Development of Few-Shot Learning Capabilities in Artificial Neural Networks When Learning Through Self-Supervised Interaction

Viviane Clay, Gordon Pipa, Kai-Uwe Kühnberger, and Peter König

Abstract—**Most artificial neural networks used for object recognition are trained in a fully supervised setup. This is not only resource consuming as it requires large data sets of labeled examples but also quite different from how humans learn. We use a setup in which an artificial agent first learns in a simulated world through self-supervised, curiosity-driven exploration. Following this initial learning phase, the learned representations can be used to quickly associate semantic concepts such as different types of doors using one or more labeled examples. To do this, we use a method we call fast concept mapping which uses correlated firing patterns of neurons to define and detect semantic concepts. This association works instantaneously with very few labeled examples, similar to what we observe in humans in a phenomenon called *fast mapping*. Strikingly, we can already identify objects with as little as one labeled example which highlights the quality of the encoding learned self-supervised through interaction with the world. It therefore presents a feasible strategy for learning concepts without much supervision and shows that through pure interaction meaningful representations of an environment can be learned that work better for few-short learning than non-interactive methods.**

Index Terms—**Embodied AI, enactivism, reinforcement learning, representation learning, fast mapping, few-shot learning.**

I. Introduction

ARTIFICIAL neural networks (ANNs) by now excel at many complex tasks in fields such as visual, auditory, and natural language processing and often even outperform humans [1]. However, the types of mistakes that are observed in ANNs are often quite different from the mistakes humans would make. For instance, many image processing networks have been shown to be vulnerable to adversarial attacks [2], [3], [4] which means, small perturbations in the pixel values, often not even visible to the human eye, can lead to a misclassification. Also, certain natural images with easily interpretable image content to the human eye have been shown to trick most state-of-the art image classification networks [5] and a general over-reliance of ANNs on texture instead of object shape has been demonstrated [6], [7]. This shows that many networks do not have a true understanding of objects like humans do but over-fit on overall color, texture, and background cues.

There are big differences between the learning process in humans and learning in ANNs which leads to differences in behavior and the tasks that they excel at [8]. By focusing on the differences in the computational task that needs to be solved (interactive learning with weak supervision), we postulate that making these factors in the learning of ANNs more similar to human learning will lead to performance and errors more similar to what we observe in humans. All living beings interact in some way with the world. This makes it possible to learn more stable concepts about the world, relations between objects, and sensory-motor contingencies [9]. ANNs are often trained with no interaction with the world as well as fully supervised, beginning with the final task and with no gradual acquisition of knowledge. This is in stark contrast to what we believe to be the case in humans. Piaget proposes the development of a child to be split into several stages [10] and especially during the first stages knowledge is largely acquired through weakly-supervised interaction with the environment [11]. Although this model is not without criticism [12] the general idea that humans develop skills gradually over their lifespan through interaction with the world is widely accepted.

Humans have the ability to perform fast mapping, which is a phenomenon first detailed in children by Susan Carey and Elsa Bartlett in 1978 [13]. Fast mapping describes the observation that children can learn new concepts, words, or facts after minimal exposure to them. It has been demonstrated that a single exposure to a new word can be sufficient to lead to the child remembering the word a week later [13]. This means the child can make an instant association between word and meaning. This ability has also been found in other species such as dogs [14]. Some later studies found that for fast mapping to be successful, additional memory aids and specific learning conditions are needed [15], [16]. For fast concept learning of more abstract concepts in older children and adults even more cognitive mechanisms seem to

Manuscript received 3 February 2021; revised 18 August 2023; accepted 23 September 2023. Date of publication 9 October 2023; date of current version 5 December 2023. This work was supported in part by the funds of a research training group provided by the DFG under Grant GRK2340. Recommended for acceptance by K. Fragkiadaki. *(Gordon Pipa, Kai-Kühnberger, and Peter König are senior authors.) (Corresponding author: Viviane Clay.)*

Viviane Clay was with the Institute of Cognitive Science, University of Osnabrück, 49090 Osnabrück, Germany, and also with Numenta, Redwood City, CA 94063 USA (e-mail: vkakerbeck@uos.de).

Gordon Pipa and Kai-Uwe Kühnberger are with the Institute of Cognitive Science, University of Osnabrück, 49090 Osnabrück, Germany (e-mail: gpipa@uos.de; kkuehnbe@uni-osnabrueck.de).

Peter König was with the Institute of Cognitive Science, University of Osnabrück, 49090 Osnabrück, Germany. He is now with the Institute of Neurophysiology and Pathophysiology, University Medical Center Hamburg-Eppendorf, 20251 Hamburg, Germany (e-mail: pkoenig@uos.de).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3323040, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3323040

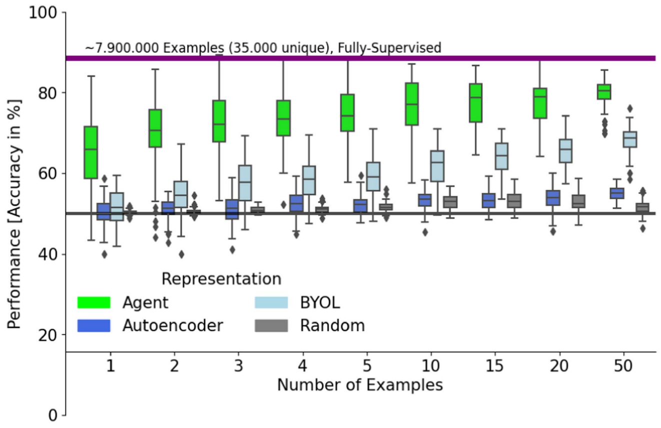

Fig. 1. Performance of recognizing a specific door (level door) in a 168 × 168 RGB image with 1-50 labeled examples. A comparison is shown between different learning setups used to learn the image embedding. The embedding learned self-supervised through interaction with a virtual world is shown in green. This is compared to a self-supervised representation learned without interaction using an autoencoder (dark blue) or contrastive learning (light blue). Grey boxes show the baseline performance of using a random network. The purple line indicates an upper bound performance, obtained from a fully supervised classifier trained on six million labeled examples. The grey line indicates chance performance (50%). The box plots show median performance as well as first and third quartile performance over 100 random sets of examples.

be used ranging from analogical comparisons [17], Bayesian reasoning [18] to abstraction by varying prior knowledge [19]. In general the amount of labeled examples needed for a human to learn a new word-meaning-association is several orders of magnitudes smaller than what is needed for classical ANNs [20]. Several one-shot and few-shot learning approaches have been developed for ANNs in an attempt to classify new observations based on only one or few labeled examples [21], [22], [23], [24]. However, the few-shot task is usually only a generalization to new classes of a task that has previously been trained for, using a large data set of other classes. This means that these approaches still require large labeled or weakly labeled data sets of objects other than the tested ones to learn a meaningful embedding space.

When learning through interaction, the agent creates its own data set by adjusting its action policy to maximize future rewards. Previous work shows that allowing an ANN to actively sample the data it uses to improve a pre-trained object recognition network can improve performance beyond that of ANNs learning from only static data sets [25]. The agent does not only create its own data set but can actively select the data distribution in a way that it improves its performance on the task at hand.

Here we investigate whether training an ANN more similar to how infants may learn leads to phenomena similar to those found in humans. We focus on self-supervised, interactive learning and look at the phenomenon of fast mapping. We propose to learn a meaningful embedding of visual observations purely through self-supervised interaction within a simulated world. This interactive learning period can then be followed by a supervised, fast mapping like, association between representations and concepts using very few labeled examples. Overall, this learning setup (shown in Fig. 2) seems more similar to how we appear to learn and does not require large amounts of labeled training data.

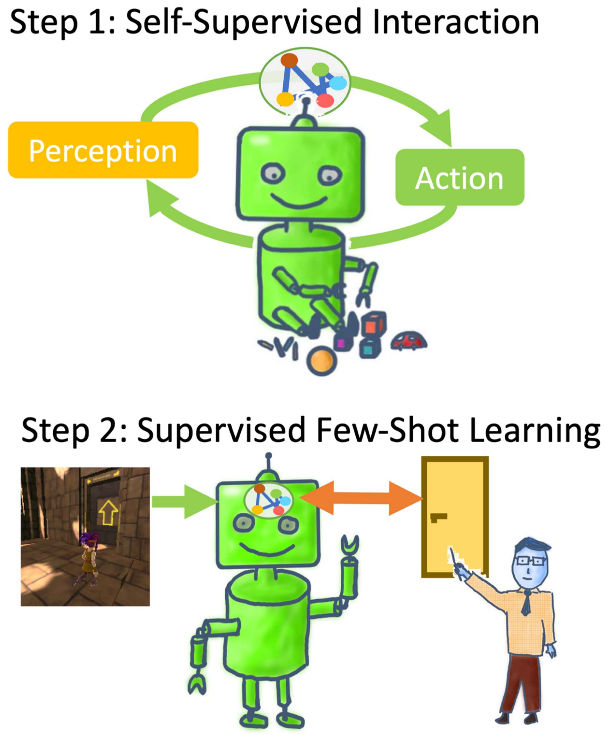
Step 1: Self-Supervised Interaction
Perception
Action
Step 2: Supervised Few-Shot Learning

Fig. 2. Learning procedure. First, the agent learns an embedding of high-dimensional visual space through self-supervised interaction with an environment (top). Then, this embedding can be used to perform few-shot learning (bottom). To show the advantage of interactive learning, we replace step one with other, non-interactive representation learning methods in our experiments. We also show that step 2 can be performed using different few-shot probing methods.

II. Methods

A. Network Structure and Training

To demonstrate the advantages of interactive, self-supervised learning we contrast four deep neural networks with each other. All four networks have the same network architecture between input and encoded state. They differ in their output and the objective that they are trying to optimize as well as the way in which they are optimized. We take a look at three networks which are trained without any human supervision and one network trained fully supervised. The three self-supervised conditions comprise one agent, trained using deep reinforcement learning, one autoencoder, and one contrastive learning method called BYOL [27]. There are of course many other self-supervised learning methods out there [28], [29], [30] but we chose these as they produced state-of-the-art results (at the time of writing) without requiring negative examples or a network structure and learning parameters that do not match the interactive agent.

The agent learns through interaction with the world, outputting actions and optimizing a curiosity objective [31]. The autoencoder learns to compress the input into an encoded state and decode this representation again into an image, optimizing a reconstruction error between input and output. The contrastive learning network is optimized to represent two augmentations (such as random crop, horizontal flip, color distortion, and Gaussian blur) of the same image with a similar encoding.

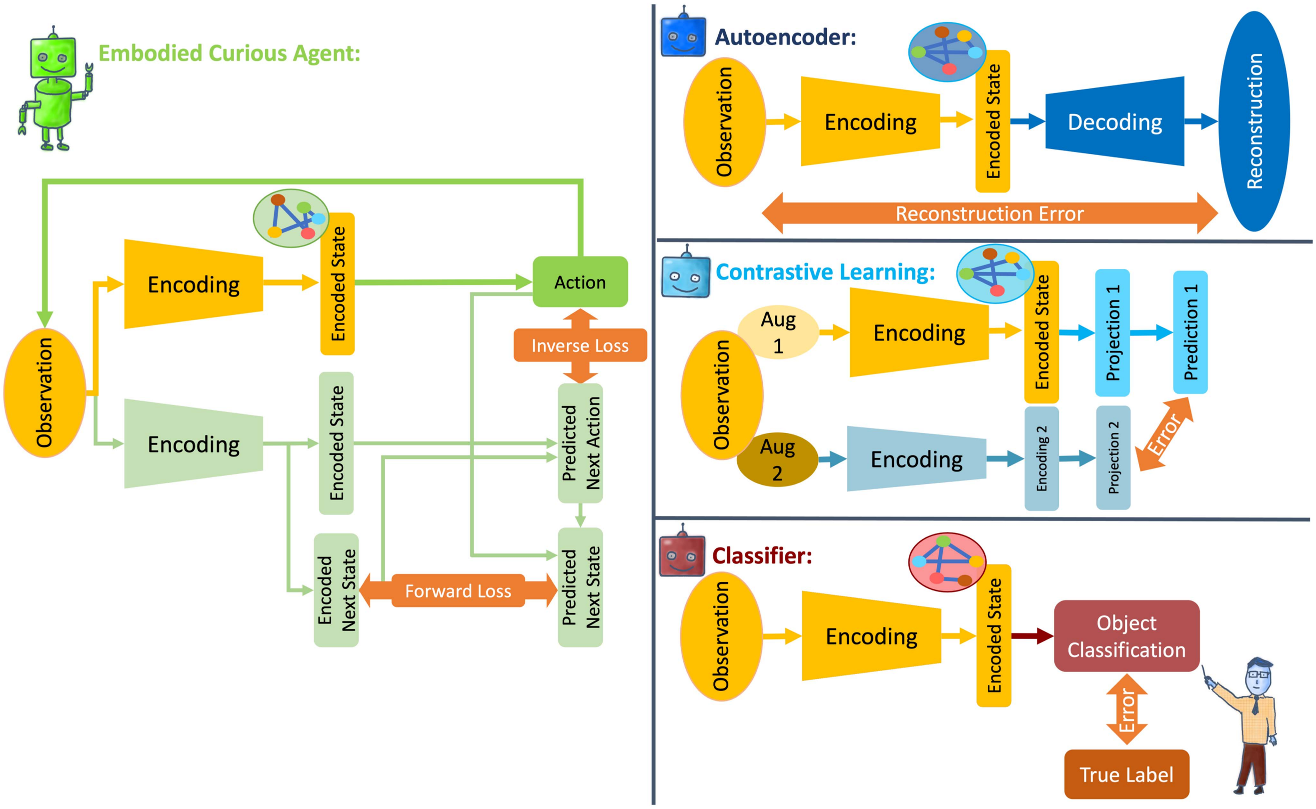

Fig. 3. Network structure of the four learning conditions compared in this paper. For simplicity the individual layers are omitted. The conditions compared are interactive, self-supervised learning (left); passive, self-supervised learning through reconstruction error (top right) or using contrastive learning (middle right); and passive, fully-supervised learning (bottom right). Yellow elements have the same architecture in all conditions. The encoded state marked in yellow is used for analysis and fast concept mapping. All conditions learn on 168 × 168 × 3 images from the obstacle tower environment, using the Adam optimizer and a batch size of 256. For the code used for network training and analysis and more details on network parameters, see the links in A.2, available online. The interacting agent uses PPO [26] to optimize the action policy with respect to the curiosity reward (a combination of forward and inverse loss). For simplicity the details of PPO are not shown in this figure.

This is done by using the encoding of the first augmentation to predict the encoding of the second augmentation which is produced by a copy of the first network with a rolling average of its weights [27]. The fully-supervised network is trained on an object classification task in which the task is to output the presence of 8 different concepts in the input image which are the same concepts as above with additional option to classify the image as containing 'no door' and 'puzzle piece'.[1]

All four networks receive a visual input of size 168 × 168 × 3, encode it to an encoded state of size 256 and then use this encoded state to solve their respective task. All networks have the exact same layers and structure between the input and encoding layer (for details see A.2), available online. In the following experiments we will look at the four encoded states (activations in the last dense layer before task specific output), how they differ, and how well we can extract semantic concepts from them.

The four networks are trained with observations collected in the obstacle tower environment [32]. We did not modify this environment in any way for the experiments here. The environment is a simulated 3D maze with a time limit and several obstacles. It consists of randomly generated levels which are made up out of several rooms connected by doors. Doors have visually marked properties such as leading to a next room, a next level, or only opening with a key or after solving a spatial puzzle. The time limit can be extended by entering new levels and by picking up blue time orbs. The visual theme can vary across levels and the illumination of different rooms is selected randomly. An agent in this environment receives a visual camera input, taken at its current position, as well as a vector of size 8 with auxiliary information such as time left, current floor number, and number of keys holding. In this paper, we only look at the encoding created of the visual input. Note that the 8 values of auxiliary information were part of the original benchmark environment but do not contain any reward information, are not used as a supervisory signal, and do not contribute to the embedding of the visual input. The autoencoder, BYOL learner, and the classifier receive camera images collected by a trained agent in the obstacle tower environment as their input in random order. Therefore all four networks are trained on 168 × 168 × 3 RGB images from the obstacle tower environment (examples in

[1] No door is excluded from the analysis above as it is a negation of the other door concepts and is implicitly expressed in the door concepts not exceeding the detection threshold. Puzzle piece is also excluded because the curious agent does not perform well enough to reach the higher levels at which the puzzles appear.

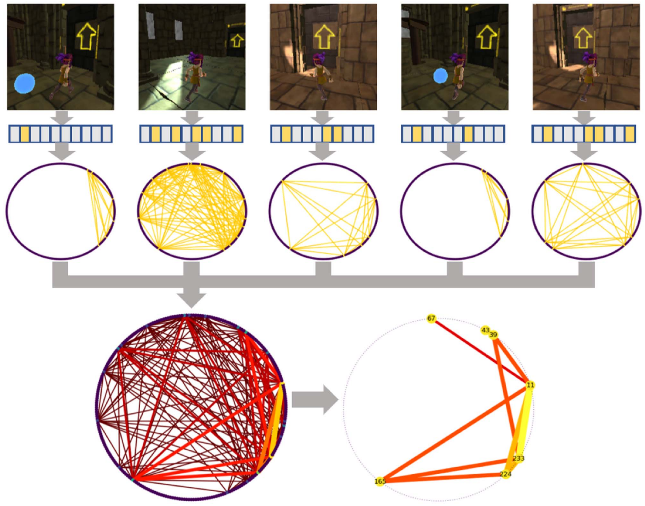

Fig. 4. Fast concept mapping. Demonstrated on the concept 'level door', using five example images of the concept and the encodings of the trained agent network. First, the encodings corresponding to the input images are extracted and rewritten as a connectivity graph (top rows). Next, the sum of the five connectivity graphs is calculated (bottom left). Color represents the connection strength which is the number of examples in which the connection is present. The concept is then defined be the P strongest connections, here ten,[2] and their weights (bottom right).

Figs. 4 and 10 (in the supplementary material, available online)) and we are taking a closer look at their respective encodings of these images after training.

The agent network shown in Fig. 3 represents the interactive learning condition. In our interactive learning condition, interaction happens through a physically simulated body with a closed loop between action and perception. The input observation determines which action the network produces and this action in turn influences the next observation. The network learns to produce actions that lead to rewards. Rewards are conventionally received from the environment for achieving certain goals such as walking through a door, entering a new level or picking up an object. Here, we want to look at learning without any external supervision, so we omit all rewards from the environment and instead replace them with internal rewards. These internal rewards are produced by a curiosity module introduced in [31] and visualized in matte green in Fig. 3.

The curiosity module comprises another neural network which optimizes two objectives. The first one is to predict the next encoded state based on the current encoded state and the current action. The difference between the predicted next state and the actual next state is called the forward loss. The second objective of this network is to determine the action that was performed between the current state and the next state. The difference between the inferred action and the actual action performed is called the inverse loss. As both tasks are performed using the same encoded state, optimizing the inverse loss makes sure, that this encoded state of the curiosity network contains action relevant information. The curiosity network tries to minimize the inverse loss and the forward loss by making accurate predictions about the next state and the performed action [31].

The action network (shown in bright green) in contrast receives the inverse loss and the forward loss as a reward weighted with 0.8 and 0.2 respectively [31]. This enforces the action network to learn a policy which produces actions that lead to new, unpredicted observations. It implicitly rewards the network for navigating through the world, entering new rooms and new floors as these behaviors lead to new observations. An agent with a policy that stays in only one room would have a small forward and inverse loss as the curiosity network can make good predictions in this known environment. However, the action network would not receive many rewards from the curiosity network as it does not receive any unpredicted observations. Therefore the action networks' policy is adapted to enter through doors and explore new floors to receive the curiosity networks' rewards. Using only intrinsic curiosity the agent can learn a policy that navigates through the 3D tower environment without any external supervision.

The agent network is optimized using proximal policy optimization [26]. Therefore, the action network does not only produce action probabilities but also a value estimate for each time step. This value estimate is used to update the network weights in a way that the cumulative reward is maximized.

To keep conditions as comparable as possible and only vary the type of learning, the three networks not only have the same structure from input to encoding but also use the same optimizer (Adam optimizer [33]) and the same batch size (256) to update their network weights. This allows us to compare between interactive, self-supervised, and fully supervised training conditions.

B. Fast Concept Mapping

We propose fast concept mapping (FCM) as a simple probing method to directly read out the encoding of a concept from a representation using a few examples of the concept. This readout only works if the concept is encoded uniquely in the representation, but if it is, it can be read out with very few labeled examples. This is only one possible probing method and we show later that other existing methods also produce similar results. However, FCM has the advantage that it only requires positive examples and only looks at a small subset of the encoding neurons during inference.

To read out a concept using FCM, a small number of labeled examples of the concept are needed. Fig. 4 shows FCM on the example of the concept 'level door' which is a door with a yellow arrow that leads to the next level of the environment. The example images should be different from each other and representative of the concept. It is no problem if also other concepts are present in some of the examples. In the example shown here, five instances of the concept 'level door' are used to extract the concept definition. To do this, the five examples are given to the encoding network and the encoded representations X of the images are extracted. Next, for each pair of neurons that is active together in each of the five representations we assume a connection between them. If we only have one encoding, then this means that all active neurons in the encoding have connections to each other. In the following step, the sum of all connections from the example input encodings is calculated.

This means, that neurons that are active together in multiple of the example inputs now have a stronger connection to each other. To define the concept 'level door' the P strongest connections are taken from the summed-up connectivity graph. In this example, we take the ten strongest connections. These ten pairs of neurons as well as the normalized strength of their connection now define the concept for 'level door'. The same procedure can be repeated for any other concept one would like to extract from the encoding, resulting in a connectivity definition for each concept.

In other words, if E are the list of neurons in an encoding and X are a list of activations $x = f_E(input_c)$ of neurons in E in response to N examples of concept C, then concept C is defined as $(\text{combinations}_c, \text{weights}_c)$ where

$$\text{combinations}_c = \underset{(i,j)\in\binom{E}{2}}{\overset{P}{\text{argmax}}} \sum_{x\in X} \alpha(x_i) * \alpha(x_j), \tag{1}$$

and

$$\text{weights}_c = \left\{ \sum_{x\in X} \alpha(x_i) * \alpha(x_j) : (i,j) \in \text{combinations}_c \right\}, \tag{2}$$

$\binom{E}{2}$ is a list of all possible neuron combinations in encoding E. P denotes the pattern complexity, which is how many pairs are used to define the concept. $\text{argmax}^P_{(i,j)\in\binom{E}{2}}$ are therefore the P strongest connections between neurons for the X examples. α is a binary activation function which denotes whether neuron i is active or not such that $\alpha(x_i) * \alpha(x_j)$ is one if both neurons are active and zero otherwise. If x is already a binary activation pattern, then α is not necessary. In this paper we use $\text{weights}_c = \{\frac{w}{\sum \text{weights}_c} : w \in \text{weights}_c\}$ to normalize the weights for more intuitive threshold selection.

To detect the extracted concepts in a new, unlabeled input, one first needs the connectivity graph for this new input. This graph can be obtained in the same way as during concept extraction, by getting the encoding of the input from the trained network and making a connection between every pair of neurons that is active together in response to the input. To compare the new inputs' graph with the concept definition one looks for how much evidence for the concept is found in the new graph. This is done by adding up the normalized weights of every connection in the concept definition that can also be found in the new graph. If every connection from the concept definition would also be present in the new graph, meaning that all pairs of neurons in the definition are active in response to the new input, then the evidence for this concept would be one. If no connection from the concept definition is present, then the evidence for this concept is zero. A threshold θ is set, which determines how much evidence for a concept needs to be found in a new input encoding for this concept to be classified as present in the input. An investigation into the effect of the threshold on concept detection is provided in the next section.

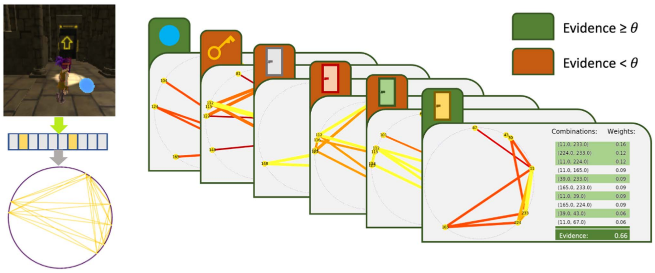

Fig. 5. Fast concept mapping inference. To detect concepts in an unlabeled input the connectivity graph is extracted from the encoding corresponding to the input (left). Each learned concept can then be compared to this graph and every connection from the concept definition present in the new input adds more evidence for the concept. How much evidence is added is determined by the weight of the connection in the concept definition. If the evidence is above a threshold, the concept is classified as present in the new input (right).

This means, in a new example with neuron activations x, the concept C is present if

$$\sum_{(i,j),w\in C} \alpha(x_i) * \alpha(x_j) * w \geq \theta, \tag{3}$$

where θ is the threshold of how much evidence needs to be present for the concept and C is the concept definition extracted in (1) and (2).

In the example shown in Fig. 5, we find that six of the ten connections in the concept definition for 'level door' are also active in the test image encoding shown on the left side. If we now add up the weights for these six connections, we get an evidence of 0.66 for this concept in the image. As this is above our threshold, we classify the level door as present in the image. We now repeat the same procedure on all the other concepts that were extracted and detect additionally to the level door the concept 'blue time orb'. For the other concepts we do not find enough evidence in the encoding and therefore classify them as not present.

III. Results

A. Structure in Representations Learned Through Interaction

All results are obtained from artificial neural networks trained on 168 × 168 × 3 dimensional visual observations from a 3D maze environment shown in the supplementary material, available online, (Fig. 10) and described in the methods Section II-A. The investigated encodings are the activations of 256 neurons in the last hidden layer before action selection for the agent, deconvolution and reconstruction for the autoencoder, projecting and predicting the projection of the second augmentation for BYOL, and classification in the classifier. The network structure leading up to this representation is the same for all three conditions and described in detail in the methods Section II-A and the appendix A.2, available online.

As shown in a previous study [34] conducted in the same environment, the agent learns a sparse and meaningful encoding of its high dimensional visual input. As opposed to the previous study, this is achieved without any external rewards from the

[2] It may be difficult to see all ten connection in the figure due to their overlaps. In Fig. 5 (bottom right) all ten connections are listed together with their connection strengths.

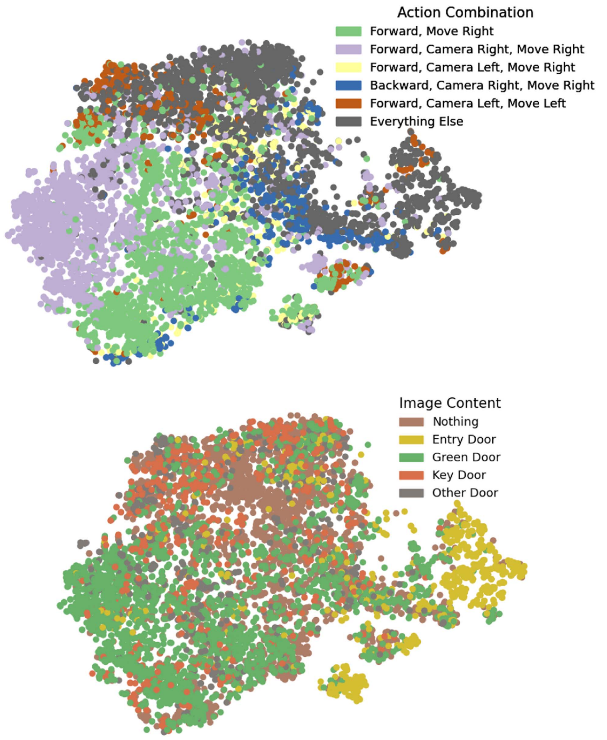

Fig. 6. Meaningful structure in the image encodings of the interactive, curious agent. t-SNE projection of latent representations colored by action (top) and colored by type of door (bottom). Each dot represents the encoding of one input image (168 × 168 × 3) in the hidden layer of the trained agent (256 neurons). The grey colored dots in the action-colored projection summarize 34 action combinations. The other five colors show the five most common action combinations.

environment, using only intrinsic curiosity as a learning signal [31]. In a set of 8400 frames collected in the 3D environment there are an average of 8 neurons active in each frame[3] (min $= 0$, max $= 29$, var $= 13.12$) which is 3.15% of the 256 neurons in the visual encoding. 91.02% of the neurons in the visual encoding are active in at least one frame of the test run. The most active neuron is active in 54.71% of the frames. This is a very sparse encoding of the 84,672-dimensional visual input with a wide variety of selective activations in the hidden layer.

With an unsupervised dimensionality reduction method such as t-SNE [36] one can investigate whether meaningful structure can be found in the encodings. In Fig. 6 the 8,400 test encodings of dimensionality 256 are projected into two-dimensional space using t-SNE. In such a strong reduction of dimensionality not all information can be preserved but nevertheless one can see some structure regarding actions (top) and objects (bottom) in the encodings. Especially the most common action combinations as well as the level entry door are encoded distinctly even in the two-dimensional projection. In our FCM approach we make use of the full dimensionality of the encodings to extract even more structure and to be able to disentangle overlap between object and action encodings.

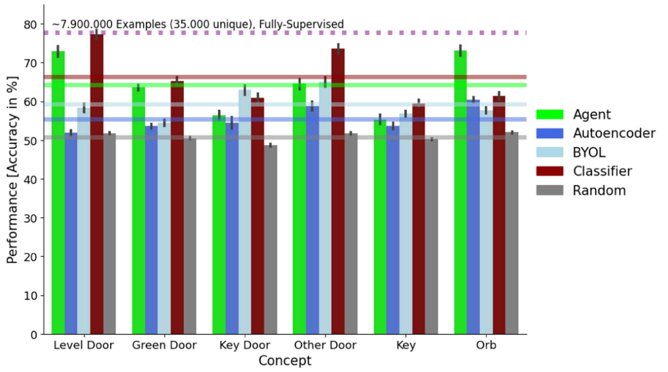

Fig. 7. Accuracy for six different concepts comparing representations learned by the interactive, curious agent, an autoencoder, BYOL, a classifier, and a random network. Five random example images are used to extract each concept with a pattern complexity P of 10. As threshold θ we use the optimal threshold for each condition (see threshold comparison in Fig. 8). 95% confidence intervals for 100 random sets of five examples are indicated for each bar. The dotted line shows the average performance of the classifier using its fully supervised trained read-out layer instead of the few-shot probing method. Horizontal lines indicate average performance over all concepts for the five conditions.

B. *Few-Shot Learning on the Different Representations*

After demonstrating that the learned representation shows some meaningful structure in action and object space (Fig. 6), we want to see whether concepts can be extracted automatically from the encodings with few labeled examples. To do this we design a procedure called fast concept mapping (FCM) which is described in detail in the methods section. The general idea is to take some example images of a concept (for example five images of a door) and look at consistent activation patterns in the learned encodings of these examples. These consistent patterns then define the concept and can be detected in the activations corresponding to novel observations. Here we use five random example images for each of the six different concepts. Then we test the extracted concepts on 250 test images of the concept and 250 randomly sampled from the other concepts such that chance performance is at 50%. We compare the performance in detecting the extracted concepts between the interactive agent and the four control conditions. The autoencoder, BYOL, and the classifier representations are trained as described in methods. The random condition uses the randomly initialized weights and network structure as used at the start of training of all other conditions. Performance shown in this section refers to the ability to extract concepts from the different learned representations of the visual input. It has nothing to do with the performance on the task that the networks were originally trained on. Therefore we compare here whether different training objectives (interactive and self-supervised, self-supervised, fully supervised, none) lead to a difference in the ability to extract concepts.

Fig. 7 shows the performance of FCM on the six concepts for the four representations learned under different conditions. FCM on the agent representation outperforms FCM on the autoencoder, BYOL, and the random representation. The agent's

[3] Active is defined as an absolute activation bigger than the average activation of this neuron. This definition of activation α compared to the original activation strengths and a universal threshold can be seen in Fig. 11 in the supplementary material, available online. The adaptive threshold is biologically and theoretically motivated [35] and helps with the much more dense encoding of the autoencoder, BYOL, and classifier.

performance differences between concepts can be explained by its learning progress. Concepts that cannot be extracted well, such as the key and the key door, are not as well learned in the policy as the other four concepts. While the agent can already pick up time orbs and navigate through level doors and green doors to reach level five, it has trouble when needing to pick up a key to go through the key door, which is a task introduced at this point. The key and key door concepts do not seem to be encoded well yet (and have been observed less often at this point) and are therefore also harder to extract. Overall, FCM works best on the agent representation and the classifier representation with some performance differences between concepts.

The comparison to the performance of the classifier read out layer, indicated by the dotted line, shows that the learned representations most likely contain more object information which could be read out with a more sophisticated method. The classifier uses a fully supervised trained read-out layer for this, which is fitted to extract the object information using over 7.9 million labeled examples. As the output of the classifier is based on the embedding of the classifier we can see directly from the comparison between the red line and the dotted line how much improvement in performance a better probing method gives and how much depends on the encoding.

To investigate how few examples are needed to extract the concepts from the learned representations we measure the performance when using 1, 2, 3, 4, 5, 10, 15, 20, and 50 examples. The effect of the number of labeled examples on the performance is shown in Fig. 1 for the concept 'level door'. One can see that already with only one labeled example the agent achieves a performance significantly above chance. Showing more labeled examples leads to an increase in performance for the agent. Also the autoencoder and BYOL performance increases with more labeled examples but even with 50 labeled examples they are still outperformed by the agent. The fully supervised classifier read-out layer outperforms the agent but has been trained with several orders of magnitude more labeled examples. Impressively, the agent can even reach the performance of the fully supervised classifier when given the right examples, as indicated by the whiskers on the box plots. A more targeted concept extraction using representative instances of a new concept to learn it could therefore lead to even better performance.

1) Parameter Selection: To perform FCM two parameters need to be selected. The first parameter is the pattern complexity P which is the number of neuron pairs that is used to define the concept during concept extraction. The second one is the threshold θ which determines how much evidence for a concept is needed to classify it as present during inference. Finding the best values for these parameters requires more than the five examples used for the previous results. However, once the ideal parameters are found for one concept they can be used for all other concepts as we can observe the same trend in all concepts (see Fig. 8(b) and (d)).

FCM is surprisingly robust regarding the pattern complexity parameter. As shown in Fig. 8(a), using a higher pattern complexity does not have a strong effect on the performance. FCM makes the implicit assumption that a concept is encoded with a small pattern of consistent neuron activations and not with a large set of neurons in the encoding. If only a few neurons encode a concept, then using a higher pattern complexity to define the concept does not add further information. Especially in the sparse agent representations where on average only 8 neurons are active in each frame, it does not help much to add more neuron pairs to define a concept.

The performance of FCM can be more sensitive to the selection of the threshold parameter. As shown in Fig. 8(c), the choice of the threshold determines the quality of concept extraction. Depending on how many neurons an encoding used to encode a concept, the ideal threshold is closer to zero or 100 percent. When comparing the ideal threshold on the agent representations for the different concepts (Fig. 8(d)) a similar trend for all concepts can be observed. However, there are some concepts where the threshold has little effect on performance (for example key door and key). These concepts therefore could not be extracted effectively from the representation.

2) FCM Compared to Other Methods: The method introduced in Section II-B is just one of many ways one could extract the encoded patterns for different concepts. Here we look at a few alternative methods that can be applied using only a few labeled examples. While FCM can also be used to see whether a concept is generally encoded in a representation, there are better methods out there for this analysis if number of examples is not an issue [37]. Here we are interested in how suitable the learned representations are for few-shot learning.

Overall, FCM is comparable in performance to support vector machines and linear regression (Fig. 9). However, those two methods also require negative examples to learn a decision boundary which is not necessary in the approach used here. FCM is outperformed by using the cosine similarity of new observations to the mean encoding of the example images used to learn the concept. However, cosine similarity as well as SVM, decision trees and linear regression all use the entire encoding to make a classification. FCM is more efficient in this regard, only looking at the P neuron pairs that define the concept. This seems more biologically realistic as it only requires a hand-full of read-out connections to the most consistently active pairs of neurons to detect a concept. Additionally, this should be more robust to concept overlap (such as multiple concepts being present in one image) and noise in the example activations as it discards all activations that are not consistently active while the other methods still take them into account. Unfortunately, the relatively low complexity of the environment used here did not allow for testing this.

Whether one uses the N most consistently active pairs or single neurons does not seem to make a large difference in performance here. The ranking of the different conditions remain the same when using single activations (see bottom row of Fig. 9). We use pairs of neurons here as this may lend a larger representational capacity of the embedding to encode more distinct concepts in a more complex world. However, this was not tested here as learning in an environment with such a large number of objects was not feasible. Triplets performed slightly worse than single or pair activations. This is possibly due to it being rarer that three neurons consistently fire together in the example images, especially in the agents' sparse representation where only an

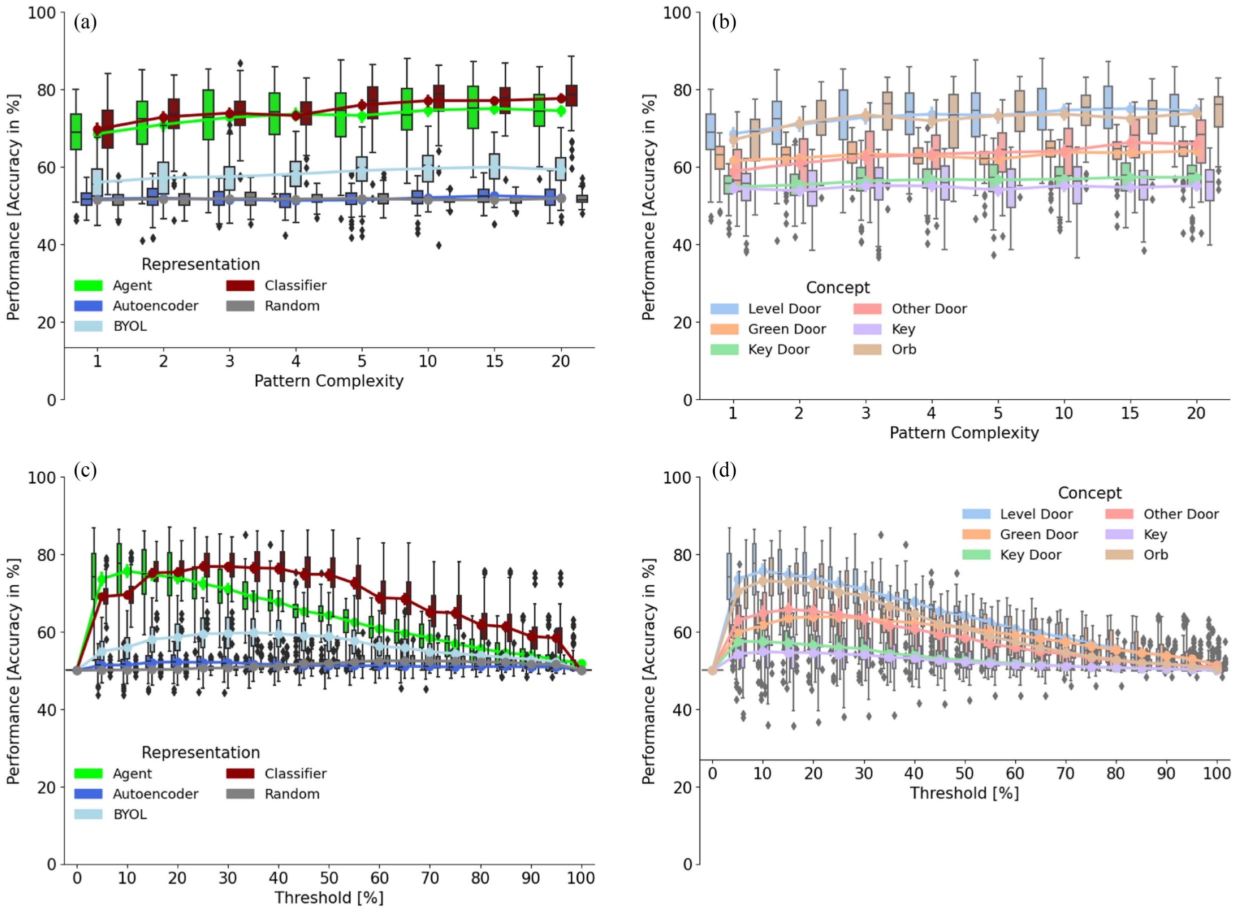

Fig. 8. Effect of pattern complexity P and threshold θ on concept mapping performance. The box plots show median performance as well as first and third quartile performance over 100 random sets of examples. The point plots show the point estimate mean and its 95% confidence interval. For all experiments five examples are used for concept extraction. In the pattern complexity comparison (a and b) the optimal threshold for each condition is used. In the threshold comparison (c and d) a pattern complexity of 10 is used. (a) Comparison between different representations dependent on the pattern complexity used to define the concept 'level door'. (b) Comparison of pattern complexity for each concept on the agent representation. (c) Threshold (amount of evidence needed for concept detection) comparison on the concept level door for all conditions. (d) Threshold comparison for all concepts in the agent.

average of eight neurons are active in each frame and each frame can contain several concepts.

Overall, the main results of the differences between representation learning methods and the strength of interactive learning can be replicated irrespective of the method used for concept extraction (see Fig. 9). FCM is a simple and easy to implement method that does not rely on negative examples or taking all neuron activations into account during inference and therefor could also be implemented in biological systems which is why we choose it here.

IV. Discussion

The most obvious advantage of the method introduced here is the small number of labeled examples that is required to learn a concept, with an above chance performance with as little as one labeled example. As opposed to other few-shot learning methods [21], [22], [23], [24], one does not only require very few examples of the test classes but also the representation learning works without any labeled examples. This makes the learning comparable to what we observe in humans. An interactive, curiosity driven learning period leads to a meaningful representation of the world, which can then be used to perform new tasks such as object classification with few supervised learning examples.

The fast concept mapping method introduced here is a fast association between consistent neuron firing patterns and semantic concepts. It works instantaneously without the need for weight optimization and gradient descent. The method of looking for correlated firing patterns that persist over different instances of the same concept is simple enough that it could be implemented with biological neurons [35], [38]. The success of this method assumes that the representation that it is applied to uses a unique and consistent encoding of the concept. We showed that an agent who learns through self-supervised interaction with the world can learn such an encoding of action relevant concepts. Extracting a new concept does not require any retraining and does not influence the concepts that have already been extracted.

Even though in theory, arbitrarily many concepts can be extracted if they are encoded, this is not so easy in practice. The interactive learning phase can require long training times and due to the absence of supervision it cannot be explicitly chosen which and how many concepts will be encoded. Learning in

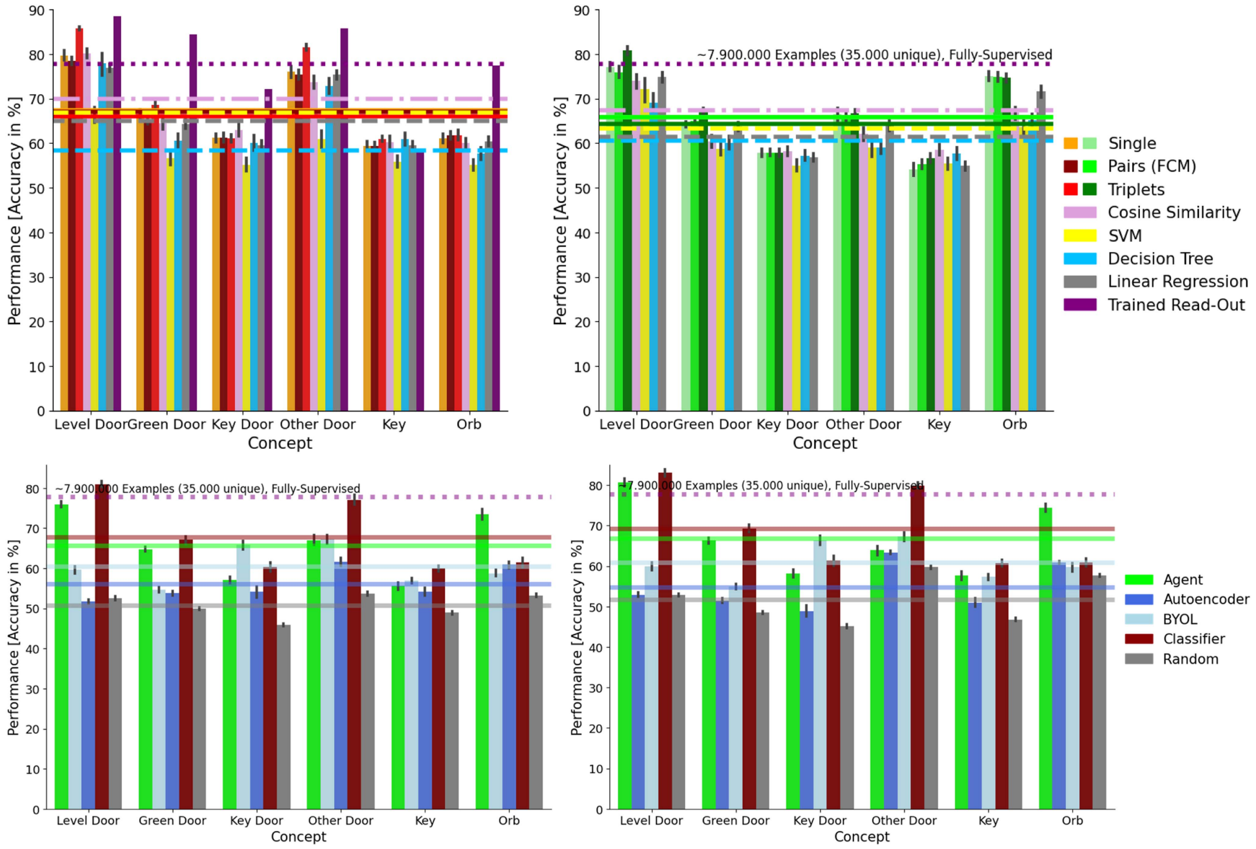

Fig. 9. Comparison of few-shot probing methods on the classifier representations (top left) and the agents' representations (top right). In the single condition, a concept is defined by the P most consistent single neuron activations. Pairs refers to the FCM implementation used in the main manuscript. Triplet defines concepts by consistent triples of activations instead of pairs. Bars show results from 100 random sets of 6 examples for each concept and condition. For support vector machines (SVM), decision trees and linear regression 3 positive and 3 negative examples are used. Horizontal lines indicate the average performance over all concepts for a specific few-shot probing method. Dashed lines indicate conditions where negative examples were used. Solid lines indicate conditions where during inference only a small subset of neurons is taken into account. The dotted line indicates the upper-bound performance, measured by the performance of a fully supervised classifier. The bottom row shows the same as Fig. 7 but looking at the top two performing alternatives to pairwise FCM. Both using single neuron activations instead of pairs to define concepts (bottom left) or cosine similarity (bottom right) result in the same overall ranking between the different representations and support the main message of the paper.

this interactive, self-supervised setup is incredibly difficult and sample efficient learning in more complex environments is still an area of active research [39]. However, human learning also takes place over a long time frame, which may lead to more stable and robust representations that avoid current pitfalls of artificial neural networks.

It would be nice to evaluate this method in an environment with a wider range of interactable objects. Interesting environments for such experiments have been introduced recently [40], [41]. However, since reinforcement learning using only curiosity is incredibly slow and in the environment presented here already takes place over the course of multiple weeks, a more complex environment is not included here. The difficulty of interactive, unsupervised learning is the main practical limitation of the methods used in this paper.

A natural next step would be to use the extracted concepts and introduce them back into the interactive learning setup as an additional aid in the decision-making process. This can on the one hand, help with learning a more efficient policy due to the compressed information of concepts in the observations. Additionally, it could lead to more refined encodings of the concepts themselves and a higher accuracy in detecting them. Ultimately, it could achieve language grounding without explicit enforcement through the constant presence of a language grounding task, as it is currently done [42], [43], [44].

Another extension could be to perform reasoning on the levels of the concepts. Some concepts, such as the different types of doors, are sub concepts of a higher-level concept. Other concepts can be combined with each other, such as 'left' and 'right' or 'far away' and 'close' can be combined with the different objects. Some concepts can be applied to certain types of objects, for example doors can be 'opened' or 'closed'. Due to the independent populations of active neurons in response to different concepts in a good encoding, all these concepts could be extracted independently using the FCM method and put into logical relations, either automatically deduced from activity overlap or by hand.

V. Conclusion

Overall, we have shown that representations learned through embodiment with no external supervision encode meaningful information about the content of high dimensional visual input. These representations can be associated with semantic labels almost as well as representations that were optimized fully supervised for object classification. This association works fast and robustly with few randomly chosen labeled examples, similar to the ability of children to perform fast mapping. We find that concepts such as different types of doors and other action-relevant objects are encoded even though no semantic concepts were ever explicitly taught during training. Our results show the viability of a new approach to train ANNs, inspired by the way humans seem to learn. This approach focuses on self-supervised learning through interaction and only transitioning later on to supervised concept learning with few labeled examples.

Author Contributions

Viviane Clay conceived the idea for the experiments and the fast concept mapping procedure, implemented the training, FCM and analysis and took the lead in writing the manuscript. All authors discussed the experiment design and results and contributed to the final manuscript. Peter König, Gordon Pipa and Kai-Uwe Kühnberger supervised the project.

Competing Interests Statement

The authors declare no competing interests.

Acknowledgment

Additional resources for the review of the paper were provided by Numenta. We would like to thank the reviewers for many insightful comments that significantly improved this manuscript.

References

[1] M. Alam, M. D. Samad, L. Vidyaratne, A. Glandon, and K. M. Iftekharuddin, “Survey on deep neural networks in speech and vision systems,” 2019, *arXiv:1908.07656*.

[2] C. Szegedy et al., “Intriguing properties of neural networks,” 2013, *arXiv:1312.6199*.

[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” 2016, *arXiv:1607.02533*. [Online]. Available: http://arxiv.org/abs/1607.02533

[4] S. Qiu, Q. Liu, S. Zhou, and C. Wu, “Review of artificial intelligence adversarial attack and defense technologies,” *Appl. Sci.*, vol. 9, p. 909, Mar. 2019.

[5] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, “Natural adversarial examples,” 2019, *arXiv:1907.07174*. [Online]. Available: http://arxiv.org/abs/1907.07174

[6] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, “Deep convolutional networks do not classify based on global object shape,” *PLOS Comput. Biol.*, vol. 14, no. 12, pp. 1–43, Dec. 2018. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1006613

[7] W. Brendel and M. Bethge, “Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet,” in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=SkfMWhAqYQ

[8] A. M. Zador, “A critique of pure learning and what artificial neural networks can learn from animal brains,” *Nature Commun.*, vol. 10, no. 1, 2019, Art. no. 3770. [Online]. Available: https://doi.org/10.1038/s41467-019-11786-6

[9] A. K. Engel, A. Maye, M. Kurthen, and P. König, “Where’s the action? The pragmatic turn in cognitive science,” *Trends Cogn. Sci.*, vol. 17, pp. 202–209, 2013.

[10] J. Piaget, *The Origins of Intelligence in Children*, vol. 8. New York, NY, USA: International Universities Press, 1952.

[11] J. Piaget, *Judgment and Reasoning in the Child*. Oxford, U.K.: Harcourt, Brace, 1928.

[12] Z. Babakr, P. Mohamedamin, and K. Kakamad, “Piaget’s cognitive developmental theory: Critical review,” Aug. 2019. [Online]. Available: https://www.researchgate.net/publication/335219854_Piaget%27s_Cognitive_Developmental_Theory_Critical_Review

[13] S. Carey and E. Bartlett, “Acquiring a single new word,” in *Proc. Stanford Child Lang. Conf.*, 1978, pp. 17–29.

[14] J. Kaminski, J. Call, and J. Fischer, “Word learning in a domestic dog: Evidence for “fast mapping”,” *Science*, vol. 304, pp. 1682–1683, 2004.

[15] P. Gurteen, P. Horne, and M. Erjavec, “Rapid word learning in 13-and 17-month-olds in a naturalistic two-word procedure: Looking versus reaching measures,” *J. Exp. Child Psychol.*, vol. 109, pp. 201–17, Jun. 2011.

[16] J. S. Horst and L. K. Samuelson, “Fast mapping but poor retention by 24-month-old infants,” *Infancy*, vol. 13, no. 2, pp. 128–157, 2008.

[17] D. Gentner et al., “Rapid learning in a children’s museum via analogical comparison,” *Cogn. Sci.*, vol. 40, pp. 224–240, Jan. 2016.

[18] J. B. Tenenbaum, “Bayesian modeling of human concept learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 59–68.

[19] D. W. Braithwaite and R. L. Goldstone, “Effects of variation and prior knowledge on abstract concept learning,” *Cogn. Instruct.*, vol. 33, no. 3, pp. 226–256, 2015. [Online]. Available: https://doi.org/10.1080/07370008.2015.1067215

[20] Y. Zhang and C. Ling, “A strategy to apply machine learning to small datasets in materials science,” *npj Comput. Mater.*, vol. 4, no. 1, 2018, Art. no. 25. [Online]. Available: https://doi.org/10.1038/s41524-018-0081-z

[21] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HkxLXnAcFQ

[22] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep face recognition: A survey,” in *Proc. 31st Conf. Graph. Patterns Images*, 2019, pp. 471–478.

[23] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.

[24] G. R. Koch, “Siamese neural networks for one-shot image recognition,” 2015. [Online]. Available: https://www.researchgate.net/publication/321994829_Siamese_Neural_Networks_for_One-shot_detection_of_Railway_Track_Switches

[25] D. S. Chaplot, H. Jiang, S. Gupta, and A. Gupta, “Semantic curiosity for active visual learning,” 2020, *arXiv:2006.09367*. [Online]. Available: https://arxiv.org/abs/2006.09367

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347

[27] J. Grill et al., “Bootstrap your own latent: A new approach to self-supervised learning,” 2020, *arXiv:2006.07733*. [Online]. Available: https://arxiv.org/abs/2006.07733

[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[29] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.

[30] X. Liu et al., “Self-supervised learning: Generative or contrastive,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[31] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” 2017, *arXiv:1705.05363*. [Online]. Available: http://arxiv.org/abs/1705.05363

[32] A. Juliani et al., “Obstacle tower: A generalization challenge in vision, control, and planning,” Feb. 2019. [Online]. Available: http://arxiv.org/abs/1902.01378

[33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017, *arXiv:1412.6980*.

[34] V. Clay, P. König, K.-U. Kühnberger, and G. Pipa, “Learning sparse and meaningful representations through embodiment,” *Neural Netw.*, vol. 134, pp. 23–41, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608020303890

[35] J. Leugering and G. Pipa, “A unifying framework of synaptic and intrinsic plasticity in neural populations,” *Neural Comput.*, vol. 30, no. 4, pp. 945–986, Apr. 2018. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29342400/

[36] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[37] R. Fong and A. Vedaldi, "Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," 2018, *arXiv:1801.03454*. [Online]. Available: http://arxiv.org/abs/1801.03454

[38] J. C. Magee and C. Grienberger, "Synaptic plasticity forms and functions," *Annu. Rev. Neurosci.*, vol. 43, pp. 95–117, 2020.

[39] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends Cogn. Sci.*, vol. 23, pp. 408–422, May 2019.

[40] A. Szot et al., "Habitat 2.0: Training home assistants to rearrange their habitat," 2021, *arXiv:2106.14405*.

[41] C. Li et al., "iGibson 2.0: Object-centric simulation for robot learning of everyday household tasks," 2021, *arXiv:2108.03272*. [Online]. Available: https://arxiv.org/abs/2108.03272

[42] K. M. Hermann et al., "Grounded language learning in a simulated 3D world," 2017, *arXiv:1706.06551*.

[43] F. Hill, S. Clark, K. M. Hermann, and P. Blunsom, "Understanding early word learning in situated artificial agents," 2019, *arXiv:1710.09867*.

[44] F. Hill, O. Tieleman, T. von Glehn, N. Wong, H. Merzic, and S. Clark, "Grounded language learning fast and slow," 2020, *arXiv:2009.01719*.

[45] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2016, *arXiv:1511.07289*.

[46] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*. [Online]. Available: http://arxiv.org/abs/1710.05941

[47] A. Juliani et al., "Unity: A general platform for intelligent agents," 2018, *arXiv:1809.02627*. [Online]. Available: http://arxiv.org/abs/1809.02627

[48] K. Ota, D. K. Jha, and A. Kanezaki, "Training larger networks for deep reinforcement learning," 2021, *arXiv:2102.07920*. [Online]. Available: https://arxiv.org/abs/2102.07920

[49] V. Clay, "Data from neural network training in the obstacle tower environment to investigate embodied, weakly supervised learning," vol. 2, 2020. [Online]. Available: https://data.mendeley.com/datasets/zdh4d5ws2z/1

Viviane Clay received the BSc and MSc degrees in cognitive science from University Osnabrück, in 2017 and 2018 respectively, and the PhD degree in computational cognition from the University Osnabrück, in 2022. She received awards for both her bachelor's and master's thesis and was awarded the science award of lower-saxony, in 2018. She also received the AI talent award of lower-saxony, in 2019 and the Mendeley FAIRest dataset award for data from her most recent publication in Neural Networks. Her first scientific publication has reached more than 270 citations since its publication, in 2019. Her research interests are in biologically inspired machine learning, reinforcement learning, and embodied AI.

Gordon Pipa received the physics and electrical engineering degree from the RWTH Aachen, and the PhD degree in computer science, in 2006. After that he held a Group leader position in Prof. Wolf Singer Department of Neurophysiology, Max Planck Institute for Brain Research, Frankfurt am Main, Germany and a junior fellow position with the Frankfurt Institute for Advanced Studies (FIAS) until 2010 after which he was habilitated in Biology with the TU Darmstadt. Between 2007 and 2009 he was a research fellow with Prof. Emery Brown with a joint appointment with the Department of Brain and Cognitive Sciences, MIT, Cambridge, and the Department of Anesthesia and Critical Care with Massachusetts General Hospital, Boston. He became full professor (W3) and chair of the Neuroinformatics Department, Institute of Cognitive Science, University of Osnabrueck in Germany with the age of 36. His research focus is on neuroinspired artificial intelligence, and neuromorphic computing. He also works on developing AI systems that support the human in complex situations and the application of AI and machine learning to complex data structures, such as time series.

Kai-Uwe Kühnberger received the master's degree in linguistics, philosophy, and German literature from the University of Tübingen, in 1996, and the PhD degree in computational linguistics from the University of Tübingen, in 2002. He was a visiting scholar with Indiana University, Bloomington from 1997 to 1999. From 1999 to 2001 he was a research associate with the University of Tübingen (computational linguistics), from 2001 to 2003 research associate with Osnabrück University (artificial intelligence), and from 2003 to 2009 junior professor for artificial intelligence with Osnabrück University. Since 2009 he is University Professor for Artifical Intelligence with Osnabrück University. He got the PhD Award of the University of Tübingen, in 2002, he was a SICSA fellow (Scottish Informatics and Computer Science Alliance), in 2009, and was awarded an IBM Faculty Award for achievements in Cognitive Computing, in 2016. He published more than 130 papers with international conferences and in international journals. He was involved as PI in many research projects funded, for example, by the EU, the German Research Foundation, different German federal ministries, and foundations. He served as a survey editor for the journal *Cognitive Systems Research* (2008 to 2019) and was a member of the program committee of most major AI conferences. His major research interests are cognitively inspired AI technologies, computational creativity, hybrid systems, and AI applications.

Peter König received the degree in physics, in1985, and in medicine, in1987. He worked as a stipend and scientific assistant with Wolf Singer with the Max-Planck Institute for Brain Research (Frankfurt, D), as a senior fellow with the Neuroscience Institute (La Jolla, California) working with Gerald Edelman, and with the ETH/University (Zürich, CH) working with Kevan Martin and Rodney Douglas. In 2003 he accepted the chair of NeuroBioPsychology with the Institute of Cognitive Science (Osnabrück, D). He received an ERC Advanced Investigator Grant jointly with Andreas Engel and holds a guest professorship with the University clinics Hamburg Eppendorf. He is a member of the Hamburger Akademie der Wissenschaften, and of the Max-Planck School of Cognition. Further, he serves on the advisory board of the Bernstein Zentrum Berlin and the Vienna Doctoral School. He published more than 200 articles in peer-reviewed journals that were cited more than 26.000 times. He founded three spin-off companies (WhiteMatter Labs GmbH, feelSpace GmbH, and SciCovery GmbH). His research focuses on the investigation of embodied cognition with experimental and theoretical approaches.