To appear in *Advanced Robotics*
Vol. 36, No. 1-2, November 2021, 85–99

ARTICLE

Multi-modal shared module that enables the bottom-up formation of map representation and top-down map reading

Wataru Noguchi[a*], Hiroyuki Iizuka[a,b], and Masahito Yamamoto[a,b]

[a] *Faculty of Information Science and Technology, Hokkaido University, Japan*;

[b] *Center for Human Nature, Artificial Intelligence, and Neuroscience, Hokkaido University, Japan*

(*Received 30 April 2021; accepted 30 September 2021*)

Humans create internal models of an environment (i.e., cognitive maps) through subjective sensorimotor experiences and can also understand spatial locations by looking at an external map as a symbol of an environment. We simulate the development of the cognitive map from sensorimotor experiences and grounding of the external map in a single deep neural network model. Our proposed network has a shared module that processes the features of multiple modalities (i.e., vision, hearing, and touch) and even external maps in the same manner. The multiple modalities are encoded into feature vectors by modality-specific encoders, and the encoded features are processed by the same shared module. The proposed network was trained to predict the sensory inputs of a simulated mobile robot. After the predictive learning, the spatial representation was developed in the internal states of the shared module, and the same spatial representation was used for predicting multiple modalities, including the external map. The network can also perform spatial navigation by associating the external map with the cognitive map. This implies that the external maps are grounded in subjective sensorimotor experiences, being bridged through the developed internal spatial representation in the shared module.

Keywords: cognitive map; multimodal learning; predictive learning; deep neural networks; symbol grounding

1. Introduction

Generally, humans have knowledge regarding their location in the environment. Spatial understanding is considered to be achieved by having a map-like internal representation called a

*Corresponding author: w.noguchi@ist.hokudai.ac.jp

cognitive map in the brain [1]. In support of this idea, neuroscience studies have revealed the existence of neurons that fire in response to spatial locations [2, 3]. Because specific neurons fire in response to particular locations, we can obtain information regarding our location from these spatial neural activities by using associations between neural activities and physical locations. It is known that spatial neural activities are achieved by integrating low-level sensorimotor signals in a bottom-up manner [4]. This means that the cognitive map is a bottom-up internal spatial representation integrated with sensorimotor signals.

On the other hand, we are also able to detect our location by reading external maps, which are presented physically and printed on paper. Unlike a cognitive map, which is an internal representation in the brain, external maps exist physically and can be shared with anyone. As we interpret symbols to understand them, we need to interpret external maps to understand what is where and where we are. This is because real space is represented abstractly as an external map. In other words, to read a map, we need to associate the internal representations of the cognitive map with the symbolic external map. Once we establish the associations, we can plan the sensorimotor sequences to arrive at destinations designated on the external maps or even communicate with other people regarding locations by pointing to the external maps.

The problem of how the external maps are associated with the internal representation of cognitive maps can be considered as a symbol grounding problem [5], that is, whether a robot can understand the meaning of symbols that humans use. In this paper, we consider an external map as a symbol, and understanding a symbol, that is, reading external maps, implies the ability to understand the destination designated by others on the symbolic external map. This means that given an external map as a symbol, it can be transformed into a subjective experience by reading the external map. Reading external maps consequently enables navigation to the locations shown on the external maps from the current location.

An attempt to challenge the symbol grounding problem from a practical point of view, rather than a philosophical argument, has been made in robotics as a constructive approach. For example, there are approaches for building robots that can learn the existing symbol systems [6, 7]. Nakamura et al. proposed a multimodal LDA that performs unsupervised categorization on multimodal sensory inputs and organizes internal representations as object categories [8]. This has been extended to associate multimodal sensory inputs with words via an internal representation [9]. The association process can be considered as symbol grounding.

Regarding approaches to developing models that obtain spatial understanding, there are deep

neural network models that can obtain the cognitive map or learn to read maps. There are studies that model the development of internal spatial representations solely through the predictive learning of sensory inputs without a pre-given map [10, 11]. There is also an approach that teaches deep neural networks to read external maps in a top-down manner to navigate the environment [12]. These models can be classified as either bottom-up, which builds up the cognitive map without teaching signals, or top-down, which learns the association between the pre-given external map and the navigation behaviors with teaching signals. However, to the best of our knowledge, there is still no model that can develop cognitive maps in a bottom-up manner and ground external maps in sensorimotor experiences simultaneously.

In this paper, we propose a model that obtains a cognitive map through subjective experiences and grounding an external map as a symbol in subjective experiences by associating the external map with the cognitive map. We propose a shared module network model that can develop a common internal spatial representation, that is, the cognitive map, from different modalities, and even associate the cognitive map with external maps. The shared module in the proposed network is used in a shared manner over different modalities to calculate the predictions of future sensory inputs from the motions and past sensory inputs. Through the shared module, subjective sensorimotor inputs and external maps are associated.

The proposed network was implemented as a deep neural network following a previous study [10]. An overview of the simulations is shown in Figure 1. First, we show that out model can obtain and share a cognitive map among different modalities, namely, vision, hearing, and touch. The learning process of our model follows the predictive coding/processing [13], which has attracted attention as a theory that provides a unified explanation of brain functions. Our model was trained to predict the sensory inputs. The training results show that the internal states of our model are self-organized to have a two-dimensional structure and represent the spatial locations of the robot as a cognitive map. Then, learning is conducted to associate the cognitive map with an external map. Predictive learning is also conducted for the external map in the same manner as other modalities using the shared module, and the same internal spatial representation of the cognitive map is used for the external map. During the predictive learning on the external map, visual sensations and abstract motions on the external map are processed in the same manner as the actual sensations and motions by using the same shared module. Finally, we show that the external map is grounded in subjective experiences by performing a navigation experiment. Because the actual and abstract motions are associated with the shared module,

the path planning on the external map is naturally translated to actual navigation behaviors. The results show that the external map, the symbol of physical space, is grounded in subjective sensorimotor experiences.

2. Related works

2.1 *Simultaneous localization and mapping*

In the field of computer vision and robotics, research into the self-localization method in the environment is conducted in a SLAM framework [14], where an agent simultaneously estimates the self-location and environmental structure. The map of the environment is constructed from sensory sequences, such as RGB images, depth sensors, and odometry. In the SLAM framework, knowledge of the spatial coordinates of the environment (i.e., 2D or 3D coordinates of the space) is manually incorporated into the system as the spatial coordinate is known to the designer. The current location and environmental map are estimated based on known geometric structures. Unlike such a general SLAM system, we intentionally remove the assumption of the spatial structure from the proposed model to investigate how spatial representation is obtained in the model.

2.2 *Spatial recognition by deep neural networks*

Unlike traditional SLAM methods, deep neural networks are often trained to navigate environments without any predefined spatial coordinate systems, and can obtain the internal model of the environment naturally while achieving given tasks such as spatial navigation [15, 16]. There are two types of deep neural network models, depending on whether or not a specialized and arbitrarily designed architecture is used. The model proposed in [17] has a two-dimensional network topology of memory neurons to represent a two-dimensional spatial map. The spatial layout of the environment was obtained in the memory neurons through learning for path planning. In contrast, there are models that obtain spatial representation without any specialized architectures to have two-dimensional representation [10, 11]. Spatial representations are obtained through predictive learning of visual inputs, which change corresponding to spatial locations. By simply receiving visual inputs and predicting future vision, the internal states in the deep neural network are maintained to correspond to their location in the physical space. Their stud-

ies show that the number of effective dimensions in the internal states matches the number of dimensions in the space of the physical environment. Thus, integrating the localized information of sensory inputs in a given environment can form a spatial representation of the entire environment. These studies consider only the bottom-up formation of the spatial representation.

Meanwhile, there is also a model that learns to read an external map that is given in a top-down manner [12]. The deep neural network learns to navigate complex mazes through reinforcement learning. The network learns to estimate a temporal local map from the subjective vision, while the ground truth map is given as a supervision signal. Subsequently, the probability of an agent's location over the global map is calculated based on the similarity between the estimated local map and regions of the global map. The network always depends on the given global map for spatial localization. Thus, the bottom-up formation of the spatial representation is not required.

2.3 *Multimodal learning by deep neural networks*

Deep neural network models for learning multimodal data have been studied [18, 19]. For example, deep neural networks can retrieve or generate visual images from text captions in a cross-modal manner and obtain better classification performance by integrating multimodal information than when using a single modality. Multimodal association and integration abilities are realized by the high-level abstraction of deep neural networks. In other words, multimodal association and integration can be considered as the grouping ability of feature vectors at the deepest layers. While the values of input vectors, coming through different modalities, are generally irrelevant to each other, deep neural networks can discover the properties of correlations and co-occurrences over different modalities. As an approach to multimodal learning, deep neural networks are trained to minimize the differences between the feature vectors transformed from the input vectors of various modalities so that the networks can recognize the environment consistently, regardless of the type of input modalities [20, 21]. Other studies train the network to jointly encode inputs of different modalities into a single feature vector and reconstruct one or all input modalities from the feature vector [18, 19]. The feature vector includes the integrated features of multiple modalities.

Multimodal learning models are not limited to deep neural networks. For example, probabilistic generative models can be used to learn joint probability of multimodal sensory inputs including latent variables [8]; the latent variables can be considered as categories as a result of

multimodal integration and inference among modalities can be performed through the latent variables. Another study proposed a probabilistic generative model that performs multimodal learning integrated with SLAM to construct spatial concepts [22]. It is easier to introduce structural assumptions over latent variables to form meaningful internal representation by probabilistic generative models than deep neural networks while hand engineered data pre-processing is often necessary in probabilistic generative models. It should be noted that deep probabilistic generative models can be used as an integration of deep neural networks and probabilistic generative models [23] while the current study does not investigate these approaches.

Closely related to our study, Aytar and others proposed sharing modules among different modalities to obtain aligned representations between various modalities [20]. The networks are trained in a supervised manner to minimize the difference in the feature vectors extracted from the paired inputs of different modalities such as image-text and image-sound pairs. As a result, the shared module succeeded to obtain modality-agnostic representations. We also propose the use of shared module and modality-specific encoders to learn multimodal sensory inputs. However, in contrast to [20], our proposed network is just trained in a self-supervised manner to predict the future sensory inputs, which are spatially embedded and given according to the locations of the results of movements in the environment. There is no teaching signal to form modality-agnostic representations as the previous study uses. In our experiments, we show that the multi-modal sensory inputs are naturally associated according to their spatial locations through predictive learning.

3. Developing internal spatial representation across multiple modalities

In the following text, we detail the experiments performed on our proposed network models. First, in Section 3, the proposed prediction network with a shared module is described. The network learns to predict a simulated mobile robot's sensory sequences of multiple modalities and how the internal spatial representation is developed is investigated. In Section 4, the network is trained to predict external map sequences, and the external map is associated with the internal spatial representation in the shared module. In Section 5, the network is trained to perform the robot's navigation by looking at the external map wherein the navigation goal is presented. The network learns to generate motion sequences to navigate the robot to a goal, based on internal spatial representation. It is demonstrated that navigation is possible because the internal spatial

representation is associated with both subjective sensorimotor experiences and external maps.

3.1 *Prediction network with a shared module*

We propose a deep neural network model that develops an internal spatial representation through predictive learning of the robot's sensory inputs. It was shown that RNN can obtain internal spatial representation, namely, the cognitive map, in its internal states through predictive learning [10, 24]. Our proposed model is basically similar to the model in [10]. A distinctive feature of our current model in this paper is that it has a shared module for processing features of multiple modalities in the same manner. By having a shared module, the network can learn to share internal representations across multiple modalities [20].

The network receives sensory inputs $\boldsymbol{x}_t^s$ and outputs prediction $\hat{\boldsymbol{x}}_{t+1}^s$, where s denotes the modalities of sensory inputs ($s \in \{vision, hearing, touch\}$). The network consists of an encoder and decoder for sensory inputs, and the LSTM [25] as a shared module (Figure 1 (a)). Different encoders and decoders are used for different modalities; meanwhile, LSTM is shared across modalities. This implies that the prediction is independently performed for each modality, while there is a constraint that the prediction is always performed through the shared LSTM module for all modalities. Sharing LSTM indicates that the feature vectors, which have the same fixed number of dimensions, encoded from different modalities, are processed in the same manner as in the shared LSTM. Specifically, the same receptor neurons and neural weights are used to process the features of multiple modalities.

At each step of the prediction process for modality s, the encoder Enc^s receives the sensory input $\boldsymbol{x}_t^s$ and the encoded sensory feature $\boldsymbol{f}_t^s$ becomes an input to the shared LSTM along with motion $\boldsymbol{m}_t$, and the prediction of future sensory input $\hat{\boldsymbol{x}}_{t+1}^s$ is generated by the decoder Dec^s as follows:

$$\boldsymbol{f}_t^s = \mathrm{Enc}^s(\boldsymbol{x}_t^s), \tag{1}$$

$$(\boldsymbol{h}_t^s, \boldsymbol{c}_t^s) = \mathrm{LSTM}(\mathrm{Mask}(\boldsymbol{f}_t^s; prob^{mask}), \boldsymbol{m}_t, \boldsymbol{h}_{t-1}^s, \boldsymbol{c}_{t-1}^s), \tag{2}$$

$$\hat{\boldsymbol{x}}_{t+1}^s = \mathrm{Dec}^s(\boldsymbol{h}_t^s), \tag{3}$$

where $\boldsymbol{h}_t^s$ and $\boldsymbol{c}_t^s$ are the hidden and cell states of the LSTM and Mask is a mask operation. We refer to hidden states $\boldsymbol{h}_t^s$ as just internal states, and the internal states are analyzed in the

experiments. The mask operation sets all elements of the encoded feature vector to zero with probability $prob^{mask}$. If the mask operation is applied, the sensory features $\boldsymbol{f}_t^s$ are not provided to LSTM. The masking probability $prob^{mask}$ is a hyperparameter, and we set it to a high value such as 0.99. Note that, at the first time step, the mask operation is not applied and the LSTM always receives $\boldsymbol{f}_t^s$ regardless of $prob^{mask}$. This masking operation makes the network perform sensory predictions using motion inputs without relying strongly on sensory inputs. It has been shown that such predictive learning, with sensory input masking, encourages the network to obtain an internal representation of the external space through sensory prediction [10, 24].

The network learns to minimize errors between the predicted and actual future sensory input. The prediction loss $\mathcal{L}_{pred}^s$ for modality s is computed as follows:

$$\mathcal{L}_{pred}^s = \sum_{t=1}^{T} \mathrm{MSE}(\boldsymbol{x}_{t+1}^s, \hat{\boldsymbol{x}}_{t+1}^s), \tag{4}$$

where $\mathrm{MSE}(\boldsymbol{x}_{t+1}^s, \hat{\boldsymbol{x}}_{t+1}^s)$ is the mean-squared error between $\boldsymbol{x}_{t+1}^s$ and $\hat{\boldsymbol{x}}_{t+1}^s$, and T is the length of the training sequence.

3.2 *Mobile robot simulation*

Our proposed model is tested in a simulated environment. Figure 2 (b) shows the simulated environment where the mobile robot moves around. The environment is a room surrounded by walls. The mobile robot is equipped with an omni-wheel, which allows it to move in any direction of 360 degrees on the floor of the room. The robot also has an omnidirectional camera and captures the visual images that cover all directions in one sight (Figure 2 (d)). The robot does not change its direction, and consequently, it captures the same visual images if it is at the same location. The colored corners of the walls work as visual landmarks. Further, there are also landmarks for hearing and touch (Figure 2 (c)). The audio sources, for the sense of hearing, were placed on each corner of the room; in total, there were four audio sources. Each audio source makes a sound with a constant frequency, and the robot can distinguish between sounds of different frequencies. The sound pressures decrease depending on the distances from these audio sources (Figure 2 (e)). As landmarks for sense of touch, there are bumps on the floor. There were a total of 16 bumps; four bumps were placed near each corner of the room. Each bump elicits different touch sensory inputs for the robot. The intensity of the touch sensory input

depends on the distance to the bump, as in the case of audio sources (Figure 2 (f)). However, unlike the case of audio, the size of each bump is small and cannot be sensed if the robot moves away from the bumps.

The robot motor commands were the two-dimensional displacement of the robot at each step. Two-dimensional motor commands were used as motion inputs $\boldsymbol{m}_t$ for the network. The visual images are RGB images with size of 16×64. The dimensionality of hearing and touch sensory inputs are four and 16, respectively, corresponding to the number of frequencies of the audio sources and bumps.

3.3 *Experiments*

The sensorimotor sequences for training the network were collected using the behavioral data of the robot that moved according to a predefined rule. The robot was initially placed randomly at a certain location and moved straight toward a randomly selected destination. The robot moved one unit at each step. When the robot reached its destination, a new destination appeared randomly. The robot kept going to the destinations until a sequence end. The sensorimotor data for vision, audio, tactile, and action were stored while the robot moved around. The 1,000 sensorimotor sequences comprising consecutive 100 steps were collected and used for network training. When collecting data on touch inputs, the robot is placed in one of the bump regions at the initial step so that the robot can know where it is at the beginning. After the training, the network's internal representations are analyzed. For the analysis, we collected 100 sensorimotor sequences independent of the training sequences.

The shared LSTM module has 128 hidden units. The structure of encoders and decoders are visualized in Figure 3. Enc^{vision} consists of three convolution layers of 16, 32, and 64 output channels with a kernel size of 3×3 and a stride size of 1, and two fully connected layers with a size of 64. Each convolution is followed by max pooling for halving the spatial dimension of the convolutional layer's output. For each convolution, zero padding is performed for vertical padding, and loop padding is performed for horizontal padding. Loop padding indicates that the outside of the edge on the visual image is padded by the values of the other edges. Loop padding is used for processing omnidirectional camera images, which have a loop structure in the horizontal dimension. Dec^{vision} comprises three fully connected and four convolution layers. The size of the first and second fully connected layers is 64, and that of the last one is selected such

that the size of the generated vision matches the input vision. The convolution layers have 32, 16, 16, and 3 output channels, respectively, and their kernel sizes are 3×3 and stride sizes are 1. Outputs of each of the first three convolutions are up-sampled and these spatial dimensions are doubled. Padding is performed in the same manner as Enc^{vision}. The channels of the last convolution corresponds to RGB channels. The encoders and decoders for hearing and touch (i.e., $\mathrm{Enc}^{hearing}$, $\mathrm{Dec}^{hearing}$, Enc^{touch}, and Dec^{touch}) all consist of three fully connected layers that have a size of 64, except for the last layer of the decoders. The sizes of the last layers of $\mathrm{Dec}^{hearing}$ and Dec^{touch} match the size of the inputs $\boldsymbol{x}_t^{hearing}$ and $\boldsymbol{x}_t^{touch}$, respectively. Each convolution or fully connected layer is followed by batch normalization. ReLU activation function is used for each layer, except for the last layers of decoders where logistic sigmoid activation function is used. Note that these network configurations such as number of layers and number of nodes are not intensively tuned, and the current configuration may not be minimal one. For example, the encoders and decoders for hearing and touch could have fewer layers. However, the requirements for the network is to prepare sufficient capacity to learn because the overfitting does not matter in the current simulation.

The network learns 100 times for visual prediction and 300 times for hearing and touch prediction over the training sequences. The parameters of the network are updated using the Adam optimizer [26] with a learning rate of 0.001 and a batch size of 10 for training. The masking probability $prob^{mask}$ is 0.99. The input masking is applied not only during training but also during analysis.

3.3.1 Learning results

First, the network was trained on visual predictive learning. The internal representation of the network, acquired as a result of the learning, is visualized in a two-dimensional space using independent component analysis (ICA) (Figure 4). Each point of the visualized internal states is colored according to the spatial location of robot. For coloring, RGB values are assigned to each location. Red, blue, green, and yellow correspond to the four corners of the room, and linearly interpolated colors are assigned to other locations (Figure 4 (a)). The internal states $\boldsymbol{h}_t$ in the shared module are colored in the ICA space by the color corresponding to the robot's location. Figure 4 shows that the internal states $\boldsymbol{h}_t^{vision}$ are arranged in correspondence with the spatial location of the robot as in previous studies [10]. Especially, two independent components found by ICA correspond to x and y coordinates of the robots. In other words, the network maintains

the internal states in the memory where the robot is. The arrangements of the internal states extend in two dimensions in the same manner as in the external world. The layout of the internal states maintains the structure of the physical space that corresponds to its distance. This self-organization of internal spatial representation indicates that the network can recognize the spatial structure and layout of the environment. The visual prediction was achieved using this spatial representation. As the predictive learning was conducted under the condition of visual input masking, the result also implies that the network recognizes how the robot's location changes according to its motion.

After visual predictive learning, the network was trained on hearing and touch predictive learning using the trained shared LSTM module. In this case, only the encoders and decoders were trained, while the parameters of the shared LSTM were fixed. After training, the internal states of the trained network were visualized in the same manner as the vision. The same ICA space is used to map the internal states. The internal states for hearing and touch prediction $\boldsymbol{h}_t^{hearing}$ and $\boldsymbol{h}_t^{touch}$ were mapped onto the same space for the visual prediction case (Figures 4 (c) and (d)). The figure also shows that the internal states for hearing and touch prediction were organized in the same manner as for visual prediction. This result implies that the internal spatial representation, developed during visual predictive learning, is also used for hearing and touch prediction. In other words, the encoders and decoders are trained to re-use the internal structure of the shared LSTM module where the spatial representation emerges. Because all of the sensory input sources, including visual landmarks, audio sources, and floor bumps, are localized in the same environment, the network is required to associate them with the internal spatial representation to predict hearing or touch inputs. However, it is not trivial to use the same representation for hearing and touch prediction because the encoders and decoders can make arbitrary associations between sensory inputs and internal states. Our results show that the developed spatial representation can also be reused for hearing and touch prediction by sharing LSTM without any additional teaching signals[1].

[1]Please note that the predictive learning of vision, hearing, and touch can be done simultaneously not in an ordered way as in the current results. Because the prediction can be successful for all of three modalities by using the spatial representation as obtained in the current simulation, the spatial representation will emerge even when the predictive learning for three modalities is conducted simultaneously.

4. Learning to predict external map sequences based on the developed internal spatial representation

In the above experiments, the network was trained to predict the sensory inputs that the robot senses when the robot moves around in the room, and the cognitive map is developed in the internal states of the shared module. Subsequently, we trained the network on an external map, where the environment was drawn. The network used in this section has the trained shared LSTM module which obtained the internal cognitive map in the previous experiment. Then, this predictive learning on the external map makes an association between the internal cognitive map and the external map.

The training is conducted in the form of prediction on the sequences of the external map, which are the simulated walk on the visually presented external map as described below, similar to other modalities. However, the external map is completely different from other modalities of vision, hearing, and touch in the sense that the external map is not a subjective sensory input of the robot itself. The robot needs to associate the external map with the internal spatial representation to use the external map similar to humans.

4.1 *External maps and virtual walk*

The experimental setup here assumes that the robot is looking at the external map as human do. While looking at the external map, we take a virtual walk on the external map, tracing the external map with a finger representing our location. For example, we can know on the external map, if we move this way, there is a school here, and if we move that way, there is a restaurant. The virtual walk is independent of the current real location. To simulate the virtual walk, the location of the virtual walk is represented by the gray square on the external map, and the top view becomes inputs as external map vision (Figure 5 (a)). The gray square moves around freely regardless of the real current location. We call these sequence of top-view external maps for the virtual walk external map sequence (Figure 5 (b)). The size of each external map is 64×64, and it has RGB channels for color. The external maps as inputs are denoted by $\boldsymbol{x}^{map}$ ($s = map$).

4.2 *Prediction network for external maps*

The network was built to be able to predict the external map sequences for the virtual walk. The prediction of the external map sequences is called external map prediction here. The shared LSTM that was trained for visual prediction was also used, and the encoder and decoder were constructed for the external map (Enc^{map} and Dec^{map}) (Figure 5 (c)). Because the virtual walk on the external map is independent of the robot's physical motion, no physical motion is input to the shared module. To compensate for the absence of motor inputs, an additional module is used to internally generate motor inputs. The additional module is called the motion generator Gen^m. The motion generator comprises an LSTM and a fully connected layer. At the current time step t, the motion generator receives the feature of future external map input $\boldsymbol{f}_{t+1}^{map}$, and outputs the vectors $\hat{\boldsymbol{m}}_t$ with the same dimensions as the robot's motion, $\hat{\boldsymbol{m}}_t = \mathrm{Gen}^m(\boldsymbol{f}_{t+1}^{map})$. It means that, to generates $\hat{\boldsymbol{m}}_t$, the motion generator can use the history of the external map inputs from time step 1 to $t+1$ stored in LSTM memory[1]. The reason why the feature of future step $\boldsymbol{f}_{t+1}^{map}$ is input to Gen^m is that the generated motion $\hat{\boldsymbol{m}}_t$ should explain the change between current and future of external map inputs $\boldsymbol{x}_t^{map}$ and $\boldsymbol{x}_{t+1}^{map}$. The encoded features $\boldsymbol{f}_t^{map}$ as inputs for the shared LSTM were masked as previously described, while that for the motion generator was not masked. Thus, the motion generator network learns to generate virtual motion on the external map to ensure that the shared LSTM module can update its internal state to change the prediction. Note that the dimension of the generated motion is smaller than that of the external map inputs, and directly passing the information of the prediction target $\boldsymbol{x}_{t+1}^{map}$ to the shared LSTM through the generated motion is not possible.

4.2.1 Additional learning loss for correspondence

To use the external map to recognize the robot's location, correspondence between the locations represented by the external map and the internal spatial representation developed within the network should be maintained. As the external map sequence of the virtual walk is separated from the robot's subjective sensory experiences, there is no guarantee that the same internal state represents the same spatial locations for both the external map and vision prediction. Additional learning is required to achieve correspondence between the external map and vision prediction for the representations of spatial locations. We assume that the correspondences

[1]To use the input of the first time step ($t = 1$), the motion generator receives the input twice, $\boldsymbol{f}_1^{map}$ and then $\boldsymbol{f}_2^{map}$ at $t = 1$.

are given as the supervised signal. The correspondences are given as pairs of vision and external map $(\boldsymbol{x}_p^{vision}, \boldsymbol{x}_p^{map})$ for the robot location $p \in P$, where P is a set of spatial locations. $\boldsymbol{x}_p^{vision}$ is the vision of the robot taken at p, and $\boldsymbol{x}_p^{map}$ is the external map where the gray square is drawn at the location corresponding to p. We collect pairs of vision at four locations $P = \{(5,5), (5,-5), (-5,5), (-5,-5)\}$ (Figure 6).

Given the pairs $(\boldsymbol{x}_p^{vision}, \boldsymbol{x}_p^{map})$, the correspondence loss $\mathcal{L}_{corr}$ for learning correspondences of spatial locations is defined as follows:

$$\mathcal{L}_{corr} = \mathrm{MSE}(\boldsymbol{h}_p^{vision}, \boldsymbol{h}_p^{map}), \tag{5}$$

where $\boldsymbol{h}_p^s = \mathrm{LSTM}(\mathrm{Enc}^s(\boldsymbol{x}_p^s), \mathbf{0})$ for each $s \in \{vision, map\}$. To obtain $\boldsymbol{h}_p^s$, the masking operation Mask is not used, and the motion input $\boldsymbol{m}$ is replaced by a zero vector $\mathbf{0}$. The network is trained to minimize the prediction loss $\mathcal{L}_{pred}^{map}$ and the correspondence loss $\mathcal{L}_{corr}$. It should be noted that the loss for directly supervising motions is not used.

4.3 *Experiments*

Predictive learning on the external map was performed using the trained shared LSTM in the previous experiment. The gray square moved around on the external map according to the same rule as in the previous experiment. The 1,000 sequences, consisting of consecutive external maps for 100 steps, were collected. We also collected 100 sequences that were independent from the training sequences for analysis after training.

The structure of the motion generator and the map encoders and decoders are visualized in Figure 7. The motion generator Gen^m consists of an LSTM with 128 units and a fully connected layer of the same size as $\boldsymbol{m}_t$, and tanh activation is used for the generated motion. Enc^{map} and Dec^{map} are convolutional neural networks, similar to Enc^{vision} and Dec^{vision}. Enc^{map} has four convolution layers of 16, 32, 64, and 64 channels, and Dec^{map} has five convolution layers of 64, 32, 16, 16, and 3 channels. The other settings for Enc^{map} and Dec^{map} are the same as Enc^{vision} and Dec^{vision}, except that loop padding is not used.

The network learns 100 times over the training sequences for external map prediction. Only the parameters of Enc^{map}, Dec^{map}, and Gen^m are trained. The Adam optimizer was used with a learning rate of 0.001 and a batch size of 10, as previously described. In this case, the masking

probability $prob^{mask}$ is 0.8[1].

4.3.1 Learning results

The internal states $\boldsymbol{h}_t^{map}$ of the trained network are visualized by ICA, as previously described (Figure 8). The internal states were mapped to the same space, as in Figure 4. The two-dimensional structure can be observed in the mapped space (Figure 8 (a)), and the correspondences of the spatial location between vision and external map were obtained despite the fact that only four correspondences were given. The arrangements of the internal states for vision (Figure 4 (b)) and for external map (Figure 8 (a)) are almost the same in the same ICA space although there is some gaps in area covered by the internal states. This indicates that the network represents the external map using the internal spatial representation, while the same states are used to represent the corresponding locations between vision prediction and external map prediction. In other words, an external map is associated with the cognitive map, which is developed through subjective experiences in a bottom-up manner. Consequently, locations shown in the external map are also associated with physical locations through the internal cognitive map. We considered that the network obtained the map reading ability, namely, the ability to recognize the physical location indicated in the external map.

The network was also trained without using the correspondence loss (Figure 8 (b)). Although the arrangement of the internal states for the external map prediction does not correspond to that of the vision prediction, a two-dimensional structure can be observed. The trained network used the two-dimensional structure of the internal spatial representation to predict the external map without supervision of correspondence between vision and external map. This implies that sharing the LSTM enables the network to find the spatial structure even from such inputs disconnected from the robot's own experiences.

We also trained the prediction network without the trained LSTM to verify the contribution of sharing LSTM. In this case, the parameters of the LSTM are initialized randomly and optimized. Consequently, the learned internal state becomes distorted; in other words, it does not have a two-dimensional structure, as shown in Figure 9. This means that it is difficult to find the spatial structure underlying the external map without sharing the LSTM, i.e., without transferring the spatial structure obtained in the vision prediction.

[1] We empirically found that the learning is not successful with extremely high masking probability such as $prob^{mask} = 0.99$

5. Navigating by reading map

In this section, we show that the proposed network which obtained the map reading ability can be extended to perform spatial navigation to reach locations indicated in the external map. The experiment will show that the external map is grounded in subjective sensorimotor experiences by demonstrating that the network can read the external map to know locations and perform navigation in physical space.

The navigation goal is presented by the external map $\boldsymbol{x}^{map}_{p^{goal}}$, where the gray square is at the goal location p^{goal}. The network learns to generate the motion sequence of the robot to reach the goal. We performed navigation learning through internal simulations inside the networks based on the internal spatial representation. However, we will show that the network could navigate the robot to the goal in physical space, although navigation learning is conducted by the network's internal simulation.

5.1 *Navigation networks*

The network for navigation is shown in Figure 10. First, the network receives the external map input where the gray square is at the goal $\boldsymbol{x}^{map}_{p^{goal}}$ and generates internal states $\boldsymbol{h}^{map}_{p^{goal}}$, which is used as the goal state of the internal simulation of navigation. The network then receives the vision $\boldsymbol{x}^{vision}_{p^{start}}$ when the robot is placed at the start location p^{start} and generates the internal states $\boldsymbol{h}^{vision}_{p^{start}}$, which is used as the initial state of the internal simulation of navigation; $\boldsymbol{h}_{nav,t} = \boldsymbol{h}^{vision}_{p^{start}}$ at time 0. The target of navigation learning is to update the current internal states $\boldsymbol{h}_{nav,t}$ to make it closer to the goal state $\boldsymbol{h}^{map}_{goal}$. To update the internal states, an additional navigation module Gen^{nav} is introduced. The navigation module receives the difference between the goal state and current state $\boldsymbol{h}^{map}_{p^{goal}} - \boldsymbol{h}_{nav,t}$ to know how far the current state is from the goal state. The navigational motion $\boldsymbol{m}^{nav}_t$ is generated as follows:

$$\boldsymbol{m}^{nav}_t = \text{Gen}^{nav}(\boldsymbol{h}^{map}_{p^{goal}} - \boldsymbol{h}_{nav,t}). \tag{6}$$

During navigation training, only the parameters of the navigation module were optimized, and the other parameters were fixed. The navigation module is trained to minimize the following

navigation loss $\mathcal{L}_{nav}$:

$$\mathcal{L}_{nav} = \sum_{t=1}^{T_{nav}} \mathrm{MSE}(\boldsymbol{h}^{map}_{p^{goal}}, \boldsymbol{h}_{nav,t}), \tag{7}$$

where T_{nav} is the duration of single trial of the navigation.

5.2 *Experiment*

We collected 1,000 pairs of vision and external maps $(\boldsymbol{x}^{vision}_{p^{start}}, \boldsymbol{x}^{map}_{p^{goal}})$ for navigation learning, and 100 pairs for evaluation. $\boldsymbol{x}^{vision}_{p^{start}}$ and $\boldsymbol{x}^{map}_{p^{goal}}$ were randomly sampled independently, and the distances from $\boldsymbol{x}^{vision}_{p^{start}}$ to $\boldsymbol{x}^{map}_{p^{goal}}$ are not fixed. The navigation module is a single linear fully connected layer of the same size as $\boldsymbol{m}_t$, and tanh is used as the activation function. The navigation module is trained 100 times over the training pairs by Adam optimizer. Learning rate is 0.001 and batch size is 10. T_{nav} is set to 30.

5.2.1 Learning results

Figure 11 (a) shows the example trajectories of the robot navigating with the motion generated by the network after learning. It can be confirmed that the robot is approaching the goal. Figure 11 (b) also shows the transition of the internal states during navigation, demonstrating that the current internal state also approaches the target internal state in the internal state space. The navigation performance was quantitatively evaluated on the evaluation data. Errors between the goals and the robot's locations at final steps after T_{nav} steps were calculated; the average error was 2.2 units and the standard deviation was 0.9 units, and the error is as small as about 1/10 of the length of one side of the region. Note that the size of the room where navigation was performed was 20×20 units. It is considered that the errors of this navigation is mainly caused by the gap between the internal state arrangement between for vision and for external map. The navigation results show that the gaps in the internal states for the external map and vision are sufficiently small to perform navigation by comparing the internal states. In other words, the map reading ability, achieved by the network, was sufficient for performing navigation. The results show that the network can use the external map to understand the target location and navigate there, and in that sense, the external map is grounded in the robot's subjective experiences.

6. Discussion

The internal spatial representations, i.e., the cognitive map, were developed through the predictive learning of bodily sensations and motions. The sensory inputs of different modalities are processed in the single shared module. The bodily motions were given as fragmentary motor commands at each time step, such as moving to the north or east, and the sensory inputs were snapshot images for vision, sound pressure for hearing, and tactile sensations for touch at each time. Although the sensory input modalities were different, they were consistent in that they were sensations that were generated under the restraint of a moving body in an environment. Consequently, the shared module successfully processed and unified different types of information into a spatial representation. We then forced this shared module to predict events on an external map, as the abstract space, which is detached from the physical space, in the same manner as the bodily sensations and motions. In other words, this forces the network system to superpose the events on the physical and abstract spaces by providing the constraint of sharing the main module. Because the spatial representations of the two spaces are represented as a single internal representation, the physical space can be transformed into an abstract space and vice versa. In other words, the current location in the physical space can be understood as the location on an external map, and the location indicated on the external map can be reached using first-person sensory input and motion. We have shown this in our simulations. The development of common representations produced by shared modules allows abstract systems to be grounded in physical bodily systems.

The symbol grounding problem is the question of how an abstract system can be connected to a physical system if it exists apart from the physical system. Certainly, there is a way to complete the explanation with materialism without mentioning the abstract system. However, considering that we use language and logic, it seems more beneficial to explain how the abstract system can be grounded in the physical system. For this problem, we present a method of grounding. Physical and abstract systems share modules to be processed to form a common representation. It has also been shown from the perspective of machine learning engineering that sharing modules facilitates representation learning and improves classification accuracy [20]. We predict that similar shared modules will exist in the brain due to the following advantages: smooth connection between physical and abstract systems, engineering learning efficiency, and reduction of computational resources in the brain through reuse of modules.

Finally, we describe the limitations of the proposed model. One limitation is that, in the current simulations, the model learns a simple environment and an external map where no obstacles or walls exist, and it can be much more difficult than the current settings. A simulation study showed that, even in such complex environments, the cognitive map-like internal spatial representation could be obtained through predictive learning similar to ours [24]; however, grounding of external maps was not considered. To achieve grounding of an external map with walls or obstacles, it is necessary to understand that the drawings of walls or obstacles on the external map indicate the physical objects in subjective experiences. One possible approach to this problem is to explore the environment while referring to the external map, while the current model learns subjective sensory inputs and external maps separately. Another limitation is that the processing paths for subjective vision and external maps in the current model were separated by their design. Humans view physical space and external maps as symbols of space through the same vision, and can recognize external maps as abstract symbols of the space even if they are seamlessly embedded in the physical space. To recognize and use external maps in the real world as humans, the ability to appropriately interpret presented visual sensations of the actual environment or external map as abstract symbols is required. This might be achieved by introducing a more adaptive architecture of neural networks, such as external memory or attention [27, 28].

7. Conclusion

In this paper, we propose a deep neural network model that can develop the internal spatial representation, that is, the cognitive map, from sensory inputs of multiple modalities, that is, vision, hearing, touch, and even external maps that are separated from the agent's own subjective experiences. By using the shared module where the spatial representation is obtained, the network naturally uses the same internal spatial representation to predict the input sequences of multiple modalities. The network can also navigate using an external map. It is considered that symbol grounding occurred in our proposed model, where the network obtained the cognitive map as the internal spatial representation of the environment in a bottom-up manner and the external map as the symbol was associated with the robot's subjective experiences through the obtained cognitive map. Our study shows that processing multiple kinds of inputs in a unified way, with a shared module, promotes symbol grounding, which is an essential aspect of cognition.

Funding

This work was supported by JSPS KAKENHI Grant Number JP20H04989.

References

[1] E. C. Tolman, "Cognitive maps in rats and men.," *Psychological review*, vol. 55, no. 4, p. 189, 1948.

[2] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat.," *Brain research*, 1971.

[3] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–806, 2005.

[4] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, "Path integration and the neural basis of the'cognitive map'," *Nature Reviews Neuroscience*, vol. 7, no. 8, pp. 663–678, 2006.

[5] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[6] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, and H. Asoh, "Symbol emergence in robotics: a survey," *Advanced Robotics*, vol. 30, no. 11-12, pp. 706–728, 2016.

[7] T. Taniguchi, E. Ugur, M. Hoffmann, L. Jamone, T. Nagai, B. Rosman, T. Matsuka, N. Iwahashi, E. Oztop, J. Piater, *et al.*, "Symbol emergence in cognitive developmental systems: a survey," *IEEE transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 494–516, 2018.

[8] T. Nakamura, T. Nagai, and N. Iwahashi, "Multimodal object categorization by a robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2415–2420, IEEE, 2007.

[9] T. Nakamura, T. Nagai, and N. Iwahashi, "Grounding of word meanings in multimodal concepts using lda," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3943–3948, IEEE, 2009.

[10] W. Noguchi, H. Iizuka, and M. Yamamoto, "Cognitive map self-organization from subjective visuomotor experiences in a hierarchical recurrent neural network," *Adaptive Behavior*, vol. 25, no. 3, pp. 129–146, 2017.

[11] A. Laflaquière and M. Garcia Ortiz, "Unsupervised emergence of egocentric spatial structure from sensorimotor prediction," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[12] G. Brunner, O. Richter, Y. Wang, and R. Wattenhofer, "Teaching a machine to read maps with deep

reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[13] A. Clark, "Whatever next? predictive brains, situated agents, and the future of cognitive science," *Behavioral and Brain Sciences*, vol. 36, no. 3, p. 181–204, 2013.

[14] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[15] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, *et al.*, "Learning to navigate in complex environments," in *ICLR*, 2017.

[16] S. Chiappa, S. Racaniere, D. Wierstra, and S. Mohamed, "Recurrent environment simulators," in *ICLR*, 2017.

[17] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2017.

[18] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, pp. 689–696, 2011.

[19] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of robot behavior using deep neural networks," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 721–736, 2014.

[20] Y. Aytar, C. Vondrick, and A. Torralba, "See, hear, and read: Deep aligned representations," *arXiv preprint arXiv:1706.00932*, 2017.

[21] J.-B. Alayrac, A. Recasens, R. Schneider, R. Arandjelović, J. Ramapuram, J. De Fauw, L. Smaira, S. Dieleman, and A. Zisserman, "Self-supervised multimodal versatile networks," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 25–37, Curran Associates, Inc., 2020.

[22] A. Taniguchi, Y. Hagiwara, T. Taniguchi, and T. Inamura, "Online spatial concept and lexical acquisition with simultaneous localization and mapping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 811–818, IEEE, 2017.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[24] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin,

"Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[28] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

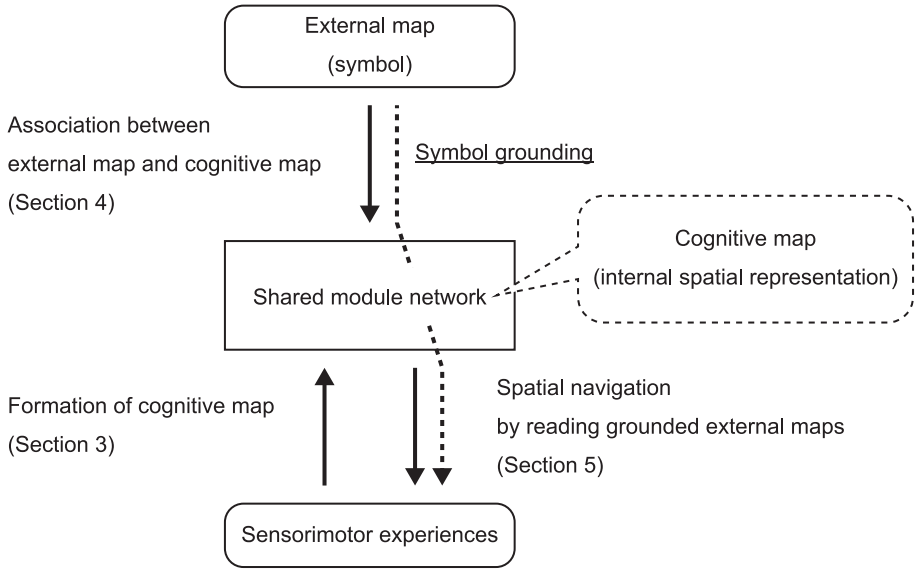

Figure 1. The concept of symbol grounding in our simulation.

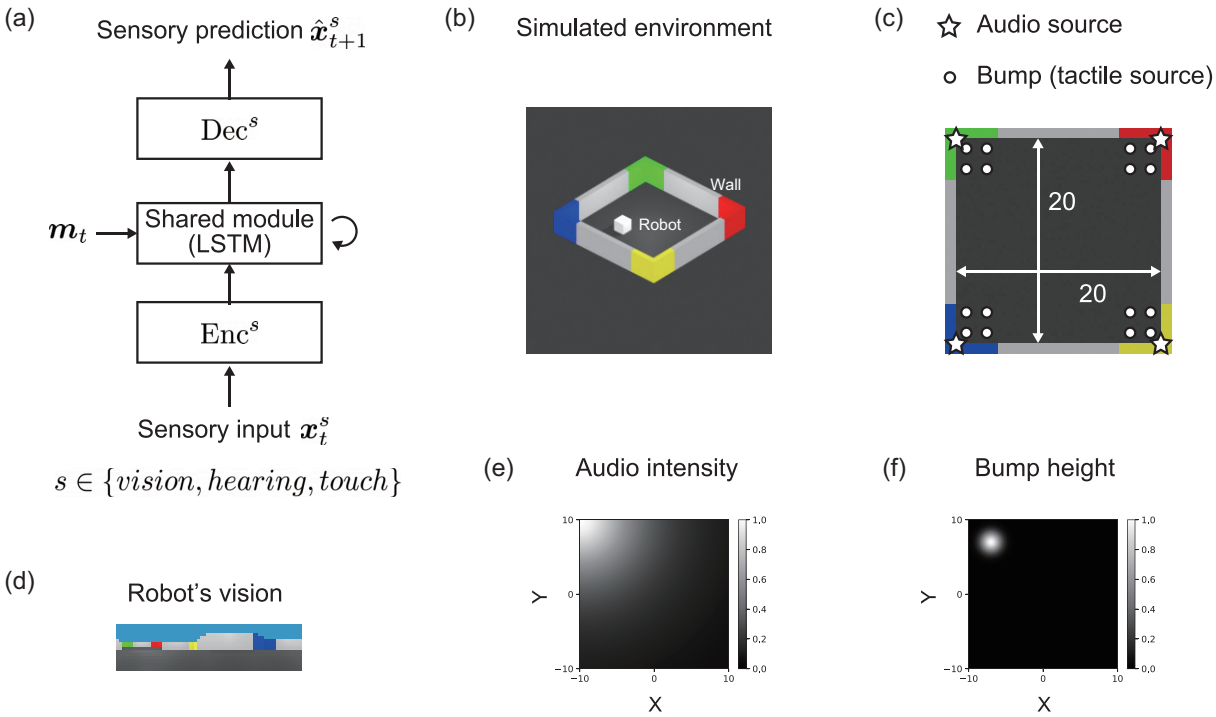

Figure 2. Simulated environment and prediction network: (a) The prediction network. (b) Simulated environment. (c) Arrangement of audio sources and floor bumps. The audio sources are located at (10, 10), (10, -10), (-10, 10), and (-10, 10). The floor bumps are located at (9, 9), (7, 9), (9, 7), (7, 7), (-9, 9), (-7, 9), (-9, 7), (-7, 7), (-9, -9), (-7, -9), (-9, -7), (-7, -7), (9, -9), (7, -9), (9, -7), and (7, -7). (d) Example of robot vision. (e),(f) Audio intensity from one of the audio sources and bump height of one of bumps. They are visualized as heatmap in two-dimensional space of the simulated environment.

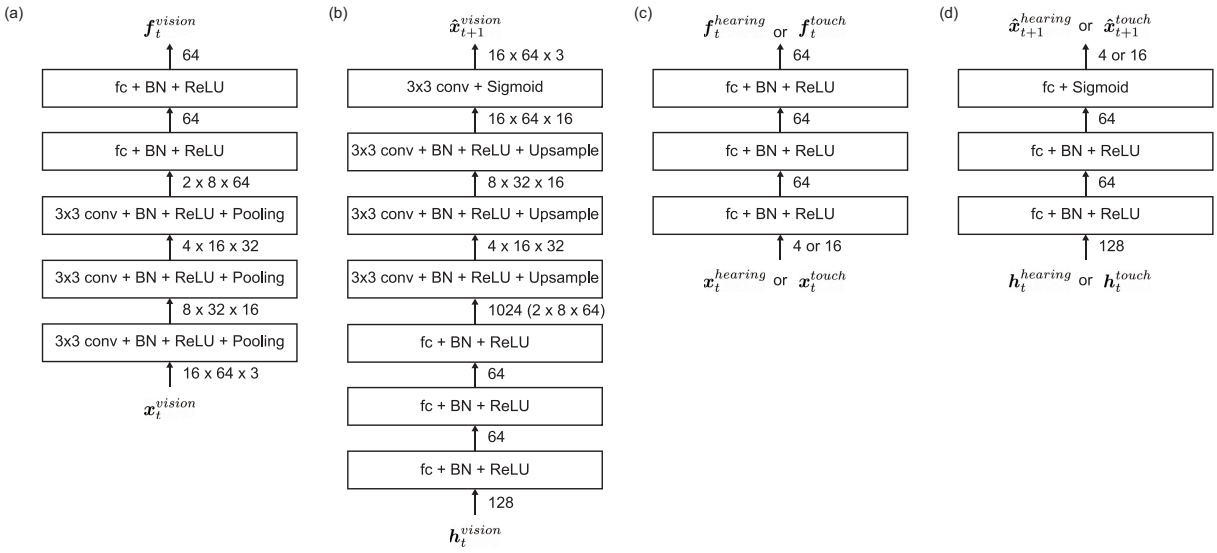

Figure 3. The detailed structures of (a) Enc^{vision}, (b) Dec^{vision}, (c) $\text{Enc}^{hearing}$ and Enc^{touch}, and (d) $\text{Dec}^{hearing}$ and Dec^{touch}.

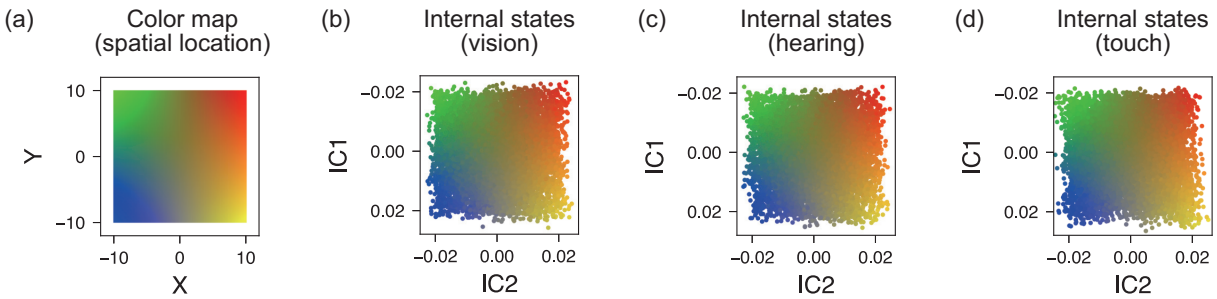

Figure 4. Internal states of the trained shared LSTM. (a) Color map of the spatial location for coloring internal states. (b, c, d) Internal states for each modality. (b) Vision, (c) hearing, and (d) touch. Two independent components found by ICA are shown. No notable characteristic was observed in other independent components. Note that the vertical axis for the internal states, which corresponds to the first independent component, is inverted to make it easier to compare with the color map.

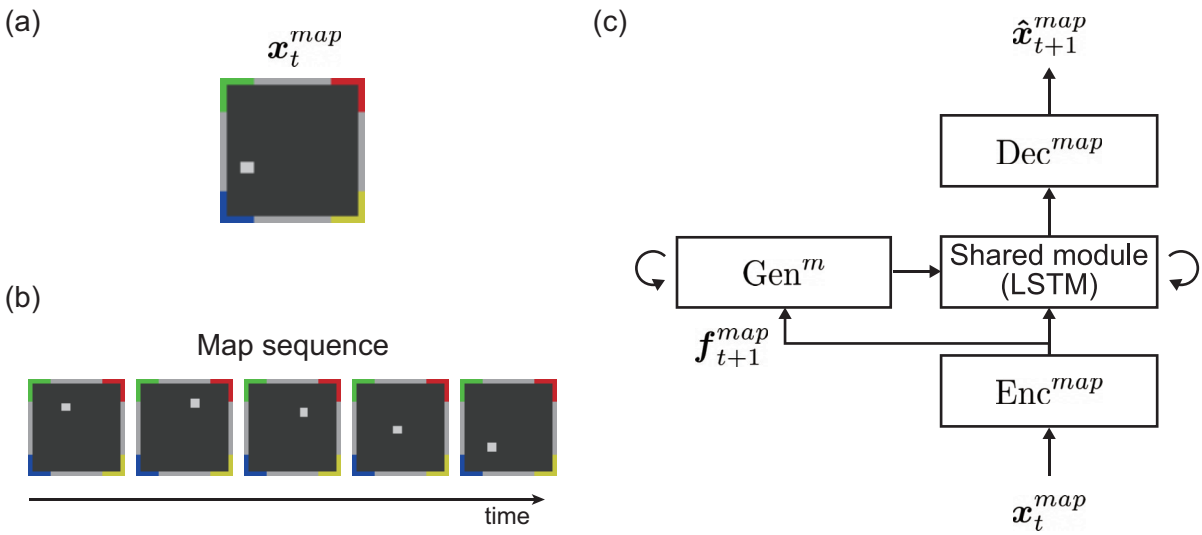

Figure 5. (a) External map and (b) external map sequence using the virtual walk. (c) Prediction network for the external map.

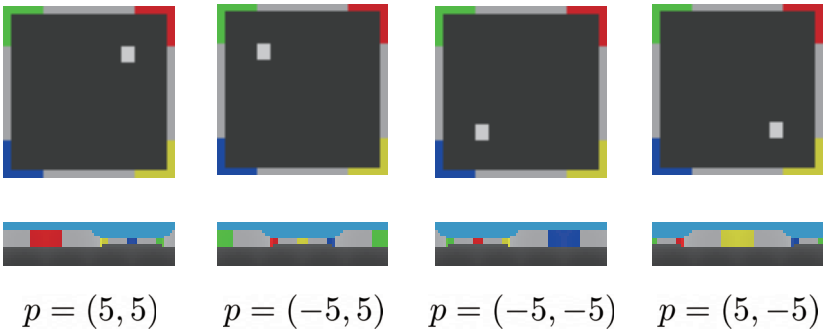

Figure 6. Pairs of the external map (top) and vision (bottom) in which the gray square and the robot are at the corresponding location p. The pairs are collected for the four locations.

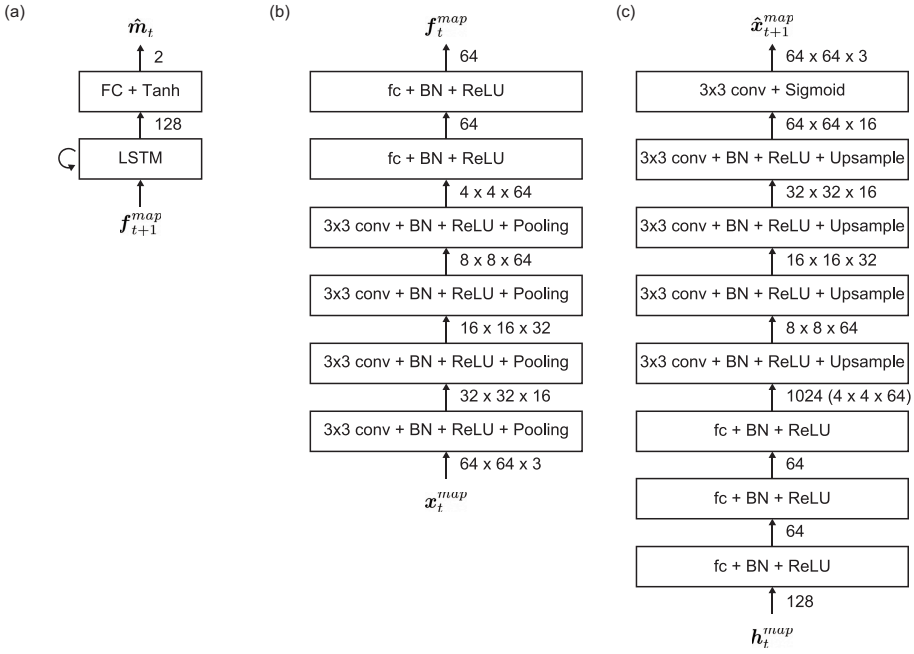

Figure 7. The detailed structures of (a) Gen^m, (b) Enc^{map}, and (c) Dec^{map}.

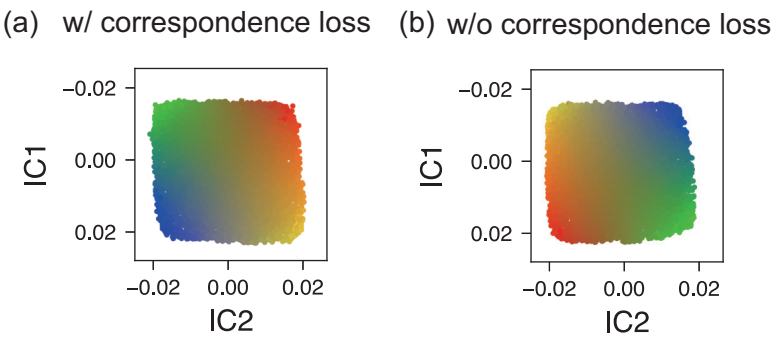

Figure 8. Internal states for the external map prediction. (a) Internal states of the network trained with correspondence loss and (b) without correspondence loss. The internal states are visualized in the same manner as in Figure 4.

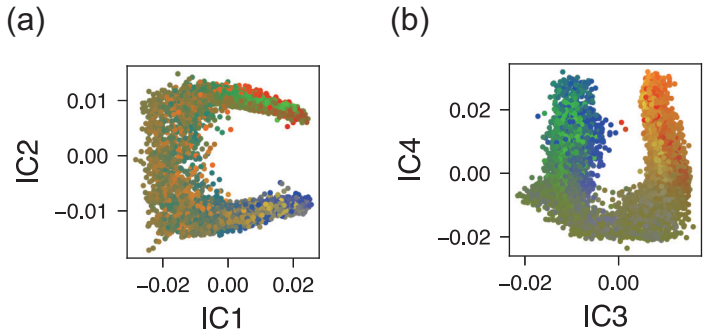

Figure 9. Internal states for the external map prediction in the case where the LSTM was not trained by visual prediction. In this case, four independent components found by ICA are shown: (a) first and second independent components and (b) third and fourth independent components.

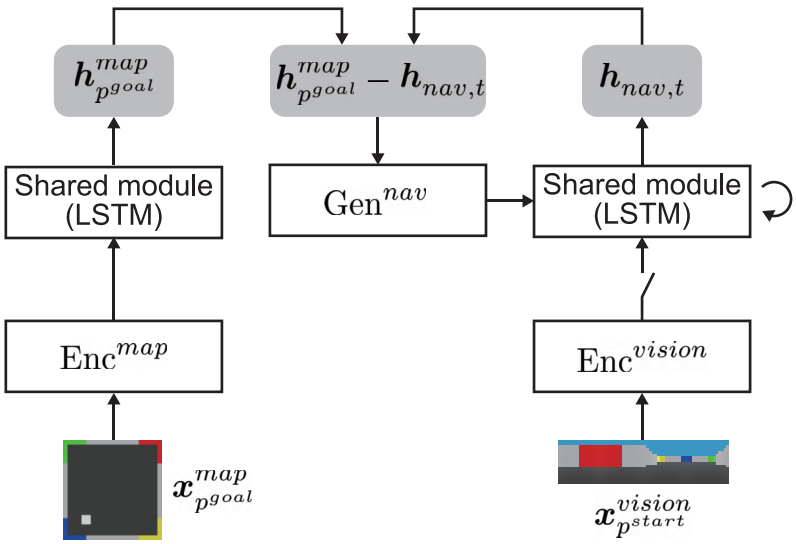

Figure 10. Network with the navigation module. The switch-like illustration from the output of the Enc^{vision} module indicates that the input $\boldsymbol{x}^{vision}_{p^{start}}$ is input only at the initial step of navigation.

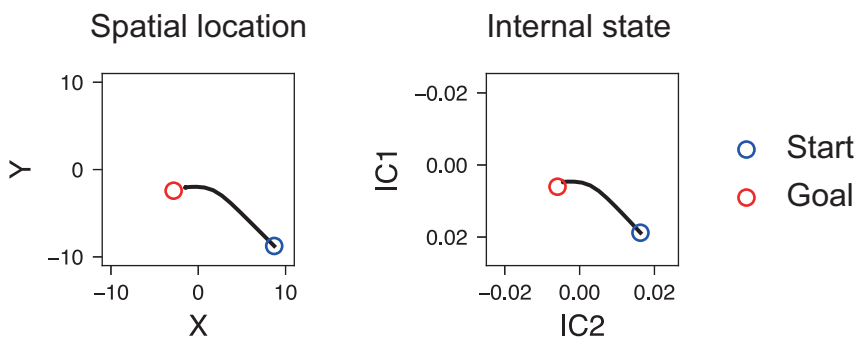

Figure 11. Trajectories of the robot's spatial location and internal states for navigation. Internal states are mapped onto the same ICA space, as shown in Figure 4.