Goal-Directed Behavior under Variational Predictive Coding: Dynamic Organization of Visual Attention and Working Memory

Minju Jung[1,2], Takazumi Matsumoto[2] and Jun Tani[2,*]

Abstract—Mental simulation is a critical cognitive function for goal-directed behavior because it is essential for assessing actions and their consequences. When a self-generated or externally specified goal is given, a sequence of actions that is most likely to attain that goal is selected among other candidates via mental simulation. Therefore, better mental simulation leads to better goal-directed action planning. However, developing a mental simulation model is challenging because it requires knowledge of self and the environment. The current paper studies how adequate goal-directed action plans of robots can be mentally generated by dynamically organizing top-down visual attention and visual working memory. For this purpose, we propose a neural network model based on variational Bayes predictive coding, where goal-directed action planning is formulated by Bayesian inference of latent intentional space. Our experimental results showed that cognitively meaningful competencies, such as autonomous top-down attention to the robot end effector (its hand) as well as dynamic organization of occlusion-free visual working memory, emerged. Furthermore, our analysis of comparative experiments indicated that the introduction of visual working memory and the inference mechanism using variational Bayes predictive coding significantly improved the performance in planning adequate goal-directed actions.

I. INTRODUCTION

Goal-directed behavior is the ability to generate optimal action plans to achieve goals and to execute the action plans generated. Understating how humans recognize goals of others by observation and translate the goals into goal-directed actions is a crucial step toward general intelligence. Goal-directed behavior is becoming increasingly important in the robotics literature because future robots, such as collaborative and home assistant robots, need to execute a variety of goal-directed behaviors, rather than a single fixed one [1], [2], [3]. However, even a simple goal-directed behavior for robots, such as grasping a cup, is not a trivial problem, because the location and shape of a cup need to be extracted visually, and hand-eye coordination must be programmed or learned before action execution [4].

The capacity to anticipate action outcomes based on knowledge of the causal structure of the environment, which is known as mental simulation, is crucial for goal-directed behavior in the brain [5], [6]. Due to the importance of mental simulation in the brain, various neurorobotics models of mental simulation have been proposed [7], [8]. In a similar vein, one promising approach in the robotics literature to address goal-directed behavior is based on predictive models (Fig. 1). Finn and Levine [1] showed that predictive
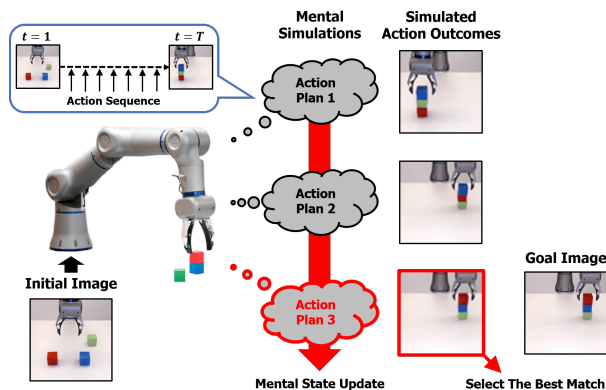
Fig. 1. Goal-directed action planning via iterative mental simulation. A robot mentally simulates different courses of action by updating a robot's mental state and then selects a plan based on the simulated action outcome, which is most similar to the desired goal. However, mental simulation becomes more challenging as the length of a plan T increased.

models can be used for robot planning by selecting an action sequence that maximizes the probability of the desired outcome. However, the following limitations exist: (1) the model was trained to predict vision, but not action, (2) actions were estimated by a stochastic optimization algorithm, which is not part of the predictive model, and (3) a plan was estimated within a short period of time. Choi et al. [2] proposed a neural network model based on the predictive coding framework [9], [10]. The model was trained to encode high-dimensional visuomotor sequences into corresponding initial states in low-dimensional latent intentional space. By doing so, planning can be defined as finding an initial state that mentally generates a visuomotor sequence to achieve a given goal. Therefore, mental simulation is a key capacity to plan effectively. However, predicting high-dimensional visuomotor sequences becomes more challenging as sequences become longer, even though it is essential for complex goal-directed action planning. Furthermore, Choi et al. [2] reported that deterministic predictive coding requires a large number of training samples for good generalization. How can we address long-term visuomotor prediction and generalization problems in the predictive coding framework for better goal-directed action planning?

Humans have two kinds of vision: central vision and peripheral vision. Central vision allows us to focus on important visual stimuli from a narrow receptive field at high resolution, while peripheral vision has a wide receptive field at low resolution. By coordinating the two complementary vision types, humans are able to deal with a high-dimensional raw visual stream efficiently. Especially, goal-directed behavior focuses visual attention on goal-relevant stimuli and ignores

irrelevant stimuli in a top-down manner. Visual information perceived by the retina is first stored in iconic memory, but it rapidly disappears [11]. Therefore, some visual information must be transferred to visual working memory for information integration across visual attention shifts and future usage. Working memory is a cognitive brain system that temporarily stores and manipulates an active representation of information to provide an interface between perception, long-term memory, and action [12]. Visual working memory is the specific type of working memory for visual information and involves mental simulation or manipulation [13], [14].

In this paper, we employ top-down visual attention and visual working memory to address the difficulty of long-term visuomotor prediction in the predictive coding framework. The proposed model has two visuomotor stream sub-networks: dorsal and ventral. The segregation of the dorsal and ventral visuomotor streams is based on the source of the visual information. Dividing high-dimensional visual processing into low-dimensional peripheral and central visual processing reduces the degree of difficulty of long-term visual prediction and the computational burden for training and planning. For visual prediction, we propose a visual prediction module that merges peripheral and central visual prediction into a single visual prediction with the help of an external visuospatial memory, called background memory, working as visual working memory. Background memory preserves long-term visuospatial information, which is crucial for long-term action planning.

Since the proposed model is under the predictive coding framework, the proposed model is trained to generate multiple visuomotor sequences depending on the corresponding initial state. However, unlike a deterministic predictive coding framework, we incorporate a variational Bayes approach [15] into predictive coding framework, called variational predictive coding, to enhance generalization capability with a small number of training samples. In the block stacking experiment, the results showed that the proposed model can plan appropriate visuomotor sequences that achieve goals by updating initial states in the direction of prediction error minimization. Without any supervision, smooth pursuit tracking of the end effector by top-down visual attention as well as maintenance and manipulation in background memory is self-organized and varied depending on goals. Also, the proposed model outperformed, in terms of visual and motor prediction, baseline models: the model without background memory and the model based on deterministic predictive coding. It indicates that the background memory and variational predictive coding are essential for complex goal-directed action planning.

II. METHOD

By extending the predictive coding model proposed in [2], we propose a neural network model that consists of a visual prediction module and three sub-networks: dorsal visuomotor stream, ventral visuomotor stream, and integration networks (Fig. 2). Inspired by the exploitation/exploration framework [16] for the segregation of visual pathways into dorsal and
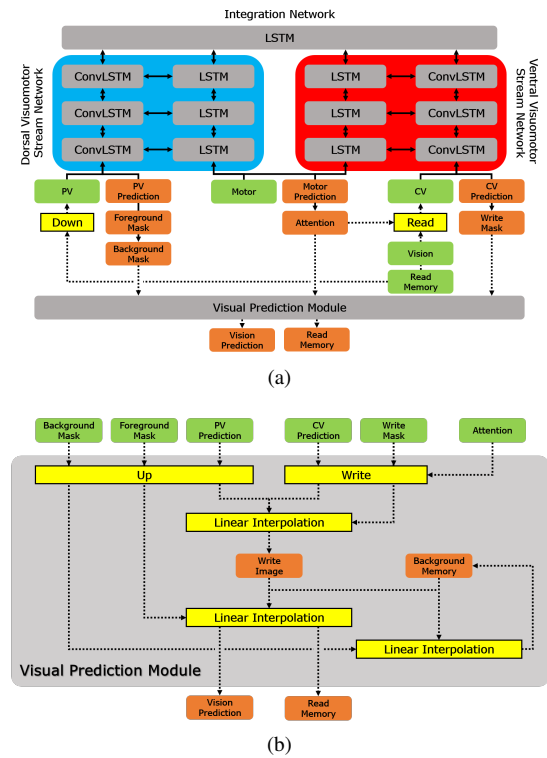

Fig. 2. Our proposed model for goal-directed, long-term planning. (a) Overall model architecture. (b) Visual prediction module.

ventral streams, the dorsal and ventral visuomotor stream networks are responsible for peripheral and central vision, respectively. However, unlike [16], [17] arguing that the dorsal stream is better suited for action, both visuomotor streams of the proposed model involve the processing of motion as in [18]. The size and location of the attentional focus for central vision vary continuously over time in response to changing environments and goals. Each visuomotor stream network consists of visual and motor pathways. Each pathway has multiple recurrent layers with bottom-up and top-down connections between successive layers in the same pathway and a lateral connection between layers from the visual and motor pathways at the same level. The integration network integrates visuomotor information from two visuomotor streams. Furthermore, the visual prediction module seamlessly merges peripheral and central visual predictions into a single visual prediction with background memory that preserves visuospatial information to predict future action outcomes.

A. Dorsal Visuomotor Stream

The dorsal visuomotor stream network consists of visual and motor pathways. For the visual pathway, we employ the network having several stacked convolutional long short-term memory (ConvLSTM) [19] layers to extract spatio-temporal features at multiple scales within a video. Each layer receives inputs from three sources: bottom-up, lateral, and top-down connections. The bottom-up connection carries information from an environment through vision to update internal states of the network and the top-down connection propagates the prediction or belief of an environment from higher to lower

layers. The lateral connection brings information from the motor pathway at the same level to enhance the interaction between visual and motor pathways. The convolution operation with stride is used for the bottom-up connection in order to reduce the size of feature maps. The lateral and top-down connections utilize the deconvolution or transposed convolution operation for expanding the size of the feature maps. The bottom layer receives peripheral visual information $\mathbf{v}_{d,t}^0$ downsampled from the original image $\mathbf{v}_t^0$ and read memory $\mathbf{r}_t$ at each time step t. The neural activations $\mathbf{v}_{d,t}^l$ in lth layer at each time step t are computed as follows:

$$\mathbf{v}_{d,t}^0 = downsample([\mathbf{v}_t^0, \mathbf{r}_t]) \qquad (1)$$

$$\mathbf{v}_{d,t}^l = \begin{cases} ConvLSTM(\mathbf{v}_{d,t}^{l-1}, \mathbf{m}_{d,t-1}^l, \mathbf{a}_{t-1}), & \text{if } l = L \\ ConvLSTM(\mathbf{v}_{d,t}^{l-1}, \mathbf{m}_{d,t-1}^l, \mathbf{v}_{d,t-1}^{l+1}), & \text{otherwise} \end{cases} \qquad (2)$$

where $[\cdot]$ represent the concatenation operation along the channel dimension; $ConvLSTM(\cdot)$ represents the convolutional LSTM layer receiving three parameters that correspond to bottom-up, lateral, and top-down inputs, respectively; $\mathbf{m}_{d,t-1}^l$ and $\mathbf{a}_{t-1}$ represent neural activations of the motor pathway network in lth layer and the integration network at the previous time step, respectively. L represents the number of layers of the visuomotor stream network.

At each time step t, the visual pathway network generates the prediction of peripheral vision $\mathbf{pv}_{t+1}$ for the next time step, the foreground mask $\mathbf{g}_{f,t+1}$, and the background mask $\mathbf{g}_{b,t+1}$ as follows:

$$\begin{pmatrix} \mathbf{pv}_{t+1} \\ \mathbf{g}_{f,t+1} \\ \mathbf{g}_{b,t+1} \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \end{pmatrix} deconv(\mathbf{v}_{d,t}^1) \qquad (3)$$

where $deconv(\cdot)$ represents the deconvolution or transposed convolution operation.

For the motor pathway, the network having stacked LSTM [20] layers is used. As in the visual pathway network, each LSTM layer has bottom-up, lateral, and top-down connections. The neural activations $\mathbf{m}_{d,t}^l$ in lth layer at each time step t are computed as follows:

$$\mathbf{m}_{d,t}^l = \begin{cases} LSTM(\mathbf{m}_{d,t}^{l-1}, \mathbf{v}_{d,t-1}^l, \mathbf{a}_{t-1}), & \text{if } l = L \\ LSTM(\mathbf{m}_{d,t}^{l-1}, \mathbf{v}_{d,t-1}^l, \mathbf{m}_{d,t-1}^{l+1}), & \text{otherwise} \end{cases} \qquad (4)$$

B. Ventral Visuomotor Stream

The architecture of the ventral visuomotor stream network is same as the dorsal one. However, the crucial difference between two visuomotor stream networks is that the ventral network receives central visual information rather than peripheral visual information. For reading and writing central visual information, a spatial transformer network (STN) [21] is used. STN is a module that generates attention parameters for cropping, translating, and scaling of an image. Specifically, central visual information $\mathbf{v}_{v,t}^0$ is obtained from the original image $\mathbf{v}_t^0$ and read memory $\mathbf{r}_t$ by using STN with the attention parameters $\boldsymbol{\alpha}_t$ at each time step t as follows:

$$\mathbf{v}_{v,t}^0 = read([\mathbf{v}_t^0, \mathbf{r}_t], \boldsymbol{\alpha}_t) \qquad (5)$$

where $read(\cdot)$ represents the function to focus the sub-region of a given image by using STN with the attention parameters $\boldsymbol{\alpha}_t$ consisting of s, t_x, and t_y for scaling, x-translation, and y-translation, respectively. The neural activations of the visual pathway $\mathbf{v}_{v,t}^l$ and the motor pathway $\mathbf{m}_{v,t}^l$ in lth layer at each time step t are computed in the same way as Eq. (2) and (4), except for replacing $\mathbf{v}_d$ and $\mathbf{m}_d$ to $\mathbf{v}_v$ and $\mathbf{m}_v$, respectively.

At each time step t, the visual pathway network generates the prediction of central vision $\mathbf{cv}_{t+1}$ for the next time step and the write mask $\mathbf{g}_{w,t+1}$ as follows:

$$\begin{pmatrix} \mathbf{cv}_{t+1} \\ \mathbf{g}_{w,t+1} \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \end{pmatrix} deconv(\mathbf{v}_{v,t}^1) \qquad (6)$$

C. Dual Visuomotor Stream Integration

The integration network, having a single LSTM layer, is placed on top of the two visuomotor stream networks. Outputs of two visuomotor stream networks in the top layer are fed into the integration network. Hence, the integration network can integrate information from different modalities (vision and motor) and from different scales (peripheral and central vision). Integrated information containing the abstraction of an environment and the history of conducted actions. This abstract information in the integration network is projected to two visuomotor stream networks in order to predict the next visuomotor inputs and to interact with those networks. The neural activations $\mathbf{a}_t$ at each time step t are computed as follows:

$$\mathbf{a}_t = LSTM(\mathbf{v}_{d,t}^L, \mathbf{v}_{v,t}^L, \mathbf{m}_{d,t}^L, \mathbf{m}_{v,t}^L) \qquad (7)$$

D. Visuomotor Generation

Since the proposed model has two visuomotor stream networks, these need to be merged for visuomotor generation. The neural activations of dorsal and ventral motor pathways are used to generate the attention parameters $\boldsymbol{\alpha}_{t+1}$ and motor prediction $\mathbf{m}_{t+1}$ for the next time step as follows:

$$\boldsymbol{\alpha}_{t+1} = MLP(\mathbf{m}_{d,t}^1, \mathbf{m}_{v,t}^1) \qquad (8)$$

$$\mathbf{m}_{t+1} = MLP(\mathbf{m}_{d,t}^1, \mathbf{m}_{v,t}^1) \qquad (9)$$

where $MLP(\cdot)$ represents the multi-layer perceptron.

To generate a visual prediction $\mathbf{v}_{t+1}$ at the next time step, we propose a visual prediction module having three stages (Fig. 2(b)). First, the peripheral visual prediction $\mathbf{pv}_{t+1}$ and central visual prediction $\mathbf{cv}_{t+1}$ from the dorsal and ventral visuomotor stream networks, respectively, are merged into a single image, called a write image, $\mathbf{w}_{t+1}$. Since the resolutions of peripheral and central vision are lower than of the original vision, the peripheral visual prediction $\mathbf{pv}_{t+1}$, foreground mask $\mathbf{g}_{f,t+1}$, and background mask $\mathbf{g}_{b,t+1}$ are upsampled; and the central visual prediction $\mathbf{cv}_{t+1}$ and write mask $\mathbf{g}_{w,t+1}$ are transformed to the original image using the attention parameters $\boldsymbol{\alpha}_{t+1}$ before merging. Note that the attention parameters $\boldsymbol{\alpha}_{t+1}$ are shared for writing and reading as in [22]. Hence, the attention parameters $\boldsymbol{\alpha}_{t+1}$ will be used for reading the next image $\mathbf{v}_{t+1}^0$ as shown in Eq. (5). For smooth blending, the write mask $\mathbf{g}_{w,t+1}$ is used for the linear

interpolation between the peripheral and central visual predictions. Second, the visual prediction $\mathbf{v}_{t+1}$ is computed by interpolation between the write image $\mathbf{w}_{t+1}$ and background memory $\mathbf{bg}_t$ weighed by the foreground mask $\mathbf{g}_{f,t+1}$. The foreground masked background memory, called read memory $\mathbf{r}_{t+1}$, used for the visual prediction is given as a visual input for next time step. The background memory is important to preserve the central visual predictions at high resolution over time. The central visual prediction at the current time step can be overwritten at the next time step by occlusion, but it needs to be restored when occlusion is resolved. Finally, the next background memory $\mathbf{bg}_{t+1}$ is updated by the current background memory $\mathbf{bg}_t$ and write image $\mathbf{w}_{t+1}$ with the background mask $\mathbf{g}_{b,t+1}$. All three stages explained above are computed as follows:

$$
\begin{aligned}
\mathbf{w}_{t+1} = (1 - f_w(\mathbf{g}_{w,t+1}, \boldsymbol{\alpha}_{t+1})) \odot f_{up}(\mathbf{pv}_{t+1}) \\
+ f_w(\mathbf{g}_{w,t+1}, \boldsymbol{\alpha}_{t+1}) \odot f_w(\mathbf{cv}_{t+1}, \boldsymbol{\alpha}_{t+1})
\end{aligned}
\tag{10}
$$

$$
\mathbf{r}_{t+1} = (1 - f_{up}(\mathbf{g}_{f,t+1})) \odot \mathbf{bg}_t
\tag{11}
$$

$$
\mathbf{v}_{t+1} = \mathbf{r}_{t+1} + f_{up}(\mathbf{g}_{f,t+1}) \odot \mathbf{w}_{t+1}
\tag{12}
$$

$$
\mathbf{bg}_{t+1} = (1 - f_{up}(\mathbf{g}_{b,t+1})) \odot \mathbf{bg}_t + f_{up}(\mathbf{g}_{b,t+1}) \odot \mathbf{w}_{t+1}
\tag{13}
$$

where $\odot$ represents the element-wise multiplication, $f_{up}(\cdot)$ represents the upsampling function, and $f_w(\cdot)$ represents the write function for transforming an attended sub-region to an original image.

E. Sampling Initial States Based on Variational Bayes

Initial sensitivity is one of important characteristics of recurrent neural network (RNN) models [23]. The proposed model is trained with the mapping between the training visuomotor sequences and corresponding initial states. Once the training is finished successfully, the model is able to generate multiple visuomotor sequences based solely upon the corresponding trained initial states without any external input. Hence, the trained initial state can be considered an intention, which should be prepared before executing goal-directed actions, as in the brain [24]. Most previous models utilizing initial sensitivity were trained using a deterministic approach [23], [25], [2]. However, deterministic models have difficulty dealing with stochasticity and uncertainty in an environment, leading to blurry predictions [15], [26], [27]. Hence, we use a variational Bayes approach [15] for training the initial states in order to capture the full distribution of outcomes. The initial state $\mathbf{s}_0^n$ for the nth training sample is sampled using the reparametrization trick [15] as follows:

$$
\mathbf{s}_0^n = \boldsymbol{\mu}_0^n + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_0^n
\tag{14}
$$

where $\boldsymbol{\mu}_0^n$ and $\boldsymbol{\sigma}_0^n$ represent the mean and standard deviation of the initial state $\mathbf{s}_0^n$, respectively, and $\boldsymbol{\epsilon}$ represents an auxiliary noise sampled from a standard normal distribution.

F. Training

The proposed model is trained to mentally generate multiple visuomotor sequences with corresponding initial states. There are two types of initial state: prior and posterior. The prior initial state is shared for all training samples, but posterior initial states are provided for each training sample. Each initial state is parameterized by the mean and standard deviation, which are updated directly during the training phase. Note that we train the prior as well as posterior initial states as in [27]. The weights, biases, and initial states of the model are updated to minimize both the visuomotor prediction error and the Kullback-Leibler divergence between prior and posterior initial states, which is same as maximizing the variational lower bound as in the variational auto-encoder (VAE) [15]. The loss L^n for the nth training sample is defined as follows:

$$
L^n = L_v^n + L_m^n + \beta D_{KL}(q_{\boldsymbol{\phi}^n}(\mathbf{s}_0^n) || p_{\boldsymbol{\theta}}(\mathbf{s}_0))
\tag{15}
$$

$$
L_v^n = \sum_{t=1}^{T^n} L_{v,t}^n = \sum_{t=1}^{T^n} (\mathbf{v}_t^n - \tilde{\mathbf{v}}_t^n)^2 + (f_{up}(\mathbf{pv}_t^n) - \tilde{\mathbf{v}}_t^n)^2
\tag{16}
$$

$$
L_m^n = \sum_{t=1}^{T^n} L_{m,t}^n = \sum_{t=1}^{T^n} (D_{KL}(\mathbf{m}_t^n || \tilde{\mathbf{m}}_t^n))
\tag{17}
$$

where $\boldsymbol{\phi}^n$ and $\boldsymbol{\theta}$ represent parameters of the posterior initial state for the nth training sample and prior initial state, respectively; $D_{KL}(\cdot)$ represents the Kullback-Leibler divergence; β represents the hyper-parameter for controlling the balance between minimizing the reconstruction error and fitting the prior [27]; T^n represents the length of the nth target visuomotor sequence; and $\tilde{\mathbf{v}}_t^n$ and $\tilde{\mathbf{m}}_t^n$ represent the nth target visual and motor values at each time step t. The loss L^n is for the case of a single training sample, but it can be easily extended to the case of a mini-batch training samples.

During the training phase, we follow the closed-loop training method used in [25] to improve the mental simulation capability of the model. In the closed-loop training, the visual and motor predictions are fed into the model as input for the next time step. However, since it is hard to train the model using complete prediction feedback, external and predicted visuomotor values are blended together (weighted by 0.1 and 0.9, respectively) as in [25].

G. Planning

During the planning phase, the loss L^n in Eq. (15) is no longer available because only a few initial visuomotor values and a final image are given to the model. Therefore, the loss L_p^n for planning the nth testing sample is defined as follows:

$$
L_p^n = \sum_{t=1}^{T_g} (L_{v,t}^n + L_{m,t}^n) + L_{v,T_e}^n + \beta D_{KL}(q_{\boldsymbol{\psi}^n}(\mathbf{s}_{n,0}) || p_{\boldsymbol{\theta}}(\mathbf{s}_0))
\tag{18}
$$

where $\boldsymbol{\psi}^n$ represents parameters of the initial state for the nth testing sample, T_g represents the number of initial target visuomotor values given to the model, and T_e represents the

end step for visuomotor generation. Note that only initial states are updated to minimize the loss L_p while the weights and biases of the model are fixed during the planning phase.

III. RESULTS

A. Implementation Details

Each visual pathway network of both dorsal and ventral visuomotor stream networks consisted of three ConvLSTM layers having 16, 32, and 64 feature maps, respectively. The convolution kernel, stride, and padding sizes were set to 5×5, 2×2, and 2×2, respectively, for bottom-up connections. The deconvolution kernel, stride, and padding sizes were set to 6×6, 2×2, and 2×2, respectively, for top-down connections, except from the integration network. For lateral connections from the motor pathway network and the top-down connection from the integration network, the convolutional kernel size was set the same as the resolution of feature maps at each layer in the visual pathway network. The resolution of peripheral and central vision were set to half the original vision. Each motor pathway network of both visuomotor stream networks consisted of three LSTM layers having 512, 256, and 128 neurons, respectively. To generate motor and attention at each time step as shown in Eq. (8) and (9), MLP with one hidden layer of 256 neurons, layer normalization (LN) [28], and a rectified linear unit (ReLU) activation function was used. The integration network consisted of a single LSTM layer having 512 neurons.

For training, Adam optmizer [29] was applied to train the weights, biases, and initial states of the model over 1000 epochs. The learning rate of the weights, biases, and prior initial state was set to 5e-4. Since each posterior initial state is updated only when its corresponding training sample is given to the model, the learning rate of the posterior initial states was set to 0.015. Mini-batch size was set to 10. The hyper-parameter β in Eq. (15) and (18) was set to 1e-5. An L2-norm weight decay of 1e-4 was applied while updating the model parameters to prevent over-fitting. Because of the exploding gradient problem, we employed a gradient clipping method [30] which rescales the L2-norm of the gradient to a threshold whenever the L2-norm exceeds that threshold. Here, the threshold was set to 0.2. Each mean and standard deviation of both prior and posterior initial states were initialized to 0 and 1.

B. Evaluation Protocol

During the planning phase, the model inferred the posterior initial state of a testing sample by iteratively updating the initial state toward to minimize the loss L_p defined in Eq. (18). T_g and T_e were set to 5 and 100. The prior initial state was used to initialize the initial state as a starting point of planning. The error regression was repeated over 50 epochs. At each epoch, 16 visuomotor sequences were generated based on 16 initial states sampled by using the reparameterization trick in Eq. (14) with the mean and standard deviation of the current initial state, and then the loss L_p was computed for each sequence. The sequence having the minimum loss L_p over 16 visuomotor sequences

was used for updating the initial state. We repeated the above error regression procedure 6 times and selected the visuomotor sequence having the minimum loss L_p as an optimal plan.

C. RGB Block Stacking

1) Experiment Setting: We conducted a robot manipulation experiment using a Torobo Arm robot manufactured by Tokyo Robotics. Torobo Arm has a total of 8 joints including the end effector, of which 6 joints were used in the following experiments. Each motor joint was converted to a 10 dimensional sparsely encoded vector as in [25] to reduce the overlap between motor sequences caused by the low dimensionality of the motor. The task space for the robot was approximately $30cm^2$, and an RGB camera was fixed facing both the task space and the robot. While collecting data, the robot joint angles and video frames were sampled at 20Hz. All data were temporally downsampled 7 times to reduce computational and memory costs. All video frames were rescaled to 64×64 pixels and normalized to -1 to 1.

For the experimental task, we used three colored cubes (red (R), blue (B), and green (G)) size of $5cm^3$. Data collection was automated according to the following steps: (1) randomly sampling a location for each block from a $n \times n$ grid, (2) having the robot initialize the task by placing each block at the corresponding location, (3) returning the robot to the home position, and (4) executing the stacking operation. The block location grid size n was set to 10 for training and 8 for testing to check the generalization capability of the model with an unlearned location distribution. For the stacking operation, we sampled three possible combinations of blocks as goal states, which the robot should achieve. Depending on the sampled goal state, the robot generated the corresponding motor trajectory while recording both video and motor sequence. There are total six possible combinations for stacking three color blocks, but we allowed only four stacking combinations (RGB, RBG, GRB, and GBR) for training by excluding remaining two stacking combinations (BRG and BGR) when the blue block be the base block. During the testing phase, these two unlearned stacking combinations were used to evaluate the generalization capability of the model. We used 300 videos for training and 45 videos for testing. The objective of the experiment is whether the model can generate correct action planning or not when a goal image is given. The task is difficult to achieve because the model needs to recognize what is the goal state based on a given goal image and then generates a correct visuomotor sequence for completing the goal.

2) Visual Prediction Mechanism: Due to the segregation of peripheral and central vision and the background memory, the visual prediction mechanism of the proposed model is much more complex than previous unstructured predictive models [1], [2]. Therefore, we qualitatively analyzed how the proposed model mentally generated visual sequences after planning was finished (Fig. 3). The peripheral visual prediction covered the full receptive field, but its quality was low due to the resolution limitation. On the other hand,

Fig. 3. Example of visual mental simulation. (a) Peripheral visual prediction. (b) Central visual prediction. (c) Background memory. (d) Visual prediction.

the quality of central visual prediction was high by focusing on a narrow receptive field, called an attention window. By utilizing a dynamic top-down visual attention, central vision maximally sampled visual information relevant to the given goal. Interestingly, we observed that the attention window tracked the robot end effector without any supervision. A possible interpretation can be made in terms of a retinal slip, which is created by the difference between eye velocity and the motion of a target object. During tracking motions, the eye accelerates in the direction of target motion to reduce the retinal slip. As in previous work [31], the proposed model tried to locate the attention window on the end effector to minimize the vision loss, which can be interpreted as the retinal slip caused by the movement of the end effector. This is analogous to smooth pursuit eye movements in primates.

In order to effectively predict high-resolution visual sequences, the visual prediction module mentally generates the visual prediction by merging the peripheral and central visual predictions, which complement each other with regard to resolution and the size of a receptive field, with the background memory. Background memory is a core part of the visual prediction module for long-term visuospatial information maintenance. As shown in Fig. 3(c), background memory focused the block configuration at each time step and ignored robot movement, which often caused block occlusion. With background memory, the model was able to store the information of the block configuration even when the block(s) were completely occluded by the robot and to use it for visuomotor prediction in the future. This demonstrates that background memory works similarly to visual working memory, based solely on learning. Finally, Fig. 3(d) shows the visual prediction generated by the visual prediction module. The visual prediction looks sharper than the peripheral visual prediction (especially for the gripper and blocks) because of the central visual prediction and background memory.

3) Visuomotor Contingency: One of the characteristics of the proposed model is that it predicts both visual and motor sequences jointly compared with previous work in which motor sequences were used as auxiliary inputs for video prediction [1], [26], [32]. Fig. 4(a) shows that the visual and motor predictions were highly correlated. It means that the

proposed model learned the relationship between visual and motor sequences, called visuomotor contingency. Therefore, visual prediction error minimization leads to motor prediction error minimization and vice versa, which is the reason that goal-directed visuomotor planning is possible only with a goal image.

4) Importance of Background Memory: To investigate the importance of background memory for visuomotor planning, we compared the planning performance between the proposed model and its variant without background memory in a qualitative and quantitative manner. In the model without background memory, background memory was removed in the visual prediction module and read memory was not given to the model as an input. For quantitative evaluation, we used two different evaluation metrics: perceptual loss [33] and dynamic time warping (DTW) [34] for visual and motor prediction evaluation, respectively. It has been reported that conventional pixel-wise metrics for video prediction, such as PSNR and SSIM, are far from human perception because these assume pixel-level independence [26], [32]. Hence, we used a perceptual metric proposed in [33] as the evaluation metric of visual prediction. DTW [34] is used for the evaluation metric of motor prediction because the predicted motor sequence can be longer or shorter than the target motor sequence, which is unknown until it is finished. DTW measures the similarity between two sequences, which may vary in time or speed.

We hypothesized that elimination of background memory will severely degrade the performance of planning because background memory takes charge of visuospatial information maintenance. For a qualitative comparison between the proposed model and ablations of the proposed model, we illustrated the visuomotor sequences that were mentally generated by each models (Fig. 4). As we hypothesized, the model without background memory failed to generate the green block after the robot picked the blue block due to occlusion. To compensate for the green block that had disappeared, the model created an imaginary blue block where the blue block was originally located after the blue block was picked, and then picked this imaginary blue block for the second stacking. Obviously, the corresponding motor prediction was different from the target motor sequence
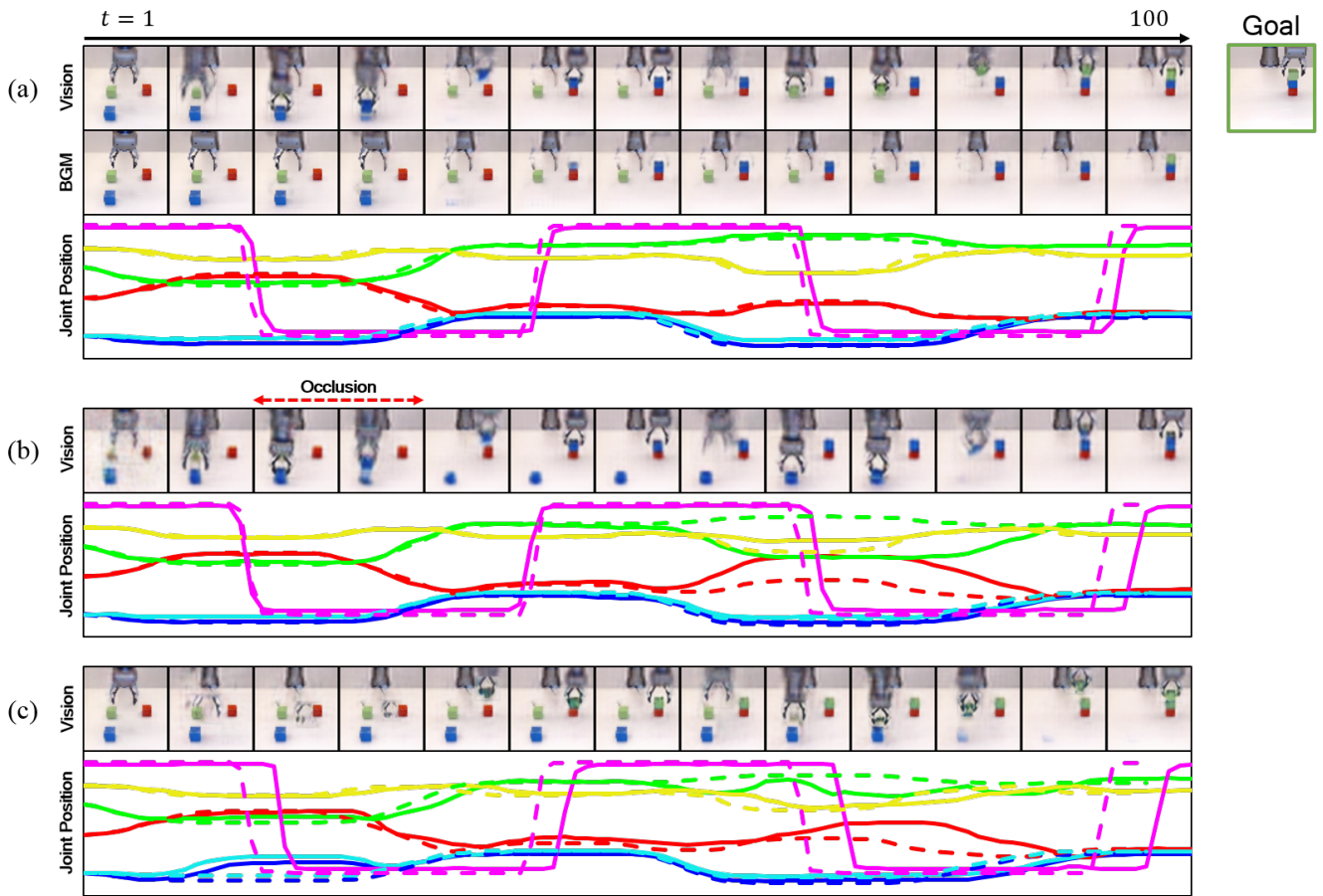
Fig. 4. Qualitative comparison of planning performance. Solid and dashed lines represent the predicted and target motor sequences, respectively. (a) Proposed model. (b) Model without background memory. Without background memory, the model forgets the green block during occlusion (red dotted arrow). (c) Deterministic model. The model fails to find an adequate plan due to over-fitting. Abbreviation: BGM, background memory.
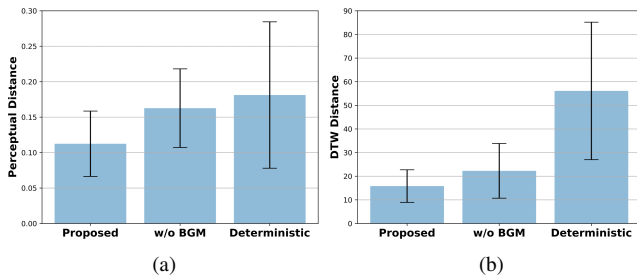

Fig. 5. Quantitative comparison of planning performance. (a) Visual prediction evaluation. (b) Motor prediction evaluation. Bars and error bars represent the means and standard deviations, respectively.

during the second half of mental simulation due to the repetition of the blue block picking and failed to achieve the desired goal. On the other hand, in the proposed model, the visual prediction immediately recovered the green block after the occlusion was resolved and the motor prediction was well matched with the target motor sequence. The reason is that the background memory preserves and updates the information of the block configuration, which is used by the visual prediction module for visual prediction of future actions. Fig. 5 shows a quantitative comparison of the effect of background memory. The proposed model consistently outperformed the model without background memory in terms of visual and motor predictions. Both qualitative and

quantitative results demonstrated that background memory is essential for visuomotor planning.

5) Deterministic versus Stochastic Initial States: As mentioned in [2], deterministic predictive models require a huge number of training samples to achieve good generalization. However, it is difficult and time-consuming to prepare an enough training samples, especially for real robot experiments. Furthermore, the required number of training samples exponentially increases as the complexities of tasks and models increase. To address this limitation, we proposed a stochastic predictive model under variational predictive coding because Bayesian approaches prevent over-fitting on few training samples [35]. We compared the deterministic and stochastic predictive models to verify the effect of variational Bayes in the predictive coding framework. The stochastic model sampled initial states from reparameterized normal distributions in Eq. (14). In the deterministic model, initial states were parameterized only using a mean without a standard deviation, because there is no sampling process for initial states. The losses for training and planning of the deterministic model were the same as the stochastic model, but the Kullback-Leibler divergence between prior and posterior initial states was computed with a fixed standard deviation set to 1. During the planning phase, error regression was repeated over 500 epochs without sampling.

In qualitative analysis, we discovered that the deterministic model had difficulty finding plausible visuomotor sequences, which indicates the lack of generalization. Fig. 4(c) shows that the robot disappeared while picking the first block and the robot's picking movements were not accurate. Also, the motor prediction little resembled the target motor sequence. In quantitative analysis, the stochastic model significantly outperformed the deterministic model as shown in Fig. 5. Note that, in the case of the (stochastic) model without background memory, the model failed to plan the correct visuomotor sequence, but the predicted visuomotor sequence seemed at least plausible. These results indicate that the variational predictive coding framework provides better generalization capability than the deterministic one.

IV. Conclusion

In this paper, we proposed a neural network model under variational predictive coding incorporating top-down visual attention and visual working memory for goal-directed, long-term planning through mental simulation. The experimental results for the task of block stacking demonstrated that the proposed model was able to plan long-term, high-dimensional visuomotor sequences by Bayesian inference in low-dimensional latent intentional space. Furthermore, the proposed model clearly outperformed ablated variants that excluded visual working memory or variational Bayes in a qualitative and quantitative manner. Our results showed that the proposed model self-organized smooth pursuit-like movements that tracked the end effector, mentally manipulated visuospatial information stored in visual working memory, and achieved remarkable generalization ability in a sample-efficient manner by employing variational Bayes. The future study should involve with replanning during action execution in an online manner.

References

[1] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[2] M. Choi, T. Matsumoto, M. Jung, and J. Tani, "Generating goal-directed visuomotor plans based on learning using a predictive coding type deep visuomotor recurrent neural network model," *CoRR*, vol. abs/1803.02578, 2018.

[3] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[4] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[5] J. Grzes and J. Decety, "Functional anatomy of execution, mental simulation, observation, and verb generation of actions: A meta-analysis," *Human Brain Mapping*, vol. 12, no. 1, pp. 1–19, 2001.

[6] G. Pezzulo, M. A. van der Meer, C. S. Lansink, and C. M. Pennartz, "Internally generated sequences in learning and executing goal-directed behavior," *Trends in Cognitive Sciences*, vol. 18, no. 12, pp. 647–657, 2014.

[7] A. D. Nuovo, V. M. D. L. Cruz, and D. Marocco, "Special issue on artificial mental imagery in cognitive systems and robotics," *Adaptive Behavior*, vol. 21, no. 4, pp. 217–221, 2013.

[8] K. Seepanomwan, D. Caligiore, A. Cangelosi, and G. Baldassarre, "Generalisation, decision making, and embodiment effects in mental rotation: A neurorobotic architecture tested with a humanoid robot," *Neural Networks*, vol. 72, pp. 31– 47, 2015.

[9] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, pp. 79–87, 1999.

[10] K. Friston, "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience*, vol. 11, pp. 127–138, 2010.

[11] M. Coltheart, "Iconic memory and visible persistence," *Perception & Psychophysics*, vol. 27, no. 3, pp. 183–228, 1980.

[12] A. Baddeley, "Working memory: looking back and looking forward," *Nature Reviews Neuroscience*, vol. 4, pp. 829–839, 2003.

[13] V. K. Sims and M. Hegarty, "Mental animation in the visuospatial sketchpad: Evidence from dual-task studies," *Memory & Cognition*, vol. 25, no. 3, pp. 321–332, 1997.

[14] P. Roelfsema and F. De Lange, "Early visual cortex as a multiscale cognitive blackboard." *Annual Review of Vision Science*, vol. 2, pp. 131–151, 2016.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[16] B. R. Sheth and R. Young, "Two visual pathways in primates based on sampling of space: Exploitation and exploration of visual information," *Frontiers in integrative neuroscience*, vol. 10, p. 37, 2016.

[17] M. A. Goodale and A. Milner, "Separate visual pathways for perception and action," *Trends in Neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[18] T. Schenk and R. D. McIntosh, "Do we have independent visual streams for perception and action?" *Cognitive Neuroscience*, vol. 1, no. 1, pp. 52–62, 2010.

[19] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

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

[21] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[22] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, k. kavukcuoglu, and G. E. Hinton, "Attend, infer, repeat: Fast scene understanding with generative models," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[23] R. Nishimoto and J. Tani, "Learning to generate combinatorial action sequences utilizing the initial sensitivity of deterministic dynamical systems," *Neural Networks*, vol. 17, no. 7, pp. 925–933, 2004.

[24] K. Shima, M. Isoda, H. Mushiake, and J. Tanji, "Categorization of behavioural sequences in the prefrontal cortex," *Nature*, vol. 445, pp. 315–318, 2006.

[25] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment," *PLOS Computational Biology*, vol. 4, no. 11, pp. 1–18, 2008.

[26] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *CoRR*, vol. abs/1710.11252, 2017.

[27] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," *CoRR*, vol. abs/1802.07687, 2018.

[28] L. J. Ba, R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[30] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 2013.

[31] L. P. Wijesinghe, J. Triesch, and B. E. Shi, "Robot end effector tracking using predictive multisensory integration," *Frontiers in Neurorobotics*, vol. 12, p. 66, 2018.

[32] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *CoRR*, vol. abs/1804.01523, 2018.

[33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[35] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, pp. 452–459, 2015.