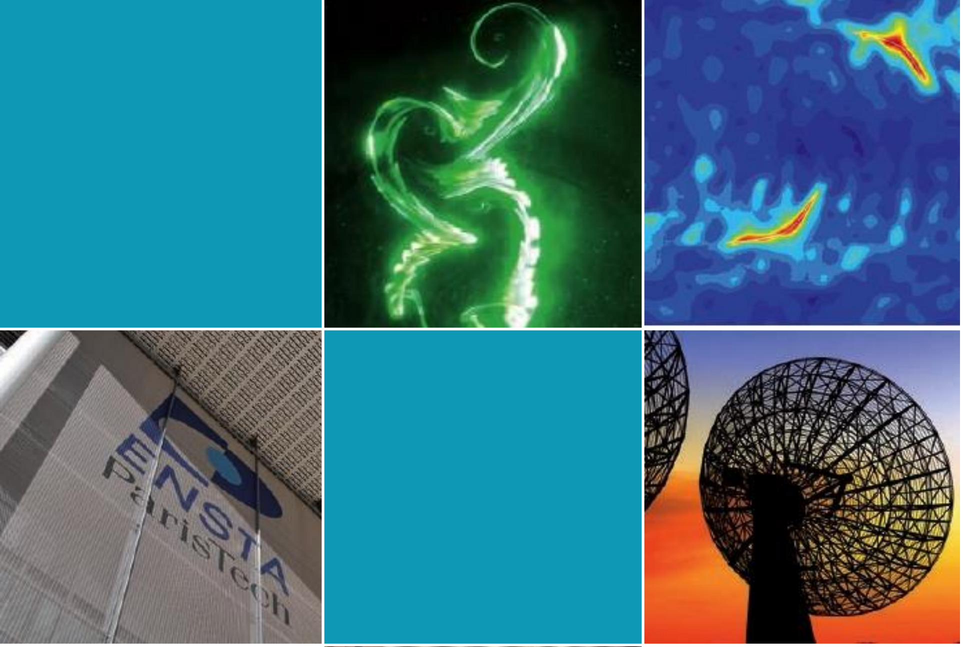


Sciences et techniques avancées

Improving data efficiency for machine learning in robotics

David Filliat
U2IS
ENSTA Paris

École Nationale Supérieure de Techniques Avancées

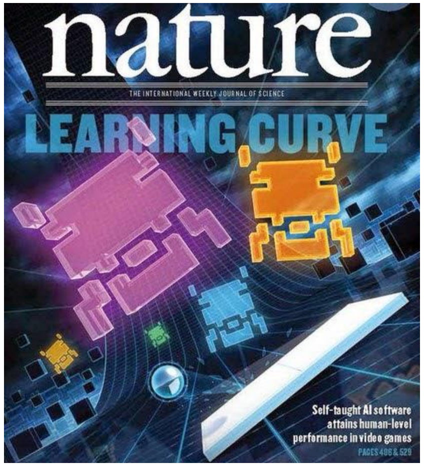
ENSTA
IP PARIS

Machine Learning and Robotics

Machine Learning is important for robotics / intelligent vehicles

- Perception
  - Vision, Lidar, Multi-sensor systems
  - Object/obstacle detection, road/path detection, sensor fusion, mapping, localization…
- Decision
  - Trajectory prediction, manoeuvre prediction
  - Risk estimation
- Action
  - Learning model for MPC / tuning ‘classical’ controllers
  - Imitation learning (e.g.: end to end driving)
  - Reinforcement learning

RL in Robotics vs Games



Deep Reinforcement Learning

- Big (cheap) data
- Slow Learning (millions of interactions with environment ➔ simulation)
- Learn one task defined by researcher
- Quite unstable (hyperparameters, …)

Reinforcement Learning for robotics

- Little (expensive) data
- Would need fast incremental learning during interaction with real world
- Learn multiple tasks
- No researcher → autonomous learning
- Needs to be stable, robust

Machine Learning and Robotics

Robotics constraints

- Data are expensive (vs games), robots are slow, break easily
- Search (behaviour) space are huge, enough (iid) data difficult to gather
- Incremental learning, multi-tasks learning …

How to improve efficiency of machine learning on real robots ?

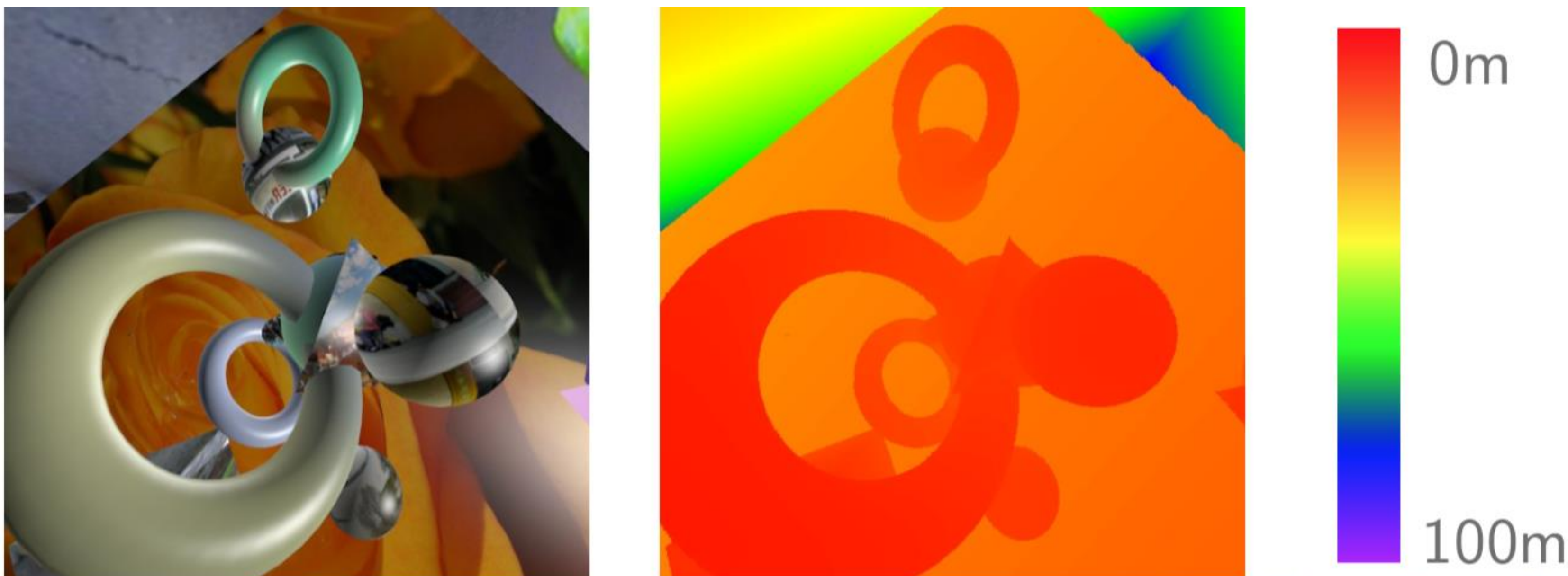
- Learn in simulation and transfer to real life
- Use auxiliary tasks to accelerate learning in real life
- Learn compact representation to accelerate learning in real life

All of the above ?

Exploiting simulated data

Estimating depth from video

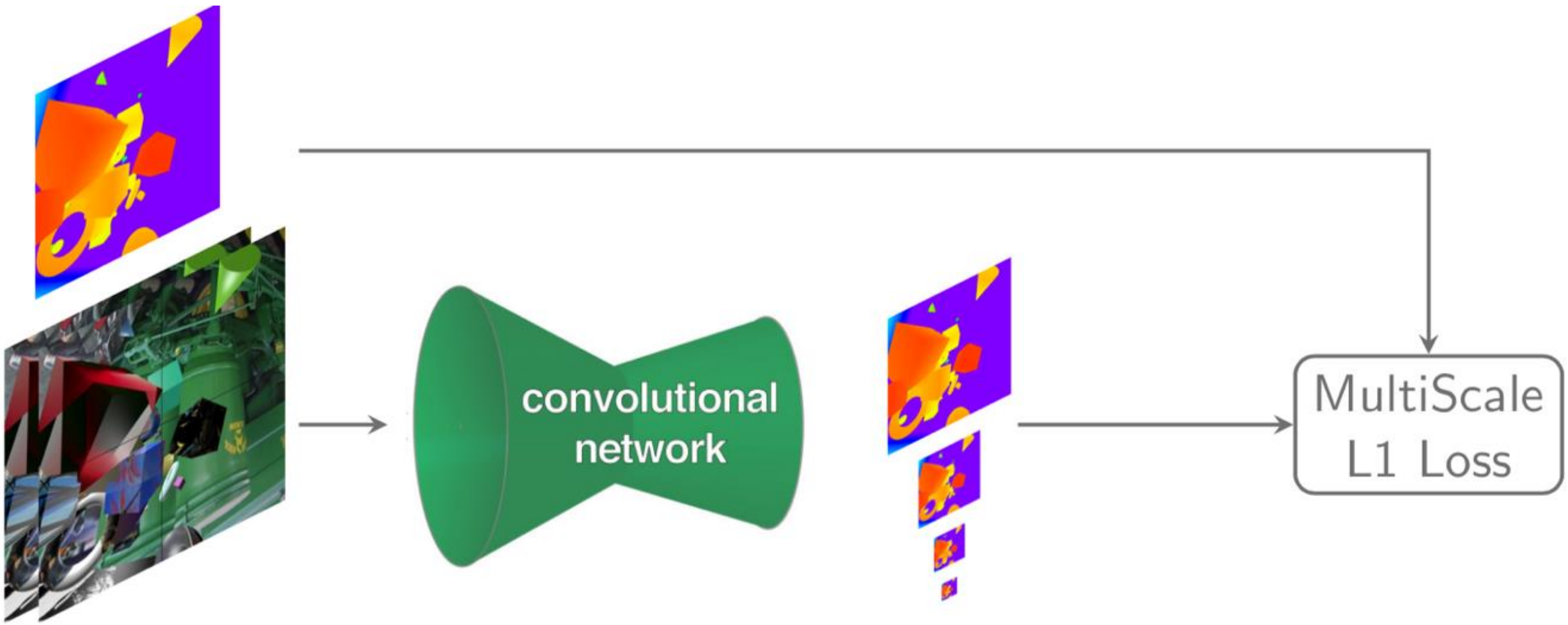
- Input multiple images / randomize appearance -> focus on optical flow
- Exploit stabilized orientation of UAV -> simplify depth prediction
- Generate database in simulation with randomized shapes/textures



Exploiting simulated data

Estimating depth from video

- Train a neural network to predict depth (supervised)



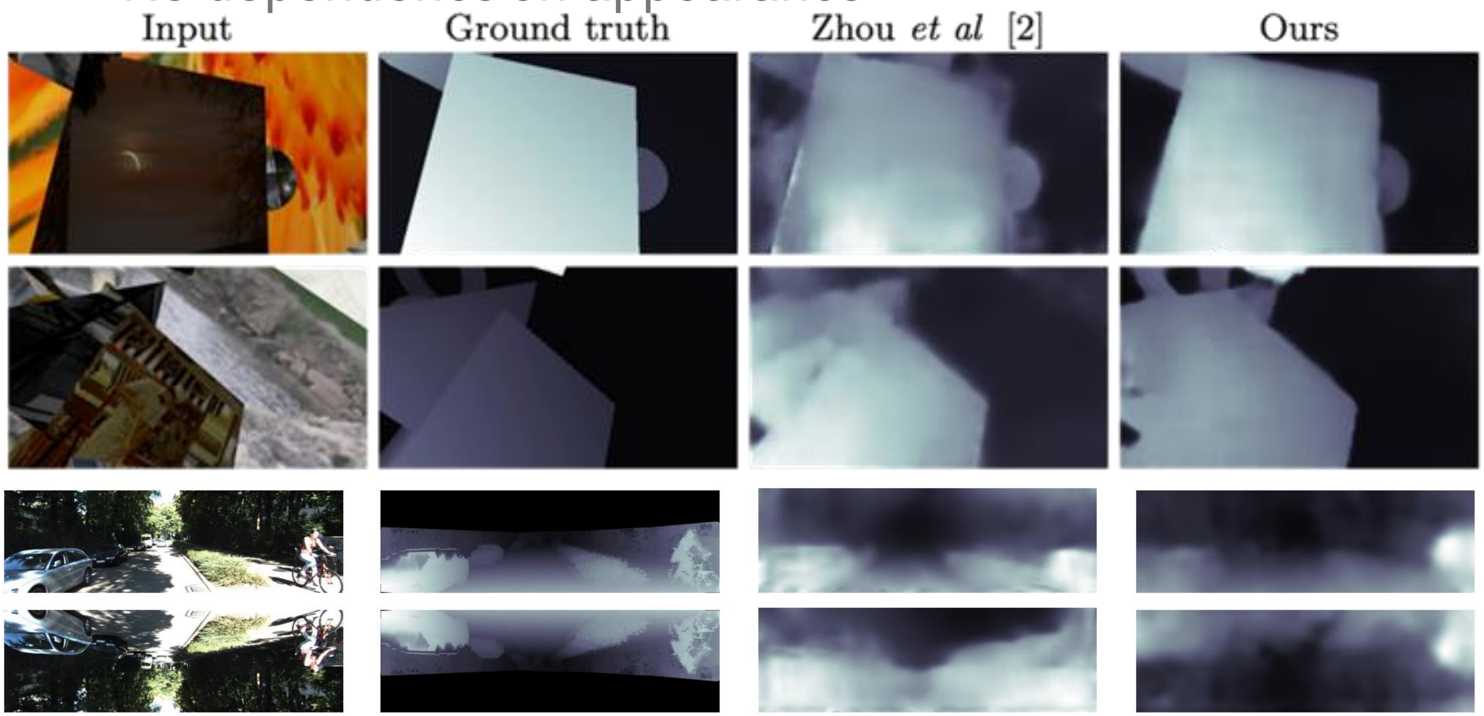
- CNN regularizes around POE and uniform areas

[Pinard & al., ECMR17]

Exploiting simulated data

Estimating depth from video

- Additional unsupervised fine tuning using photometric consistency for transfer on real data
- No dependence on appearance



[Pinard & al., ECCV18]

Exploiting simulated data

Transferring from simulation to reality

- Pinard & al. is just an example in a simple transfer case
- Many other techniques :
  - Domain randomization
  - Domain adaptation
  - Learning residual models of sim/real differences
  - …
- See e.g., https://twitter.com/sim2realAIorg

Using auxiliary tasks

End to end driving

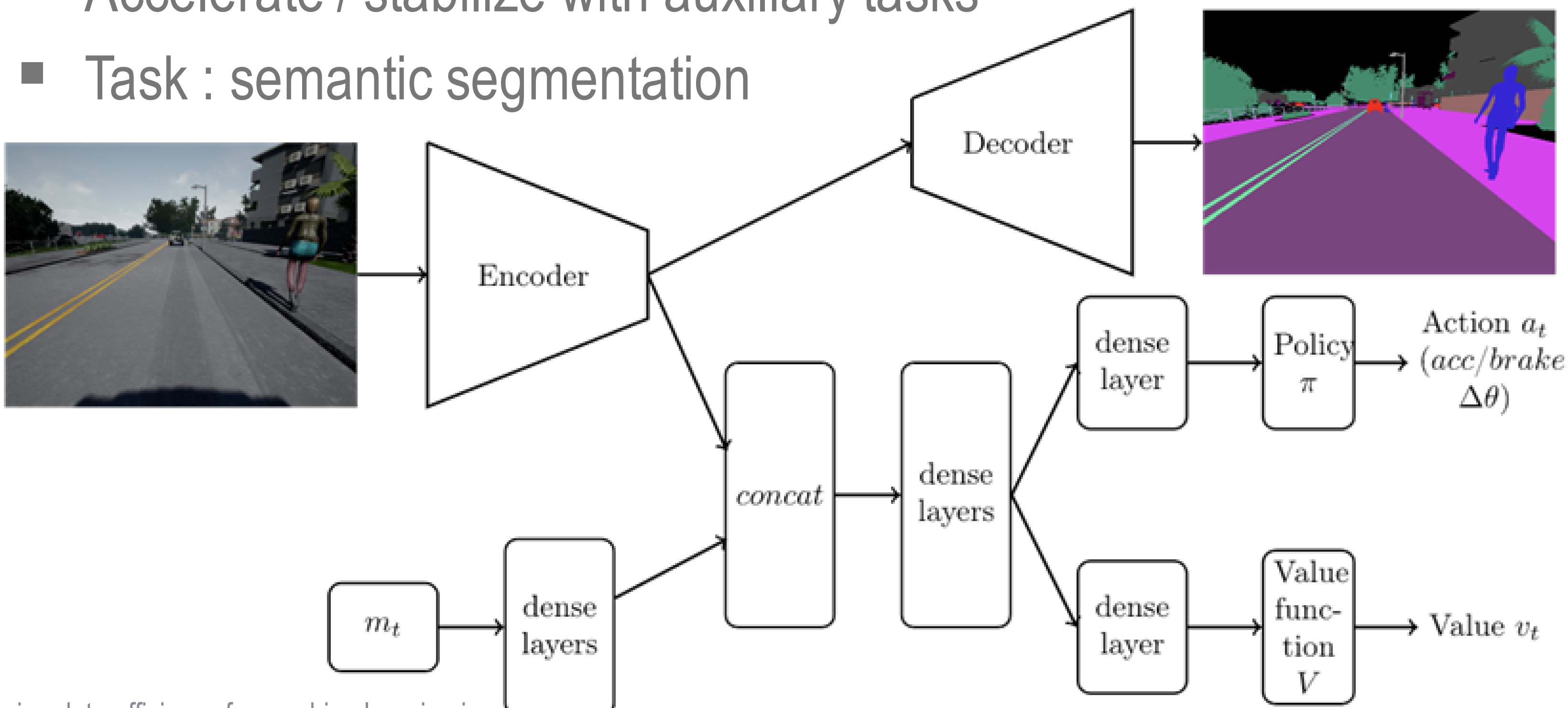
- Learn to follow road, avoid obstacles, negociate intersections from image
- Using CARLA simulator

Using auxiliary tasks

Learning in simulation

[Carton et al. 21]

- Perform Reinforcement Learning (PPO) in simulation
- Learn action choice to maximize long term reward
- Accelerate / stabilize with auxiliary tasks
- Task : semantic segmentation



Using auxiliary tasks

Auxiliary tasks improve generalization

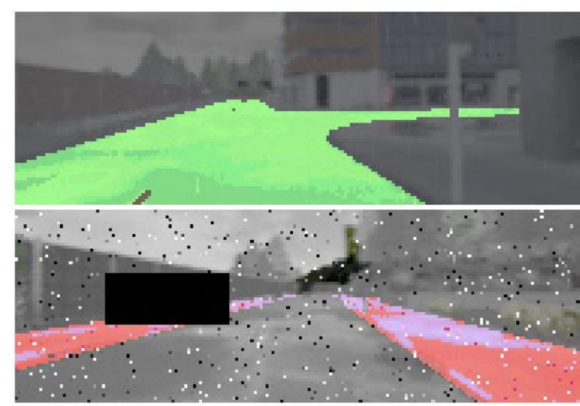
- Simulation to simulation transfer with different conditions

	Training Condition	New Weather	New Town	New Town & Weather
RL [9]	14%	2%	3%	6%
CIRL [16]	**93%**	86%	53%	68%
Auxiliary task	90%	**92%**	**78%**	**68%**

- Better than (semantic) data augmentation and pre-training

Training	Training conditions	New weather	New town	New Town & weather
No da	34%	6%	9%	2%
Classic da	57%	60%	22%	4%
Da w/ seg	67%	60%	34%	28%

Training	Training conditions	New weather	New town	New Town & weather
Pretraining	82%	**98%**	49%	40%
Auxiliary task	**90%**	92%	**78%**	**68%**



Using auxiliary tasks

Auxiliary tasks improve generalization

- Simulation to simulation transfer with different conditions

Training conditions

New town, new weather

Using auxiliary tasks

Auxiliary tasks

- Semantic segmentation improves stability, training speed, generalization
- But supervised task, not so useful in real life
- Would be better with unsupervised tasks (i.e., without human labels)

=> State Representation Learning

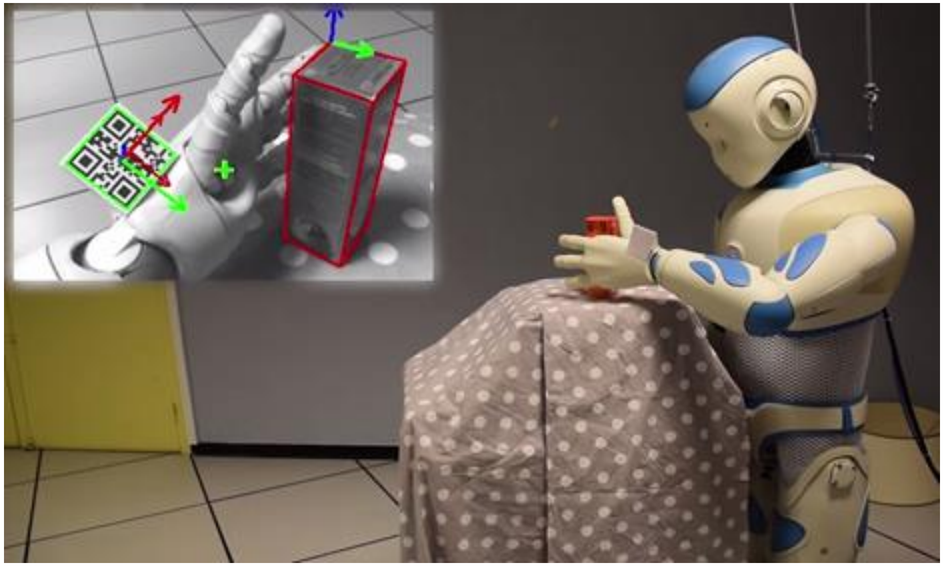
States ?

Often, robot controllers require simple, 'high-level', low dimension inputs (the 'state' of the robot/world)

- E.g., grasping: object position, gripper position
  driving: road direction, obstacle positions, …

Vision based control requires filtering to get this information

- Many solutions, often hand-crafted, task specific



Why learning states ?

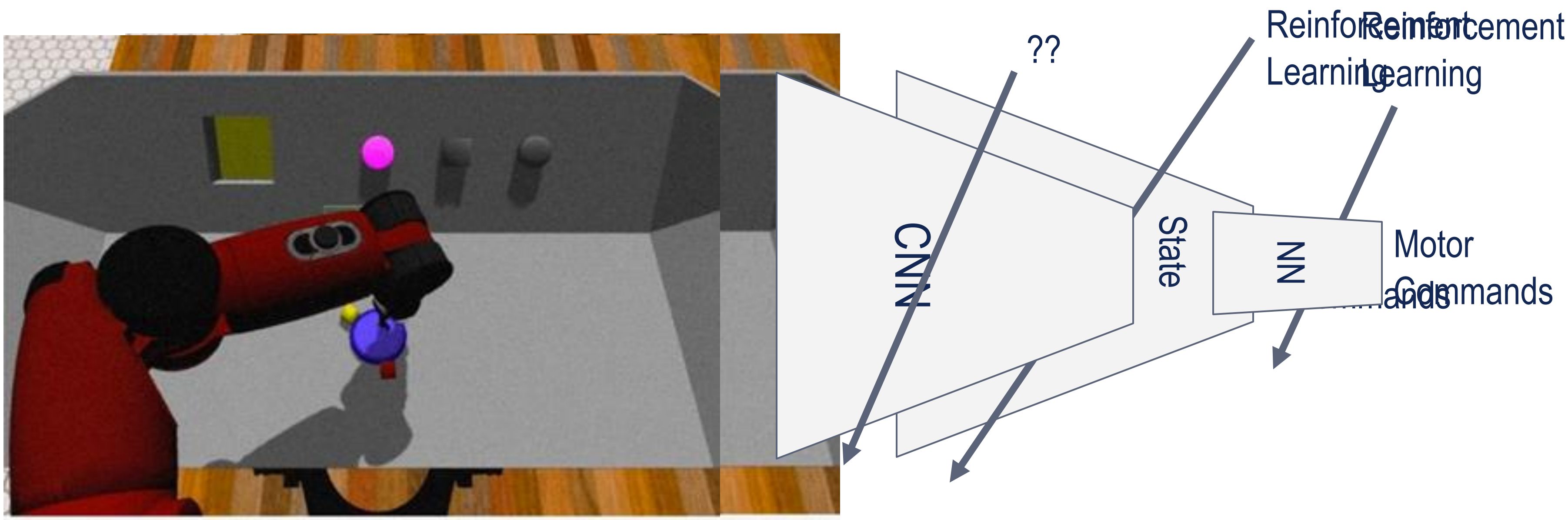
DREAM approach
[Doncieux et al., FiN18]

Facilitate adaptation to new task

- Discover the relevant state from exploration/demonstrations

Controllers are easier to train in such lower dimension

- Possibly faster than end-to-end; Could help transfer across tasks

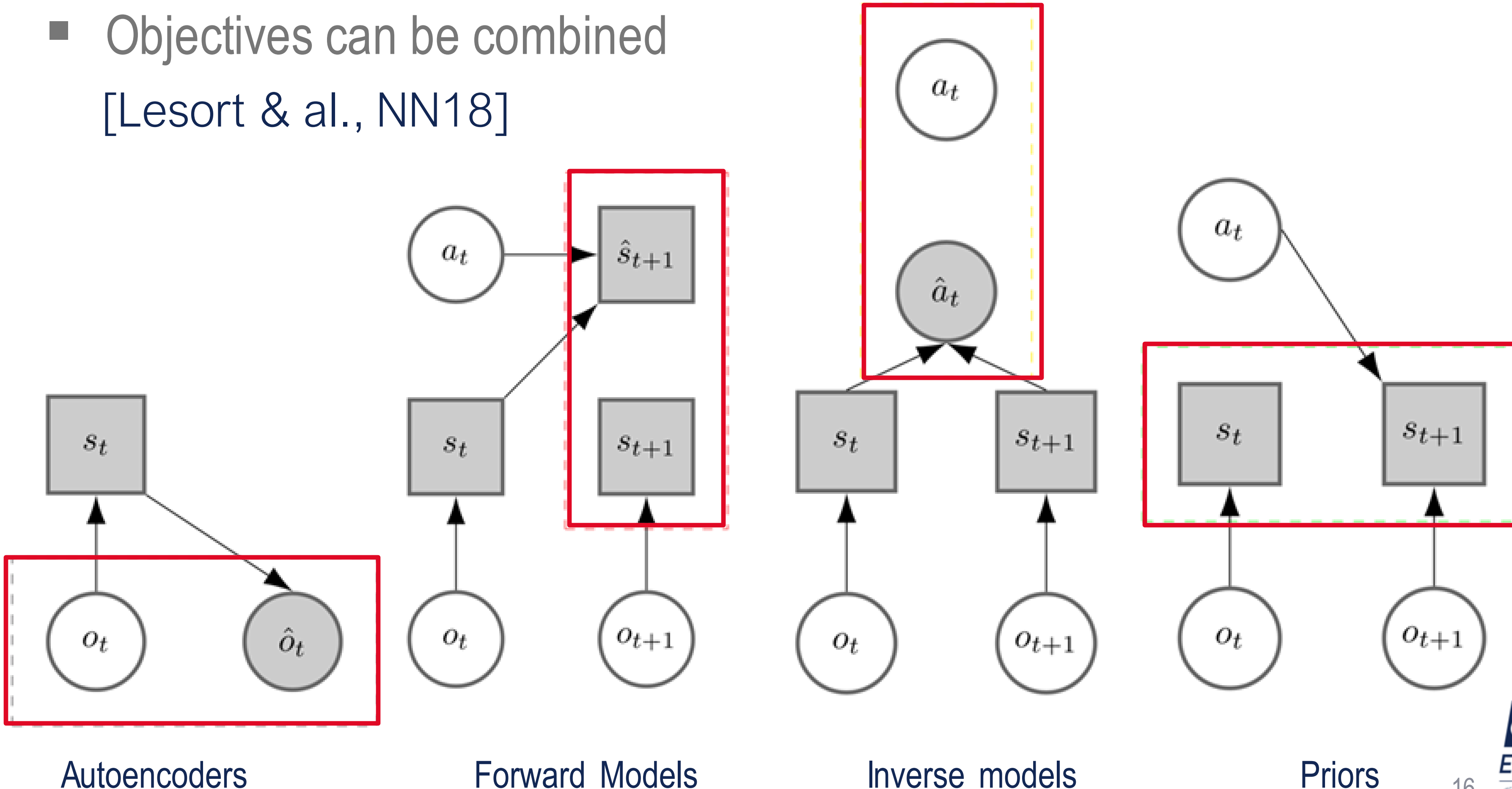


SRL approaches

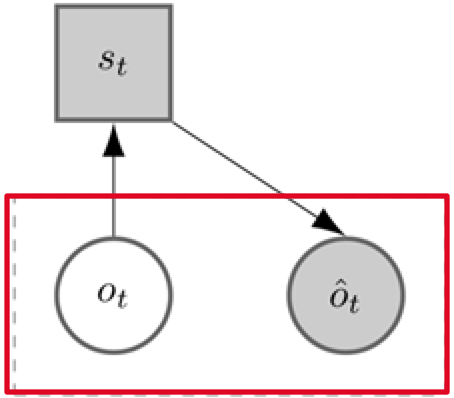
Learning state representation using self-supervision

- Several objectives can be exploited without human labelling
- Objectives can be combined

[Lesort & al., NN18]



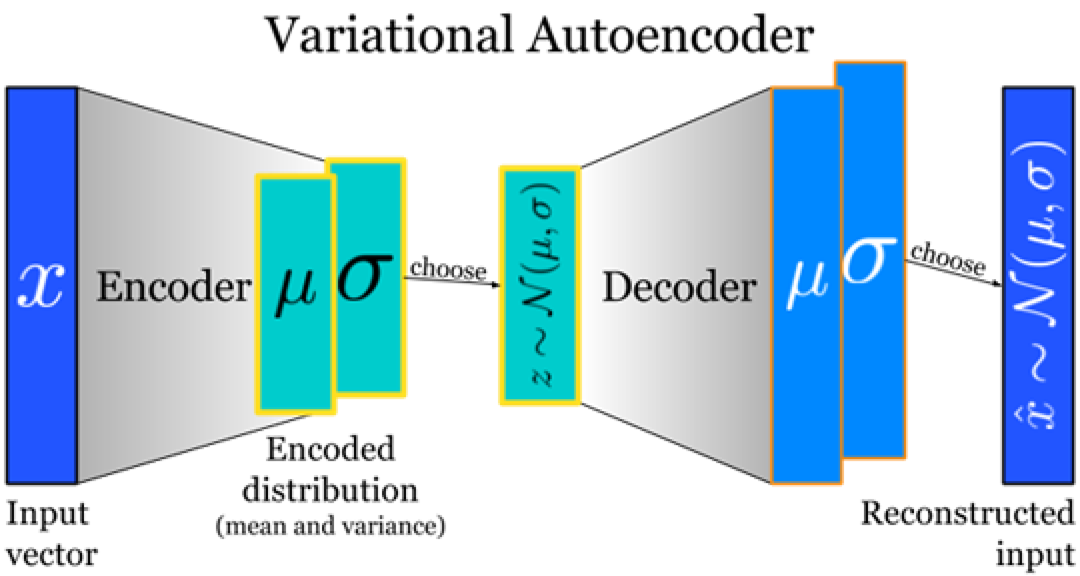
Reconstructing the observation



Train a state that is sufficient for reconstructing the input observation

- AE, DAE, VAE
- (Bi)GANs

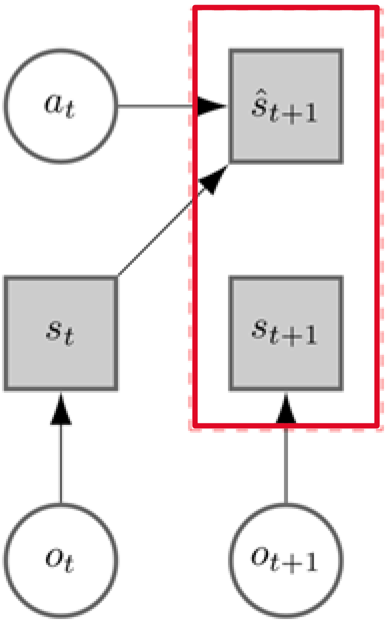
Downside: sensitive to irrelevant variations (wrt actions)



Forward models

Find state from which it is easy to predict next state

- Additional constraints to avoid fixed representations (AE, triplet loss…)
- Impose constraints on forward model (e.g., linear model)



Naturally discard irrelevant features

Model may be useful

- in model based RL
- In planning

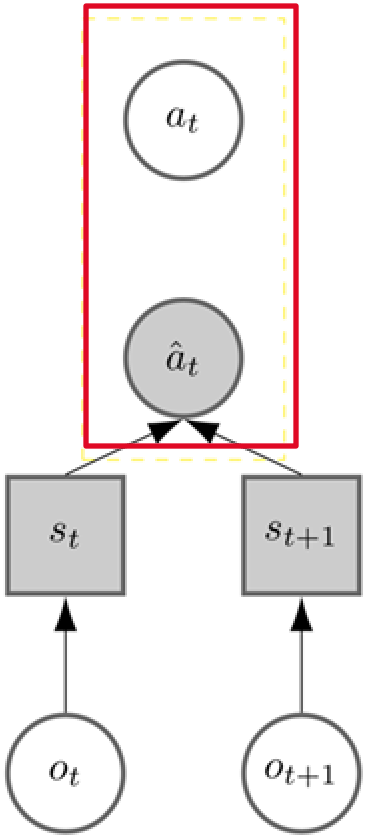
Inverse models

Find a state sufficient to recover action from 2 observations

- Impose constraints on model (e.g., linear model)

Focus on states that can be controlled

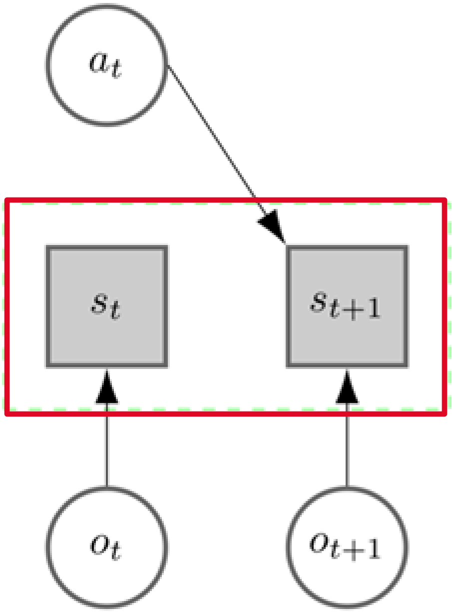
Useful for a direct control model



Prior models

Encode high-level constraints on the states

- Temporal continuity
- Controllability
- Inertia
- etc….



May exploit rewards

Robotic Priors

[Jonschkowski et. al. 2015]

Use *a priori* knowledge to learn representations relevant to the task

- **Temporal coherence Prior:** *Two states close to each other in time are also close to each other in the state representation space.*

$$L_{Temp}(D, \hat{\phi}) = \mathrm{E}[\| \Delta \hat{s}_t \|^2] \ , \tag{1}$$

- **Proportionality Prior:** *Two identical actions should result in two proportional magnitude state variations.*

$$L_{Prop}(D, \hat{\phi}) = \mathrm{E}[(\| \Delta \hat{s}_{t_2} \| - \| \Delta \hat{s}_{t_1} \|)^2 | a_{t_1} = a_{t_2}] \ , \tag{2}$$

- **Repeatability Prior:** *Two identical actions applied at similar states should provide similar state variations, not only in magnitude but also in direction.*

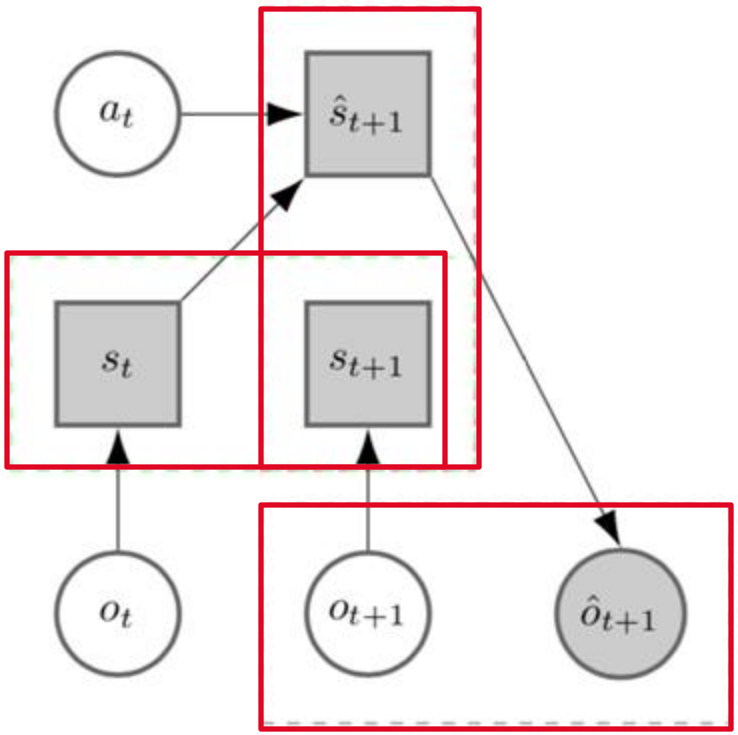
$$L_{Rep}(D, \hat{\phi}) = \mathrm{E}[e^{-\|\hat{s}_{t_2} - \hat{s}_{t_1}\|^2} \| \Delta \hat{s}_{t_2} - \Delta \hat{s}_{t_1} \|^2 | \ a_{t_1} = a_{t_2}] \ , \tag{3}$$

- **Causality Prior:** *If two states on which the same action is applied give two different rewards, they should not be close to each other in the state representation space.*

$$L_{Caus}(D, \hat{\phi}) = \mathrm{E}[e^{-\|\hat{s}_{t_2} - \hat{s}_{t_1}\|^2} \ | \ a_{t_1} = a_{t_2}, r_{t_1+1} \neq r_{t_2+1}] \ , \tag{4}$$

Mixing objectives

Integrating several approaches



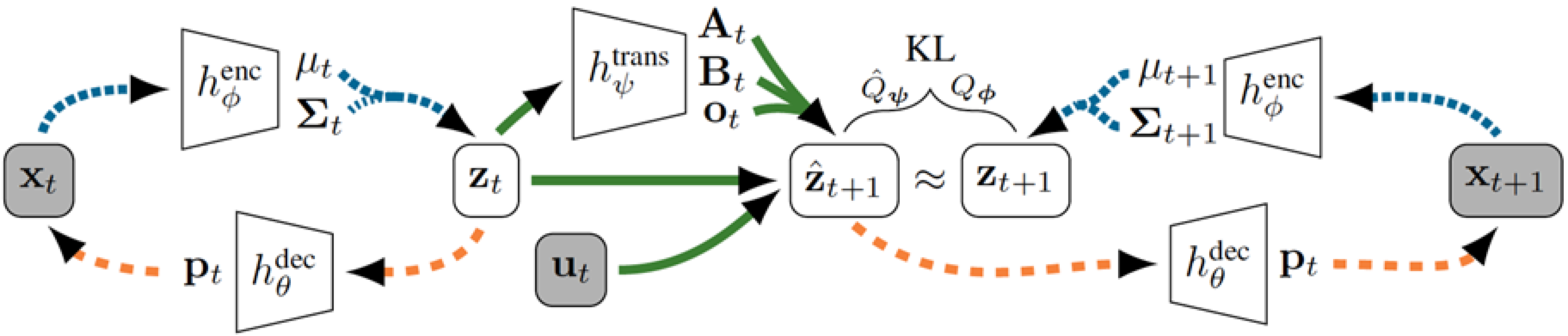
Embed to Control (E2C)

[Watter'18]

Multiple objectives

- Reconstruct observation using VAE
- Learn a locally linear forward model
- Exploit this forward model in optimal control setting

$$\hat{s}_{t+1} \sim \mathcal{N}(\mu = W * \hat{s}_t + U * a_t + V, \sigma)$$



SRL : state of the art

[Lesort et al, NN18]

State Representation Learning for Control: An Overview

Timothée Lesort[1, 2], Natalia Díaz-Rodríguez[1], Jean-François Goudou[2], and David Filliat[1]

[1]U2IS, ENSTA ParisTech, Inria FLOWERS team, Université Paris Saclay, Palaiseau, France.,
{timothee.lesort, natalia.diaz, david.filliat}@ensta-paristech.fr
[2]Thales, Theresis Laboratory, Palaiseau, France.,
{jean-francois.goudou@thalesgroup.com

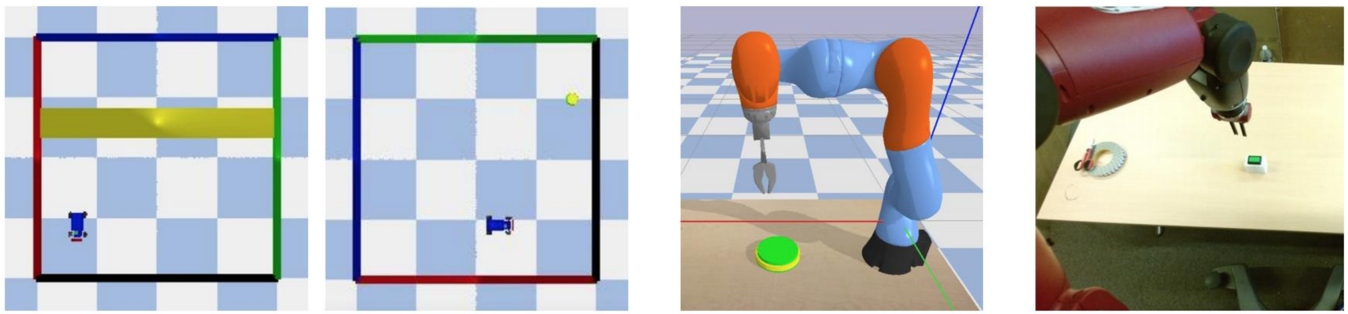
Contents

State representation learning Toolbox

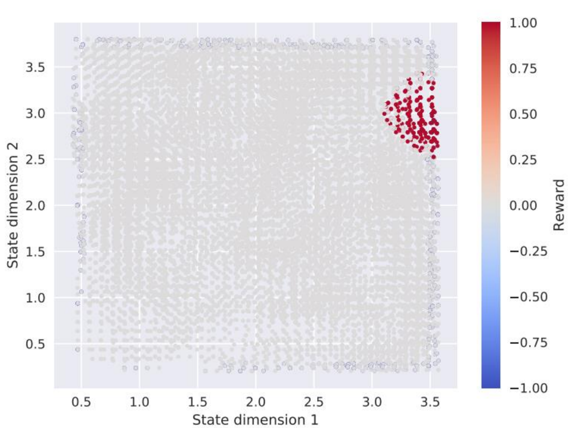
SRL Toolbox https://github.com/araffin/robotics-rl-srl

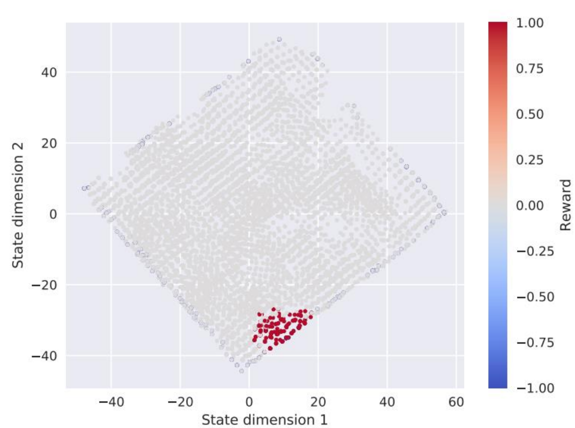
[Lesort et al. 18]
[Raffin et al. 18]]

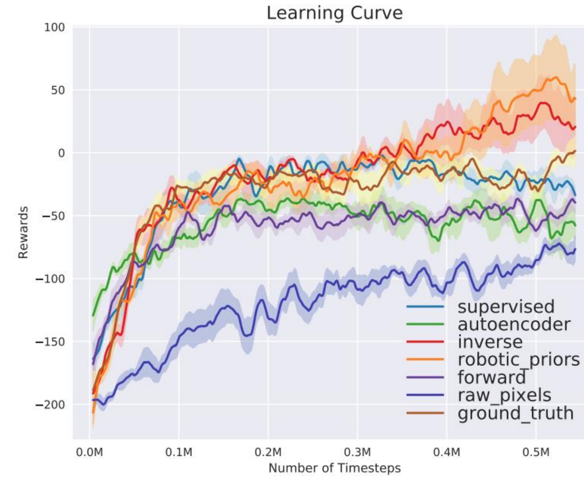
- Set of environments



GT states (Env. 2) | Learned States | RL Performance







ENSTA IP PARIS

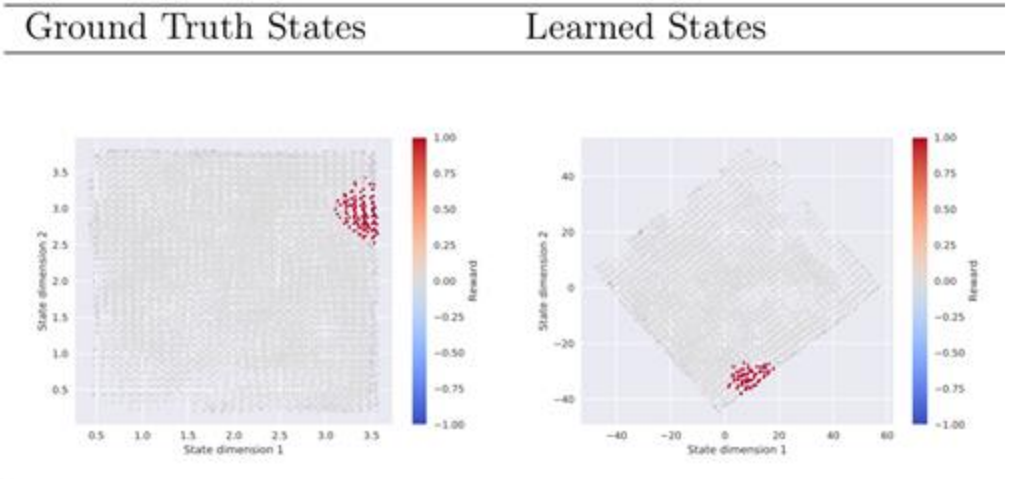
SRL Toolbox

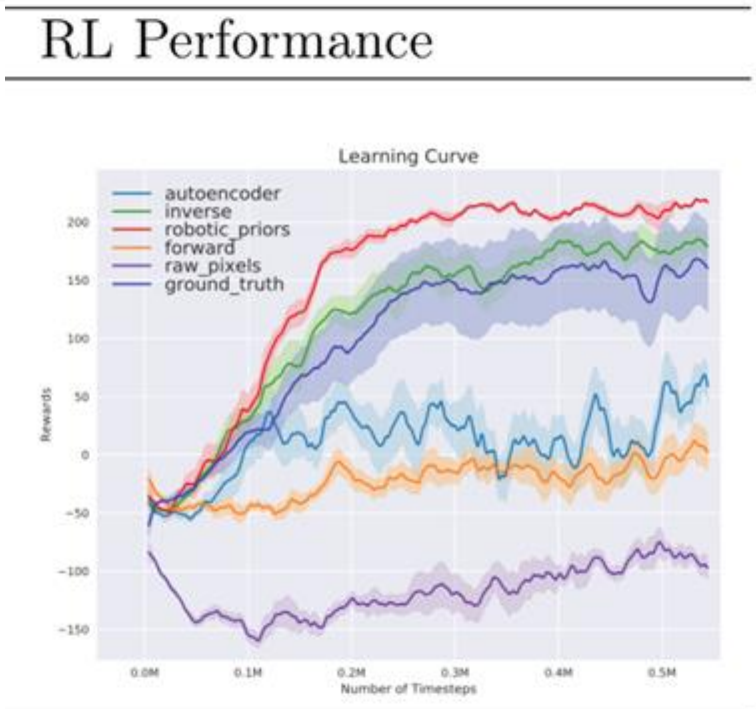
A set of baselines

- Auto Encoders
- Variational Auto Encoders
- Robotic priors
- Forward Models
- Inverse models

A set of evaluation tools

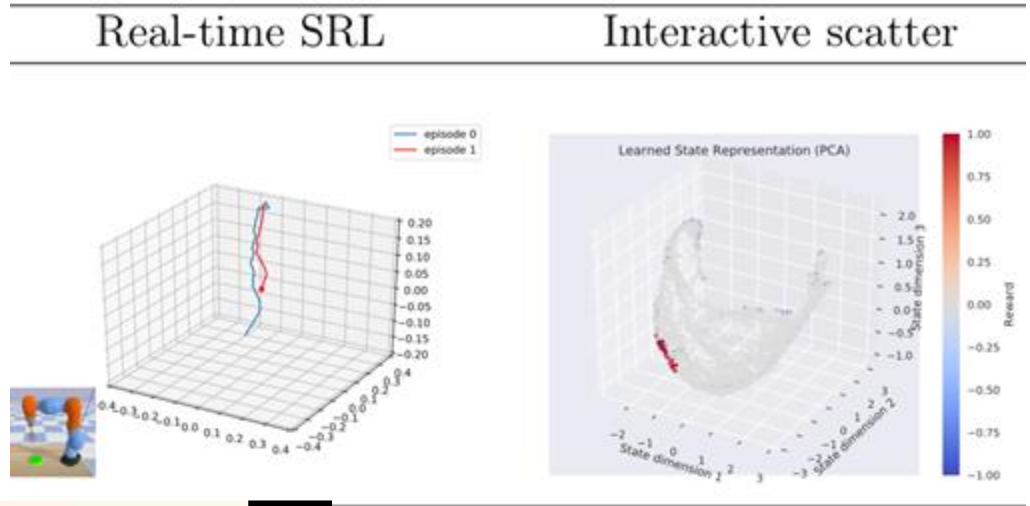
- RL (Stable Baselines)
- PPO, CMA-ES, ARS, …
- KNN-MSE
- Ground truth correlation



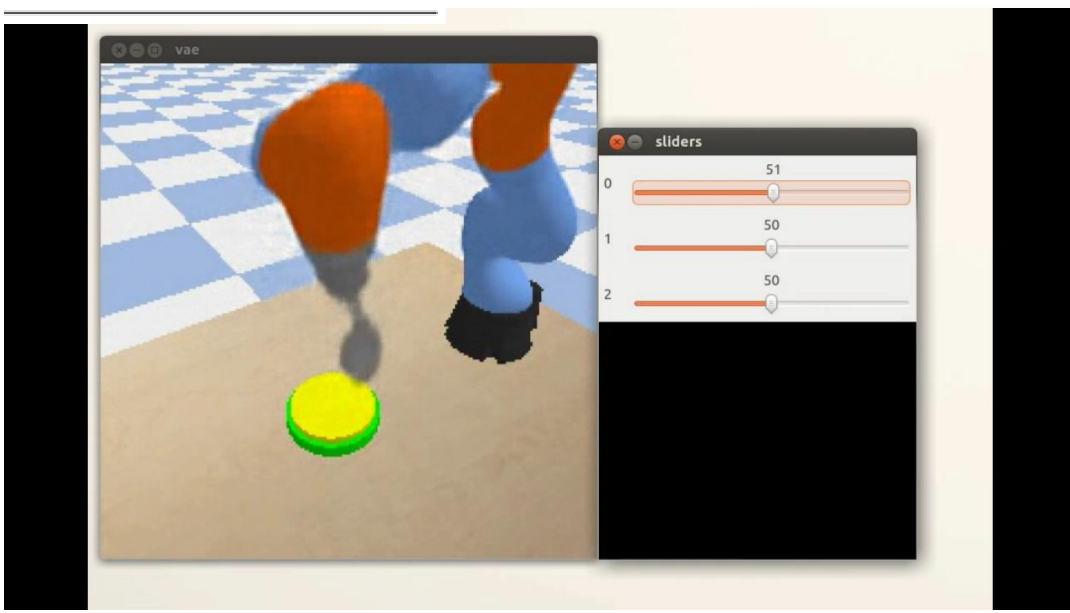


SRL Toolbox

A set of visualization tools

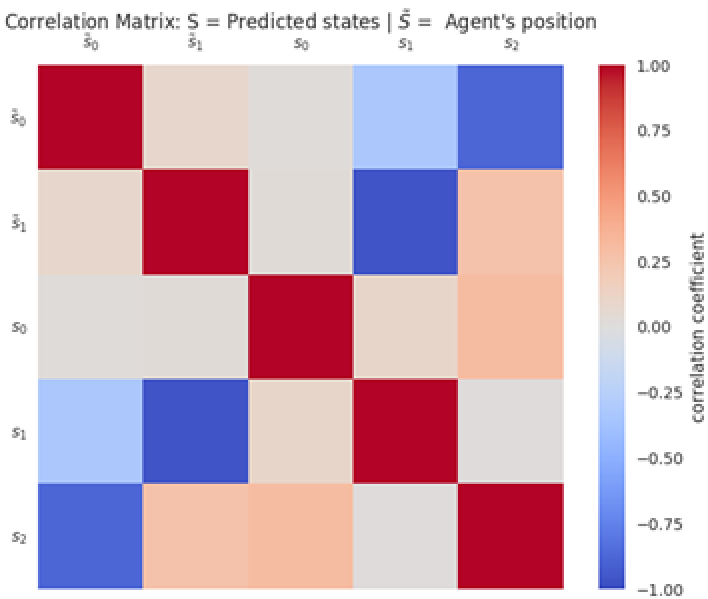


Latent visualization

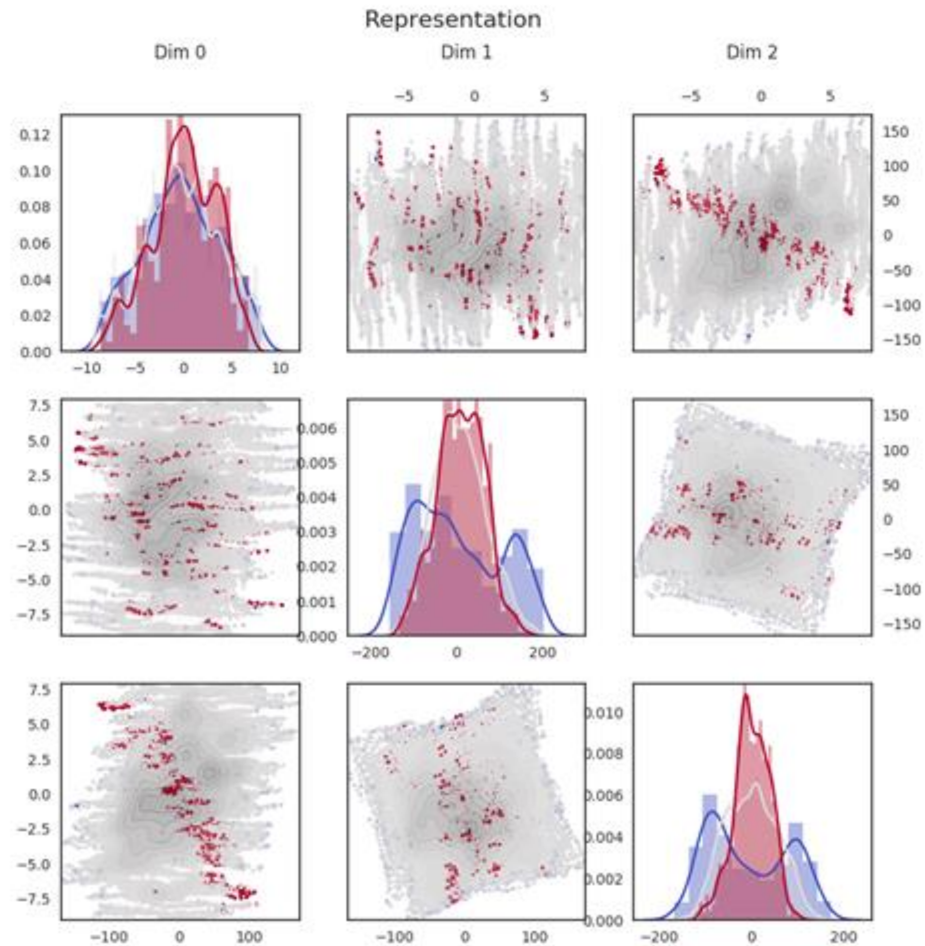


SRL Toolbox

A set of visualization tools



State / GT correlation



State vs State plot

SRL Toolbox

Some lessons learned

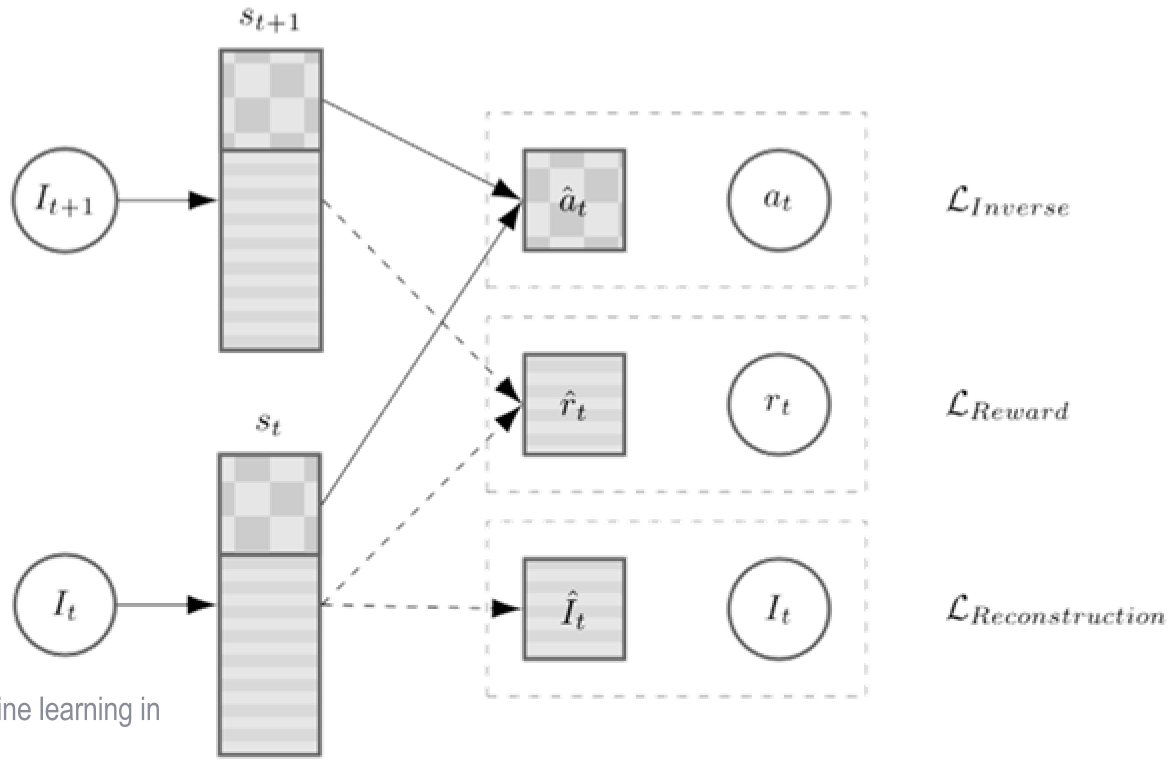
- Many methods' performance is quite task specific
  - E.g. robotic priors fail on robotic arms
- Autoencoders/VAE work quite well if extreme (small or large) noise
- Predicting a forward and inverse model often efficient
- Random states often reasonably efficient
- SRL + RL usually more efficient than end-to-end RL

- Encoding robot state AND environment state may be difficult
  - E.g. robotic priors work with fixed goal, but not moving goal

SRL - Split model

Learning structured state representation [Raffin et al. SPIRL19]

- Structure / disentangle / split state representation
- Forward/inverse models -> robot state
- Autoencoder/reward -> environment state

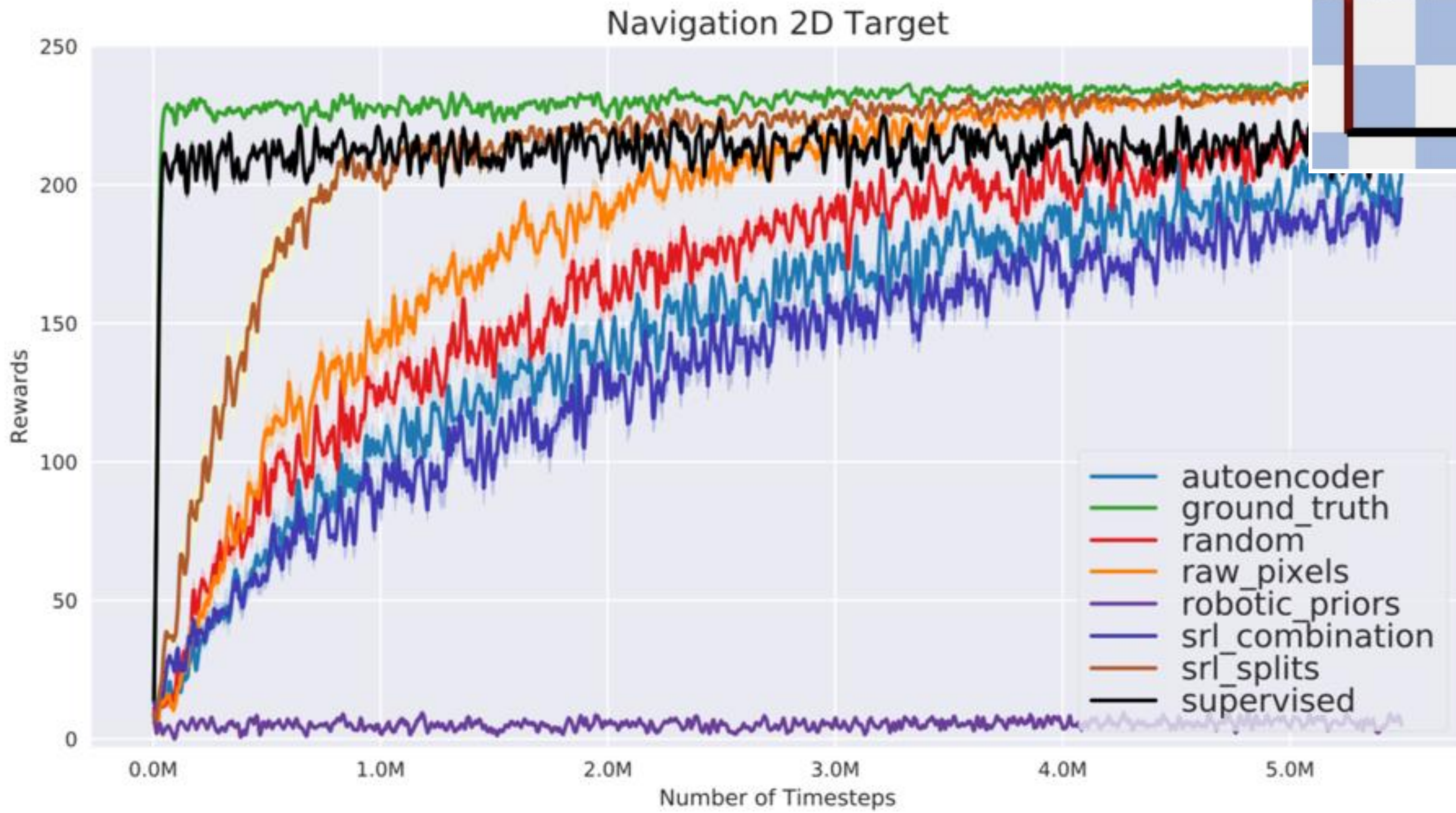


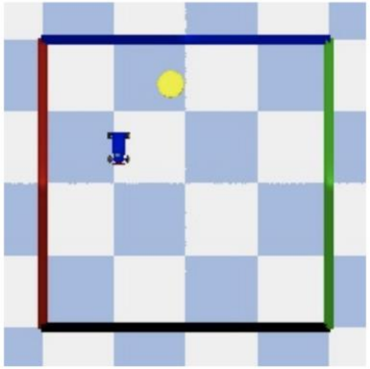
SRL - Split model

Learning structured state representation

[Raffin et al. SPIRL19]

- Can learn representation with moving goal
- Better sample efficiency / robustness



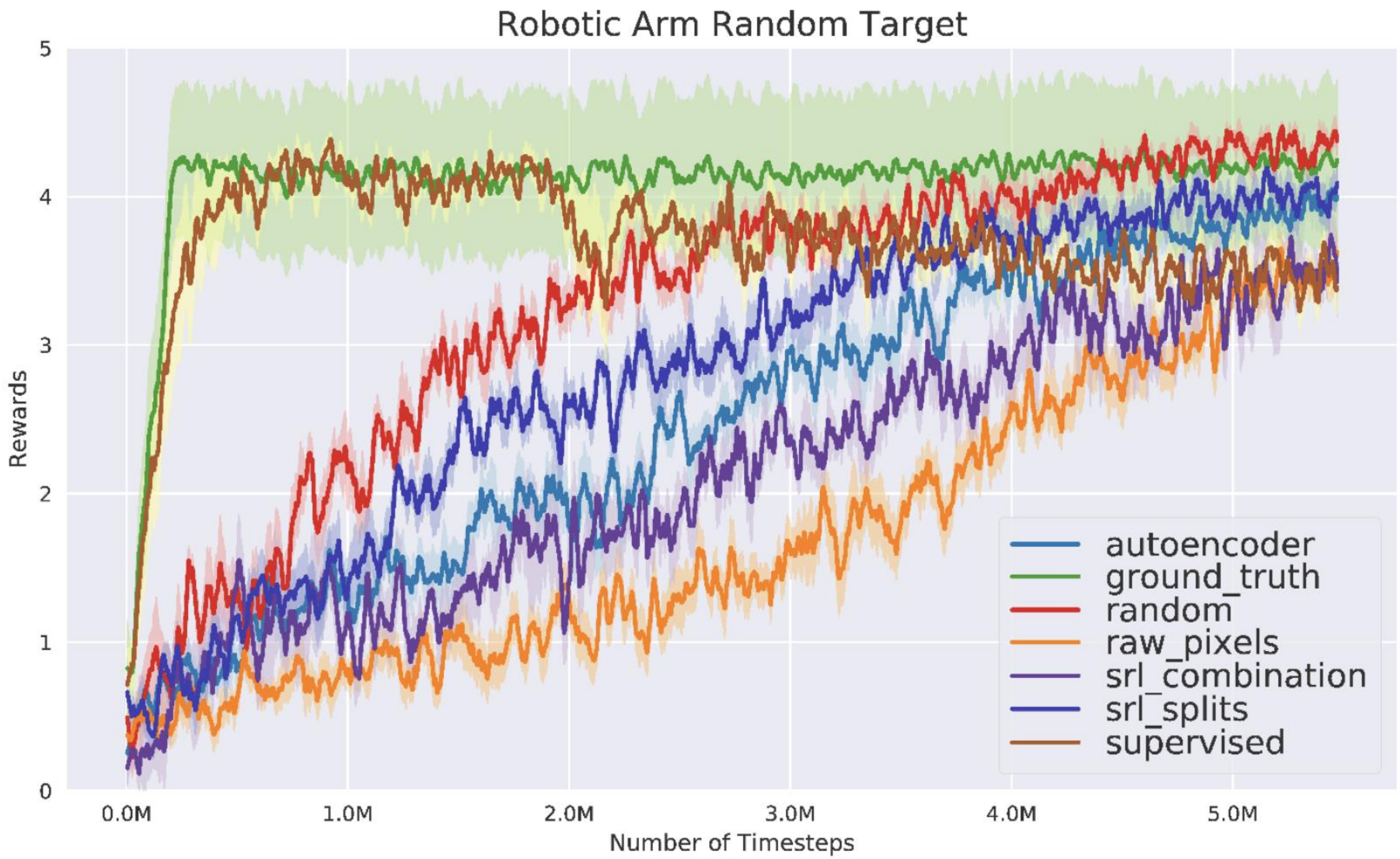


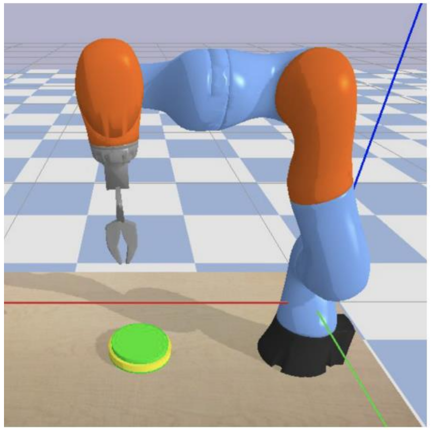
SRL - Split model

Learning structured state representation

[Raffin et al. SPIRL19]

- But not so efficient on more complex tasks





Conclusion

Machine learning has difficulties linked to the robotics context, but can exploit constraints/knowledge

Take advantage of the domain

- Learn in simulation using easy to simulate features (e.g. 3D motion)
- Exploit constraints on relevant info (low dim, controllable, predictable…)
- Exploit unsupervised (self supervised) learning

Many approaches

- Many existing approaches that can be combined
- Proposed a new way to combine AE & models

Perspectives

Very active domain

- Many Sim to Real transfer approaches (domain randomization, domain adaptation, …)
- Many new state representation learning approaches associated to unsupervised pretraining of CNNs
- Some fixed representation may be useful (e.g., Fourrier features)

  [Brellman et al. 21]
- Define / improve representation disentanglement (explicability)
- Merge everything ?
  - Supervised/self supervised pre-training in simulation with SRL, randomization, …
  - Ensure disentanglement/interpretability in simulation
  - Fine tuning on real data with continual learning and SRL as auxiliary tasks

Behind these results

Students

- Florence Carton
- Hugo Caselles-Dupré
- Timothée Lesort
- Clément Pinard
- Antonin Raffin
- Ashley Hill
- René Traoré

Colleagues

- Laure Chevalley
- Natalia Diaz
- Michael Garcia Ortiz
- Jean François Goudou
- Antoine Manzanera
- Quoc Cong Pham
- Jaonary Rabarisoa

Projects

- H2020 DREAM
- H2020 VeriDREAM

https://www.veridream.eu/

References

[Pinard & al., ECMR17] **Multi range Real-time depth inference from a monocular stabilized footage using a Fully Convolutional Neural Network** Clément Pinard, Laure Chevalley, Antoine Manzanera, David Filliat European Conference on Mobile Robotics,

[Pinard & al., ECCV18] **Learning structure-from-motion from motion** Clément Pinard, Laure Chevalley, Antoine Manzanera, David Filliat ECCV GMDL Workshop, Sep 2018

[Lesort et al, NN18] **State Representation Learning for Control: An Overview.** Timothée Lesort, Natalia Díaz-Rodríguez, Jean-François Goudou, David Filliat Neural Networks, Elsevier, 2018, 108, pp.379-392.

[Raffin et al, SPIRL19] **Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics** Antonin Raffin, Ashley Hill, René Traoré, Timothée Lesort, Natalia Díaz-Rodríguez, David Filliat SPiRL 2019 : Workshop on Structure and Priors in Reinforcement Learning at ICLR 2019.

[Lesort et al, IF20] **Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges.** Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., & Díaz-Rodríguez, N. (2020). Information fusion, 58, 52-68.

[Caselle-Dupré et al., IJCNN21] **S-trigger: Continual state representation learning via self-triggered generative replay.** Caselles-Dupré, H., Garcia-Ortiz, M., & Filliat, D. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7).

[Carton et al. 21] **Using Semantic Information to Improve Generalization of Reinforcement Learning Policies for Autonomous Driving** Florence Carton, David Filliat, Jaonary Rabarisoa, Quoc Pham IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops

[Brellman et al. 21] **Fourier Features in Reinforcement Learning with Neural Networks** David Brellmann, Goran Frehse, David Fiilliat, submitted IJCNN 2022