S-RL Toolbox: Environments, Datasets and Evaluation Metrics for State Representation Learning

Antonin Raffin, Ashley Hill, René Traoré, **Timothée Lesort**, **Natalia Díaz**, David Filliat,

{firstname.lastname}@ensta-paristech.fr

U2IS, ENSTA ParisTech, Université Paris Saclay, Palaiseau, France. Autonomous systems and Robotics http://asr.ensta-paristech.fr/, Inria FLOWERS team https://flowers.inria.fr/

State Representation Learning

State representation learning (SRL) [2] aims at learning compact representations from raw observations in robotics and control applications. Approaches used for this objective are auto-encoders, learning forward models, inverse dynamics or learning using generic priors on the state characteristics. However, the diversity in applications and methods makes the field lack standard evaluation datasets, metrics and tasks. We provide a set of environments, data generators, robotic control tasks, metrics and tools to facilitate iterative state representation learning and evaluation in reinforcement learning (RL) settings.

SRL Methods and RL Algorithms

Using RL notation, SRL corresponds to learning a transformation ϕ from observation o_t to state s_t. Then we learn a policy π that takes state s_t as input and outputs action a_t

$$o_t \xrightarrow{\phi} s_t \xrightarrow{\pi} a_t$$

S-RL toolbox provides the following **SRL methods**:

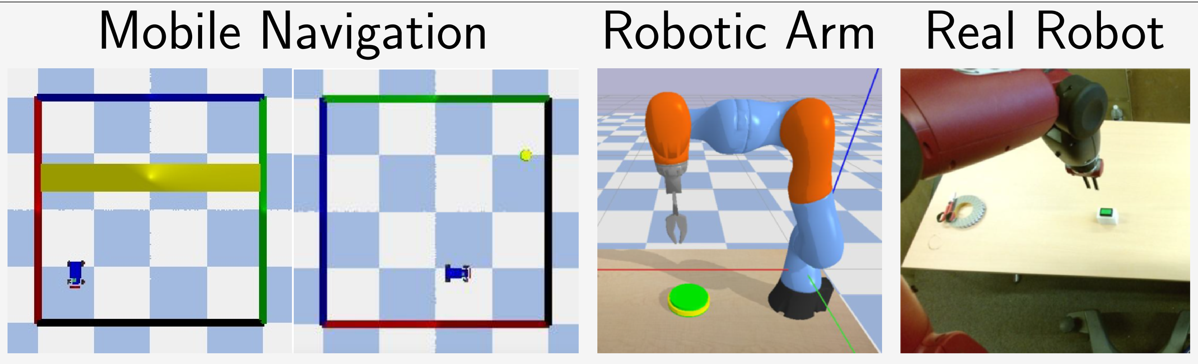
- **Auto-Encoders:** Compress the observation into a low dimensional state that is sufficient to reconstruct the observation.
- **Forward Models:** Integrate dynamics of the world by predicting state s_{t+1} given state s_t and action a_t.
- **Inverse Models:** Predict action a_t given two successive states s_t and s_{t+1}.
- **Robotic Priors:** Encode prior knowledge about the world as losses.
- **Hybrid Approaches:** *Combined* and *Split* SRL models: Combine the objectives above by averaging them on a single embedding or using splits of the state representation.

Baselines SRL methods to compare with include **Raw Pixels**, **Ground Truth**, **Supervised learning** and **Random Features**.

SRL Datasets and Environments

A set of environments with incremental difficulty was designed to assess SRL algorithms for robotics control. They all follow the interface defined by OpenAI Gym, which makes integration with RL algorithms easy and fast:

Mobile Navigation | Robotic Arm | Real Robot



Qualitative Evaluation

Real-time SRL | Interactive scatter | Latent visualization

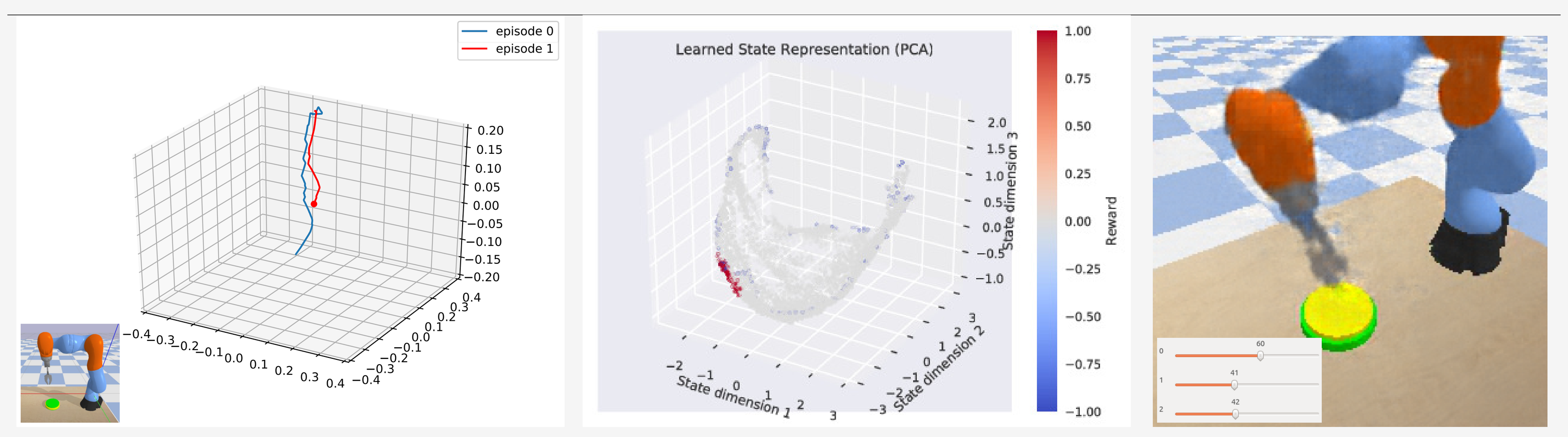


Figure 1: Visual tools for analysing SRL; Left: Live trajectory of the robot in the state space. Center: 3D scatter plot of a state space; clicking on any point displays the corresponding observation. Right: reconstruction of the point in the state space defined by the sliders (See videos).

Quantitative Evaluation

The visualization tools are quite useful for low-dimensional spaces. However, for state spaces with a high number of dimensions, looking at the correlation matrix becomes impractical. Therefore, we introduce the Ground Truth Correlation (GTC) metric that allows to compare the model's ability to encode relevant information:

$$GTC_{(i)} = \max_j |\rho_{s,\tilde{s}}(i,j)| \in [0,1] \qquad (1)$$

with $i \in [\![0, |\tilde{s}|]\!]$, $j \in [\![0, |s|]\!]$, $\tilde{s} = [\tilde{s}_1; \ldots; \tilde{s}_n]$, and $\tilde{s}_k$ being the k^{th} dimension of the ground truth state vector. The mean of GTC allows to compare learned states using one scalar value: $GTC_{mean} = \mathbb{E}[GTC]$.

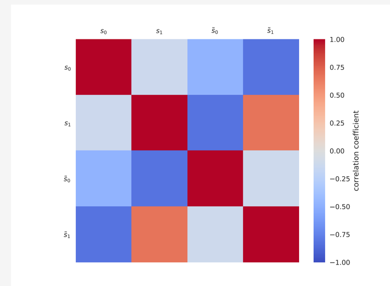


Figure 2: Correlation matrix for mobile robot navigation dataset (static target)

Learned states can also be evaluated with 8 **RL algorithms** trained on them:

- **A2C, ACKTR, ACER, DQN, DDPG, PPO1, PPO2, TRPO** from *Stable-Baselines* [1] -a fork of OpenAI baselines.
- Augmented Random Search (**ARS**), Covariance Matrix Adaptation Evolutionary Strategy (**CMA-ES**) and Soft Actor Critic (**SAC**).

Experiments

Ground Truth States | Learned States | RL Performance

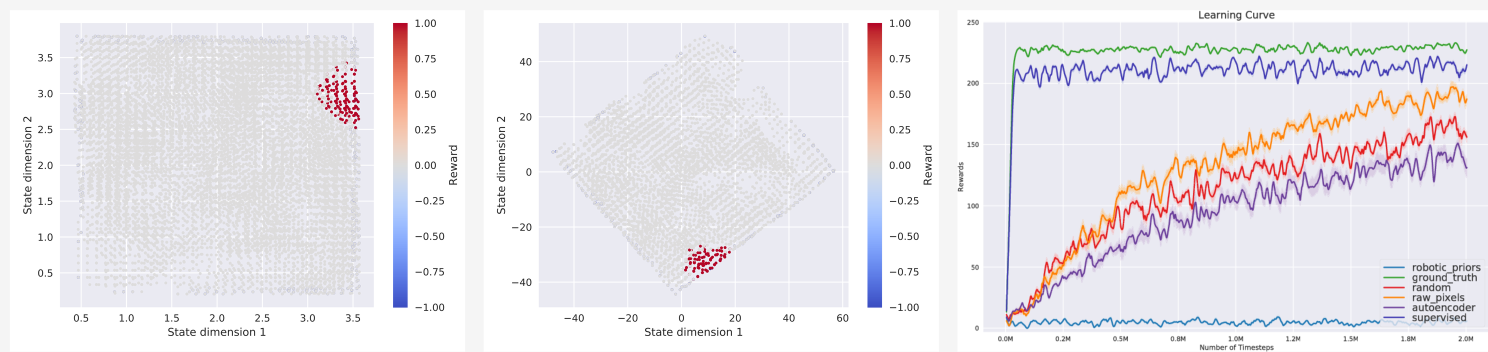


Table 1: Ground truth states (left), states learned (Inverse and Forward) (center), and RL performance evaluation (10 runs mean and std error on PPO) (right) in the mobile robot environment. Colour denotes the reward, red for positive, blue for negative and grey for null reward (left and center).

GroundTruthCorrelation	x_{robot}	y_{robot}	x_{target}	y_{target}	*Mean*	*RL*
Robotic Priors	0.2	0.2	0.41	0.66	0.37	5.4 ± 3.1
Random	0.68	0.65	0.34	0.31	0.50	163.4 ± 10.0
Supervised	0.69	0.73	0.70	0.72	0.71	213.3 ± 6.0
Auto-Encoder	0.52	0.51	0.24	0.23	0.38	138.5 ± 12.3
GT	1	1	1	1	1	229.7 +- 2.7

Table 2: GTC and GTC_{mean}, and mean reward performance in RL (using PPO) per episode after 2 millions steps, with standard error for each SRL method in mobile robot with random target environment.

Conclusion

S-RL toolbox facilitates fast iteration, interpretability and reproducibility, with a set of qualitative and quantitative metrics and interactive visualization tools for robotics control on SRL methods.

- **Repo**: https://github.com/araffin/robotics-rl-srl

Acknowledgement: This work is supported by the EU H2020 DREAM project (Grant agreement No 640891) http://www.robotsthatdream.eu

References

[1] A. Hill, A. Raffin, R. Traoré, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu.
Stable baselines.
https://github.com/hill-a/stable-baselines, 2018.

[2] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat.
State representation learning for control: An overview.
Neural Networks, 2018.