IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Multimodal Variational DeepMDP: An Efficient Approach for Industrial Assembly in High-mix, Low-Volume Production

Grzegorz Bartyzel[1,2] *Member, IEEE*

***Abstract*—Transferability, along with sample efficiency, is a critical factor for a reinforcement learning (RL) agent's successful application in real-world contact-rich manipulation tasks, such as product assembly. For instance, in the case of the industrial insertion task on high-mix, low-volume (HMLV) production lines, transferability could eliminate the need for machine retooling, thus reducing production line downtimes. In our work, we introduce a method called Multimodal Variational DeepMDP (MVDeepMDP) that demonstrates the ability to generalize to various environmental variations not encountered during training. The key feature of our approach involves learning a multimodal latent dynamic representation. We demonstrate the effectiveness of our method in the context of an electronic parts insertion task, which is challenging for RL agents due to the diverse physical properties of the non-standardized components, as well as simple 3D-printed blocks insertion. Furthermore, we evaluate the transferability of MVDeepMDP and analyze the impact of the balancing mechanism of the *generalized Product-of-Expert*, which is used to combine observable modalities. Finally, we explore the influence of separately processing state modalities of different physical quantities, such as pose and 6D force/torque (F/T) data.**

***Index Terms*—Reinforcement Learning, Representation Learning, Assembly, Industrial Robots, Sensor Fusion**

I. Introduction

In modern manufacturing, two distinct types of production system have emerged: low-mix, high-volume, and high-mix, low-volume. The first is characterized by a linear and highly specialized production system, exemplified by the automotive industry, where dedicated production lines are designed for specific vehicles that have been assembled for extended periods. In contrast, the latter is more prevalent in industries such as electronic manufacturing, where a multitude of devices with lower production volumes, such as specialized devices, are produced. Lines designed for HMLV production must allow quick re-tooling of automated machines, such as robotized stands, to minimize production downtime.

Manuscript received: July, 1, 2024; Revised September, 7, 2024; Accepted October, 21, 2024.

This paper was recommended for publication by Editor J. Kober upon evaluation of the Associate Editor and Reviewers' comments. The research was carried out in collaboration with the company Fitech as part of the project funded by the Polish National Center for Research and Development. Project title: "Intelligent robot for autonomous handling of assembly of electronic components based on artificial intelligence and neural networks" and number: POIR.01.01.01-00-0123/19.

[1]Department of Automatic Control and Robotics, AGH University of Krakow, al. Adama Mickierwicza 30, 30-359 Kraków, Poland gbartyzel@agh.edu.pl

[2]Fitech, ul. Kościelna 5, Sucha Beskidzka, 34-200, Poland

Digital Object Identifier (DOI): see top of this page.

Various methods have been proposed to improve the transferability of robotized machines, including those that take advantage of deep learning systems [1] and reinforcement learning algorithms [2], [3]. However, the majority of approaches demonstrating the ability to generalize across tasks often rely on complex preprocessing, which affects the inference time during the insertion process. The execution time plays a crucial role in the assembly production lines, along with reducing machine retooling time.

In the present work, we introduce a novel RL-based method called Multimodal Variational DeepMDP, which has been designed to address efficient insertion tasks on HMLV production lines. MVDeepMDP has demonstrated notable transferability to various non-standardized electronic parts assembled using through-hole technology (THT), and to custom-designed 3D-printed blocks. We position our method as an extension of the well-known DeepMDP algorithm [4], which improves the performance of RL algorithms by incorporating learning of dynamic representation and reward prediction. A key difference between those algorithms is that we designed the MVDeepMDP to handle multimodal observation spaces effectively.

The principal contributions of our work are as follows:

1) Using *generalized Product-of-Expert* [5] to compute joint latent representation leads to better extraction of task-relevant information from the observations
2) Incorporating the learning of dynamic representation and reward prediction for each observation modality into the process of learning the joint dynamic representation improves the transferability of the RL agent.
3) The independent processing of state-based modalities from various sensors leads to a more informative latent representation compared to the approach of concatenating them and jointly processing them.

II. Related Work

A. Contact-rich Assembly

The manipulation of objects in contact-rich tasks, such as picking up objects, inserting pegs, or grinding, has been a subject of study for many years [6] due to its significance in manufacturing. Traditional manipulation methods rely on proprioception feedback and force control systems [7], [8] that are based on an estimated mathematical model of the robot. However, these solutions are often designed for specific types of objects and require adjustments for new geometries.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Recently, RL methods have been applied to industrial assembly tasks. Initial studies [9], [10], [11] have involved training neural network policies with haptic and proprioception feedback. However, the reliability of proprioception feedback is frequently compromised by real-world disturbances, making it challenging to train an RL agent. Therefore, research on visual [12], [13], [14], [15] and multimodal [2] feedback has been conducted. These approaches produce robust policies that can quickly adapt to disturbances and handle objects not seen previously.

However, the aforementioned works often assume training the RL policy from scratch, limiting their transferability to other tasks or elements. An approach to address this limitation is to improve training by integrating learning from human demonstrations [16], [17], [18] or by transferring a model trained in simulation to the real world [19]. More advanced techniques involve context-learning methods [20], [21] or improving RL policies through weak supervision, such as additional image segmentation [3], [22]. These approaches demonstrate the ability to adapt rapidly or even transfer knowledge to previously unseen tasks.

B. Representation Learning in RL

It is well known that basic RL algorithms suffer from sample-inefficiency and a limited ability to generalize across different environmental variations. To address this challenge, several solutions have been investigated. One group of methods involves observation augmentation [23], [24]. However, applying a highly complex augmentation reduces the stability of training RL agents. In addition, there are algorithms that make use of contrastive learning [25] or weak supervision [26], [27]. Another approach is model-based reinforcement learning, where algorithms learn the latent dynamic representation of the observable environment in conjunction with reward prediction or observation reconstruction [4], [28], [29], [30].

III. METHODS

We propose the Multimodal Variational DeepMDP method to achieve transferability in HMLV production lines, such as the assembly of electronic products. Our solution combines the model-based DeepMDP [4] algorithm with the *generalized Product-of-Experts* [5] to effectively handle multimodal observation spaces. MVDeepMDP can be seamlessly integrated with any standard off-policy RL algorithm by incorporating an auxiliary objective elaborated in Section III-D. In this work, we used Soft Actor-Critic (SAC) [31], a state-of-the-art off-policy RL algorithm designed for continuous action spaces, as our base algorithm.

In this section, we outline the environment with a multimodal observation for assembly tasks. Following this, we provide an overview of the model's architecture and a brief introduction to the SAC. Lastly, we will comprehensively explain the training procedure used for MVDeepMDP.

A. Electronic Parts Insertion Environment

In this work, we examine the transferability of the RL-based method to different objects from the same domain of elements used during training, as well as to objects from an entirely distinct domain. To train the RL agent, based on the environment proposed by Bartyzel et al. [32], we have designed an environment with a multimodal observation space. In the context of the robotic area, multimodality refers to the use of multiple sensors to capture different aspects of the environment. In our case, the multimodal observation space includes following modalities: two RGB images of size 128×128 acquired from cameras mounted on the robot's end-effector, the tool's pose relative to a target reference frame, tool's 6D twist, and 6D F/T data from a sensor attached to the robot's end-effector. Euler's angles represent the orientation; therefore, the pose is six-dimensional.

The industrial robot operates in a cylindrical workspace with a radius 15 mm and a limitless height. In addition, we have restricted the allowable tool rotation to $45°$ to ensure the safety of the tool components. The reference frame of the workspace is the manipulated object insertion pose.

During the insertion trial, the RL agent computes a relative translation along the XYZ-axes $[\Delta x_t, \Delta y_t, \Delta z_t]$ and a rotation along the Z-axis ψ_t^z of the tool center point (TCP). The action space range is $[-2.0, 2.0]$ mm for translation and $[-1°, 1°]$ for rotations. The target pose of the tool is given in the workspace reference frame, transformed into a robot base reference frame, and then transmitted to the control system. To ensure safe execution during training, the industrial robot is controlled using the Cartesian admittance control system [33].

For the experiments, we used a reward function and an episode termination logic proposed in the work [32]. We found this function to be suitable for the problem addressed in this work.

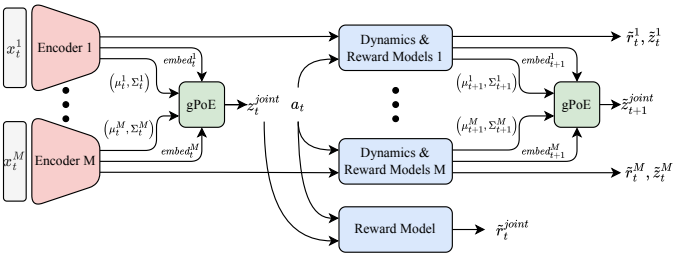


Fig. 1: Overview of the MVDeepMDP model architecture. For each observation modality, a latent representation is computed. The obtained latent representations are then processed by a gPoE to obtain a joint multimodal latent representation. This joint representation is used as input for the actor and critic networks.

B. Model Architecture

An overview of MVDeepMDP integrated with the off-policy actor-critic algorithm is depicted in Fig. 1. In this framework, each modality is preprocessed by an independent stochastic encoder [34] that computes the mean and standard deviation of the Gaussian distribution, from which a 128-dimensional latent representation is sampled. Depending on the type of observation modality, the encoder is a fully connected network with two dense layers of size 512 or a four-layer convolutional

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

neural network (CNN) with a filter size of $\{32, 64, 128, 256\}$ followed by spatial softmax [12]. The transition models and the shared reward decoder follow the design of the encoder models for state-based modalities. For each neural network in MVDeepMDP, we apply LayerNorm, followed by the SiLU activation function.

To compute a joint multimodal latent representation from the M modalities $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$, existing solutions [35] use a *Product-of-Experts* (PoE). In this approach, the modalities are equally weighted. The formula for computing the PoE is as follows:

$$P(\mathbf{z}) = \frac{1}{Z} \prod_{i=0}^{M} p\left(\mathbf{z}|\mathbf{x}_i\right) \tag{1}$$

Hinton et al. [36] pointed out that training the model that utilizes PoE by maximizing the likelihood is challenging due to the renormalization term Z. However, assuming a special case of Gaussian experts $p_i\left(\mathbf{z}|\mathbf{x}_i\right) = \mathcal{N}\left(\mu_i(\mathbf{x}_i)|\Sigma_i(\mathbf{x}_i)\right)$, the output distribution still remains Gaussian, with mean and variance defined as follows:

$$\mu_{PoE}(\mathbf{z}) = \Big(\sum_i \mu_i(\mathbf{z}) T_i(\mathbf{z})\Big) \Big(\sum_i T_i(\mathbf{z})\Big)^{-1} \tag{2}$$

$$\Sigma_{PoE}(\mathbf{z}) = \Big(\sum_i T_i(\mathbf{z})\Big)^{-1} \tag{3}$$

where $T_i(\mathbf{z}) = \Sigma_i(\mathbf{z})^{-1}$ is the precision of the i-th Gaussian expert in $\mathbf{z}$. However, the PoE approach tends to result in the output distribution being influenced more by highly confident experts than less confident ones, potentially impacting the model's ability to generalize. To address this limitation, we propose using the *generalized Product-of-Expert* [5], which introduces a weighted mechanism to balance the confidence of each expert:

$$P(\mathbf{z}) = \frac{1}{Z} \prod_i p^{\lambda_i(\mathbf{z})}\left(\mathbf{z}\right) \tag{4}$$

As previously, assuming the Gaussian expert, we define the gPoE mean and the variance as:

$$\mu_{gPoE}(\mathbf{z}) = \Big(\sum_i \mu_i(\mathbf{z}) \lambda_i(\mathbf{z}) T_i(\mathbf{z})\Big) \Big(\sum_i \lambda_i(\mathbf{z}) T_i(\mathbf{z})\Big)^{-1} \tag{5}$$

$$\Sigma_{gPoE}(\mathbf{z}) = \Big(\sum_i \lambda_i(\mathbf{z}) T_i(\mathbf{z})\Big)^{-1} \tag{6}$$

The weights λ_i are parameterized by the function $f\left(\mathbf{x}_1, ..., \mathbf{x}_M\right)$ based on the input modalities. In our architecture, the feature extraction part of the encoders is shared with the parameterization function $f\left(\mathbf{x}_1, ..., \mathbf{x}_M\right)$. The obtained weights are normalized using the softmax function to distribute the importance across available modalities, ensuring that for each latent dimension $\sum_i \lambda_i = 1$. Having introduced gPoE, we can formulate a joint latent representation as $\mathbf{z}_{joint} = \mathcal{N}\left(\mu_{gPoE}\left(\mathbf{z}_1, ..., \mathbf{z}_M\right), \Sigma_{gPoE}\left(\mathbf{z}_1, ..., \mathbf{z}_M\right)\right)$ or, more generally, $\mathbf{z}_{joint} \sim q\left(\mathbf{z}_{joint}|\mathbf{x}_1, ..., \mathbf{x}_M\right)$.

The computed joint multimodal latent distribution serves as an input observation for actor and critic networks. However, in the proposed framework, the encoder backbone does not propagate the policy's gradient to ensure stable training. For all neural networks of the RL agent, the same model design is used, consisting of two dense layers of size 512 with ReLU activation function.

C. Reinforcement Learning

We model our problem within a standard RL framework [37], where the interaction between the agent and the environment is represented as a Markov Decision Process (MDP). At each discrete time step, the RL agent computes the action $\mathbf{a}_t$ based on the observed state $\mathbf{x}_t$. Following the execution of action $\mathbf{a}_t$, the RL agent transitions to a new state $\mathbf{x}_{t+1}$ and receives a scalar reward value r_t that evaluates the quality of the decision made. The goal of such an agent is to maximize the cumulative return $R = \sum_{t=0}^{T} \gamma^t r_t$ throughout the interaction with the environment, where T is the length of the planned trajectory and $\gamma \in (0, 1)$ is a discount factor.

This work incorporates Soft Actor-Critic [31] as an RL algorithm integrated within MVDeepMDP due to its effectiveness in solving continuous action space problems, such as robotic manipulation. As mentioned above, the observed state $\mathbf{x}_t$ is first pre-processed by the encoders. Therefore, the joint latent representation $\mathbf{z}_t^{joint} \sim q_\phi^{gPoE}\left(\mathbf{z}_t \middle| \mathbf{x}_t^1, ..., \mathbf{x}_t^M\right)$ serves as an input to the RL agent. In SAC, neural networks are used to represent the soft Q-function $Q_\theta(\mathbf{z}_t^{joint}, \mathbf{a}_t)$ and the policy $\pi_\theta(\mathbf{a}_t \mid \mathbf{z}_t^{joint})$, where θ is a learnable set of parameters. SAC maximizes entropy to facilitate effective exploration during training. The neural networks' parameters are optimized by given forumlas:

$$\mathcal{J}_Q(\theta) = \mathop{\mathbb{E}}_{\substack{\mathbf{z}_t^{joint} \sim q_\phi, \\ \mathbf{a}_t \sim \mathcal{D}}} \left[\frac{1}{2}\left(Q_\theta\left(\mathbf{z}_t^{joint}, \mathbf{a}_t\right) - y_t\right)^2\right] \tag{7}$$

$$\mathcal{J}_\pi(\theta) = \mathop{\mathbb{E}}_{\substack{\mathbf{z}_t^{joint} \sim q_\phi, \\ \mathbf{a}_t \sim \pi_\theta}} \left[\alpha \log \pi_\theta\left(\mathbf{a}_t \middle| \mathbf{z}_t^{joint}\right) - \min_{i \in \{1,2\}} Q_{\theta_i}\left(\mathbf{z}_t^{joint}, \mathbf{a}_t\right)\right] \tag{8}$$

where $\mathcal{D}$ is an experience replay buffer that stores collected transitions, $y_t = r_t + \left(\min_{i \in \{1,2\}} Q_{\bar{\theta}_i}\left(\mathbf{z}_{t+1}^{joint}, \mathbf{a}_{t+1}\right) - \alpha \log \pi_\theta\left(\mathbf{a}_{t+1} \mid \mathbf{z}_{t+1}^{joint}\right)\right)$ denotes the temporal difference (TD) target. $Q_{\bar{\theta}}$ is a target soft Q-function updated via exponential moving average, and α is an entropy temperature coefficient.

D. Learning Objective

Our proposed method involves learning a latent representation of the multimodal observation space. For each modality, MVDeepMDP computes a separate latent representation $\left\{\mathbf{z}_t^1, ..., \mathbf{z}_t^M\right\}$ and predicts the transition in the latent space $\left\{\widetilde{\mathbf{z}}_{t+1}^1, ..., \widetilde{\mathbf{z}}_{t+1}^M\right\}$ and the reward r_t. Additionally, we sample the joint latent representation from $\mathbf{z}_t^{joint} \sim q_\phi\left(\mathbf{z}_t^{joint} \middle| \mathbf{x}_t^1, ..., \mathbf{x}_t^M\right)$ and the joint transition $\widetilde{\mathbf{z}}_{t+1}^{joint} \sim p_\phi\left(\mathbf{z}_t^{joint} \middle| \mathbf{z}_{t+1}^1, ..., \mathbf{z}_{t+1}^M; \mathbf{a}_t\right)$. The components of the model, which were thoroughly introduced in Section III-B, are optimized jointly to maximize the variational lower bound (ELBO [38]). In the proposed method, the bound includes

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1: Multimodal Variational DeepMDP

Initialize model parameters $\phi, \theta_\pi, \theta_Q$ with random weights
Initialize replay buffer $\mathcal{D}$
for *iteration* $t = 1, 2, ..., T$ **do**
  Sample latent variables $\mathbf{z}_t^{joint} \sim p_\phi\left(\mathbf{z}_t \middle| \mathbf{x}_t^1, ..., \mathbf{x}_t^M\right)$
  Sample action $\mathbf{a}_t \sim \pi_\theta\left(\mathbf{a}_t \middle| \mathbf{z}_t^{joint}\right)$
  Execute action $\mathbf{a}_t$ and observe reward r_t and next state $\left\{\mathbf{x}_{t+1}^1, ..., \mathbf{x}_{t+1}^M\right\}$
  Store transition $\left(\left\{\mathbf{x}_t^1, ..., \mathbf{x}_t^M\right\}, \mathbf{a}_t, r_t, \left\{\mathbf{x}_{t+1}^1, ..., \mathbf{x}_{t+1}^M\right\}\right)$ in $\mathcal{D}$
  Sample a minibatch of transitions from $\mathcal{D}$
  Update model network using (11)
  Update Q networks using (7)
  Update policy network using (8)
  Update target networks: $\bar{\theta} \leftarrow (1-\tau)\bar{\theta} + \tau\theta$
end

reward reconstruction terms and a KL regularizer for the transition:

$$\begin{aligned}\mathcal{J}_{DYN}(\phi) = \frac{1}{M}\sum_{i=0}^{M} & D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{z}_{t+1}^i \mid \mathbf{x}_{t+1}^i\right) \,\|\, p_\phi\left(\widetilde{\mathbf{z}}_{t+1}^i | \mathbf{z}_t^i, \mathbf{a}_t\right)\right) \\ & + D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{z}_{t+1}^{joint} \middle| \mathbf{x}_{t+1}^1, ..., \mathbf{x}_{t+1}^M\right) \,\middle\|\, p_\phi\left(\widetilde{\mathbf{z}}_{t+1}^{joint} \middle| \mathbf{z}_t^1, ..., \mathbf{z}_t^M; \mathbf{a}_t\right)\right)\end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{J}_{REW}(\phi) = & \frac{1}{M}\sum_{i=0}^{M} \log p_\phi\left(r_t \middle| \mathbf{z}_t^i, \mathbf{a}_t\right) \\ & + \log p_\phi\left(r_t \middle| \mathbf{z}_t^{joint}, \mathbf{a}_t\right)\end{aligned} \tag{10}$$

$$\mathcal{J}_M(\phi) = \mathcal{J}_{REW}(\phi) - \beta\mathcal{J}_{DYN}(\phi) \tag{11}$$

The parameters of the encoders and transition models are updated using stochastic backpropagation [34]. The impact of the KL regulator is controlled by β [39]; in this work, we set β = 1. The optimization procedure for MVDeepMDP is outlined in Algorithm 1.

IV. Experiments

In HMLV production scenarios, such as the assembly of a printed circuit board (PCB), it is crucial that automated machines quickly adapt to new products. Considering robotized solutions that utilize reinforcement learning, such an adjustment can be achieved via quick training from scratch or using a method with high generalizability.

To assess transferability, we designed two test scenarios that reflect potential production use cases. The first scenario involves the RL agent attempting to insert the electronic part used during training into insertion places with a varying PCB layout. The second scenario challenges the RL agents to perform insertion trials on different types of components.

A. Experimental Setup

We conducted experiments on the laboratory stand described in the work [32]. This setup contains the Universal Robot UR5e-series industrial robot, the servoelectric gripper, a 6D F/T sensor, and a custom-made tool with two vision sensors. The vision tool is attached to the robot end effector with cameras oriented at 30° to the tool axis. Therefore, the acquired images are focused on the picked electronic element and the surroundings near the tip of the tool. The captured raw images are of size 1024×512 pixels in RGB format. To reduce the computational cost, we downsampled the images to 128×64 pixels and resized them to 128×128 pixels. The samples of the acquired views are presented in Fig. 2. The parameters of the admittance control system that controls the industrial robot remained unchanged.

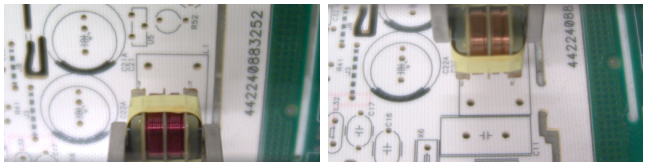


Fig. 2: Views acquired from the vision system attached to the robot's end effector. **Left:** View 1. **Right:** View 2.

Within the robot's workspace, we placed three PCB panels and a test mockup for 3D printed blocks. The selected PCBs varied in layout, laminate color, and hole clearance. The target positions for 3D printed blocks were measured manually, while the target positions for the PCBs were acquired from the CAD models. Those positions were projected onto the robot's workspace using the fiducial markers. To thoroughly examine the generalizability of the RL agent, we chose 7 varieties of electronic parts (Fig. 3a) and 6 types of 3D printed blocks (Fig. 3b). The selected objects varied in terms of their shape, appearance, and, in case of the electronic parts, the number of leads and arrangement of leads. The electronic parts were placed on the 3D printed trays, from where the industrial robot picked them throughout the training. The 3D printed blocks were placed in the test mockup, where the robot was supposed to insert the electronic parts.

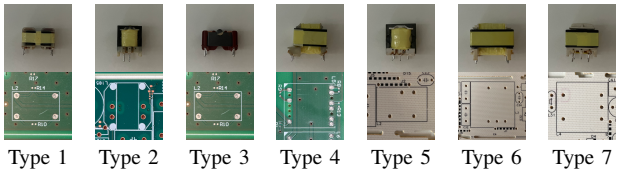
We utilized ROS 2 middleware to control the industrial robot and operate peripheral devices. Additionally, RLlib was employed as the software to manage the learning process and implement the RL agents. All experiments were conducted on a workstation equipped with an AMD Threadripper 1950X CPU, 64 GB of RAM, and a single NVIDIA RTX 4070 GPU.

B. Training Procedure

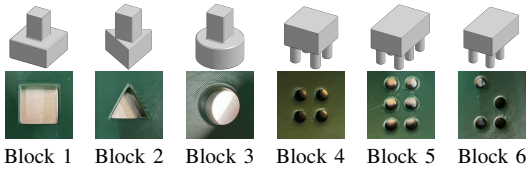
Throughout the training process, the RL agent's task was to insert the electronic part type 1 into the corresponding insertion place. The agent was trained for 50000 steps with AdamW [40] optimizers, with learning rates of $3e-4$ and weight decays of 0.01. Based on Bartyzel et al.'s [32] proposal, we used the Ape-X framework [41] to ensure smooth control of the industrial robot during the training process. This framework allows asynchronous data collection in relation to the update steps. With abovementioned settings, the training process of MVDeepMDP lasted approximately 3 hours, with noticeable convergence observed after 18 minutes.

In each episode, a component was already grasped and positioned close to the insertion place on the PCB. As opposed to the training process presented in the work [32],

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a)



(b)

Fig. 3: Overview of the objects used in the experiments. (a) Electronic parts. (b) 3D printed blocks.

in our experiments, the initial pose of the robot tool was sampled from uniform distributions with higher boundary values: $p^{xy}_{noise} \sim \mathcal{U}(-5, 5)$ mm and $\psi^{z}_{noise} \sim \mathcal{U}(-15°, 15°)$. Furthermore, during the initial training phase in this study, the robot began 2 mm above the surface of the PCB, and this distance gradually increased to 20 mm as training progressed, instead of the starting height remaining constant throughout training.

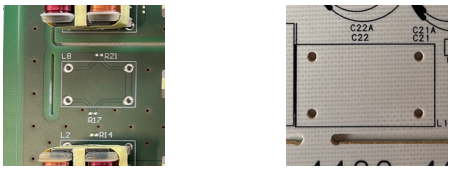
C. First Test Scenario: Robustness

In this experiment, we evaluate whether our method can insert elements that were used during training under disturbance of the background view. The disturbance can be seen as a differing PCB layout or as a partially assembled PCB. This is an important feature from a production perspective, especially in HMLV production lines, where the products are frequently altered. To provide a comparative evaluation, we benchmark our method against the following current best methods in the literature:

SAC [31]: This off-policy actor-critic algorithm serves as a benchmark for state-of-the-art model-free learning methods.
DrQ [26]: This is an extension of SAC that applies a random shift augmentation to visual modalities.
SVEA [24]: This is an extension of DrQ that applies additional random convolution augmentation to visual modalities. The so-called hard augmentation in SVEA is applied only in the objective of the critic.

TABLE I: Evaluation results of robustness against background disturbances.

	Success rate [%]		**Mean time [s]**	
	Layout 2	Layout 3	Layout 2	Layout 3
SAC	4	11	8.94	5.25
DrQ	66	5	4.98	8.93
SVEA	**100**	**100**	**1.77**	1.86
PAD	99	60	2.38	7.89
DeepMDP	**100**	73	2.07	4.50
DBC	**100**	99	1.81	2.61
MVDeepMDP (PoE)	95	5	3.60	9.04
MVDeepMDP (gPoE)	**100**	**100**	1.81	**1.77**



(a) (b)

Fig. 4: The PCB layouts used to evaluate the robustness of the RL agents against background disturbance. (a) **Layout 2**: Layout with already-assembled surrounding elements. (b) **Layout 3**: Layout with different design and hole clearance.

PAD [27]: This is an extension of the actor-critic algorithms that, through self-supervision, adapts to the test environment. The self-supervised task is inverse dynamics prediction.
DeepMDP [4]: This algorithm forms the foundation of our MVDeepMDP. It merges a semi-supervised objective with an RL agent, training a transition model to predict future latent representations and a decoder to predict a reward.
DBC [29]: This algorithm learns an invariant representation based on bisimulation metrics that is directly imposed on the latent space.
MVDeepMDP-PoE: This is an option of our method that utilizes PoE instead of gPoE to compute the joint latent representation of the observed modalities.

For a fair comparison, we used the feature extractor design outlined in Section III-B for all the algorithms.

The trained policies were evaluated for their robustness to changes in the background. For each scenario and policy, we conducted 100 insertion trials. The initial pose at the beginning of each episode was randomly sampled from a uniform distribution as detailed in Section IV-B. Throughout the evaluation, we recorded data on the insertion status (success or failure) and assembly time of the successfully completed trial. Based on these data, we computed the success rate and averaged the assembly time for each test case. The results of this experiment are presented in Table I. It is evident from the results that most of the methods were effective in inserting electronic parts into the target place with the background layout shown in Fig. 4a. This was an expected outcome, as this layout was used during training, although with additional surrounding components. However, when it came to inserting components into the PCB with the layout shown in Fig. 4b, the best performing methods were identified as MVDeepMDP, DBC and SVEA. In this case, not only did the color of the PCB's laminate differ but also the hole's clearance, making it a more challenging task.

D. Second Test Scenario: Transferability

Next, we subjected methods against the second test scenario. This test evaluates the transferability of methods trained with electronic parts type 1 to other types of electronic parts (see Fig. 3a). This experiment is more challenging than the previous one, as the selected test components differ in the number of leads and the arrangement of leads. In addition to the overall physical properties of the elements, it is noteworthy that for each component, there is a specially designed PCB

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II: Evaluation results of transferability to unseen object types from the same domain.

	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7
			Success rate [%]			
SAC	6	7	1	15	10	5
DrQ	2	0	28	18	13	27
SVEA	43	95	84	12	68	64
PAD	1	32	81	18	6	52
DeepMDP	4	93	91	81	29	31
DBC	56	97	80	79	90	21
MVDeepMDP (PoE)	35	74	50	42	51	38
MVDeepMDP (gPoE)	**87**	**100**	**100**	**98**	**99**	**85**
			Mean time [s]			
SAC	8.07	8.71	9.10	8.23	6.34	8.93
DrQ	8.57	9.20	7.38	7.96	8.42	7.56
SVEA	6.43	2.76	3.41	8.41	4.99	5.07
PAD	13.85	10.94	5.09	11.91	13.27	8.17
DeepMDP	9.02	2.54	2.92	4.02	7.43	7.37
DBC	6.04	2.91	3.82	4.31	3.72	8.11
MVDeepMDP (PoE)	7.45	5.10	6.24	6.54	6.57	7.04
MVDeepMDP (gPoE)	**3.59**	**2.15**	**1.83**	**2.79**	**2.13**	**3.55**

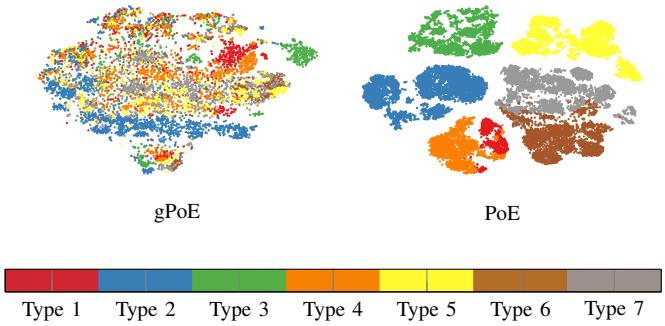


Fig. 5: t-SNE visualization of the learned latent representations for each electronic part.

TABLE III: Transferability to objects from a different domain.

	Block 1	**Block 2**	**Block 3**	**Block 4**	**Block 5**	**Block 6**
Success Rate [%]	100	100	100	100	100	100
Mean Time [s]	1.79	2.42	1.68	1.93	1.89	1.82

layout that can introduce disturbances in policy inference. As indicated previously, the HMLV production line would benefit from solutions that can rapidly adapt to new products, thus minimizing downtime.

Similarly to the first test scenario, we conducted 100 insertion trials for each selected test component and policy and collected the success rates alongside the average assembly times. The results obtained are reported in Table II. It is evident that our method outperformed all the compared algorithms. In four out of the six test cases, the success rate was nearly 100% or even reached 100%, while for the remaining two electronic parts, the success rate was close to 90%. Among the compared algorithms, only SVEA and DBC achieved relatively high results. Furthermore, analysis of the recorded assembly times indicates that our method quickly identified the target position during the trials, allowing it to produce a correct sequence of commands.

From the results obtained, it is evident that learning dynamic representation or incorporating additional hard augmentations during training significantly improves the transferability to other types of electronic parts. While soft augmentations such as shift improve the algorithm's sample efficiency, they do not improve its generalization capability. In addition, an intriguing finding is the performance of the PAD algorithm, which performs adaptation during deployment. However, the metrics indicate that learning the inverse dynamics alone is not sufficient for effective adaptation in the insertion task.

E. Analysis of the gPoE Balancing Effect

To assess the impact of the gPoE balancing mechanism on generalizability, we utilized t-distributed Stochastic Neighbor Embedding (t-SNE) [42] to clusterize the multimodal latent representation of MVDeepMDP and its PoE counterpart. t-SNE is a method that transforms high-dimensional Euclidean distances into conditional probabilities, capturing similarities between adjacent data points. We categorized the clusters according to the electronic parts described in Fig. 3a. The resulting visualization is shown in Fig. 5. We immediately observed that the clusters generated from the multimodal latent representations obtained by the PoE represent electronic parts. Upon closer inspection, we noticed that certain clusters overlap, indicating potential shared representations. However, in the case of the multimodal latent representation acquired from the gPoE, the t-SNE output resembled a single cluster without a distinct separation of the components. Therefore, this observation led us to conclude that the weight-balancing mechanism improves the capacity to learn task-relevant features, a critical factor in achieving generalization.

F. Transferability to Other Domains

In previous experiments, MVDeepMDP demonstrated its effectiveness in transferring the learned policy to new objects within the same domain, which is particularly useful for HMLV production lines where elements of the same type are frequently changed. However, there are cases where automated machines on production lines may need to handle objects from entirely different domains. To evaluate MVDeepMDP's transferability to such scenarios, we conducted an additional experiment. In this experiment, the RL agent trained to insert electronic part type 1 was tested by inserting 3D-printed blocks. The results of this experiment are presented in Table III. Our method successfully inserted all the blocks, with a mean assembly time ranging from 1.68 s to 2.42 s. It is worth noting that the clearance of the insertion place for the 3D-printed blocks was larger than that for the electronic parts. Nonetheless, the results indicate that MVDeepMDP could be effectively adapted to the new domain, demonstrating its high transferability.

G. Impact of Independently Processing State-based Modalities

In the context of training RL agents for robotic tasks, it is a common approach to concatenate state-based modalities, such as pose and 6D F/T data, into a unified vec-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

tor and preprocess them using a single feed-forward neural network. However, due to the disparate nature of such modalities in representing physical quantities, some features may not be effectively captured during training. To overcome this challenge, Haffner et al. [30] suggested preprocessing the concatenated state-based modalities with the function $\text{symlog}(x) \doteq \text{sign}(x) \ln(|x| + 1)$ before inputting them into the feed-forward neural network. This function is designed to scale down the magnitude of physical quantities that are represented by large values, such as force data.

In our final experiment, we examined the impact of processing state-based modalities separately on the performance of MVDeepMDP. The MVDeepMDP model was trained with the following variations: (1) separate processing of all modalities, (2) concatenation and joint processing of state-based modalities, and (3) concatenation and joint processing of state-based modalities with the *symlog*. The results are illustrated in Fig. 6. Based on the training results, we can observe that the MVDeepMDP trained with independently processed state-based modalities converged in approximately 6000 steps, while the other two variations required more than 10000 steps to achieve similar performance. Subsequently, we assessed the trained models using the test scenarios outlined in Section IV-D. It is evident that variants utilizing concatenated state-based modalities achieved significantly lower success rates compared to the proposed approach. This indicates that independent preprocessing of state-based modalities of different physical quantities is advantageous in extracting meaningful features, thereby improving the performance and transferability of the RL Agent.

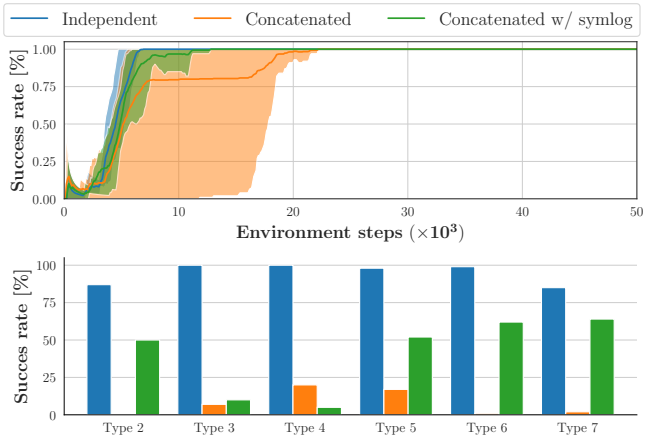


Fig. 6: Comparison of methods for processing state modalities of different physical quantities. **Top:** Training results. **Bottom:** Test results from the second test scenario.

V. Conclusions

In this work, we introduced a novel approach called Multimodal Variational DeepMDP, which is specifically designed to enhance the transferability of RL agents in HMLV production lines. The proposed method can be viewed as an extension of DeepMDP for multimodal observation. This model is optimized through learning multimodal latent dynamic representation and predicting rewards. We employed gPoE to efficiently compute the joint multimodal latent representation, which then served as input to the RL agent. A series of experiments demonstrated the effectiveness of MVDeepMDP in transferring the learned policy to new objects from the same domain as the training objects, as well as to objects from entirely different domains. The results showed that MVDeepMDP significantly outperformed state-of-the-art methods in terms of its generalization capacity. Furthermore, we thoroughly investigated the impact of the balancing mechanism of the gPoE mixing technique and the importance of separately preprocessing state-based observations. The results clearly showed that the proposed improvements in our work significantly enhance the generalization capacity of RL agents.

For future work, we plan to assess our solution in a prototype stand that is integrated into the production line. This will allow us to collect feedback for potential improvements. Additionally, we would like to expand MVDeepMDP to other contact-rich tasks, such as object grasping or robotized soldering. An interesting area for further research would involve exploring the effects of additional objectives aimed at maximizing mutual information between modalities or integrating with recurrence networks.

References

[1] O. Spector, V. Tchuiev, and D. D. Castro, "InsertionNet 2.0: Minimal Contact Multi-Step Insertion Using Multimodal Multiview Sensory Input," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6330–6336.

[2] M. Lee, Zhu, Y., P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[3] L. Xie, H. Yu, Y. Zhao, H. Zhang, Z. Zhou, M. Wang, Y. Wang, and R. Xiong, "Learning to Fill the Seam by Vision: Sub-millimeter Peg-in-hole on Unseen Shapes in Real World," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2982–2988.

[4] C. Gelada and S. Kumar and J. Buckman and O. Nachum and M.G. Bellemare, "DeepMDP: Learning Continuous Latent Space Models for Representation Learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2170–2179.

[5] Y. Cao and D.J. Fleet, "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," *CoRR*, vol. abs/1410.7827, 2014.

[6] O. Kroemer, S. Niekum, and G. Konidaris, "A Review of Robot Learning for Manipulation: Challenges and Representations and and Algorithms," *Journal of Machine Learning Research*, vol. 22, no. 30, pp. 1–82, 2021.

[7] Y. Fei and X. Zhao, "An Assembly Process Modeling and Analysis for Robotic Multiple Peg-in-hole," *Journal of Intelligent and Robotic Systems*, vol. 36, no. 2, pp. 175–189, Feb 2003.

[8] H. Park, J. Bae, J. Park, M. Baeg, and J. Park, "Intuitive peg-in-hole assembly strategy with a compliant manipulator," in *IEEE ISR 2013*, 2013, pp. 1–5.

[9] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. Ojea, E. Solowjow, and S. Levine, "Residual Reinforcement Learning for Robot Control," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6023–6029.

[10] J. Luo, E. Solowjow, C. Wen, J. Ojea, A. Agogino, A. Tamar, and P. Abbeel, "Reinforcement Learning on Variable Impedance Controller for High-Precision Robotic Assembly," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3080–3087.

[11] T. Inoue, G. D. Magistris, A. Munawar, T. Yokoya, R. Tachibana, and Ryuki, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 819–825.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

[12] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," *Journal of Machine Learning Research*, vol. 17, pp. 39:1–39:40, 2016.

[13] M. Vecerik, O. Sushkov, D. Barker, T. Rothörl, T. Hester, and J. Scholz, "A Practical Approach to Insertion with Variable Socket Position Using Deep Reinforcement Learning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 754–760.

[14] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. Ojea, E. Solowjow, and S. Levine, "Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5548–5555.

[15] Z. Liu, K. Wang, D. Liu, Q. Wang, and J. Tan, "A Motion Planning Method for Visual Servoing Using Deep Reinforcement Learning in Autonomous Robotic Assembly," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 6, pp. 3513–3524, 2023.

[16] Y. Shi, Z. Chen, Y. Wu, D. Henkel, S. Riedel, H. Liu, Q. Feng, and J. Zhang, "Combining Learning from Demonstration with Learning by Exploration to Facilitate Contact-Rich Tasks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1062–1069.

[17] K. Wang, Y. Zhao, and I. Sakuma, "Learning Robotic Insertion Tasks From Human Demonstration," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5815–5822, 2023.

[18] J. Luo, Z. Hu, C. Xu, Y. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine, "SERL: A Software Suite for Sample-Efficient Robotic Reinforcement Learning," in *2024 International Conference on Robotics and Automation (ICRA)*, 2024, pp. 16 961–16 969.

[19] C. Yuan, Y. Shi, Q. Feng, C. Chang, M. Liu, Z. Cheng, A. Knoll, and J. Zhang, "Sim-to-Real Transfer of Robotic Assembly with Visual Inputs Using CycleGAN and Force Control," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2022, pp. 1426–1432.

[20] G. Schoettler, A. Nair, J. Ojea, S. Levine, and E. Solowjow, "Meta-Reinforcement Learning for Robotic Industrial Insertion Tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9728–9735.

[21] Z. Wu, Y. Xie, W. Lian, C. Wang, Y. Guo, J. Chen, S. Schaal, and M. Tomizuka, "Zero-Shot Policy Transfer with Disentangled Task Representation of Meta-Reinforcement Learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7169–7175.

[22] I. Iturrate, E. Roberge, E. Østergaard, V. Duchaine, and T. Savarimuthu, "Improving the Generalizability of Robot Assembly Tasks Learned from Demonstration via CNN-based Segmentation," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 553–560.

[23] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, "Improving Sample Efficiency in Model-Free Reinforcement Learning from Images," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 10 674–10 681, May 2021.

[24] N. Hansen and H. Su and X. Wang, "Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021 and NeurIPS 2021 and December 6-14 and 2021 and virtual*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Vaughan, Eds., 2021, pp. 3680–3693.

[25] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: Contrastive Unsupervised Representations for Reinforcement Learning," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 5639–5650.

[26] D. Yarats and I. Kostrikov and R. Fergus, "Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels," in *9th International Conference on Learning Representations and ICLR 2021 and Virtual Event and Austria and May 3-7 and 2021*, 2021.

[27] N. Hansen and R. Jangir and Y. Sun and G. Alenyà and P. Abbeel and A.A. Efros and L. Pinto and X. Wang, "Self-Supervised Policy Adaptation during Deployment," in *9th International Conference on Learning Representations and ICLR 2021 and Virtual Event and Austria and May 3-7 and 2021*, 2021.

[28] A. Lee, A. Nagabandi, P. Abbeel, and S. Levine, "Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[29] A. Zhang and R.T. McAllister and R. Calandra and Y. Gal and S. Levine, "Learning Invariant Representations for Reinforcement Learning without Reconstruction," in *9th International Conference on Learning Representations and ICLR 2021 and Virtual Event and Austria and May 3-7 and 2021*, 2021.

[30] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering Diverse Domains through World Models," *CoRR*, vol. abs/2301.04104, 2023.

[31] T. Haarnoja and A. Zhou and P. Abbeel and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholm and Sweden: PMLR, 2018, pp. 1861–1870.

[32] G. Bartyzel and W. Półchłopek and D. Rzepka, "Reinforcement Learning With Stereo-View Observation for Robust Electronic Component Robotic Insertion," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 3, p. 57, Oct 2023.

[33] L. Villani, J. D. Schutter, and O. Khatib, "Force Control," in *Springer Handbook of Robotics*, B. Siciliano, Ed. Springer International Publishing, 2016, ch. 9, pp. 195–220.

[34] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2014.

[35] M. Wu and N.D. Goodman, "Multimodal Generative Models for Scalable Weakly-Supervised Learning," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 and NeurIPS 2018 and December 3-8 and 2018 and Montréal and Canada*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 5580–5590.

[36] G. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[37] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*. Cambridge and MA and USA: A Bradford Book, 2018.

[38] M. Jordan, Z.Ghahramani, T. Jaakkola, and L. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, Nov 1999.

[39] I. Higgins, L. Matthey, A. Pal, C. B, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *5th International Conference on Learning Representations and ICLR 2017 and Toulon and France and April 24-26 and 2017 and Conference Track Proceedings*, 2017.

[40] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *7th International Conference on Learning Representations and ICLR 2019 and New Orleans and LA and USA and May 6-9 and 2019*, 2019.

[41] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver, "Distributed Prioritized Experience Replay," in *International Conference on Learning Representations*, 2018.

[42] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.