Imitation and Mirror Systems in Robots through Deep Modality Blending Networks

M. Yunus Seker[a,*], Alper Ahmetoglu[a], Yukie Nagai[d], Minoru Asada[b], Erhan Oztop[b,c], Emre Ugur[a]

[a] *Bogazici University, Bebek, Istanbul, 34342, Turkey*
[b] *Osaka University, Suita, Osaka, Japan*
[c] *Ozyegin University, Istanbul, Turkey*
[d] *The University of Tokyo, Bunkyo-ku, Tokyo, Japan*

Abstract

Learning to interact with the environment not only empowers the agent with manipulation capability but also generates information to facilitate building of action understanding and imitation capabilities. This seems to be a strategy adopted by biological systems, in particular primates, as evidenced by the existence of mirror neurons that seem to be involved in multi-modal action understanding. How to benefit from the interaction experience of the robots to enable understanding actions and goals of other agents is still a challenging question. In this study, we propose a novel method, deep modality blending networks (DMBN), that creates a common latent space from multi-modal experience of a robot by blending multi-modal signals with a stochastic weighting mechanism. We show for the first time that deep learning, when combined with a novel modality blending scheme, can facilitate action recognition and produce structures to sustain anatomical and effect-based imitation capabilities. Our proposed system, which is based on conditional neural processes, can be conditioned on any desired sensory/motor value at any time-step, and can generate a complete multi-modal trajectory consistent with the desired conditioning in one-shot by querying the network for all the sampled time points in parallel avoiding accumulation of prediction errors. Based on simulation experiments with an arm-gripper robot and an RGB camera, we showed that DMBN could make accurate predictions about any missing modality (camera or joint angles) given the available ones outperforming recent multimodal variational autoencoder models in terms of long-horizon high-dimensional trajectory predictions. We further showed that given desired images from different perspectives, i.e. images generated by the observation of other robots placed on different sides of the table, our system could generate image and joint angle sequences that correspond to either anatomical or effect based imitation behavior. To achieve this mirror-like behavior our system does not perform a pixel-based template matching but rather benefits from and relies on the common latent space constructed by using both joint and image modalities, as shown by additional experiments. Overall, the proposed DMBN architecture not only serves as a computational model for sustaining mirror neuron-like capabilities, but also stands as a powerful machine learning architecture for high-dimensional multi-modal temporal data with robust retrieval capabilities operating with partial information in one or multiple modalities.

1. Introduction

With appropriate and sufficient amount of data, a range of sensorimotor learning tasks encountered by robots and biological systems can be solved by deep learning. However, unlike the abundance of data for image recognition and language modeling, robots and biological systems often need to harvest data themselves by either using self-exploration based learning or by observing the relevant behaviors of other agents. These two alternatives are studied in robotics and machine learning under the general titles of Reinforcement Learning (RL) [1] and Learning from Demonstration (LfD)[2]. Although the use of self-observation during self-executed actions is common for forming a reward signal in RL, it is not well addressed how to benefit the agent in a cognitive developmental sense, for example, for recognizing actions of others or forming a general imitation capacity. Learning to interact with the environment not only empowers the agent with manipulation capability but also generates information to facilitate the building of action understanding and imitation capabilities. This seems to be a strategy adopted by biological systems, in particular primates, as evidenced by the existence of mirror neurons [3, 4] in the ventral premotor cortex of those animals, which encode actions in a multi-modal fashion [5]. For example, there are mirror neurons that become active when the animal breaks a peanut, observes an experimenter do the same act or hears the sound of peanut cracking [6]. With such a system, sensed actions are mapped to one's own motor representation; and thus can bootstrap imitation, by for example, understanding the parts of an observed act in terms of the existing 'action vocabulary' of the animal, which can be reproduced in sequence yielding novel action imitation capability. Although, it is not clear whether mirror neurons play a role in imitation, as their exact function and mechanism are far from clear, computational modeling may help produce insights towards understanding them [7]. Therefore, from a scientific and also a technological point of view, it is desirable to

*Corresponding author
Email address: yunus.seker1@boun.edu.tr (M. Yunus Seker)

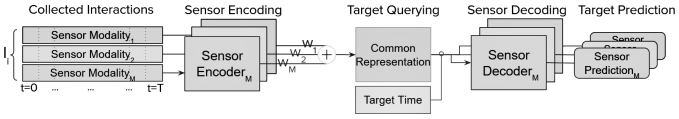

Figure 1: General architecture of a Deep Modality Blending Network

develop a neural multi-modal action representation system that can learn/store actions and recall them from partial information that might be transformed as in the case of action observation from different perspectives. In fact, there exist a range of computational models related to mirror neurons and their function in the literature ([8, 9, 10, 11, 12, 13]) that have leveraged our understanding by creating hypotheses to be tested. Now, the time is ripe for a less constrained, end-to-end and more powerful multi-modal action representation mechanism for obtaining better insights. In particular, the existing multi-modal action representation schemes based on self-observation either fall short of providing robust recognition and imitation capability or rely on feature engineering.

In this study, we improve the state of the art in multi-modal action representation by showing for the first time that deep learning, when combined with a novel modality blending scheme, can facilitate feature-engineering-free action recognition and basic imitation capabilities under perspective changes with only partial information. Moreover, the modality blending scheme produces latent representations that can sustain both anatomical and effect-based imitation capabilities. We call the developed multi-modal action representation architecture as a Deep Modality Blending Network (DMBN).

DMBN connects multiple modalities by blending them as random mixtures of modality-specific latent representations to form a common latent representation for seamless transfer from one modality to another (see Figure 1). The DMBN architecture follows an encoder-decoder structure where each modality is summarized by its corresponding encoder network, processing the sensorimotor data into a compact latent representation. While learning, not only these latent representations are formed but they are blended together into a common representation through stochastic mixture weights. After learning, using the common representation, each decoder network can predict the corresponding modality for an arbitrary desired time-step, effectively generating outcome predictions as temporal sequences for all the modalities. In this sense, the common latent layer in our network encodes representation of the complete multi-modal trajectories rather than encoding modalities in particular time steps. This feature sets our system apart from its competitors [14, 11] and give it a big advantage. To be concrete, our system can be conditioned on any desired sensory/motor value at any time-step, and can generate a complete multi-modal trajectory consistent with the desired conditioning in one-shot by querying the network for all the sampled time points in parallel. This one-shot full trajectory decoding ability makes our system very accurate as it does not suffer from the error accumulation faced by systems that need to chain next-state predictions in order to generate full trajectories.

To demonstrate the efficacy of the proposed DMBN architecture, we implemented it in a simulated manipulation setup. In this setup, an object was placed in the middle of a table, and an arm-gripper robot was set to execute grasp and push actions on it with different approach directions. The robot observed the consequences of its actions using an RGB camera from a fixed perspective, and learned the generated multi-modal sensory (visual and proprioceptive) signals as sensory trajectory distributions through the proposed DMBN architecture. After learning,

- Given desired images at any time point (such as images of objects lifted or pushed away), our system can find the joint trajectories that are required to generate changes in the environment to observe these images;
- Given joint angles at any time point(s), our system can generate the sequence of images that are expected to be observed during the execution of the action that is consistent with given angles;
- Given desired images from different perspectives, i.e. images generated by the observation of other robots placed on different sides of the table, our system can generate image and joint angle sequences that correspond to valid actions of the robot;
- Those valid actions, intriguingly correspond to either anatomical or effect based imitation behavior.

To clarify the last bullet above it would be useful to consider an example behavior observed in our simulations. Given an image that shows the snapshot of another robot on the other side of the table pulling the object to itself, our system can generate the sequence of images where its own gripper pulls the object towards itself (anatomical imitation behavior) or pushes the object towards the other side of the table (effect based imitation or goal-emulation behavior) depending on the visual cues available to the robot. In our analysis, we show that the prediction capability of the proposed DMBN system does not simply perform a pixel-based template matching but rather benefits from and relies on the common latent space constructed by using both joint and image modalities. In addition to other interesting results, our experiments clearly show that our system outperforms a recent multimodal variational autoencoder model [14] in reconstructing long-horizon high-dimensional trajectories.

The outline of this paper is as follows: in Section 2, we review the related work, in particular LfD systems as DMBN builds upon one such system and the competing multi-modal action representations. In Section 3, we describe our proposed method in detail. We explain our experiment setup in Section 4 and give experimental results in Section 5. Finally, we give a conclusion in Section 6.

2. Related Work

Imitation learning, or learning from demonstration (LfD) [2], has been a popular research topic in robotic learning [15, 16, 17, 18, 19, 20]. Various LfD methods has been proposed

based on dynamic systems and statistical modeling [21, 22, 23, 24], where the parameters in the environment can be learned with Locally Weighted Regression [25, 26, 27] and Locally Weighted Projection Regression [28]. Gaussian Mixture Models [29, 30] and Hidden Markov Models [31, 32, 33, 34] are also frequently used to learn the motion distributions from multiple demonstrations. More recently, deep neural networks also started to be used in imitation learning in order to make it possible to learn movement primitives from complex high-dimensional data [35, 36, 37, 38]. In our earlier work, we proposed Conditional Neural Movement Primitives (CNMPs) [39] as an end-to-end deep LfD architecture that can learn temporal sensorimotor distributions of complex manipulation skills. The DMBN architecture developed in the current study, builds upon CNMPs by introducing a novel mechanism for modality blending to learn a common latent representation that allows cross-modal temporal prediction with partial information.

Several works studied the emergence of the mirror neuron system (MNS) in the context of multi-modal sensor fusion. Nagai et al. [40] proposed a computational model for the early development of the MNS. In this model, the robot cannot make self-other discrimination in the early stages due to the immature visual system. As the visual system develops, the robot starts to discriminate between itself and others, yet, still retains information regarding early experiences, producing the MNS as a by-product. Noda et al. [41] used time-delay neural networks [42] as autoencoders to fuse multiple modalities and reconstruct the missing ones given others.

Copete et al. [43] also used a similar autoencoder architecture in a predictive learning context so to imagine the action of others. Jung et al. [44] proposed a top-down visual attention system to address the long-term visual prediction problem. In this system, the visual stream is divided into dorsal and ventral streams to decompose the difficulty of the problem into two sub-problems. These two streams are then merged for the visual prediction with the help of an external visuospatial memory which holds long-term visuospatial information. On the other hand, we provide a more holistic approach where there are only different submodules for different modalities. Our experiments show that DMBNs can output very accurate visual signals conditioned only on a single visual frame without any memory module. Among these studies, the learning problems considered in the work of Zambelli et al. [14] is well aligned with our study. They proposed a multimodal variational autoencoder (MVAE) [45, 46] to fuse the sensorimotor information of an iCub humanoid robot for prediction and control. They showed that by training MVAE as a denoising autoencoder [47], MVAE can predict the future sensorimotor states, reconstruct the missing modalities, and imitate based on human action observation. As MVAE is not a recurrent architecture, the temporal information should be explicitly stated in the input. To be concrete, in the training phase, the sensorimotor information at time t and $t+1$ were combined and given as input to the MVAE for reconstruction. Here, some sensorimotor information at time $t+1$ was randomly masked with -2 (as in a denoising autoencoder) in order to train the network to reconstruct the future timestep even if it was partially missing.

In the testing phase for future state predictions, states at $t+1$ were filled with mask values -2. Further steps could be predicted by feeding the output of the MVAE to the input. However, the error at one step cascades in the feedback loop as in RNNs. Therefore, the prediction power decreases as the trajectory horizon increases. This is not the case in our proposed model as DMBNs make temporal prediction in one-shot without requiring feeding back of the output as input. To concretely state, our work differs from the previous works in terms of modality fusion strategy and architecture: (1) we take a stochastic mixture of modalities to force the formation of a more regularized representation, and (2) we learn individual modalities and their mixture as long range dependencies via CNPs [48], which allow arbitrary future and past temporal predictions. These key differences yield not only a more robust and better performing multi-modal action representation system, but also give raise to interesting generalization abilities as shown with the experiments presented in the Results section.

3. Method

In this work, we propose Deep Modality Blending Networks (DMBNs), that can learn and produce sensorimotor signals by forming and exploiting multi-modal representations acquired in a latent space. Assume M = {visual, proprioception, sound, haptic ...} corresponds to sensorimotor signals from multiple modalities that an agent collects through self-observation. The agent interacts with the environment using a variety of actions to leverage the information produced by the embodied interaction of the agent with the environment. In the current implementation, the action and action parameters are sampled from a predefined action repertoire. During every interaction, the sensorimotor values are recorded at each time-step. The multi-sensorimotor interaction data set is defined as I, and the ith interaction is described as $I_i = \{(t, S_t^M)\}_{t=0}^{T}$, where t is time and S_t^M is the sensorimotor state collection for the given time-step. S_t^M consists of multiple sensorimotor data, $S_t^M = [S_t^{visual}, S_t^{joint}, S_t^{sound}, S_t^{h}$ where each member holds the corresponding state values of the sensorimotor modalities for the time-step t. Figure 2 shows the architecture of our model where the modalities in the system correspond to the *visual* and *proprioceptive* domains. These two domains are chosen specifically in order to show that our system can learn in an end-to-end fashion with both high (image) and low (joint) dimensional data and make more accurate target predictions on a long horizon compared to the sequential prediction models. In theory, all types of sensorimotor data can be included in the system with our formulation.

The aim of DMBN is to predict a conditional output distribution for a target query given a desired set of observation samples. At the beginning of each training iteration, an interaction I_d is selected randomly from the data set I. From this selected interaction, n data points of (t, S_t^M), are randomly sampled as observations. Here, n is a changing number for each training iteration that is bounded by $[1, obs_{max}]$ where obs_{max} is a hyper-parameter that decides the maximum number of sampled observations in the training. We define this sampled observation set as $O^M = \{(t_i, S_{t_i}^M)\}_i^{obs_{max}}$ where $(t_i, S_{t_i}^M) \in I_d$. On

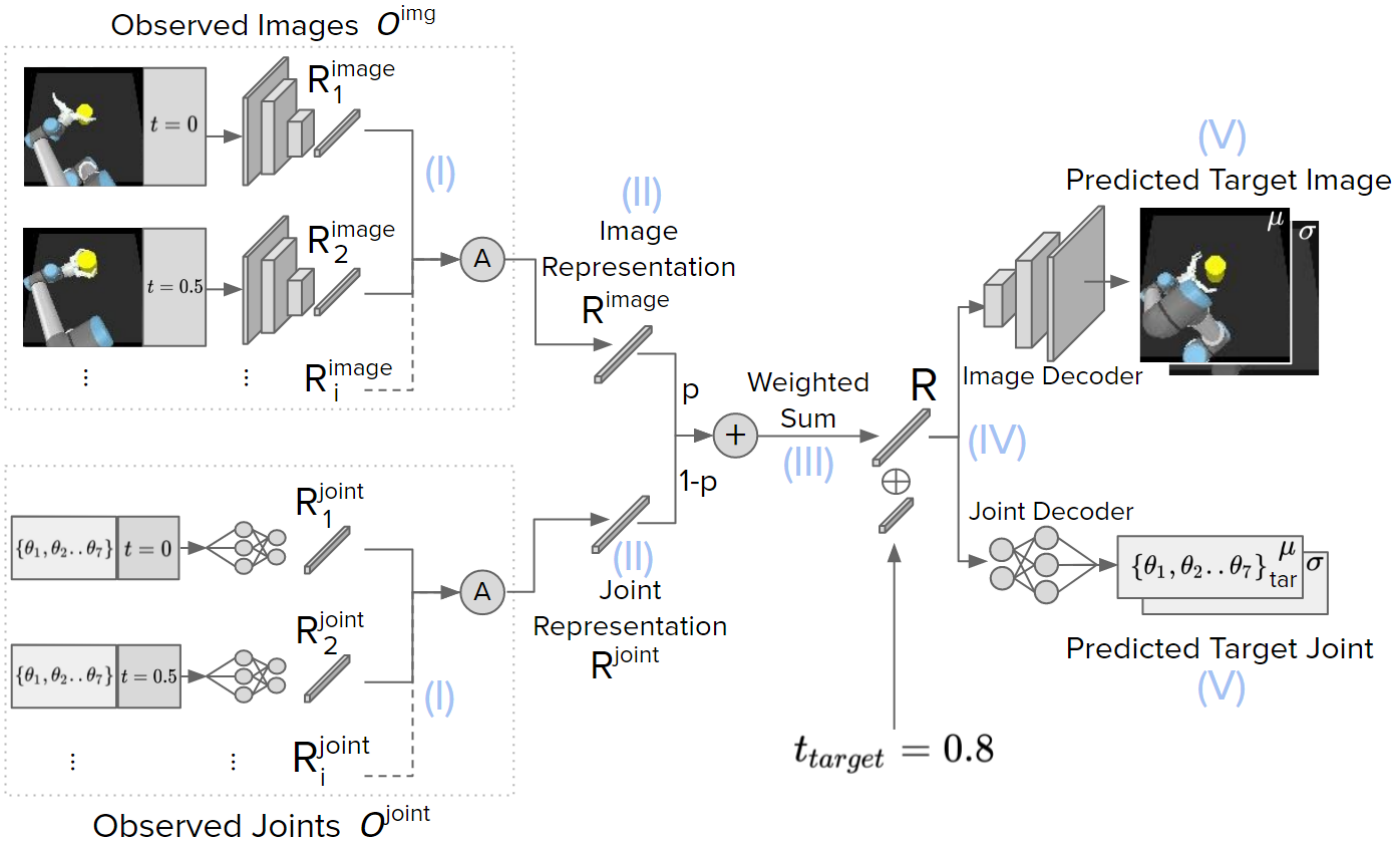

Figure 2: Proposed framework for given *visual* and *joint* modalities. Image and joint observations are turned into their latent representations separately to be used to predict the image and joint positions given at another target time step.

the left side of the Figure 2.(I), example sampled observations O^{image} and O^{joint} are shown for the image and the joint domains. Besides O^M, a target tuple $(t_{target}, S^M_{t_{target}})$ is also sampled from the same selected interaction I_d. The purpose of a training iteration is to learn distributions on t_{target} for all modalities in the system, based on the observation set O^M.

Our aim is to merge the observations of all the modality signals in a single latent space to allow information sharing for a higher quality prediction. In order to achieve this, the observations of each modality, O^m, are first turned into their latent representations R^m_i. For every modality m and every observation, latent states are calculated by the following equation:

$$R^m_i = E^m((t_i, S^m_{t_i}) \mid \theta^m) \qquad (t_i, S^m_{t_i}) \in O^m, m \in M \qquad (1)$$

where E^m is a deep encoder for the modality m with weights θ^m, and R^m_i is the latent states of its ith observation. Figure 2.(I) shows the encoded representations, R^{image}_i and R^{joint}_i, for each observation. After generating these representations, an averaged representation of each modality is calculated by:

$$R^m = \frac{1}{n}\sum_i^n R^m_i \qquad m \in M \qquad (2)$$

where n is the size of the observations of this training iteration. R^{image} and R^{joint} in Figure 2(II) hold general knowledge about their modalities, and our aim is to use these representations in a shared latent space to allow information sharing between all the modalities. To achieve that, a multi-modal general representation R that integrates all the modalities is constructed by calculating a normalized weighted average:

$$R = \frac{\sum_m^M p^m R^m w^m}{\sum_m^M p^m w^m} \qquad (3)$$

where $w^M = [w^{image}, w^{joint}, w^m, ...]$ is a vector representing the *weight* or *availability* of the individual modalities with $0 \leq w^M \leq 1$ and $w^M \neq 0$, which could be used to model cases where one modality is more reliable than the other. On the other hand, modality blending during training is achieved through the random variables $0 \leq p^m \leq 1$ that is sampled at every iteration, and obey the constraint $\sum p^m = 1$. Note that to avoid $\sum_m^M p^m w^m$ ever becoming zero (See Eq 3), we may require $p^m > 0$; but this is not an issue in practice. This follows the same intuition with dropout [49]; randomly dropping modalities forces the model to learn compact representations that can compensate for missing information. Figure 2.(III) shows this process as a two-modality setup where $w^{image} = w^{joint} = 0.5$ and $p^{image} = p$ and $p^{joint} = 1 - p$ where p is sampled uniformly from [0,1]. Note that the dimension of each R^m should be the same in order to perform summation operation between vectors, so in the first place, all the encoders must be designed to produce the latent states with the same dimensions. Once all observations are merged into one general representation, this information can be

used to infer target distributions on t_{target} for all the modalities as:

$$(\mu^m_{t_{target}}, \sigma^m_{t_{target}}) = Q^m((R, t_{target}) \mid \phi^m) \qquad m \in M \qquad (4)$$

where Q^m is a deep decoder network with weights ϕ^m that produces a distribution that consists of a mean $\mu^m_{t_{target}}$ and variance $\sigma^m_{t_{target}}$ for the modality m. Figure 2.(IV) shows the decoders, Q^{image} and Q^{joint}, and predicted distributions, $(\mu^{image}_{t_{target}}, \sigma^{image}_{t_{target}})$ and $(\mu^{joint}_{t_{target}}, \sigma^{joint}_{t_{target}})$, for two domains. The learning objective of our framework is to construct better distributions according to the given observations as in [48] and [39], so the loss term is defined as:

$$\mathcal{L} = -\sum_{m}^{M} logP(S^m_{t_{target}} \mid \mu^m_{t_{target}}, \sigma^m_{t_{target}}) \qquad (5)$$

where $S^m_{t_{target}} \in S^M_{t_{target}}$ is the target sensorimotor value for modality m at time t_{target}.

After training, the system can be requested to make predictions for all the modalities and for all the time-steps by fixing $p^M = 1/M$ and assigning O^M as novel observations. By observing the sensorimotor state at any time-step, any other time point before and after can be queried and predicted using our framework. According to the situation, if a sensorimotor modality does not seem to provide reliable signals, the weight given to that modality can be decreased by configuring availability vector w. Note that, the system can even predict missing modalities if the corresponding w^m is set to zero because of a lack of the modality. Our framework can use the shared latent space for multi-modal predictions. This also enables our framework to imitate other agents by observing their actions with, for example, vision and sound, and producing the agent own behaviour by predicting the corresponding motor signals.

4. Experiment Setup

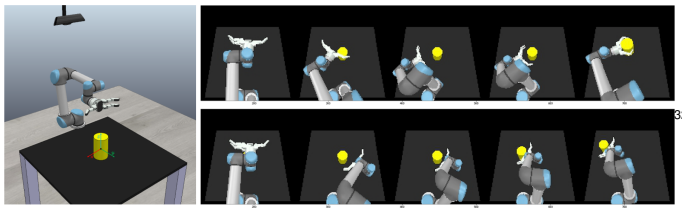

Figure 3: (Left) Experiment setup with vision sensor , UR5, and the object at the middle of the table. (Right) Example grasping and pushing actions recorded via the vision sensor.

To demonstrate the capabilities of our system, we designed an experiment where the actions of the robot can be predicted from the visual and proprioceptual observations at the beginning of the movement execution. A simulated environment was built using CoppeliaSim [50]. The setup consisted of a UR5 robot equipped with a three-finger gripper, a vision sensor, and an object on a table to be manipulated by the robot (Fig. 3 left). The action repertoire of the robot was composed of parameterized push and grasp actions that allow reaching to the object from all directions, and the data collection protocol for each action execution (interaction) was as follows. At the beginning of each interaction, the robot initialized its wide-open hand at an initial position, and an object appeared in the middle of the table (Fig. 3 right). If the selected action was push, a random pushing angle was sampled and the robot pushed the object from this angle to a predetermined fixed distance of 30cm while keeping the hand open. If the selected action was grasp, a random grasping angle was sampled and the robot started to close its hand while approaching to the object so as to grasp it and lift it to a fixed height over the table (30cm). The collected data consisted of two modalities that are *proprioception* and *vision*. The proprioceptive signals were composed of seven joint angles of the robot (6 joints of the UR5 robot and 1 hand opening joint), whereas the visual signals were 128 x 128 x 3 RGB images. Visual signals were collected via the vision sensor that was placed to the point of view of the robot (see Figure 3). In the end, 50 successful push and grasp interactions (100 in total) were collected using the simulator. The interactions were separated into train and test set with 80% and 20% ratios respectively.

5. Experimental Results

We conducted a set of experiments to test the capabilities of DMBN from different aspects. First, in Section 5.1, we verify the prediction capabilities of DMBN by generating complete image and joint trajectories conditioned only on single images. In Section 5.2, the performance of DMBN is compared with MVAE and multi-step errors made by these models are analyzed in Section 5.3. In Section 5.4, we show how the latent space of two modalities indeed blend with each other. In Section 5.5, we analyze the behavior of our model when conditioned with images from different perspective and whether it can serve as a mirror neuron system in replicating observations from different agents. We analyze whether such generalization is due to the inductive bias of the model by making two different ablation studies in Sections 5.6 and 5.7, together which lend support to the the idea that mirror neuron formation can be mediated by self-observation and modality blending with DMBN. Lastly, we test the generalization of the model by conditioning on out-of-distribution samples and include the results in Appendix A.

5.1. Long-term Prediction with Vision only

In this experiment, we verify whether our system can produce visual sequences and the corresponding joint values given a single image as an input. Note that since we take an average of latent vectors for conditioned points, we might as well give multiple images, instead of a single one, to get a more accurate prediction (see Equation 2). Here, to demonstrate the capabilities of our system even in such a scenario where the information is minimum, the system is fed with a single visual observation which is obtained just before the robot interacts with the object. The availability vector is set to one for visual modality and to zero for proprioceptive modality since the observation

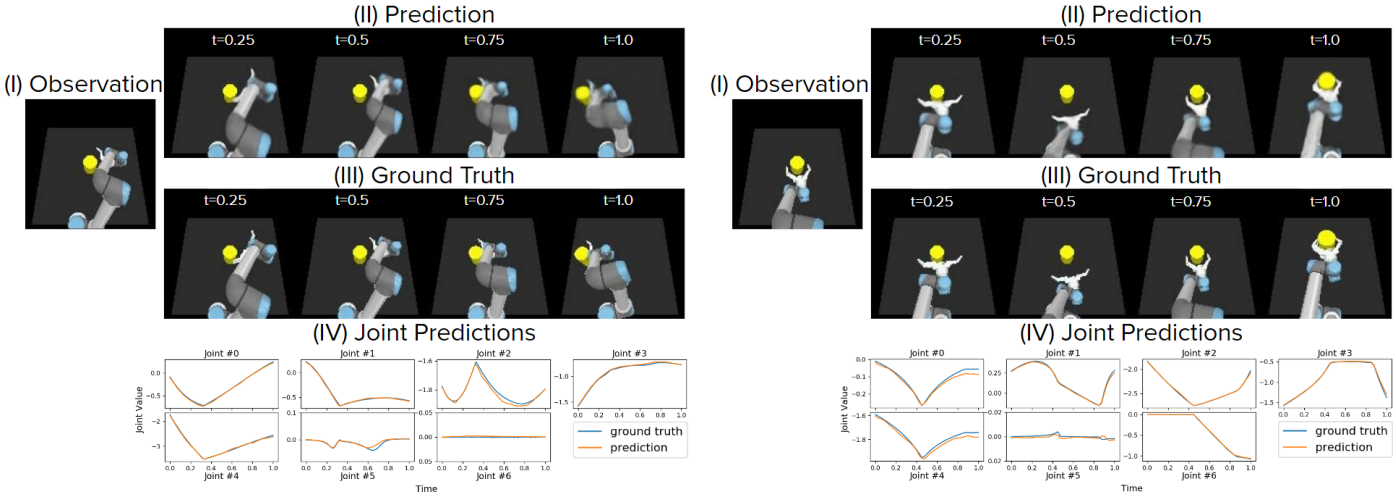

Figure 4: (I) Images that are used as observations. (II) DMBN visual predictions for the given time-steps. (III) Ground truth images for the given time-steps. (IV) DMBN 7D joint predictions for the whole action

only includes visual information. Then, the system is requested to produce visual and motor signals from the beginning to the end of the movement.

Figure 4 shows two examples of pushing and grasping actions at the left and the right of the figure respectively. Figure 4.(I) shows the obtained images that are used as observations from the test set. Figure 4.(II-III) shows the predicted images together with the ground truth at the corresponding time-steps. It can be seen that exploiting the position, orientation, and hand state of the robot extracted from the observed image, our system could successfully predict the sequences of visual and proprioceptive signals from start to finish, which are highly accurate compared to the ground truth values. It is notable that despite the fact that there was no proprioceptive observation in these two examples, our model could make accurate joint predictions from start to the end of the movement by just having access to visual modality (see Figure 4.(IV)). These results indicate that our model can use the representation encoded from an available modality to predict the signals of the other missing modalities. A more detailed quantitative analysis about cross-modality predictions is presented in the next section.

5.2. *Missing Modality Prediction as a Function of Training Set Size*

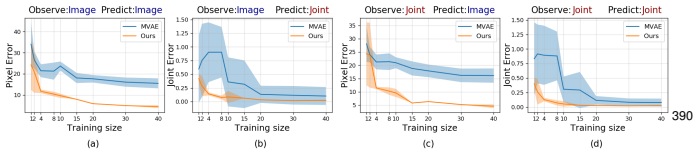

Figure 5: The prediction errors on the test set for different modality input-output pairs with the increasing size of the training data (x-axis).

In this section, we test whether DMBN can indeed perform well when there are some missing modalities. In this experiment, we used the same network in the previous experiment which is trained by using either *visual* or *proprioceptive* modalities. During the test phase, we set the availability of one of the two modalities to zero. We tested whether our system can still predict missing modalities.

We compared our method with MVAE [14] as it can handle missing modalities. We made several modifications to the original MVAE architecture to make a fair comparison. First, we added convolutional layers for the visual input pipeline. All layers in the encoder and the decoder are exactly the same as in DMBN. Therefore, the number of parameters is the same except that MVAE uses an extra fully-connected layer to combine different encoder outputs. This extra layer is not needed in DMBN since the latent representation is shared and acquired via normalized weighted summation. Second, we remove the standard deviation prediction from the decoder as it gave better results in our preliminary experiments. We did not use the KL divergence term in the loss as in [14]. Third, we randomly mask the sensorimotor data at time t and predict the data at $t+1$, in addition to other masking schemes reported in [14]. This additional masking scheme enables us to make full trajectory predictions (both forward and backward prediction) given the observation before contact. Our implementation[1] is based on [14] and their code repository[2].

We report our results in Figure 5 where the prediction accuracies with increasing number of training trajectories are shown. For the two modalities in our experimental setup, we tested four different combinations of modality masking: predicting visual states when either proprioceptive modality (Figure 5.a) or the visual modality (Figure 5.c) is missing, and predicting joint states when either proprioceptive modality (Figure 5.b) or the visual modality (Figure 5.d) is missing. We condition both DMBN and MVAE models with the observations taken from the same time-step that is right before the robot interacts with the object. Both systems predict complete visual and joint trajectories starting from $t = 0$ to $t = T$. Since DMBN is able

[1] https://github.com/alper111/multimodal-vae

[2] https://github.com/ImperialCollegeLondon/Zambelli2019_RAS_multimodal_VAE

to learn from few data, the error and its variation drop quickly even with the small training size, and it improves the accuracy while the data size is increased. For MVAE, the error slightly drops during the data size increase, yet, still far from DMBN. One reason for the error of MVAE is that it feeds the predictions back to itself as input, thus cascades the error in the long horizon. We investigate this phenomenon in the next section in detail.

5.3. Analysis of Long Horizon Predictions

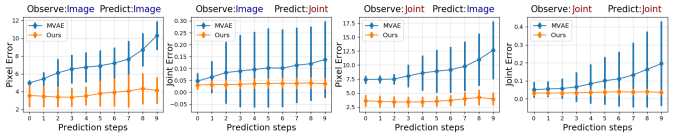

Figure 6: Multi-step prediction results. MVAE errors increase with the prediction steps due to error accumulation. However, our model preserves the error at the same level for increasing prediction steps since it predicts every timestep independent from each other

In this section, we compared the capacity of DMBN on the long horizon predictions with the MVAE method. Both models are trained using the same two modalities in the same way as in the previous section.

In [14], MVAE is used for one step ahead predictions to control the iCub humanoid robot in a closed loop. To make predictions about further time-steps, the model can be fed with its output from the previous time-step. They showed that when trained with sinusoidal data, the prediction accuracy remains the same for about 50 time-steps, and then starts to degrade. In this experiment, we compared the two methods using the data that is collected during the self-exploration which is more complex and high-dimensional. In contrast to MVAE, DMBN does not need to feed its output back to itself as input to make further predictions since we can explicitly query any time-step independently and make predictions on the long horizon directly.

We analyze the error versus the prediction step for two methods in Figure 6. The error of MVAE increases as the prediction step increases since the error is fed back in the input for future time-step predictions. However, the error of DMBN remains around the same because the model does not have a feedback loop to connect an erroneously computed output to its input, and make predictions for every time-step independently just by looking to the observations.

5.4. Multimodal latent space visualization

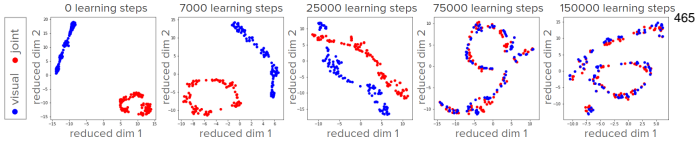

Figure 7: t-SNE visualization of latent space during training. Blue points are visual encodings, and red points are joint encodings.

In this experiment, multi-modal latent space is visualized and analyzed. As mentioned in the previous section, we trained[3] the system with the two modalities that are *visual* and *proprioceptive*. For visualization purposes, the high-dimensional representation space (128 sized vector) is reduced to two dimensions using t-SNE [51] method at different stages of the training. Figure 7 shows the t-SNE visualization of the multimodal latent space at 0, 7k, 25k, 75k, and 150k learning steps from left to right. Blue and red points indicate the samples from the visual and proprioceptive modalities, respectively. Figure 7 shows that although the different modalities are clustered and separated from each other at the beginning of the training (0 and 7k learning steps), they start to share the representations between each other after a while (25k learning steps), and turn into matching/overlapping representations in the later stages of the training (75k and 150k learning steps). Paired blue and red points in the overlapping representation space are analyzed and it is found that each paired blue-red point corresponds to two modalities recorded from the same state of the environment. These results suggest that our system can effectively learn multiple modalities in a common latent space in a way that every sensorimotor modality recorded from the same state of the environment ends up turning into the nearly same representation in the latent space. This allows our system to predict the missing modalities by using the representations produced by other available modalities, which was shown in the Section 5.2.

5.5. Imagining Own Actions by Observing Others: Emergence of Mirror Neuron System Behaviour

Figure 8: Examples of DMBN effect imitation behavior. First row: Observing the other agent just before it pushes away the object. Second row: Observing the other agent just before it grasps the object.

In this experiment, we tested our system to see if it can generate own sensorimotor data by observing another agent perform an action. In order to do that, an agent was placed on the different sides of the table and their performed actions are observed via our agent's visual sensor. Note that in the training data, interactions were only performed and recorded just by our agent, so observing other agents in the test time is a novel information which is completely outside of our training set. Since we were using only visual data as the observation, availability vector is set to one for visual modality and to zero for proprioceptive modality. Because of the fact that the observations are on another agent but the predictions are made for our agent, this prediction process can be considered as the imagination of the action of another agent for the self.

[3] Training details about the network can be found in Appendix B.

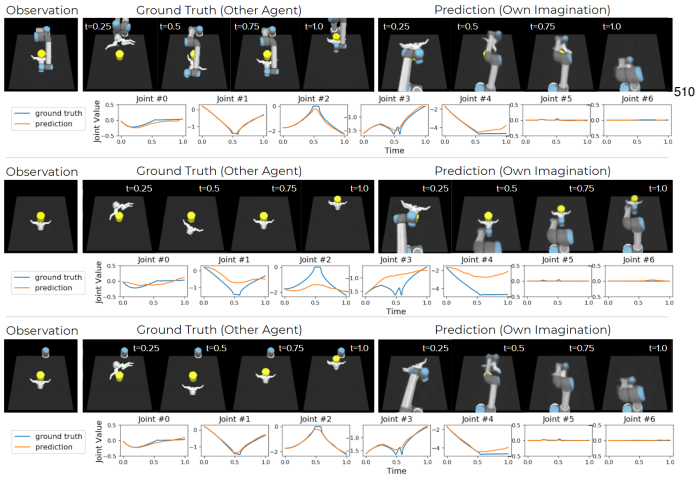

Figure 9: Examples of DMBN egocentric imitation behavior. First row: Emergence of mirror neuron behaviour where the agent observes the other agent pull the object towards itself. Second row: The agent observes a hand without the body. Third row: The agent observes a hand and a base without the arm.

Figure 8 shows the prediction results of our model in two different pushing and grasping scenarios where the observations are shown in Figure 8(a). In the first scenario, the other agent was placed on the opposite side of the table, and in the second scenario, the other agent was placed on the left side of the table. Figure 8(b) shows the visual signals during the other agent performed its action, and Figure 8(c) shows the full trajectory prediction of our system as it imagines the visual signals for itself. As it can be seen in the predictions, our agent is able to generate visual trajectories from its own perspective that matches the approaching angle and the action type in the observation, hence, imagining an action that would be an effect-based imitation of the observed action

However, when we further analyzed our model, we saw that DMBN behaves differently in some specific scenarios. Surprisingly, when the other agent pulls the object towards itself , our agent imagines an action that egocentrically imitates the observed action (Figure 9, first row) and generates motor signals that would also pull the object towards itself rather than creating the effect on the object as shown in the Figure 8. We can say that, in this particular action observation case, an emergent mirror neuron property was exhibited by our DMBN. Interestingly this behavior 'switches' so that the action imagined corresponds to effect-based imitation (i.e. *emulation*) of the observed action when the effector is removed from the interaction (Figure 9, second row). Finally, when the robot is partially revealed by disclosing the base of the robot, the system starts to understand the observed action again as bringing the object toward one's self (Figure 9, third row), thereby showing a mirror neuron response as in the first row of Figure 9.

These results show that when conditioned with the visual signals of other agents, DMBN has potential to produce output signals similar to that of a mirror neuron system. However, signals generated can correspond to either effect-based or egocentric imitation depending on the specific visual signals available from the other agent and the environment. Therefore, it is viable to modulate the behavior of DMBNs via other cognitive mechanisms, e.g. attention, to purposefully control the operation of the model.

5.6. *Template Matching According to Pixel and Latent Space Distances*

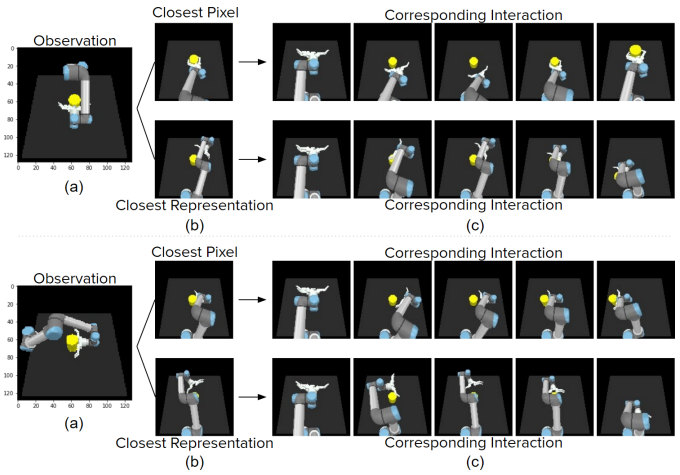

Figure 10: Closest Pixel: Pixel Space Distance; Closes Representation: Latent Space Distance

In this experiment, we aimed to see whether the mirror neuron emergence in our system was due to the rich representations constructed in the latent space during the learning, or it could be simply explained by a straightforward image based template matching. For this, first, two test cases in which true mirror response (i.e. action representation that would yield egocentric imitation) was observed was selected. Figure 10.(a) shows these two observation cases where the agents are placed on the opposite and the left side of the table, respectively. For each test case, the closest image in the training set that gives the minimum average pixel error, and the corresponding image of the closest representation that gives the minimum MSE error in the representation latent space are found and compared. Figure 10.(b) shows the corresponding closest pixel and representation images for each case respectively. Figure 10.(c) also shows the corresponding full trajectory interactions of the found results from the training set.

Results of the both examples show that the corresponding interactions of the closest pixel images do not exhibit true mirror response (i.e. the predicted signals would not yield an egocentric imitation when executed on the robot). On the other hand, the corresponding interactions of the closest latent space representations show true mirror response. These results suggest that the output signals that DMBN produces are not based on a simple image based error minimization but on rich representations that are learned during the multi-modal training with modality blending. The contribution of the deep modality blending to the mirror neuron emergence is further inspected in detail in the next section.

True Mirror Response		Success	Fail
Image + Joint Model	Case 1	10	0
	Case 2	10	0
Only Image Model	Case 1	6	4
	Case 2	4	6

Table 1: Test results of the two models in ten different training sessions with two action observation cases (see Figure 10) of demonstrating agent positioned across (Case 1) and the left side of the agent (Case 2) . Success: The model produced a signal output that corresponds to true mirror response; i.e. the execution of the action based on those signals would yield egocentric imitation. Fail: the model produced disturbed image signals

5.7. Analysis of the Contribution of Multimodal Learning to Mirror Neuron Emergence

In this experiment, we tested if deep modality blending contributes to the mirror neuron emergence in our system. To do that, a model that only uses visual modality was trained next to our model which was trained by using both visual and proprioceptive modalities. In order to prevent the training biases that can occur because of the initial network weights or sampling seeds, both models were trained 10 times with different different random initializations. After the training, both of the models were tested with two test cases and checked whether the networks produce output signals that correspond to mirror neuron emergence. The two test cases used in this experiment were the same examples as in the Experiment 5.6 where the demonstrating agent were placed at the opposite and the left side of the table (see Figure 10).

Figure 11: Example failing scenarios for the only image model. The images are disturbed and the robot arm is disappearing.

Table 1 shows the results of the two models in ten different training initializations with two test cases. Results indicate that the model that uses deep modality blending (the model with Image + Joint) produces coherent images that corresponds to egocentric imitation in every test case where the model that uses only one modality (Only Image Model) produces disturbed images on the ten test cases out of twenty. Figure 11 shows some example fail cases for the only image model where the image is disturbed or the arm of the robot is disappeared These results suggest that using deep modality blending with visual and proprioceptive modalities contribute to the emergence of mirror neuron behavior.

6. Conclusion

In this work, we proposed Deep Multi-modal Blending Network (DMBN) as a multi-modal action representation system that learns the sensorimotor signals corresponding to the actions, in a robust latent representation allowing temporal cross-modal predictions with limited information. DMBNs can generate complete signal trajectories in any desired modality even with zero information on the desired modality by using other available modalities. The performance of the network surpasses the available multi-modal learning systems due to long-range one-shot prediction capability and its novel stochastic modality blending mechanism.

DMBNs build powerful internal representations that allow surprisingly dynamic extrapolation properties, making it a strong contender as a feature-engineering-free Mirror Neuron System model. To be specif, after learning proprioception and visual signals based on self action observations, when tested with different perspective action observations, it successfully generates valid signals that represents its own actions. Depending on the visual setting, the network either acts a true mirror system matching an observed act to its own repertoire in an egocentric way, or acts as an effect-based action matching system. Thus, the network has potential to sustain egocentric and effect-based action recognition and imitation capabilities when envisioned in the cognitive system of an artificial or biological agent.

In this vein, future work should focus on developing biologically plausible and developmentally realistic end-to-end mirror neuron systems that learn along with sensorimotor skill acquisition. In the current study, we used a fixed action repertoire to systematically study the properties of DMBNs; yet in a developing artificial or biological cognitive agent, mirror neuron formation and action learning should go in parallel creating potentially non-trivial interactions worth studying. Another direction that should be pursued is to use the basic imitation capacity acquired by the model, to construct *novel imitation* capacity, where the parts of an observed novel act can be understood in terms of and matched to the existing action repertoire of the agent with the help of DMBN implementing the developing mirror neuron system. We believe that work around these directions will not only stimulate computational study of mirror neurons as a full end-to-end system but also form a framework for lifelong sensorimotor learning for social robots.

Acknowledgement

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731761, IMAGINE; was partially supported by JST CREST "Cognitive Mirroring" under grant no. JPMJCR16E2, by the International Joint Research Promotion Program of Osaka University under the project "Developmentally and biologically realistic modeling of perspective invariant action understanding" and by the Turkish Directorate of Strategy and Budget under the TAM Project number 2007K12-873. The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

References

[1] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[2] B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and autonomous systems 57 (5) (2009) 469–483.

[3] G. Di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, G. Rizzolatti, Understanding motor events: A neurophysiological study, Experimental Brain Research.

[4] G. Rizzolatti, L. Fadiga, V. Gallese, L. Fogassi, Premotor cortex and the recognition of motor actions, Cognitive Brain Research 3 (2) (1996) 131–141.

[5] E. Kohler, C. Keysers, M. A. Umilta, L. Fogassi, V. Gallese, G. Rizzolatti, Hearing sounds, understanding actions: action representation in mirror neurons, Science 297 (5582) (2002) 846–8.

[6] C. Keysers, E. Kohler, M. A. Umilta, L. Nanetti, L. Fogassi, V. Gallese, Audiovisual mirror neurons and action recognition, Exp Brain Res 153 (4) (2003) 628–36.

[7] E. Oztop, M. Kawato, M. A. Arbib, Mirror neurons: Functions, mechanisms and models, Neuroscience Letters 540 (2013) 43–55.

[8] E. Oztop, M. A. Arbib, Schema design and implementation of the grasp-related mirror neuron system, Biological Cybernetics 87 (2002) 116–140.

[9] J. Bonaiuto, E. Rosta, M. Arbib, Extending the mirror neuron system model, i - audible actions and invisible grasps, Biological Cybernetics 96 (1) (2007) 9–38.

[10] J. Bonaiuto, M. A. Arbib, Extending the mirror neuron system model, ii: what did i just do? a new role for mirror neurons, Biological Cybernetics 102 (4) (2010) 341–59.

[11] J. L. Copete, Y. Nagai, M. Asada, Motor development facilitates the prediction of others' actions through sensorimotor predictive learning, in: 2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016, pp. 223–229.

[12] J. Tani, M. Ito, Y. Sugita, Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb, Neural Networks 17 (8-9) (2004) 1273–89.

[13] Y. Demiris, M. Johnson, Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning, Connection Science 15 (4).

[14] M. Zambelli, A. Cully, Y. Demiris, Multimodal representation models for prediction and control from partial information, Robotics and Autonomous Systems 123 (2020) 103312.

[15] P. Pastor, L. Righetti, M. Kalakrishnan, S. Schaal, Online movement adaptation on previous sensor experiences, in: IROS, 2011.

[16] A. Paraschos, C. Daniel, J. Peters, G. Neumann, Using probabilistic movement primitives in robotics, Autonomous Robots 42. doi:10.1007/s10514-017-9648-7.

[17] T. Asfour, P. Azad, F. Gyarfas, R. Dillmann, Imitation learning of dual-arm manipulation tasks in humanoid robots, International Journal of Humanoid Robotics.

[18] P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, Learning and generalization of motor skills by learning from demonstration, in: ICRA, 763–768, 2009.

[19] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, J. Peters, Generalization of human grasping for multi-fingered robot hands, in: IROS, 2012.

[20] M. Mühlig, M. Gienger, J. J. Steil, Interactive imitation learning of object movement skills, Autonomous Robots.

[21] S. Schaal, Dynamic movement primitives-a framework for motor control in humans and humanoid robotics, in: Adaptive Motion of Animals and Machines, Springer, 2006, pp. 261–280.

[22] Y. Zhou, T. Asfour, Task-oriented generalization of dynamic movement primitive, in: IROS, 2017, pp. 3202–3209.

[23] S. Calinon, A tutorial on task-parameterized movement learning and retrieval, Intelligent Service Robotics.

[24] Y. Huang, L. Rozo, J. Silvério, D. Caldwell, Kernelized movement primitives, The International Journal of Robotics Research 38 (2019) 833–852. doi:10.1177/0278364919846363.

[25] C. G. Atkeson, A. W. Moore, S. Schaal, Locally weighted learning for control, in: Lazy learning, Springer, 1997, pp. 75–113.

[26] A. Ude, A. Gams, T. Asfour, J. Morimoto, Task-specific generalization of discrete and periodic dynamic movement primitives, IEEE Transactions on Robotics 26 (5) (2010) 800–815.

[27] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, A. Ude, Generalization of orientation trajectories and force-torque profiles for robotic assembly, Robot. Auton. Syst. 98 (C).

[28] S. Vijayakumar, S. Schaal, Locally weighted projection regression: Incr. real time learning in high dimensional space, in: ICML, 2000, pp. 1079–1086.

[29] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, A. Kheddar, Learning collaborative manipulation tasks by demonstration using a haptic interface, in: Advanced Robotics, 2009.

[30] A. Pervez, D. Lee, Learning task parameterized dynamic movement primitives using mixture of gmms, Intelligent Service Robotics 11 (2018) 61–78.

[31] D. Lee, C. Ott, Incremental kinesthetic teaching of motion primitives using the motion refinement tube, Autonomous Robots 31 (2-3) (2011) 115–131.

[32] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. Martinez Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell, et al., Using robotic exploratory procedures to learn the meaning of haptic adjectives, in: ICRA, 3048–3055, 2013.

[33] H. Girgin, E. Ugur, Associative skill memory models, in: IROS, 2018, pp. 6043–6048.

[34] E. Ugur, H. Girgin, Compliant parametric dynamic movement primitives, RoboticaIn press.

[35] F. Xie, A. Chowdhury, M. C. De Paolis Kaluza, L. Zhao, L. Wong, R. Yu, Deep imitation learning for bimanual robotic manipulation, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 2327–2337.

[36] A. Pervez, Y. Mao, D. Lee, Learning deep movement primitives using convolutional neural networks, in: 2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids), IEEE, 2017, pp. 191–197.

[37] R. Pahič, A. Gams, A. Ude, J. Morimoto, Deep encoder-decoder networks for mapping raw images to dynamic movement primitives, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 5863–5868. doi:10.1109/ICRA.2018.8460954.

[38] A. Droniou, S. Ivaldi, O. Sigaud, Deep unsupervised network for multimodal perception, representation and classification, Robotics and Autonomous Systems 71 (2015) 83–98.

[39] M. Y. Seker, M. Imre, J. Piater, E. Ugur, Conditional neural movement primitives, in: Proceedings of Robotics: Science and Systems, FreiburgimBreisgau, Germany, 2019. doi:10.15607/RSS.2019.XV.071.

[40] Y. Nagai, Y. Kawai, M. Asada, Emergence of mirror neuron system: Immature vision leads to self-other correspondence, in: 2011 IEEE International Conference on Development and Learning (ICDL), Vol. 2, 2011, pp. 1–6. doi:10.1109/DEVLRN.2011.6037335.

[41] K. Noda, H. Arie, Y. Suga, T. Ogata, Multimodal integration learning of robot behavior using deep neural networks, Robotics and Autonomous Systems 62 (6) (2014) 721 – 736.

[42] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang, Phoneme recognition using time-delay neural networks, IEEE transactions on acoustics, speech, and signal processing 37 (3) (1989) 328–339.

[43] J. L. Copete, Y. Nagai, M. Asada, Motor development facilitates the prediction of others' actions through sensorimotor predictive learning, in: 2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016, pp. 223–229. doi:10.1109/DEVLRN.2016.7846823.

[44] M. Jung, T. Matsumoto, J. Tani, Goal-directed behavior under variational predictive coding: Dynamic organization of visual attention and working memory (2019). arXiv:1903.04932.

[45] M. Suzuki, K. Nakayama, Y. Matsuo, Joint multimodal learning with deep generative models, arXiv preprint arXiv:1611.01891.

[46] M. Wu, N. Goodman, Multimodal generative models for scalable weakly-supervised learning, in: Advances in Neural Information Processing Systems, 2018, pp. 5575–5585.

[47] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.

[48] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, S. M. A. Eslami, Conditional neural processes, in: ICML, 2018.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[50] E. Rohmer, S. P. N. Singh, M. Freese, Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework, in: Proc. of The International Conference on Intelligent Robots and Systems (IROS), 2013, www.coppeliarobotics.com.

[51] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (86) (2008) 2579–2605.
URL http://jmlr.org/papers/v9/vandermaaten08a.html

[52] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

Appendix A. Generalization of the System to the Novel Environmental Configurations

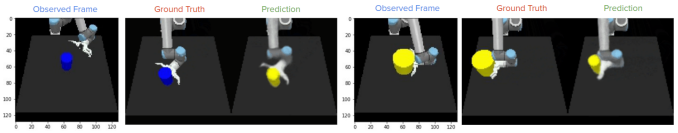

Figure A.12: Generalization performance of the proposed system in two different configurations. Left side: the color of the object is blue. Right side: the size of the object is bigger than the original one.

In this experiment, we tested our system with different novel environmental configurations that have different properties than the training set. Figure A.12 shows the generalization performances of the two different configurations. Left side of the figure shows a scenario in which the color of the object was different from the object in the training data, and the right side shows a configuration where the size of the object was changed. Despite not seeing a big or blue object in the training, our system could successfully predict the correct approaching angle and the action using the observed image in both configurations. It can be seen that the color and the size of the objects are predicted as in the configuration in the training images. This is expected since the only configuration for object in the training scene was yellow and small. Even though the object in the observed image was not the same with the training object, our system could use the knowledge that is learned in the training data to predict a correct output in its own configurations that satisfies the given observation.

Appendix B. Network Architecture and Training Details of DMBN

In this section, the network architecture and training configurations of DMBN are shown. Table B.2 and Table B.3 show the image and joint encoder architectures respectively. Table B.4 and Table B.5 show the image and joint decoder architectures respectively. DMBN is trained with Adam optimizer [52] for one million iterations with a batch size of one and a learning rate of 0.0001. We set obs_{max} to 5.

Appendix C. Network Architecture and Training Details of MVAE

In this section, the network architecture and training configurations of MVAE are shown. Table C.6 and C.7 show the image and the joint encoder architectures respectively. Table C.8 shows the shared encoder-decoder architecture. Table C.9 and C.10 show the image and joint decoder architectures respectively. MVAE is trained with Adam optimizer [52] for 200 epochs with a batch size of 128 and a learning rate of 0.001.

Layer	Input size	Output size
Conv3x3 + ReLU + MaxPool2x2	(4, 128, 128)	(32, 64, 64)
Conv3x3 + ReLU + MaxPool2x2	(32, 64, 64)	(64, 32, 32)
Conv3x3 + ReLU + MaxPool2x2	(64, 32, 32)	(64, 16, 16)
Conv3x3 + ReLU + MaxPool2x2	(64, 16, 16)	(128, 8, 8)
Conv3x3 + ReLU + MaxPool2x2	(128, 8, 8)	(128, 4, 4)
Conv3x3 + ReLU + MaxPool2x2	(128, 4, 4)	(256, 2, 2)
Flatten	(256,2,2)	1024
Dense	1024	128
Multiply (Image Coefficient)	128 * 128	128

Table B.2: DMBN Image Encoder

Layer	Input size	Output size
Dense + ReLU	8	32
Dense + ReLU	32	64
Dense + ReLU	64	64
Dense + ReLU	64	128
Dense + ReLU	128	128
Dense + ReLU	128	256
Dense + ReLU	256	128
Multiply (Joint Coefficient)	128 * 128	128

Table B.3: DMBN Joint Encoder

Appendix D. t-SNE Visualization of the Latent Space

In this section, the detailed version of the latent space is investigated. Figure D.13 shows the encodings of all of the training trajectories in the latent space.

Layer	Input size	Output size
Add (Image + Joint Representations)	128 + 128	128
Concatenate (Target Layer)	128	129
Dense + ReLU	129	1024
Reshape	1024	(256, 2, 2)
Conv3x3 + ReLU + UpSample2x2	(256, 2, 2)	(256, 4, 4)
Conv3x3 + ReLU + UpSample2x2	(256, 4, 4)	(128, 8, 8)
Conv3x3 + ReLU + UpSample2x2	(128, 8, 8)	(128, 16, 16)
Conv3x3 + ReLU + UpSample2x2	(128, 16, 16)	(64, 32, 32)
Conv3x3 + ReLU + UpSample2x2	(64, 32, 32)	(64, 64, 64)
Conv3x3 + ReLU + UpSample2x2	(64, 64, 64)	(32, 128, 128)
Conv3x3 + ReLU	(32, 128, 128)	(16, 128, 128)
Conv3x3 + ReLU	(16, 128, 128)	(8, 128, 128)
Conv3x3 + Sigmoid	(8, 128, 128)	(3, 128, 128)

Table B.4: DMBN Image Decoder

Layer	Input size	Output size
Add (Image + Joint Representations)	128 + 128	128
Concatenate (Target Layer)	128	129
Dense + ReLU	129	1024
Dense + ReLU	1024	512
Dense + ReLU	512	216
Dense + ReLU	216	128
Dense + ReLU	128	32
Dense	32	14

Table B.5: DMBN Joint Decoder

Layer	Input size	Output size
Conv3x3 + ReLU + MaxPool2x2	(6, 128, 128)	(32, 64, 64)
Conv3x3 + ReLU + MaxPool2x2	(32, 64, 64)	(64, 32, 32)
Conv3x3 + ReLU + MaxPool2x2	(64, 32, 32)	(64, 16, 16)
Conv3x3 + ReLU + MaxPool2x2	(64, 16, 16)	(128, 8, 8)
Conv3x3 + ReLU + MaxPool2x2	(128, 8, 8)	(128, 4, 4)
Conv3x3 + ReLU + MaxPool2x2	(128, 4, 4)	(256, 2, 2)
Flatten	(256, 2, 2)	1024
Dense + ReLU	1024	128

Table C.6: MVAE Image encoder

Layer	Input units	Output units
Dense+ReLU	14	32
Dense+ReLU	32	64
Dense+ReLU	64	64
Dense+ReLU	64	128
Dense+ReLU	128	128
Dense+ReLU	128	256
Dense+ReLU	256	128

Table C.7: MVAE Joint encoder

Layer	Input units	Output units
Encoder		
Concatenate (Image+Joint)	128, 128	256
Dense + Tanh	256	128 mean, 128 std
Decoder		
Dense+ReLU	128	256
Slice (for image and joint dec.)	256	128 , 128

Table C.8: MVAE shared encoder-decoder. The activation after the first decoder layer is sliced into two, and each slice is given to a different decoder.

Layer	Input size	Output size
Dense + ReLU	128	1024
Reshape	1024	(256, 2, 2)
Conv3x3 + ReLU + UpSample2x2	(256, 2, 2)	(256, 4, 4)
Conv3x3 + ReLU + UpSample2x2	(256, 4, 4)	(128, 8, 8)
Conv3x3 + ReLU + UpSample2x2	(128, 8, 8)	(128, 16, 16)
Conv3x3 + ReLU + UpSample2x2	(128, 16, 16)	(64, 32, 32)
Conv3x3 + ReLU + UpSample2x2	(64, 32, 32)	(64, 64, 64)
Conv3x3 + ReLU + UpSample2x2	(64, 64, 64)	(32, 128, 128)
Conv3x3 + ReLU	(32, 128, 128)	(16, 128, 128)
Conv3x3 + ReLU	(16, 128, 128)	(12, 128, 128)
Conv3x3	(12, 128, 128)	(12, 128, 128)

Table C.9: MVAE Image Decoder. The last activation is sliced into two (6, 128, 128) shaped tensors for mean and std. See the original implementation [14] for further details.

Layer	Input units	Output units
Dense+ReLU	128	256
Dense+ReLU	256	128
Dense+ReLU	128	128
Dense+ReLU	128	64
Dense+ReLU	64	64
Dense+ReLU	64	32
Dense	32	28

Table C.10: MVAE Joint Decoder. The last activation is sliced into two for mean and std.

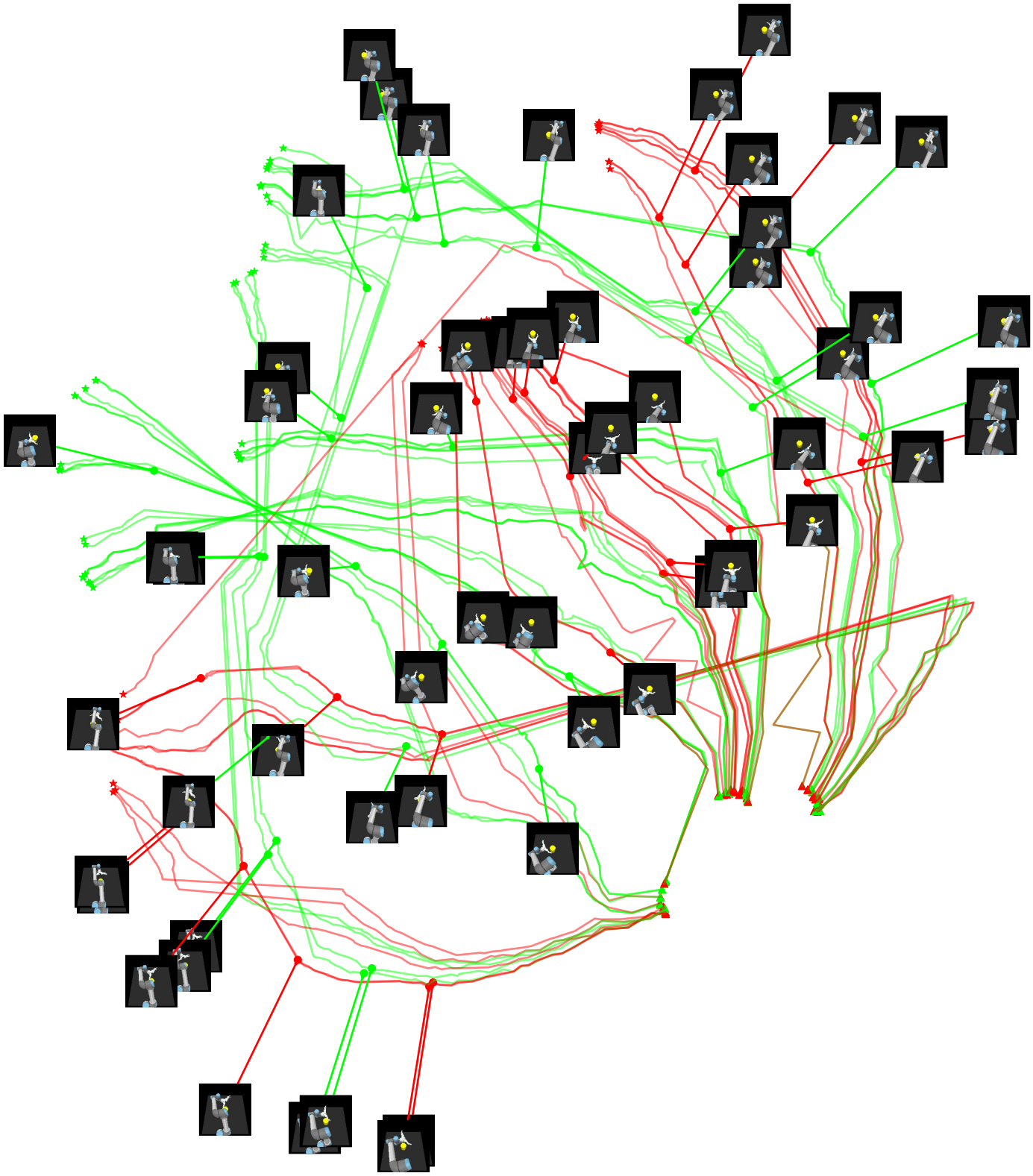

Figure D.13: t-SNE [51] visualization of the encoder output. Here, green and red represents ‘move’ and ‘grasp’ actions, respectively. The initial and the final point of a trajectory is represented with a triangle and a star, respectively.