64	32	16	4.65×10^{-5}	0	5.13×10^{-5}	0.00384		
	+full dynamics	32	64	4	1.50×10^{-4}	0	4.53×10^{-5}	0.8462	10	
	+tactile dynamics	64	32	8	1.22×10^{-4}	0.05	1.64×10^{-4}	0.23547	10	
	+full dynamics+$N_{rollouts}$	32	64	4	1.50×10^{-4}	0	1.16×10^{-4}	0.19954	4	2
Baoding	PPO(prop)	32	8	4	9.96×10^{-5}	0.05				
	PPO(prop-tactile)	32	4	8	2.02×10^{-4}	0.05				
	+full recon	16	32	4	3.68×10^{-4}	0	5.18×10^{-5}	0.058866		
	+tactile recon	64	32	8	3.61×10^{-4}	0	1.00×10^{-5}	0.2707		
	+full dynamics	32	16	4	5.47×10^{-4}	0	2.87×10^{-4}	3.686	2	
	+tactile dynamics	32	16	8	2.08×10^{-5}	0.05	1.53×10^{-4}	0.04839	3	
	+full dynamics+$N_{rollouts}$	32	16	4	5.47×10^{-4}	0	1.67×10^{-5}	1.6349	4	4

XV. LATENT TRAJECTORY ANALYSIS

Figures 14, 15, and 16 show a two dimensional latent representation of a single episode across all environments. Trajectories for RL-only and a subset of self-supervised agents are shown (tactile reconstruction, full dynamics, and tactile dynamics). The 256-dim z_t latent vector at each timestep was reduced using 2-component Principal Component Analysis (PCA). Note that the tactile activations shown are only the sum of activations in the current observation, and does not sum the history.

Baoding. The ring-like trajectory of the RL-only agent illustrates the repeated motion the agent develops. There are two tactile peaks on opposite sides of the ring, indicating symmetry in contact activations between half-rotations. The trajectory of the tactile reconstruction agent is quite different (heart-shaped, diffuse). This shows each rotation is slightly different, and there is now asymmetry between the contact activations of half-rotations. From rendering the policy, the gait is smooth like the dynamics agent but keeps the balls close together like the RL-only agent. The trajectory of the full dynamics agent is again ring-like, but with tighter bounds than the RL-only agent. Like the tactile reconstruction agent, there is contact activation asymmetry between half-rotations. Finally, the tactile dynamics agent trajectory appears to be a blend of the dynamics and tactile reconstruction trajectories.

Bounce. The latent trajectories of the self-supervised agents are highly different to the RL-only agent. From the trajectory with the time colourbar, we can see that the sequential latent states of the RL-only agent are highly discontinuous and far apart (e.g., yellow), and the agent repeats the same motion with high precision. There are two regions with non-zero tactile observations of upto 6 activations, which is understood by the 'safe' gait of raising the index and pinky finger to stabilise the ball. The gait changes completely for the self-supervised agents, which predominately uses 1 or 2 contacts. Sequential latent states are still spread out in various regions, but these regions are much more diffuse than in the RL-only.

Find. It is clear the self-supervised agents find the object faster by observing the trajectories colourised by time. Otherwise, the shape of the latent trajectories is not drastically different between RL-only and self-supervised agents. A distinction between 1 and 2 tactile activations appears in the dynamics agent trajectory.

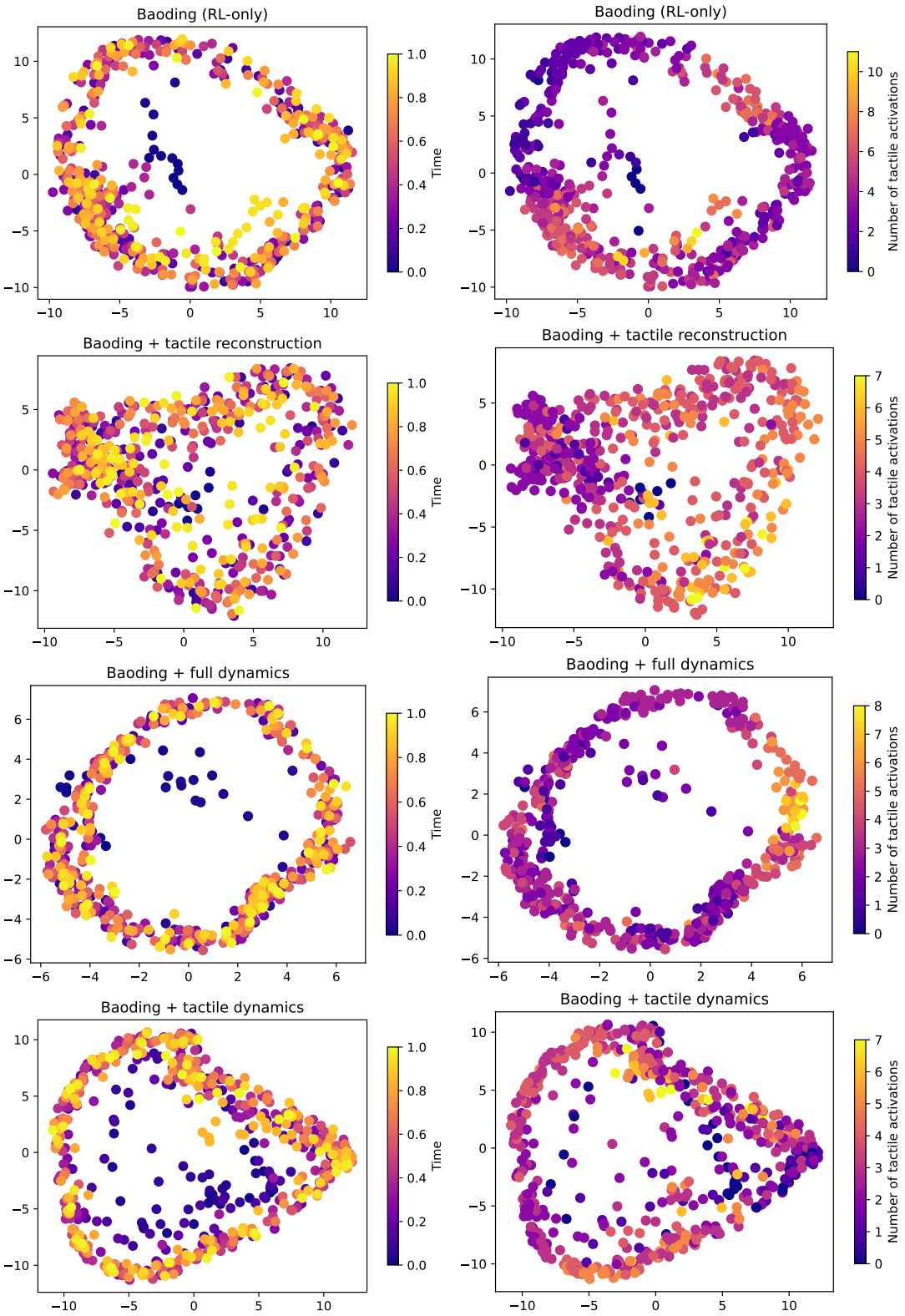


Fig. 14: Latent episode trajectory (PCA) of the best *Baoding* agents. **Left:** Samples colourised by time. **Right:** Samples colourised by summed tactile activations of the last tactile observation o_t^{tact}.

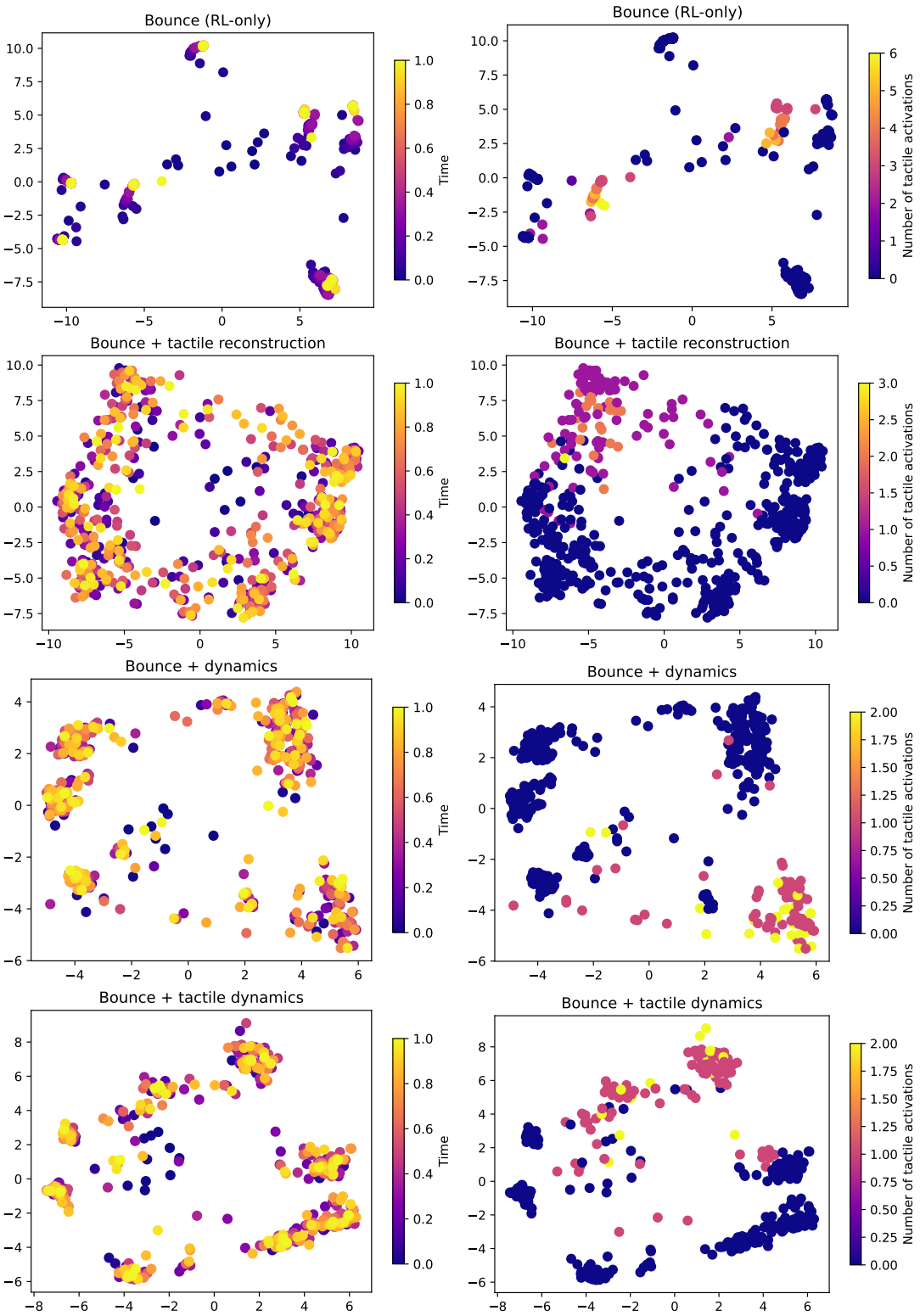


Fig. 15: Latent episode trajectory (PCA) of the best *Bounce* agents. **Left:** Samples colourised by time. **Right:** Samples colourised by summed tactile activations of the last tactile observation o_t^{tact}.

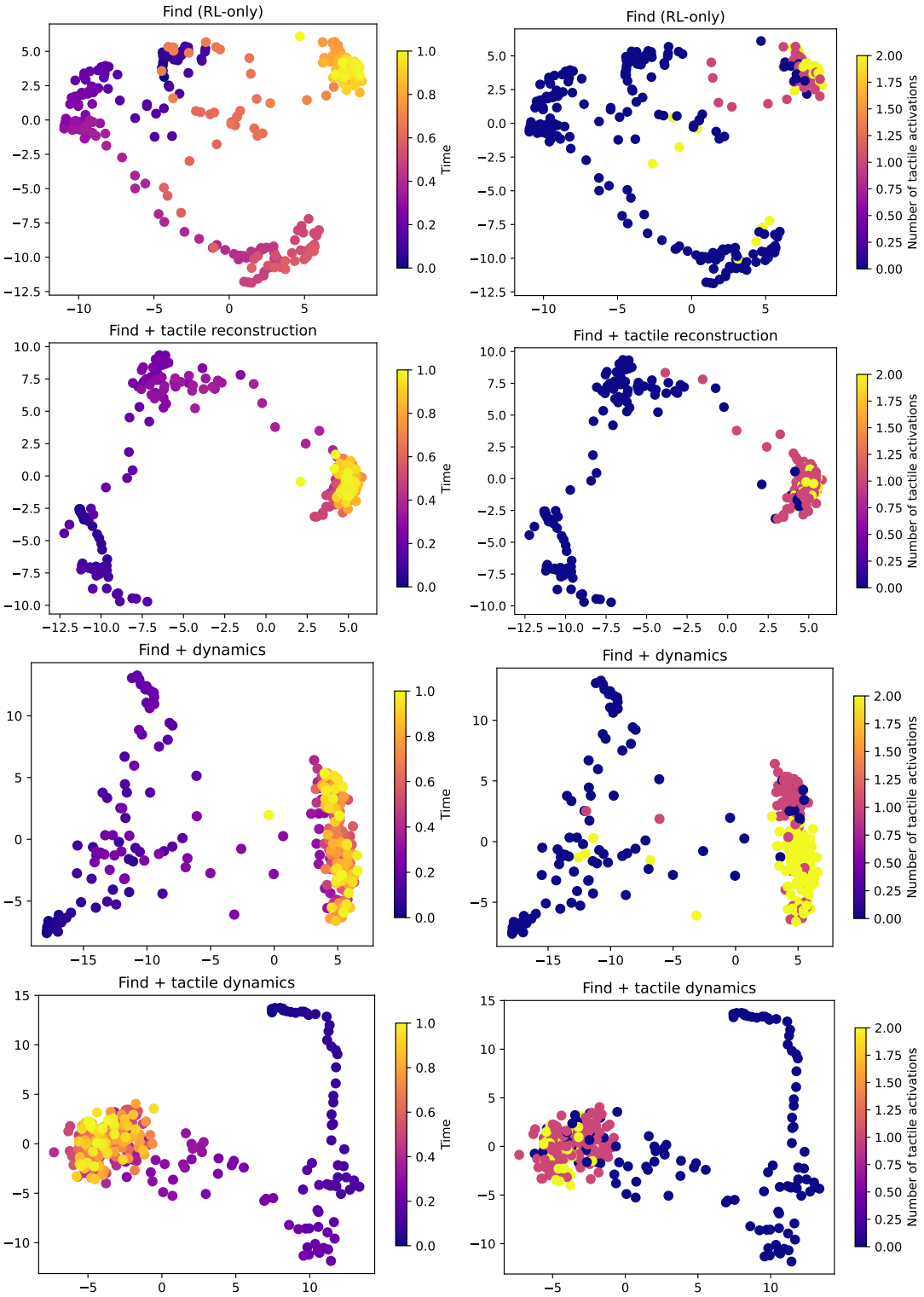


Fig. 16: Latent episode trajectory (PCA) of the best *Find* agents. **Left:** Samples colourised by time. **Right:** Samples colourised by summed tactile activations of the last tactile observation o_t^{tact}.