The Impact of Action in Visual Representation Learning

Alexandre Devillers, Valentin Chaffraix, Frédéric Armetta, Stefan Duffner, Mathieu Lefort

The Impact of Action in Visual Representation Learning

Alexandre Devillers
Univ Lyon, Université Lyon 1
LIRIS, UMR5205
Lyon, France
alexandre.devillers@liris.cnrs.fr

Valentin Chaffraix
Univ Lyon, INSA Lyon
LIRIS, UMR5205
Lyon, France
valentin.chaffraix@gmx.com

Frédéric Armetta
Univ Lyon, Université Lyon 1
LIRIS, UMR5205
Lyon, France
frederic.armetta@liris.cnrs.fr

Stefan Duffner
Univ Lyon, INSA Lyon
LIRIS, UMR5205
Lyon, France
stefan.duffner@liris.cnrs.fr

Mathieu Lefort
Univ Lyon, Université Lyon 1
LIRIS, UMR5205
Lyon, France
mathieu.lefort@liris.cnrs.fr

***Abstract*—Sensori-motor theories, inspired by work in neuroscience, psychology and cognitive science, claim that actions, through learning and mastering of a predictive model, are a key element in the perception of the environment. On the computational side, in the domains of representation learning and reinforcement learning, models are increasingly using self-supervised pretext tasks, such as predictive or contrastive ones, in order to increase the performance on their main task. These pretext tasks are action-related even if the action itself is usually not used in the model. In this paper, we propose to study the influence of considering action in the learning of visual representations in deep neural network models, an aspect which is often underestimated w.r.t. sensori-motor theories. More precisely, we quantify two independent factors: 1- whether or not to use the action during the learning of visual characteristics, and 2- whether or not to integrate the action in the representations of the current images. Other aspects will be kept as simple and comparable as possible, that is why we will not consider any specific action policies and combine simple architectures (VAE and LSTM), while using datasets derived from MNIST. In this context, our results show that explicitly including action in the learning process and in the representations improves the performance of the model, which opens interesting perspectives to improve state-of-the-art models of representation learning.**

***Index Terms*—Sensori-motor theory, Representation learning, Predictive learning, Deep learning**

I. Introduction

Sensori-motor theories are based on substantial evidence in neuroscience, developmental psychology and cognitive science. The main claim is that actions, and more especially the sensory changes induced by motor actions, play a key role in learning a predictive model of the world and in perceiving it [23]. For example, a kitten that cannot walk, i.e. it only passively receives a visual flow, will learn defective visual representations [14]. The role of action is also emphasised in the notion of affordance in psychology [9], where an object is not defined by a set of properties but by its elicited interactions for the agent. According to the sensori-motor contingencies theory, acting may even play a role in some form of consciousness [25]. These concepts are also related to the theories of enactivism and embodiment that states that the body, as the structure to interact with the world, is required for an intelligent behavior to arise [8]. While contributing to the learning of representations, the actions could also be aimed at perceiving relevant regions of the environment, which would make perception an active process. This way the actions would be required to accumulate evidence of the current state of the world as unified in the free energy principle e.g. [7]. With regard to vision, this is for example the role of saccades which allow to get successive glimpses over a visual scene [6].

Since some years, deep learning achieves state-of-the-art performance in multiple domains such as visual recognition, natural language processing, game playing etc. [20]. Initially, these data-driven approaches were mainly supervised, e.g. by using a Convolutional Neural Network (CNN) to classify objects in images [13]. Contrary to human beings that perform saccades to perceive a scene, most CNN models have a translation invariance property that allows them to process the whole image at once. Then, deep architectures have been adapted to the reinforcement learning framework [27]. Here, actions are considered through sequential decision making, but are not explicitly included neither in the perception nor in the building of representations. More recently, self-supervised approaches have emerged. They propose to use a pretext task during learning, usually making close the representations of inputs considered similar, to improve performance of a predefined task or in the context of unsupervised learning. In computer vision, some of the similar inputs generation processes can be interpreted as the resulting from movements [18]. In reinforcement learning, temporal prediction of consequences of action is often used as a pretext task.

Thus, sensori-motor theories and the recent and promising

This work was performed using HPC resources from GENCI–IDRIS and a GPU donated by the NVIDIA Corporation. We gratefully acknowledge this support. This work was financed by the Auvergne Rhônes-Alpes (AURA) region, within the Ethics.AI project (Pack Ambition Recherche). The authors would like to thank the AURA region and their partners in this project.

trend of including action-related pretext tasks in deep learning seem to point towards a benefit of action in representation learning and perception, at different degrees. Yet the precise quantification of the impact of action in representation learning is still barely known. In this article, we propose to open this research question by studying two independent factors: 1- whether or not to use action in the learning of visual features and 2- whether or not to use action in the computation of the current image representation. To keep the study tractable, we restrain ourselves to simple deep architectures as illustrative examples. Moreover, to put apart the question of the action decision process, which would introduce a retro-action loop during learning, the model will perform random actions.

Section II introduces existing works related to representation learning considering actions. In section III, we derive from our research objectives the different neural network architectures and loss functions used in our study. The protocol and hyper-parameters used and the obtained results are presented in section IV. Finally, we draw our conclusions and expose various perspectives for future works (section V).

II. Related work

Multiple works in robotics considered action while learning predictive models of the environment for achieving a variety of tasks such as object manipulation, recognition or grasping. The benefits of such interactive perception are mainly to get access to some objects characteristics requiring manipulation as weights for example and to enrich and structure the regularities in the inputs (see [5] for an in-depth survey). Considering explicitly sensori-motor contingencies can even push these properties a step further. Arranging sensori-motor schemes hierarchically leads to the learning of the complex concept of container, that could be reused across environments [12]. Sensory representation learning can be shaped by action, through the notion of compensating movements, i.e. that some displacements in the sensory inputs can be reversed via motor actions. A deep architecture, designed with this principle, is able to learn the underlying spatial structure of the input [19].

In computer vision, recent visual representation learning methods rely on either contrastive or predictive pretext tasks. In contrastive ones, models usually learn to embed multiple views of an image into similar representations [4]. The generation process used to obtain these views can be related to some form of action [18], such as *cropping* which can be linked to head movement and eye saccades. However, these methods include the actions neither to build nor to learn the representations. Such tasks can also rely on predicting the motion that led from the actual view to the future one [2], in this case the action can be seen as a supervision signal, however it is not directly integrated in the representations. For predictive tasks, they generally aim at predicting future inputs based on historical ones, as in [24] where a contrastive predictive task has been successfully applied to vision, audio, natural language processing, and reinforcement learning. Such tasks are well suited for environments with a temporal aspect, as in the context of reinforcement learning where the prediction of future observations from historical observations and actions has shown to learn good representations [11].

While most computer vision models have been focused on treating full images at once, only few works consider processing sub-parts of images. Such models that only process glimpses of images were initially introduced for computational advantage, but also open the possibility of making models actively perceiving the world by choosing where to look. This idea of processing glimpses of an image has been applied to classification two ways: either by dedicating a neural network to each glimpse w.r.t. its temporal index [26], or by letting a recurrent network learn to perform saccades in a reinforcement learning environment [22]. Later, these models have been enhanced to perform multiple object recognition, as in [3] where the model learns to classify objects from left to right by moving a virtual glimpse sensor over the image, or in [1] where the model classifies objects sequentially while determining an affine transformation to produce the next glimpse to locate the next object. Moreover, [10] also used glimpses for image generation both to "read" and "write" images, iteratively generating the result with small patches while showing strong representational and generational capacities.

III. Studied models

A. Overview

1) Problem statement: In this article we consider a system that receives visual saccades to perceive its environment. At each step, the system only takes as input a sub-part of the image and the action to come. The action defines the 2D position of the center of the next visual input. In order to decorrelate the action policy from the learned representations, it is the same for all models and consists in a random sampling from an uniform distribution. We note x_t the observed glimpse (i.e. image sub-part) at time t, a_t the next action performed, i.e. the position of the next observed glimpse x_{t+1}.

2) General overview of the model: The task the model has to perform mixes the prediction of the future visual input for a given action and the reconstruction of the current visual input. We note $\hat{x}_t$ (resp. $\hat{x}_{t+1}$) the reconstruction by the model of the glimpse x_t (resp. x_{t+1}). All the model variations that we study are relying on the same modules, each one addressing a specific point of the combined task:

- The first is a convolutional Variational Auto-Encoder (VAE) [17], which reduces the dimensionality of the current glimpse by projecting it in a latent space and then reconstructing it.
- The second is a Long Short-Term Memory (LSTM) neural network [16]. As the system only gets partial glimpses of the environment, it needs to integrate the current observation with past ones to construct a global representation of the observed image. Its output is what we consider as the representation of the current image.
- The third, which we call the recoder, is a neural network we introduce, to generate a latent embedding of the next

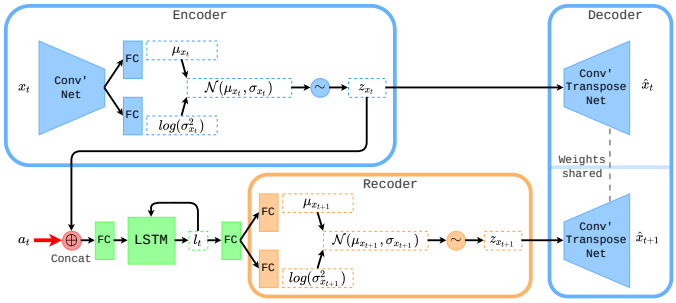

Fig. 1. PreLSTM — The input x_t passes through the encoder, transforming x_t in its latent representation z_{x_t}. Then z_{x_t} passes through the decoder, giving $\hat{x}_t$ the reconstructed input produced by the VAE. On the other side, the action a_t is concatenated with z_{x_t} and is then fed to the LSTM, which outputs l_t the global representation of the image. From this representation the recoder computes $z_{x_{t+1}}$ and finally, by passing through the decoder, the constructed prediction $\hat{x}_{t+1}$ of the next glimpse x_{t+1}.

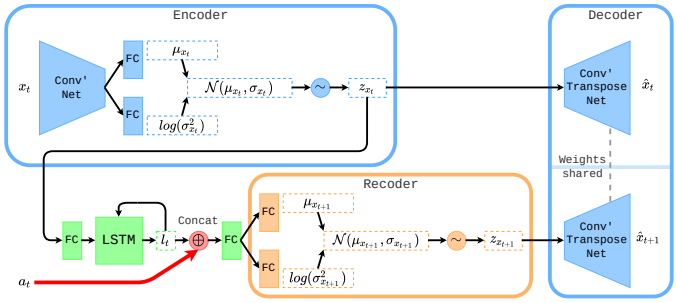

Fig. 2. PostLSTM — The whole network works the same way as in PreLSTM except that z_{x_t} is directly fed to the LSTM, and that the action a_t is concatenated with l_t before passing through the recoder.

glimpse w.r.t. the action to come and the representation from the LSTM. This embedding is used to reconstruct the next visual input. From a technical perspective, the functioning of the recoder is similar to a VAE's encoder by generating a distribution from which the recoded latent embedding is sampled. As it is the case for the VAE, this distribution is regularized.

In the next two sections, we will describe the 4 models compared in this article, that vary over two axes:

1) whether or not to use the action in the LSTM representations, i.e. to make the representations sensori-motor (Sec. III-B),
2) whether or not to use the action during the learning of the visual characteristics by the VAE's encoder, i.e. making the learning of the visual characteristics partly sensori-motor (Sec. III-C).

B. Influence of action in the representations

1) With action: The PreLSTM architecture, illustrated in Fig. 1, integrates the actions before the LSTM. While combining observed glimpses, by providing the action the LSTM will construct sensori-motor representations. Indeed, the content of the action is forced to pass through the LSTM in order to get used by the recoder, forcing the representation to be a mix of sensory and motor information. Note that to ensure that the dimensions of the VAE's latent space and the LSTM's output are constant across all model variations architectures, in addition to keep similar computational capacity between both architectures, one Fully-Connected (FC) layer is placed before and after the LSTM.

2) Without action: The PostLSTM architecture, see Fig. 2, is similar to the PreLSTM one except that it concatenates the action after the LSTM. This makes the latent representation l_t purely visual, as the action is no more directly used to construct the representation. Note that the recoder still has access to the information of the performed action and the representation as in PreLSTM.

C. Influence of action in the learning

1) With action: The first method jointly optimizes the parameters of the whole neural network during training *end-to-end*. The overall loss of the model $\mathcal{L}_{tot}$ (Eq. 3) is composed of the loss of the VAE $\mathcal{L}_{vae}$ (Eq. 1), consisting in the reconstruction of the current glimpse, and the loss of the recoder $\mathcal{L}_{rec}$ (Eq. 2) which represents the prediction of the future glimpse. While minimizing this loss, information from the action is backpropagated through the whole architecture including the visual features learned by the VAE.

$$\mathcal{L}_{vae} = ||x_t - \hat{x}_t||^2 + \beta_{vae} D_{KL}[\mathcal{N}(\mu_{x_t}, \sigma_{x_t})||\mathcal{N}(0,1)] \quad (1)$$

$$\mathcal{L}_{rec} = ||x_{t+1} - \hat{x}_{t+1}||^2 + \beta_{rec} D_{KL}[\mathcal{N}(\mu_{x_{t+1}}, \sigma_{x_{t+1}})||\mathcal{N}(0,1)] \quad (2)$$

$$\mathcal{L}_{tot} = \mathcal{L}_{vae} + \mathcal{L}_{rec} \quad (3)$$

$\mathcal{L}_{vae}$ is the loss of a Beta-VAE [15]. The first term is the Mean Squared Error (MSE) between the input glimpse x_t and its reconstruction $\hat{x}_t$. The second term, used as a regularization weighted by β_{vae}, is the KL-Divergence between the distribution created by the VAE, $\mathcal{N}(\mu_{x_t}, \sigma_{x_t})$ and the standard normal distribution, $\mathcal{N}(0,1)$. $\mathcal{L}_{rec}$ is derived from $\mathcal{L}_{vae}$, where the MSE is between the next glimpse x_{t+1} and its recoded reconstruction $\hat{x}_{t+1}$, while the regularized distribution is the one created by the recoder $\mathcal{N}(\mu_{x_{t+1}}, \sigma_{x_{t+1}})$ and is weighted by β_{rec}.

2) Without action: In order to analyze if the action has an impact on the extracted visual features, we propose a *separated* two-step learning procedure.

In the first step, the VAE is trained without actions so that the learned features are purely visual. To have a fair comparison, the prediction task, that requires actions, is replaced by a second reconstruction of the current glimpse. For this purpose, we use a temporary architecture which instead of having a *classic* recoder, has an identity recoder (see Fig. 3), that recodes the current perceived glimpse from the LSTM. The loss $\mathcal{L}_{pretrain}$ (Eq. 5), used for this first step, is composed of $\mathcal{L}_{vae}$ (Eq. 1) the loss of the Beta-VAE, but also of $\mathcal{L}_{recId}$ (Eq. 4) the loss of the identity recoder.

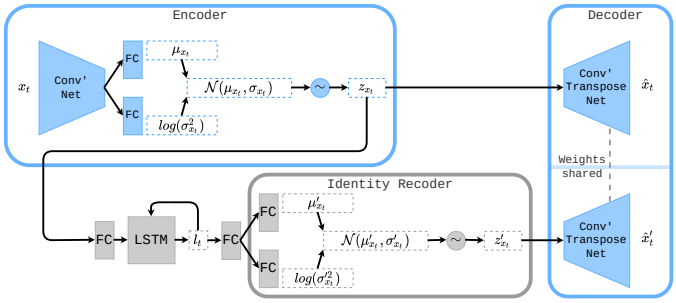

Fig. 3. Temporary architecture used in the first step (VAE training) of the two-step separated training, at the end only the weights of the encoder and decoder are kept and frozen and the identity recoder, is replaced by the one of PreLSTM or PostLSTM architectures.

$$\mathcal{L}_{recId} = ||x_t - \hat{x}'_t||^2 + \beta_{rec} D_{KL}[\mathcal{N}(\mu'_{x_t}, \sigma'_{x_t})||\mathcal{N}(0,1)] \quad (4)$$

$$\mathcal{L}_{pretrain} = \mathcal{L}_{vae} + \mathcal{L}_{recId} \quad (5)$$

In $\mathcal{L}_{recId}$, the MSE is between the input glimpse x_t and its recoded reconstruction $\hat{x}'_t$, and the distribution of the identity recoder $\mathcal{N}(\mu_{x'_t}, \sigma_{x'_t})$ is regularized.

In the second step, we replace the LSTM/identity recoder by a normal predictive one (i.e. either the PreLSTM or the PostLSTM architecture) while freezing the VAE's weights to train only the LSTM/recoder using the loss $\mathcal{L}_{rec}$ (Eq. 2).

IV. Experiments

A. Datasets

1) 28×28 *MNIST:* The MNIST digits dataset [21] is composed of 28×28 pixels images that contain centered white hand-written digits on a black background. We used a number of 15 glimpses per image for this dataset.

2) 60×60 *MNIST:* To make the digits unrecognizable at the first glimpse, we use images of MNIST resized to 60×60 pixels. This way, patches are no more digit fragments, but strokes and curves. For this dataset, we used a number of 30 glimpses per image as images are bigger.

3) 60×60 *Cluttered Translated MNIST:* In this dataset [22] (named 60×60 CT MNIST hereafter), images are 60×60 black background with a 28×28 MNIST digit randomly placed on them, and where four 8×8 clutters (extracted from other MNIST digits) are also randomly added on them. This dataset is the hardest since clutters and digit positions are totally unpredictable if never seen. This stochasticity makes the predictive task way harder. We used a number of 50 glimpses per image as the task is harder.

Note that for every dataset we split the train set in 5-folds, leading to 48k (resp. 12k) images for the training (resp. validation) and we used the test set with 10k samples.

B. Evaluation

To evaluate and compare the representations learned by the different models, we trained a classifier taking the LSTM output as input and measured the respective classification accuracy of the digits, averaged over 10 executions. We used a MLP composed of two hidden layers of 32 neurons each (one with Dropout $p = 0.5$), and the ReLU activation function. The training was done a posteriori, thus the weights of the rest of the model were frozen. This classifier was trained on all representations produced by the LSTM from the successive glimpses. Thus, we can study how the performance evolves when new glimpses are integrated in the model.

We also tracked the loss of the different models on the predictive and reconstruction tasks, and complete our quantitative evaluation with a more qualitative one based on t-SNE projections of the LSTM representations after the last glimpse.

C. Implementation details

1) Glimpses and actions: Glimpses are patches of the observed image extracted using a cropping window with a fixed size of 14×14 pixels. The position of this window is determined by the performed actions, and cannot be out of the image. Actions are 2D vectors encoding the continuous absolute position of the center of this cropping window, and they are uniformly sampled from the action space.

2) Models hyperparameters: The CNN used for the encoder (see section III) is composed of three 2D convolution layers and a FC layer, with ReLU as activation function. Convolution layers have respectively 8, 16, and 32 output channels, and kernels of size 3, 3, and 5. The output of the last convolution is flattened, and passed through the FC whose output dimension is 128. The dimension of the latent space z is 16, therefore the output size of FC generating μ and σ is also 16 for both the encoder and the recoder. The decoder is composed of a 16 to 128 FC followed by a mirror version of the encoder's CNN where input and output sizes are swapped, order is reversed and convolution layers are transposed ones. The LSTM has an input and hidden size of 128. Therefore, in the PreLSTM the FC before the LSTM has an input size of 18 (16+2) and an output size of 128, while the input size is 16 in PostLSTM (see section III-B1). Finally, the FC after the LSTM in PreLSTM has an input size of 128 and output size of 128, while the input size is 130 (128+2) in PostLSTM.

We used the Adam optimizer with a learning rate of 0.001 for both the self-supervised and the classification tasks, and have chosen $\beta_{vae} = \beta_{rec} = 0.5$ as it showed better performances. Models are trained for 200 epochs (200 epochs for the VAE then again 200 epochs for the LSTM/recoder, in the case of a separated training), while the a posteriori classification task is trained with 75 epochs for all models. We used a batch size of 128 in all configurations.

D. Results

The classification performance on the 3 datasets for the various models with increasing number of perceived glimpses is presented in Fig. 4. This metric allows us to compare the representations learned by the different models on the presence of semantic information through the ease of separation.

Firstly, we observe that the models not using the action during the learning of the VAE's encoder (-Sep suffix) perform

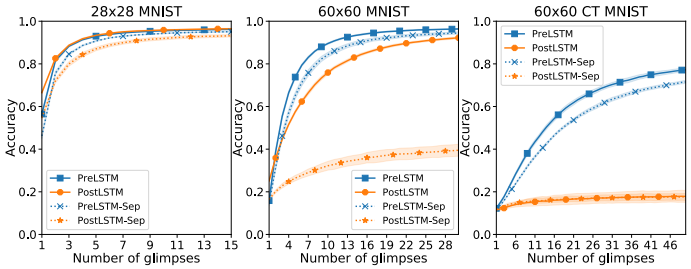

Fig. 4. Classification accuracy versus number of received glimpses, for the various models on the 3 datasets.

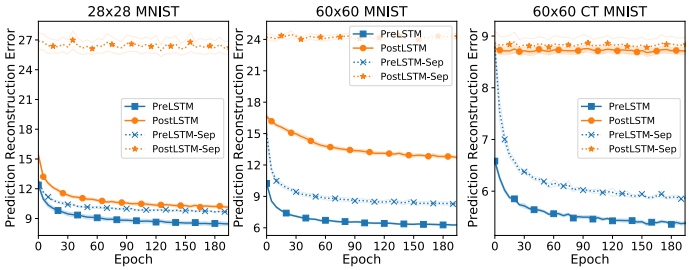

Fig. 5. Evolution of the reconstruction error for the predictive task in validation.

worse than their counterpart using the action (no suffix). The presence of this trend for both architectures and for all datasets shows that considering the action in the learning of visual characteristics seems to be beneficial for the extraction of meaningful features.

Secondly, we can see that all models integrating the action in the LSTM (PreLSTM architectures) perform better than their equivalent ones integrating the action after the LSTM (PostLSTM architectures), except for the 28×28 MNIST dataset. In this last case, they are similar when both performing an end-to-end training (no suffix), which may be due to the simplicity of the dataset. This difference of performance between Pre and Post architectures tends to show that considering the action in the representations, i.e. in the LSTM, helps to build better representations of the environment. Moreover, as this trend accentuates as the dataset becomes harder, the presence of the action in the representations seems to be more important for complex tasks. However, this difference of performance may also be explained by the fact that in the PreLSTM architectures the LSTM can use the action as an additional information to integrate the glimpses using their position in a global internal picture. Yet this may not be enough to explain all the differences as we observe the strongest difference on the 60×60 CT MNIST dataset where the position is less important as digits are small enough to get mostly captured by one glimpse. Note that all the observed trends are clearer after a certain amount of glimpses. This can be explained by the fact that the models need to temporally integrate the glimpses in order to build the representations. As all models start with an empty representation their few first representations may have similar results, but the more and the better they integrate the glimpses the better the representations would be.

The evolution during the training of the reconstruction error for the predictive task on the validation set is shown in Fig. 5. The results show the same trends as the ones on the accuracy, confirming the findings about the importance of including the action both in the learning of visual features and in the representations. However, we note that having better reconstruction loss does not necessarily imply better learned representations. For instance, the PostLSTM model on the 28×28 MNIST dataset has a higher error compared to the PreLSTM, while both have similar results on the classification task.

Finally, Fig. 6 shows the t-SNE projection of the representations for all models and for all datasets. We consider that the representations are better if the clusters are separable, i.e. with few outliers and with some space between them, and if they are expressive, i.e. clusters are spread and detailed. For each architecture, clusters of the end-to-end trained model (no suffix) are clearer and have less outliers than the ones trained in two steps (-Sep suffix). This shows that visual features learned with the action led to easier separable representations. We also observe that the models using the action to build the representations (PreLSTM and PreLSTM-Sep) are always able to cluster the representations with a varying quality depending on the dataset, where PostLSTM and PostLSTM-Sep models produce mixed representations for the hardest datasets. These results are in line with those found previously.

V. Conclusion and Perspectives

In this article, we studied the impact of action in visual representation learning in deep networks. Our questioning is raised by recent deep learning methods, which are increasingly using pretext tasks based on transformations that are action-related. Yet, these methods are not considering these *actions* to build their representations while sensori-motor theories, based on substantial evidence in many fields, claim that action is essential to perception. For this purpose, we studied and crossed two independent factors: 1- whether or not to use the action during the learning of visual features, and 2- whether or not to integrate the action in the building of image representations. By comparing these four configurations, we show that models including action during the learning of visual characteristics always perform better than their counterpart. We also observe that variations integrating the action directly in the representations tends to perform better, a trend that is more prominent for harder datasets.

These results are in line with sensori-motor theories and open perspectives to improve state-of-the-art representation learning methods by integrating the action both in the representations and during the learning. An other interesting perspective could be to study the influence of the action policy, in active learning and active perception contexts, on the learned representations. In the future, we want to extend the test-bed we elaborated for the study and make use of these first promising results to explore if it transfer to state-of-the-art representation learning methods and for more general problems studied by community (robotic, open world environments, etc.).

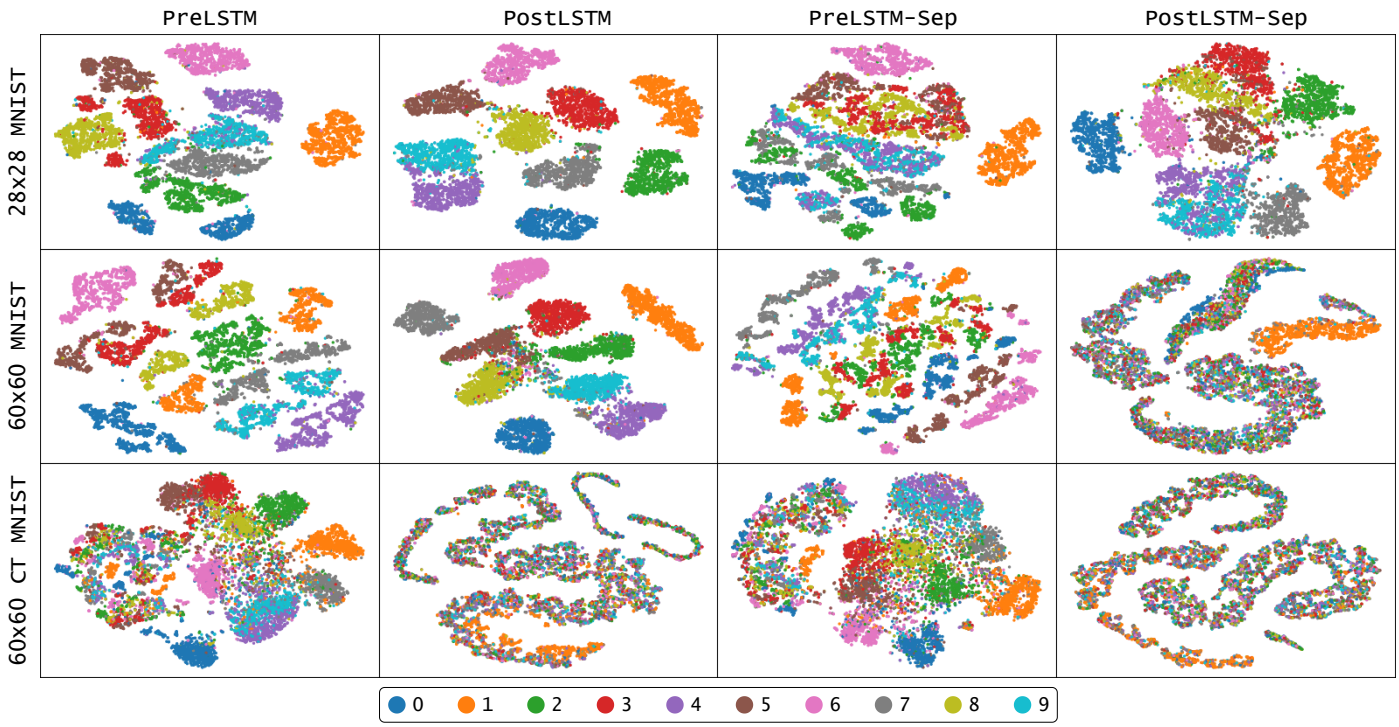

Fig. 6. t-SNE of the latent LSTM encoding for all models and datasets.

References

[1] Ablavatski, A., Lu, S., Cai, J.: Enriched deep recurrent visual attention model for multiple object recognition. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 971–978 (2017)

[2] Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE international conference on computer vision. pp. 37–45 (2015)

[3] Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755 (2014)

[4] Bardes, A., Ponce, J., LeCun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. arXiv preprint arXiv:2105.04906 (2021)

[5] Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., Sukhatme, G.S.: Interactive perception: Leveraging action in perception and perception in action. IEEE Transactions on Robotics **33**(6), 1273–1291 (2017)

[6] Friston, K., Adams, R., Perrinet, L., Breakspear, M.: Perceptions as hypotheses: saccades as experiments. Frontiers in psychology **3**, 151 (2012)

[7] Friston, K., Mattout, J., Kilner, J.: Action understanding and active inference. Biological cybernetics **104**(1), 137–160 (2011)

[8] Froese, T., Ziemke, T.: Enactive artificial intelligence: Investigating the systemic organization of life and mind. Artificial Intelligence **173**(3-4), 466–500 (2009)

[9] Gibson, J.J., Carmichael, L.: The senses considered as perceptual systems, vol. 2. Houghton Mifflin Boston (1966)

[10] Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: International Conference on Machine Learning. pp. 1462–1471. PMLR (2015)

[11] Guo, Z.D., Pires, B.A., Piot, B., Grill, J.B., Altché, F., Munos, R., Azar, M.G.: Bootstrap latent-predictive representations for multitask reinforcement learning. In: International Conference on Machine Learning. pp. 3875–3886. PMLR (2020)

[12] Hay, N., Stark, M., Schlegel, A., Wendelken, C., Park, D., Purdy, E., Silver, T., Phoenix, D.S., George, D.: Behavior is everything: Towards representing concepts with sensorimotor contingencies. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)

[13] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[14] Held, R., Hein, A.: Movement-produced stimulation in the development of visually guided behavior. Journal of comparative and physiological psychology **56**(5) (1963)

[15] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework (2016)

[16] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

[17] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[18] Laflaquière, A.: A sensorimotor perspective on contrastive multiview visual representation learning. IEEE Transactions on Cognitive and Developmental Systems (2021)

[19] Laflaquière, A., Garcia Ortiz, M.: Unsupervised emergence of spatial structure from sensorimotor prediction. arXiv e-prints pp. arXiv–1810 (2018)

[20] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[21] LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database (2010)

[22] Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. arXiv preprint arXiv:1406.6247 (2014)

[23] Mossio, M., Taraborelli, D.: Action-dependent perceptual invariants: From ecological to sensorimotor approaches. Consciousness and cognition **17**(4) (2008)

[24] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

[25] O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behavioral and brain sciences **24**(5), 939 (2001)

[26] Ranzato, M.: On learning where to look. arXiv preprint arXiv:1405.5488 (2014)

[27] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)