# Emergence of Sensory Representations using Prediction in Partially Observable Environments

Thibaut Kulak and Michael Garcia Ortiz

Softbank Robotics Europe - AI Lab
thibaut.kulak@gmail.com,mgarciaortiz@softbankrobotics.com

**Abstract.** In order to explore and act autonomously in an environment, an agent can learn from the sensorimotor information that is captured while acting. By extracting the regularities in this sensorimotor stream, it can build a model of the world, which in turn can be used as a basis for action and exploration. It requires the acquisition of compact representations from possibly high dimensional raw observations. In this paper, we propose a model which integrates sensorimotor information over time, and project it in a sensory representation. It is trained by preforming sensorimotor prediction. We emphasize on a simple example the role of motor and memory for learning sensory representations.

## 1 Introduction

Autonomous Learning for Robotics aims to endow agents with the capability to learn from and act in their environment, so that they can adapt to previously unseen situations. An agent can learn from this interaction by building compact representations of what it encounters in its environment, using information captured from a high dimensional raw sensory input and motor output.

Theories on sensorimotor prediction state that an agent learns the structure of its world by learning how to predict the consequences of its actions ( [12], [2]). The sensorimotor approach proposes to learn sensor representations and motor representations by identifying the regularities in the sensorimotor stream. However, these regularities are hard to capture: a robotic agent acts and perceives in an environment which is usually partially observable (limited field of view), noisy and ambiguous. The sensory information is not sufficient to know the exact state of the agent in its environment (similar sensory states can originate from different situations in the environment). This is in particular true for navigation tasks where an agent can observe several occurrences of very similar portions of the scenes (wall, corners) at different locations in the environment (e.g. in a maze). For these reasons, we need representations that can help disambiguate the observations and the state of the agent.

In the case of an autonomous agent, without labeled data, unsupervised learning allows to learn compression for different data streams ( [6], [16], [13]). These representations, based on the statistics of the data, reduce the dimensionality of the sensory stream, but do not inform the agent on the modalities of its potential actions in its environment, which is related to the problem of grounding knowledge in the experience of an agent [5]. In order to build representations, a classic approach is to learn forward internal models [3]: learning to predict the sensory consequences of actions. For instance a forward model of physics is learned for a real-world robotic platform in [1]. Recently, [4] proposed to build world models through learning forward models, and use them to train policies in different Reinforcement Learning environments. The authors of [9] present a complete overview of the current methods for learning representations in robotics.

In this paper, we propose to learn sensory representations using principles from sensorimotor prediction (or, forward models) and to study the properties of the learned representations. We show, on a navigation scenario, that using motor information as well as a short-term memory leads to sensory representations that correspond to richer classes of sensory stimuli encountered in the environment. Recent work also propose to learn sensory representation by sensorimotor prediction ( [4, 17]), and show that the representations learned could be succesfully used for navigation or control tasks. In this paper we are interested in studying the nature of the representations that are learned.

## 2 Sensorimotor predictive model

We train a forward model, named Recurrent Sensorimotor Encoder (**Recurrent-SM-encoder**), shown in Fig. 1, and composed of three subnetworks : (i) A sensory encoding subnetwork takes as input the sensory state $s_t$ and outputs an encoded sensory state $z_t^s$. It is composed of hidden layers followed by a stacked Long short-term memory (LSTM) network, which role is to provide a form of memory about the previous sensor states. (ii) A motor encoding subnetwork, which is a classical dense network composed of hidden layers, taking as input the motor command $m_t$ and outputting the encoded motor command $z_t^m$. (iii) $z_t^s$ and $z_t^m$ are concatenated to form the encoded sensorimotor vector $z_t^{sm}$, used as an input for a dense network, which outputs a prediction of the next sensory state $\hat{s}_{t+1}$.

We use several baselines (see Fig. 2) to evaluate the role of motor information and memory: the Sensorimotor Encoder (**SM-encoder**), doesn't have a memory, the Recurrent Sensory Encoder (**Recurrent-S-encoder**) doesn't have motor input, and the Sensory Encoder (**S-encoder**) doesn't have memory or motors. We train the proposed networks using a loss to minimize the prediction error:

$$\mathcal{L}_2 = \sum_{t=1}^{T-1} \left(\hat{s}_{t+1} - s_{t+1}\right)^2 \tag{1}$$

Where $T$ is the size of the learning batch.

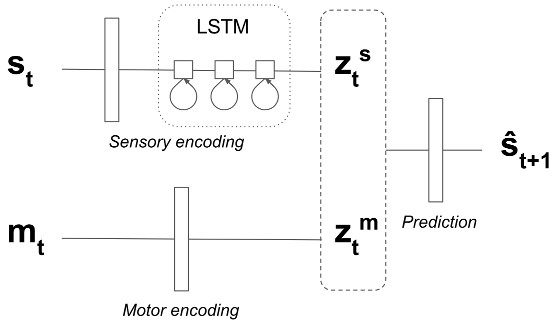
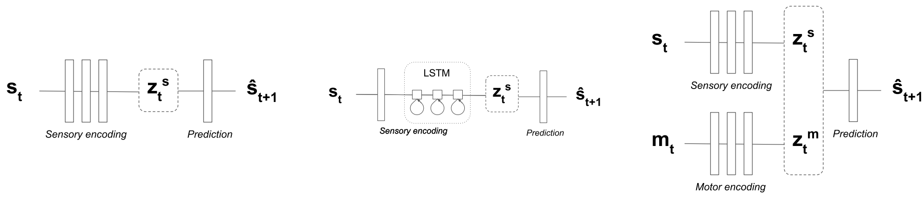



Fig. 1: Recurrent Sensorimotor Encoder




Sensory Encoder | Recurrent Sensory Encoder | Sensorimotor Encoder

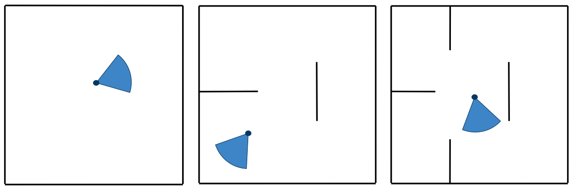
Fig. 2: Architectures of the baselines

## 3 Experimental setup

Our simulated agent (inspired from the Thymio-II robot [14]) is equipped with 5 distance sensors, evenly separated between -0.6 and 0.6 radians, with their range limited to 10 units of distance. The agent controls its translation forward (direction of the middle laser) and its rotation. One motor command $(d, r)$ is the succession of a translation $d$ and a rotation $r$. It is a planar agent moving without friction, and there is no noise on its distance sensors. We created 3 environments of size 50 units, shown on Fig. 3: **Square** is a square without walls or obstacles. **Room1** additionally contains one vertical wall and one horizontal wall. **Room2** contains one horizontal and three vertical walls.

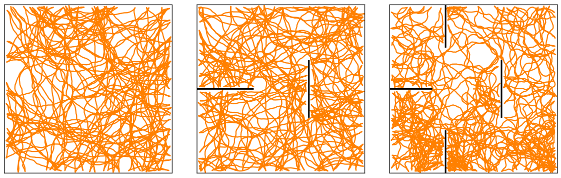
The agent moves by random translations forward and random rotations, while avoiding collisions with the walls. At each timestep, if one distance sensor value is smaller than 1 unit, the agent rotates by $r \sim \mathbf{U}(\pi - \frac{\pi}{10}, \pi + \frac{\pi}{10})$ radians ($\mathbf{U}$ denoting the uniform distribution). If not, the agent moves forward by $d \sim \mathbf{U}(0, 1)$ units, and rotates by $r \sim \mathbf{U}(-\frac{\pi}{6}, \frac{\pi}{6})$ radians. Fig.4 displays the trajectory of the agent during 10 000 steps in the different proposed environments.

We generated a sequence of 1 000 000 timesteps for each environment (each point has 5 distance sensors values and 2 motor commands), split as such: the first 80% for training, the next 10% for validation, and the last 10% for testing.



(a) Square (b) Room1 (b) Room2

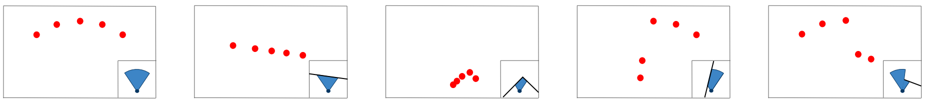
Fig. 3: The different environments created.



(a) Square (b) Room1 (c) Room2

Fig. 4: Trajectories in the environments (10 000 points)

In Fig. 5 we reconstructed, for different situations, what the agent perceives based on its sensors. Note that the agent doesn't have access to the position and angles of its distance sensors, it only receives as input a 5-dimensional real vector.



(a) Perceiving nothing (b) In front of a line (c) In front of a corner (d) With a wall at its left (e) In front of a wall's end

Fig. 5: Examples of different sensory stimuli perceived by the agent. The 5 red dots represent the distance perceived by the agent, projected in top-view.

## 4 Results

### 4.1 Numerics

Our models are trained with the Adam optimizer [8] (learning rate of 0.001). The training is stopped if the loss on the validation set doesn't decrease by 5% for 10 consecutive epochs. We use a mini-batch size of 64, and ReLUs for the activation functions. We choose arbitrarily the sensory representation space to be 10-dimensional and the motor representation space to be 5-dimensional. The number and size of layers in the different architectures are as follow: In **SM-encoder**, the sensory encoding and motor encoding subnetworks have 3 hidden layers of size 16, 32 and 64, while the prediction subnetwork has one layer of size 128. **S-encoder** is identical to the SM-encoder, without the motor encoding subnetwork. In **Recurrent-SM-encoder**, the sensory encoding and motor encoding subnetworks have 1 hidden layer of size 16, while the prediction subnetwork has one layer of size 128. The (stacked) LSTM has 3 layers with 32 units at each layer, and a truncation horizon of 20. **Recurrent-S-encoder** is identical to Recurrent-SM-encoder, without the motor encoding.

### 4.2 Sensorimotor prediction results

We report in Tab. 1 the $\mathcal{L}_2$ prediction error of the models trained on the Square environment, and tested on the three environments. First we verify that models using motor information largely outperform those without, which makes sense because motors are necessary to predict the next sensory state. We also see that models using a memory perform better compared to their memoryless counterpart, confirming that a memory is useful for accurate sensorimotor prediction. Finally, we note that the Recurrent-SM-encoder model performs best. It is to be expected, as it benefits from additional information. We verified that these observations hold when trained on Room1 and Room2.

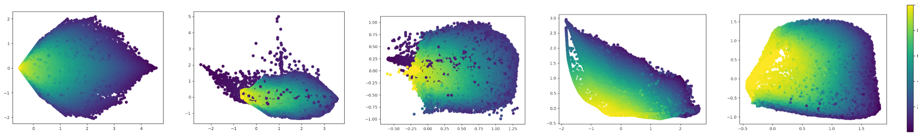
| Model | Square | Room1 | Room2 |
|---|---|---|---|
| S-encoder | 0.0374 | 0.0430 | 0.0729 |
| SM-encoder | 0.0056 | 0.0145 | 0.0257 |
| Recurrent-S-encoder | 0.0359 | 0.0407 | 0.0697 |
| Recurrent-SM-encoder | **0.0024** | **0.0105** | **0.0181** |

Table 1: Sensorimotor prediction $\mathcal{L}_2$ error of the models trained on Square tested on the test dataset of the three environments.

### 4.3 Representation spaces

We plot on Fig. 6 the representation spaces learned by our models, projected on the first two principal components extracted with a Principal Component Analysis (PCA) [7]. We color-code those spaces by the minimum value of the 5 lasers, as this gives information about the distance to the wall the agent perceives.

We observe that the models without motors group states where the agent doesn't see anything with states where the agent sees a wall from a very short distance, because its behavior (avoiding collision, see Sec.3) makes it experience sensory transitions from seeing a wall very close to seeing nothing. Without access to motor commands, the model brings those states close to each other,



(a) Sensor (b) S-encoder (c) Recurrent-S-encoder (d) SM-encoder (e) Recurrent-SM-encoder

Fig. 6: Representation spaces learned on the Square environment, colored by the minimum value of the lasers

while in reality those states are fundamentally different. We see that the portion of the representation space corresponding to the agent perceiving nothing is larger with the Recurrent-SM-encoder than with the Recurrent-S-encoder. We can interpret it as memory and the information about motor commands helping to create different states for points where the agent doesn't see anything.

### 4.4 Clusters extraction

We cluster the sensory representation spaces learned for each model, and visualize the activation of the different clusters in the environments, in order to estimate if the sensory encoding learns spatial features. We sample random sensorimotor transitions and use a kMeans algorithm [10] to extract 20 clusters from each sensory representation space. We plot for each cluster the ground truth position and orientation of 500 random data points associated with this cluster.

We show on Fig. 7, as a baseline, the 20 clusters extracted from the S-encoder representation space. We see that there are clusters corresponding to different distances/angles to the wall. As there is no memory in this model all of the configurations when the agent doesn't perceive anything are in the same cluster.

We see on Fig. 8 that the Recurrent-SM-encoder representation space trained on the Square environment contains clusters corresponding to different distances to a wall, and also a cluster corresponding to corners. We observe that we have different clusters corresponding to an absence of visual stimuli, but at different distances from a wall (when the wall is behind the agent). LSTM provides the agent with a memory of previous events, and it contains a form of spatial information. However this memory is short-term as it is relative to the previous wall that has been seen, and there is no global notion of position in the environment.

We show on Fig. 9 the clusters extracted from the Recurrent-SM-encoder model trained on the Room1 environment. We observe that in addition to clusters similar to those appearing in Square environment, there is now a cluster corresponding to wall's ends. We note, however, that when training on Room2, the cluster corresponding to wall's ends is not visible with 20 clusters extracted. We hypothesize that the layout causes the agent to be stuck in the different rooms, reducing the number of appearance of wall's ends in the database.

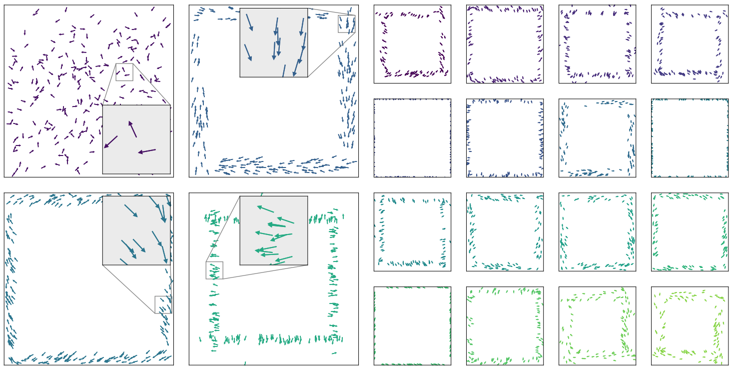


Fig. 7: S-encoder representation space clusters

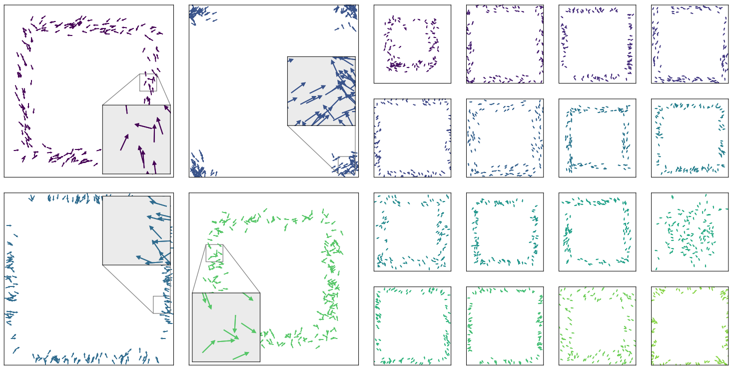


Fig. 8: Recurrent-SM-encoder representation space clusters

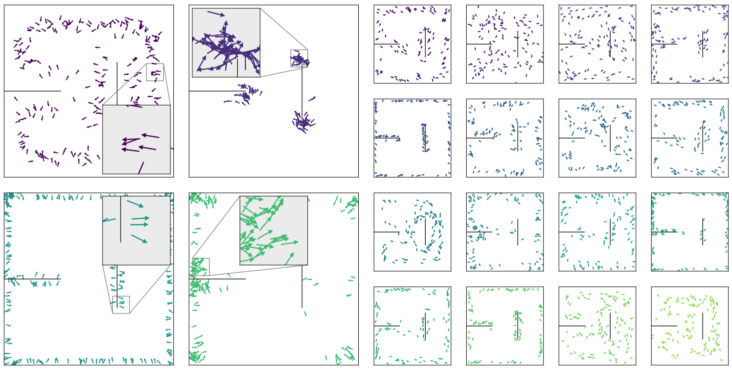
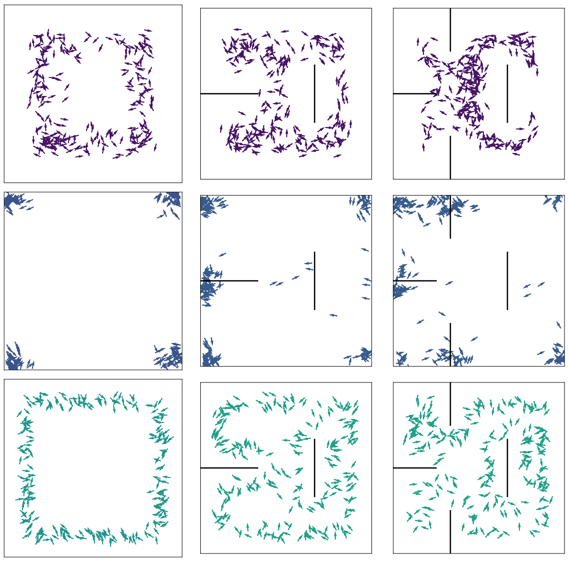


Fig. 9: Recurrent-SM-encoder representation space clusters, trained on Room1

### 4.5 Robustness to testing environment

In this experiment, we evaluate if the representations learned in one environment transfer to other environments. We train the Recurrent-SM-encoder as well as our clustering algorithm on one environment, then apply the learned representations and clusters in other environments. We show the transfer of some clusters of interest learned on Square on Fig. 10. We show on Fig. 11 the transfer of a few clusters of interest learned on Room1 to other environments. We observe that the representations learned in one environment can be used in other environments, with different spatial layouts. This is to be expected as the LSTM only captures and retains short-term information, which represents sensorimotor transitions, but do not represent different spatial layouts of the environments.



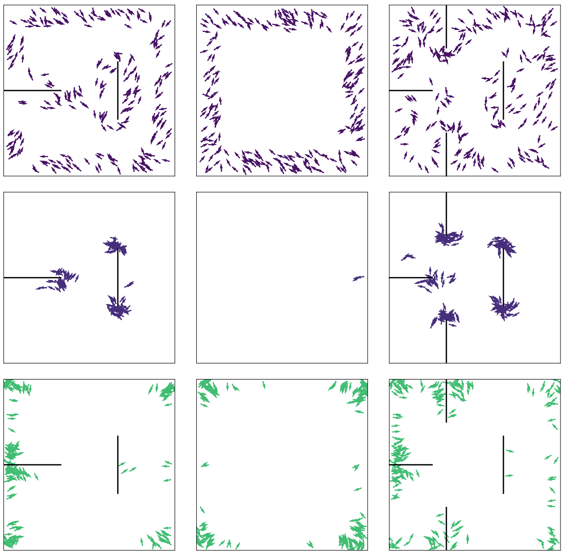
(a) Clusters in Square (b) Transferred to Room1 (c) Transferred to Room2

Fig. 10: Transferring some Square clusters

## 5 Conclusion

In this paper we proposed to use an unsupervised learning method based on sensorimotor prediction that allows an agent to acquire sensory representations by integrating sensorimotor information using recurrent neural networks.

We observed that our model extracts classes of interaction with the environment that seem qualitatively meaningful, and which contain temporal informa-



(a) Clusters in Room1 (b) Transferred to Square (c) Transferred to Room2

Fig. 11: Transferring some Room1 clusters

tion through short-term memory of previous experiences. In particular we verified that the motor commands and memory are very beneficial to learn sensory representations through prediction. We note that the clusters of the sensory representation are similar to particular cells observed in mammals, such as distance, orientation, and border cells [11]. We noticed that the representation learned on an environment can be used in other environments with different spatial layouts.

We used a generic approach, inspired from recent proposals about the nature and emergence of autonomy and intelligence through sensorimotor prediction [2]. It uses only raw data, and requires (in our simple experiment) very few engineering biases. In future works we want to investigate whether it scales to more complex environments and sensory streams, and if it can be applied on a robotic platforms in a real human environment.

One interesting possible extension would be to use the representations to learn a map of the environment. We plan to investigate how to build a graph where the nodes would correspond to particular activations of the representation, and the edges would correspond to motor commands necessary to transition from one representation to the other. We want to study the compression of this graph to obtain compact spatial representations, as proposed in [17] and [15].

In general, the proposed approach deals with very low level processing of sensorimotor streams in order to build meaningful representations. The usefulness of these representations, and how they can integrate in a cognitive architecture, would have to be demonstrated. We plan to use the learned representations in a

Reinforcement Learning task. On the one hand, the success rate at the task gives a clear quantitative evaluation. On the other hand, it will allow us to evaluate the benefits of learning representations in terms of generalization, abstraction, and transfer of knowledge across different environments.

## References

1. Agrawal, P., Nair, A.V., Abbeel, P., Malik, J., Levine, S.: Learning to poke by poking: Experiential learning of intuitive physics. In: Advances in Neural Information Processing Systems. pp. 5074–5082 (2016)
2. Friston, K.: The free-energy principle: a unified brain theory? Nature Reviews Neuroscience 11(2), 127–138 (2010)
3. Ghahramani, Z., Wolpert, D.M., Jordan, M.I.: An internal model for sensorimotor integration. Science 269, 1880–1882 (1995), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.74
4. Ha, D., Schmidhuber, J.: World models (2018), https://worldmodels.github.io
5. Harnad, S.: The symbol grounding problem (1990), http://cogprints.org/3106/
6. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
7. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of educational psychology 24(6), 417 (1933)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Lesort, T., Díaz-Rodríguez, N., Goudou, J.F., Filliat, D.: State Representation Learning for Control: An Overview. ArXiv e-prints (Feb 2018)
10. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory 28(2), 129–137 (1982)
11. Moser, M.B., Rowland, D.C., Moser, E.I.: Place cells, grid cells, and memory. Cold Spring Harbor perspectives in biology 7(2), a021808 (2015)
12. O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behavioral and brain sciences 24(5), 939–973 (2001)
13. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
14. Riedo, F., Rétornaz, P., Bergeron, L., Nyffeler, N., Mondada, F.: A two years informal learning experience using the thymio robot. Advances in Autonomous Mini Robots 101, 37–48 (2012)
15. Stachenfeld, K.L., Botvinick, M.M., Gershman, S.J.: The hippocampus as a predictive map. Nature Neuroscience 20(11), 1643–1653 (Oct 2017), http://dx.doi.org/10.1038/nn.4650
16. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
17. Wayne, G., Hung, C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J.W., Mirowski, P., Leibo, J.Z., Santoro, A., Gemici, M., Reynolds, M., Harley, T., Abramson, J., Mohamed, S., Rezende, D.J., Saxton, D., Cain, A., Hillier, C., Silver, D., Kavukcuoglu, K., Botvinick, M., Hassabis, D., Lillicrap, T.P.: Unsupervised predictive memory in a goal-directed agent. CoRR abs/1803.10760 (2018), http://arxiv.org/abs/1803.10760