Robotica (2024), **42**, pp. 3247–3262
doi:10.1017/S0263574724000626

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Robot imitation from multimodal observation with unsupervised cross-modal representation

Xuanhui Xu[1], Mingyu You[1,2], Hongjun Zhou[1] and Bin He[1,2]

[1]College of Electronic and Information Engineering, Tongji University, ShangHai, China
[2]National Key Laboratory of Autonomous Intelligent Unmanned Systems, Frontiers Science Center for Intelligent Autonomous Systems, Ministry of Education, Tongji University, ShangHai, China
Corresponding author: Mingyu You; Email: myyou@tongji.edu.cn

Received: 27 December 2023; **Revised:** 14 March 2024; **Accepted:** 28 March 2024; **First published online:** 8 November 2024

Keywords: Imitation from observation; representation learning; multimodal

Abstract

Imitation from Observation (IfO) prompts the robot to imitate tasks from unlabeled videos via reinforcement learning (RL). The performance of the IfO algorithm depends on its ability to extract task-relevant representations since images are informative. Existing IfO algorithms extract image representations by using a simple encoding network or pre-trained network. Due to the lack of action labels, it is challenging to design a supervised task-relevant proxy task to train the simple encoding network. Representations extracted by a pre-trained network such as Resnet are often task-irrelevant. In this article, we propose a new approach for robot IfO via multimodal observations. Different modalities describe the same information from different sides, which can be used to design an unsupervised proxy task. Our approach contains two modules: the unsupervised cross-modal representation (UCMR) module and a self-behavioral cloning (self-BC)-based RL module. The UCMR module learns to extract task-relevant representations via a multimodal unsupervised proxy task. The Self-BC for further offline policy optimization collects successful experiences during the RL training. We evaluate our approach on the real robot pouring water task, quantitative pouring task, and pouring sand task. The robot achieves state-of-the-art performance.

1. Introduction

One well-known and popular way for robot task learning is Imitation from Observation (IfO) [1, 2]. With IfO, robots can learn tasks directly from unlabeled videos without having access to the demonstrator's actions (e.g., the spatial position of human joints). A large number of human learning resources—for example, vast quantities of online videos of people performing different tasks—can be used for robot imitation learning since IfO does not require demonstrator's actions.

Observations of the robot are raw videos that contain a wealth of information. The IfO algorithm should exactly extract task-relevant representations. Taking the pouring water video as an example, it contains spatial position and texture of each object such as the cup, kettle, camera, microphone, table, and so on. Only the rotation angle of the kettle and the spatial position of the cup rim are task-relevant representations. Existing IfO algorithms extract features by using simple encoding networks (e.g., about three-layer CNN) [1, 3, 4] or the pre-trained networks [5, 6]. Figure 1 shows the feature heatmaps of images from the pouring water task, which are extracted by these networks. Pre-trained Resnet focuses on cups, microphone, and camera. It focuses on all appearing objects in the image since the Resnet was pre-trained for object detection. However, most of these things such as the microphone and the camera are irrelevant to the pouring water task. The robot does not need to pay attention to these objects in the process of pouring water. As for the simple encoding network, its attention is scattered and even focuses on the irrelevant background. Representations extracted by the simple encoding network may confuse

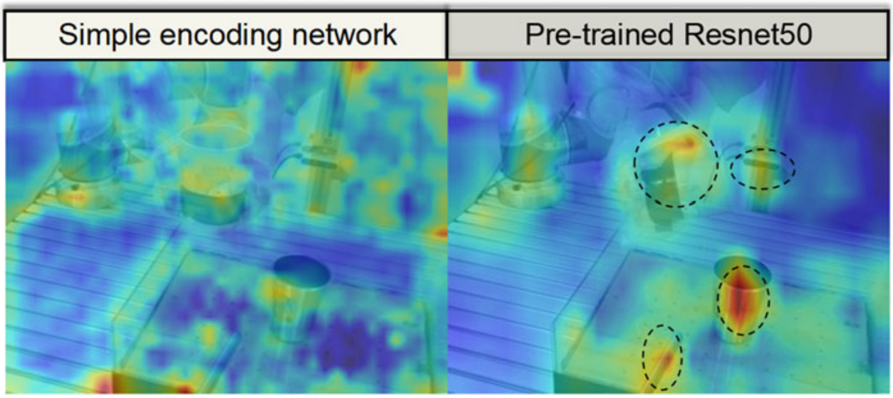


Figure 1. *The feature heatmaps of images from the pouring water task are extracted by a simple encoding network (e.g., about three-layer CNN) and a pre-trained Resnet.*

the policy. An intuitive idea to solve this problem is to design an appropriate proxy task [7] to pre-train the representation model which can extract task-relevant representations. Demonstration videos are always multimodal. They include both images and audios. Different modalities describe the same information in different ways [8]. The relationship between these modalities can be used to design an unsupervised proxy task.

Another major challenge of the IfO problem is learning efficiency because mostly IfO algorithms are based on the RL algorithm. Optimizing the RL algorithm in real world is time-consuming and laborious [9], since the training of RL algorithm requires millions of robot–environment interactions. Reducing the number of robot–environment interactions can improve the learning efficiency of the RL algorithm. Inspired by the fact that humans can learn the policy offline with their memories of the task. We present the self-behavioral cloning module, which prompts the policy to learn from its successful experiences to improve the learning efficiency. Moreover, most existing IfO algorithms [4, 10] obtain rewards by calculating the approximation between the images of robot and that of demonstrator. They suffer from the task-irrelevant representations in images. Under the current calculation approach of the reward function, the approximation of a large amount of task-irrelevant representations would occupy a great weight in the reward. Therefore, these rewards cannot accurately represent the success of the task.

We present a new approach for robot imitation from multimodal observation. Our approach contains two modules: the UCMR module and self-BC-based RL module. We design a multimodal classification task as the proxy task. The input of the proxy task is the image from the demonstration and the robot's random exploration. The label of the proxy task is the category of audio. With the proxy task, the UCMR model can focus on task-relevant representations. The self-BC-based RL module samples high-reward trajectories during the RL training, and then uses these trajectories to train the policy. The policy is optimized with a multimodal reward function, which can accurately represent whether the robot successfully imitates the task. Audios can be more efficient to signal the success of pouring water.

The main contributions of this article are as follows:

- This article presents a new approach for robot imitation from multimodal observation, which prompts the real robot to learn tasks from multimodal human learning resources.
- This article proposes the UCMR model, which learns to extract task-relevant representations through an unsupervised multimodal proxy task.
- This article proposes the Self-BC, which can improve the performance of reinforcement learning (RL) and is easy to transplant.
- This article validates the generality of the proposed method in three real-world tasks.

2. Related work

2.1. Imitation from observation

Imitation from Observation is the problem of robot imitating directly from state-only demonstrations without having access to the demonstrator's actions. Faraz Torabi et al. [11] proposed a two-phase framework named behavioral cloning from observation (BCO). BCO achieved satisfactory results in tasks from Gym and CoinRun [12]. BCO employed an inverse dynamics model to infer nonhuman operator's action from visual observations. With these action labels, its task learning process changed from unsupervised to supervised, and the representation extraction module can focus on task-relevant representations. However, the inverse dynamics model requires an auxiliary dataset with the operator's state-action pairs for training. Most IfO problems cannot meet this condition.

YuXuan Liu et al. [13] trained a translation model to extract invariant features from different contexts, which was used for task learning. After that, Faraz Torabi et al. presented GAIfO, which was based on GAN [14] and LfO. GAIfO employed an agent and a discriminator. The discriminator attempted to distinguish the source of the current action, and the agent tried to fool the discriminator. In 2019, Faraz Torabi et al. [15] proposed a LfO algorithm built on the top of GAIfO. They addressed the sample inefficiency problem by utilizing ideas from trajectory centric RL algorithms. Lukas Hermann et al. [3] presented the ACGD to adaptively set the appropriate task difficulty for the learner by controlling where to sample from the demonstration trajectories. All the feature extraction modules of these methods are simple extraction networks that are a part of policies and trained simultaneously with the control module through RL. Due to the lack of the clear supervision information—for example, demonstrator's actions, these simple extraction networks cannot focus on task-relevant representations. The extraction network, such as the GAIfO, would pay more attention to the texture of the background than the target object of operation.

Haresh Karnan et al. [5] and Rutav Shah et al. [6] employed the pre-trained Resnet to extract representations. They achieved satisfactory results in the simple real-world environments. However, the Resnet is initially pre-trained for object detection. It is expected to focus on all objects in the image, but the researches of IfO are more concerned about the action trajectory of the demonstrator and the target object of operation. Therefore, a lot of information extracted by a pre-trained Resnet is task-irrelevant. Moreover, the chaotic extracted representations would reduce the efficiency of task learning.

2.2. Robot multimodal learning

As for multimodal, Lee et al. [16] presented a cross-modal compensation model (CCM). CCM can extract representations from different modalities and fuse the representations, which perform better when one modality is corrupted or noisy. Jean-Francois Tremblay et al. [17] leveraged multiple sensors (vision, radar, and proprioception) to perceive maximal information about the vehicle's environment, which was robust to arbitrarily miss modalities at test time. Marwan et al. [18] introduced some robot grasping algorithms based on RGB-D, which have achieved great performance. These researches took advantage of supplementary information between modalities to improve the effect and robustness of the algorithm. These researches focus on vision, touch, and radar modality. However, for IfO, audio is the most common, informative, and easily accessible modality. Most demonstration videos contain audio. For pouring water task, audios can even be more efficient to signal the success of pouring water than images. The existing robotic researchers pay little attention to audios. Making use of the supplementary between images and audios can greatly improve the efficiency of the algorithm.

3. Approach

In this section, we illustrate our approach for robot imitation from multimodal observation. Our approach includes a UCMR module and a self-BC-based RL module.

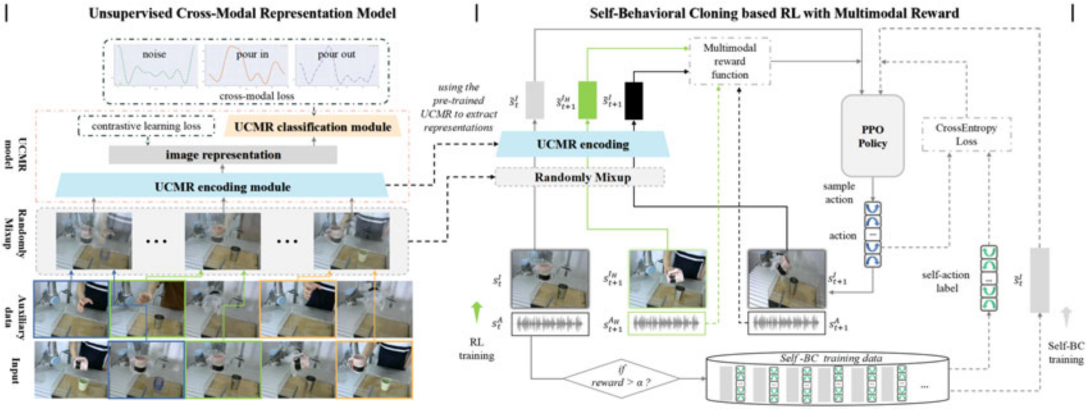


Figure 2. Our approach includes a UCMR module and a self-BC-based RL module. For the UCMR module, the inputs are the mixed images that have two performers, and the loss functions include the contrastive learning loss and the cross-modal loss. For the self-BC-based RL module, policy tasks the current image representation $\tilde{s}_t^I$ as an input to sample an action. The robot does the action in the real world. We get the next image representation $\tilde{s}_{t+1}^I$. We employ the multimodal reward to calculate the reward r. After a period of RL optimization, we collect some trajectories with rewards greater than α. These data are used to supervise the policy training with the CrossEntropy Loss.

We consider the IfO problem within Markov Decision Process (MDP) [19, 20] which is a mathematically idealized form of RL. In the IfO problem, $s_t^I \in S$ and $s_t^A \in S$ represent images and audios that the robot observes from the environment, $s_t^{I_H} \in D$ and $s_t^{A_H} \in D$ are images and audios from the multimodal human demonstration, $a_t \in A$ denotes the action which is sampled by the policy, and $r_t \in R$ corresponds to the reward which is given by a reward function in each state.

$$\pi_\theta\left(a_{t+1}|s_t^I, a_t\right) = Pr\left\{A_{t+1} = a'|S_t = s^I, A_t = a\right\} \tag{1}$$

$\pi_\theta(a_{t+1}|s_t^I, a_t)$, as shown in the Eq. 1, is an agent which learns a policy through RL. The policy represents the probability of sampling action in different state. Given any state s^I and action a, the policy represents the probability of each possible of next action a'.

3.1. Robot imitation from multimodal observation

In this article, robots are prompted to learn tasks from multimodal observations. As shown in Figure 2, we achieve this in two steps. Firstly, pre-training the UCMR module, we randomly mix the training image with an auxiliary image to get the mixed images that have two performers. Then, the contrastive learning loss is employed to train the UCMR encoding module. After that, the UCMR encoding module and UCMR classification module are trained by the cross-modal loss. Secondly, these representations that are extracted by the UCMR are used to guide the robot to learn the task with the self-BC-based RL algorithm. Policy tasks the current image representation $\tilde{s}_t^I$ as an input to sample an action. The robot does the action in the real world. We get the next image representation $\tilde{s}_{t+1}^I$. We employ the multimodal reward to calculate the reward r. After a period of optimization, we collect some trajectories with rewards greater than α. The number of trajectories collected in our experiment is 100. α is the average of the current reward values. These data are used to train the supervised policy with the crossEntropy loss. Specifically, $\tilde{s}_t^I$, $\tilde{s}_{t+1}^I$, and $\tilde{s}_{t+1}^{I_H}$ are all extracted by the UCMR encoding module.

Figure 3 shows the multimodal demonstration that includes images and audios. Figure 3(a) shows the action trajectory. For audios, we extract its formant feature. Formant refers to some areas where the energy is relatively concentrated in the frequency spectrum of audio, reflecting the resonant cavity's physical characteristics [21]. As shown in Figure 3(b), in the pouring water task, audios can be easily

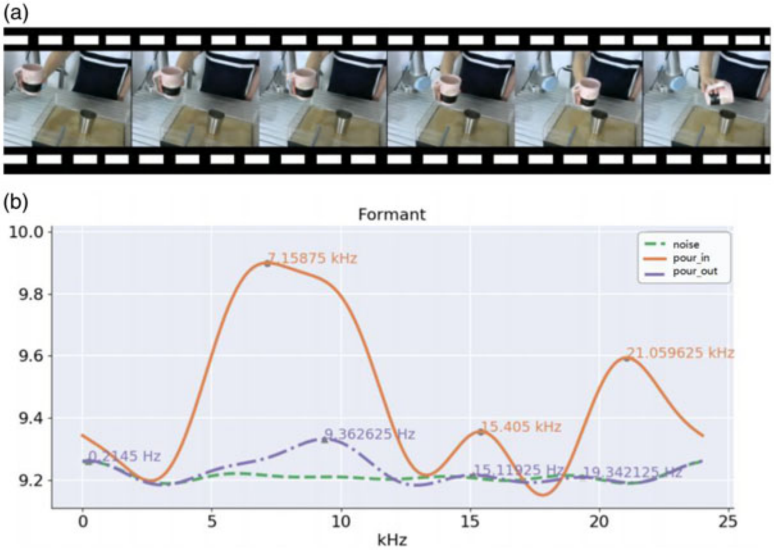


Figure 3. *(a) This shows images from the one multimodal human demonstration trajectory. (b) This shows the formants of audio in three cases: noise (no water pouring occurred), pour in (water poured into the cup), and pour out (water poured out of the cup).*

classified into three categories according to the formants: noise (no water pouring occurred), pour in (water poured into the cup), and pour out (water poured out of the cup).

3.2. Unsupervised cross-modal representation

We design a multimodal unsupervised proxy task using the interactions between images and audios. The images and audios are aligned in time. Our proxy task is an image classification task. We divide audios of pouring water into three categories: noise, pour in, and pour out. We take these categories as the labels of the image classification task. With this proxy task, the UCMR model needs to understand whether water pouring occurs in the figure and if water pouring occurs, whether water is poured into the cup. This guides the UCMR model to extract the task-relevant representations such as the spatial position of kettle and cup. Moreover, we train the UCMR model with a contrastive learning loss before the training with the cross-modal loss to improve its performance. This contrastive learning loss, which trains the model by maximizing the similarity between two augmentations of one image, is presented by Xinlei Chen [22].

Another challenge that the UCMR model needs to handle is the different joint structures of the robot and the human demonstrator. The difference in joint structure leads to a great difference in vision between robot and human even when performing the same action. This leads to the failure of the approach to obtain the reward by calculating the similarity between the image of demonstrator and that of robot. We propose Task-Performer Mixup to address this challenge. First, we collect some images of humans and robots waving their arms randomly in the task scene as the auxiliary images. In these auxiliary images, both humans and robots are empty-handed. Then, for the human image from the training data of the UCMR model, we mix it with a randomly sampled auxiliary image of robot performing action. As for the robot image, image used for mixing is the auxiliary image of human performing action. With Task-Performer Mixup, each training image of the UCMR model contains two performers as shown in Figure 2. This makes images of different task performers have no essential difference, and the information of task performers becomes the useless background. Therefore, the UCMR model can ignore the task performers, bridge the gap caused by different performers, and focus on the task-relevant information.

$$\begin{aligned} ConLoss = {} & \frac{1}{2} D_{cos}(En_{UCMR}(aug_1(I)) \\ & + F(En_{UCMR}(aug_2(I)))) \\ & + \frac{1}{2} D_{cos}(F(En_{UCMR}(aug_1(I))) \\ & + En_{UCMR}(aug_2(I))) \end{aligned} \tag{2}$$

3.3. Self-behavioral cloning-based RL

We illustrate our UCMR model in Figure 2. First, we randomly mix the training image with an auxiliary image to get the mixed images that have two performers. Then, the contrastive learning loss is employed to train the UCMR encoding module, which is shown in Eq. 2. I is the mixed images. aug_1 and aug_2 denote random data augmentation. In each epoch, aug_1 and aug_2 randomly select from “Resize”, “Cropping”, “Flipping”, “Rotation”, and “Grayscale”. D_{cos} represents cosine similarity. En_{UCMR} is the UCMR encoding module. F denotes a prediction MLP. After that, the UCMR encoding module and UCMR classification module are trained by the cross-modal loss. The mixed image’s label is the same as the category of its audio. Eq. 3 shows the cross-modal loss. i denotes the serial number of the mixed image. c represents image category. y_{ic} is a Boolean quantity. If mixed image i belongs to category c, y_{ic} equals to 1, otherwise y_{ic} equals 0. p_{ic} is the output of the UCMR classification module. The image representation which is encoded by the UCMR encoding module is used in the following task learning.

$$CroMLoss = -\frac{1}{N} \sum_{i}^{N} \sum_{c=1}^{3} y_{ic} log(p_{ic}) \tag{3}$$

After extracting the task-relevant representations by the UCMR encoding module, we propose a self-BC-based RL method to learn the task. Moreover, we use the multimodal reward function to optimize the RL algorithm.

3.3.1. Self-behavioral cloning

RL algorithms achieve satisfactory results in simulation through hundreds of thousands of robot–environment interactions. While facing some real-world tasks, the interactions are expensive and even potentially dangerous. We present self-behavioral cloning (Self-BC) to reduce the number of interactions and improve the performance of RL algorithm. The training data of the Self-BC is collected during the interactions of the RL algorithm. These data consisted of representation–action pairs $(\widetilde{s}_t^I, a)$ are from the high reward trajectories. Self-BC allows the robot to refine the core of the task from its own successful experience so as to reduce the interactions. As shown in Figure 2, PPO policy takes the current image representation $En_{UCMR}(s_t^I)$ as an input to sample an action. The robot does the action in the real world. We get the image representation $En_{UCMR}(s_{t+1}^I)$ of the next moment through the UCMR encoding module. We calculate the reward r based on the multimodal data of the demonstrator and the robot. After a period of RL optimization, we collect some trajectories whose rewards are greater than α, which is an empirical value as the training data of Self-BC. With this data, the policy can be offline optimized with the crossEntropy loss. RL and Self-BC alternately optimize the policy once in each episode. Specifically, En_{UCMR} represents the UCMR encoding module.

3.3.2. Multimodal reward

We design a multimodal reward function to take advantage of various modalities. As we illustrated in Section 3.2, the interactions between modalities are supplementary. Some information is easier to understand in one modality but confusing in the other. For example, it’s easier to figure out whether the

Algorithm 1 Training iteration procedure of self-bc-based RL

```
Initialize policy $\pi_\theta$; load the UCMR encoding module $En_{UCMR}$, multimodal reward function $R_{mm}$
for $N$ episode do
    Reset the pouring water environment
    Get the initial state $s_t^I$ and $s_t^A$
    for $T$ steps do
        timestep += 1
        Sample action $a$ according to $\pi_\theta\left(En_{UCMR}(s_t^I)\right)$
        Execute the action $a$; get the next state $s_{t+1}^I$ and $s_{t+1}^A$
        Get the $reward$ through the multimodal reward function $R_{mm}\left(En_{UCMR}(s_{t+1}^I), s_{t+1}^A,\right.$
            $\left.En_{UCMR}(s_{t+1}^{I_H}), s_{t+1}^{A_H}\right)$
        Add $a, s_t^I, s_t^A, reward, s_{t+1}^I, s_{t+1}^A$ to RL memory buffer
        if $reward > \alpha$ then
            Add $a, s_t^I$ to Self-BC memory buffer
        end if
        Assign $s_{t+1}^I$ to $s_t^I$
        if timestep % update_timestep == 0 then
            Update policy parameters on RL memory data:$\tilde{\theta} \leftarrow \theta - \lambda_\pi \hat{\triangledown} J_{PPO}^{\theta^k}(\theta)$
        end if
        if episode > start_Self-BC then
            Update policy parameters on Self-BC memory data:$\tilde{\theta} \leftarrow \theta - \gamma \hat{\triangledown} Loss_{CrossEntropy}(\pi)$
        end if
    end if
end if
return $\tilde{\theta}$
```

water is poured in according to the audio than the image. However, the information in image is richer such as the trajectory of kettle, spatial position of cup, and so on. Therefore, the multimodal reward includes the similarities of the image representations and the formants of audio, as shown in Eq. 4. D_{cos} represents the cosine similarity. $s_t^{A_H}$ and s_t^A denote the audio of demonstration and that of robot observation, respectively. γ_1, γ_2 are hyper-parameters.

$$\begin{aligned} MultimodalReward = &\gamma_1 D_{cos}(s_{t+1}^A, s_{t+1}^{A_H}) \\ &+\gamma_2 D_{cos}(En_{UCMR}(s_{t+1}^I), En_{UCMR}(s_{t+1}^{I_H})) \end{aligned} \tag{4}$$

Algorithm 1 summarizes the training procedure of the Self-BC-based RL. π_θ is the policy net and θ denotes its parameters. a denotes action. En_{UCMR} is previously trained UCMR encoding module. As for self-BC, the collected data is used to supervise and train the policy with the CrossEntropy Loss, $Loss_{CrossEntropy}$.

4. Experiments

In this article, we choose the pouring water as our task, since pouring water is a popular multimodal task in daily life. Through this task, we attempt to prove that the UCMR model can extract task-relevant representations and self-BC can improve the performance of RL algorithms.

4.1. Experiments setup

Our experiment environment is composed of a UR5 robot, a Robotiq 2F-140 Gripper, a common RGB camera, a microphone, an aluminum work table, and some cups. The workstation, used for experiments, has 2 GeForce GTX 1080 GPUs, and 1 Intel i7-10700K CPU.

Table I. *Proxy task accuracy.*

	ConLoss	CroMLoss	accuracy(%)
Net1	XXXX		16.27
Net2		XXXX	95.41
Net3	XXXX	XXXX	98.07

4.1.1. UCMR model

Both the data from human demonstrations and robot observations are multimodal, which contain images and audios. During the collection, images and audios are collected at the same time. The backbone of the UCMR model is Resnet50. During the training and evaluation, the input images are resized to 244 × 244 pixels. The UCMR model is trained by the contrastive learning loss for 100 epochs. The batch size is 64. This can improve the generalization ability of the model for small amplitude illumination and camera position changes. Then, the UCMR model is trained by the cross-modal loss for 400 epochs, and the batch size is 32. We exploit the Adam [23] as our optimizer with an initial learning rate of 0.0002.

4.1.2. Reinforcement learning

As for task learning, the policy is trained by the self-BC-based RL algorithm with the multimodal reward. It' is built on the top of PPO [24] in this article. Actually, any other RL algorithms, such TRPO [25], DDPG [26], and DQN [27] can be used as the training method. The policy takes the 512-dimensional feature, which is encoded by the UCMR model, as input. The action that the policy chooses is a six-dimensional vector. The max action step is 10. The policy network is composed of three fully connected networks: [512 × 256], [256 × 64], [64 × 6]. The reward attenuation coefficient is $\gamma = 0.99$. The policy is optimized by Adam with a learning rate of 0.0001.

4.1.3. Self-behavioral cloning

The self-BC training starts after 5000 episodes of RL learning. When the amount of data in the Self-BC data pool is greater than 1000, we start Self-BC training. The number of epochs of each training is 5 which is an experience value. After the training, the data pool is emptied, and the data are collected again. The batch size is 32, and the optimizer is SGD with a learning rate of 0.0004 [28].

4.1.4. Human demonstrations

For pouring water task, we collect 100 human demonstrations. The human demonstration numbers for quantitative pouring task and pouring sand task are 30 and 100, respectively. All demonstrations are multimodal demonstrations, including image and audio. To collect the audio of the pouring of water and sand into and out of the cup, 50 demonstrations are task success demonstrations and 50 demonstrations are task failure demonstrations (in the pouring water task and the pouring sand task). The image sampling rate is 20 Hz, and the audio window is 50 milliseconds. Moreover, we collected 1,000 images of a robot holding a cup in random motion which is used for the training of UCMR model, since the training of UCMR model needs the mixed image of robot and human.

4.2. Unsupervised cross-modal representation

In this section, we analyze the performance of the UCMR model in detail using the pouring water task. We illustrate the performance of the models which are trained with different loss function combinations, on proxy tasks. Moreover, we visualize the attention of the model in the feature extraction process.

Table I shows the proxy task accuracies with different loss function combinations. Net1 is only trained with ConLoss for 200 epochs. Net2's loss function contains only CroMLoss, and it is trained for 200

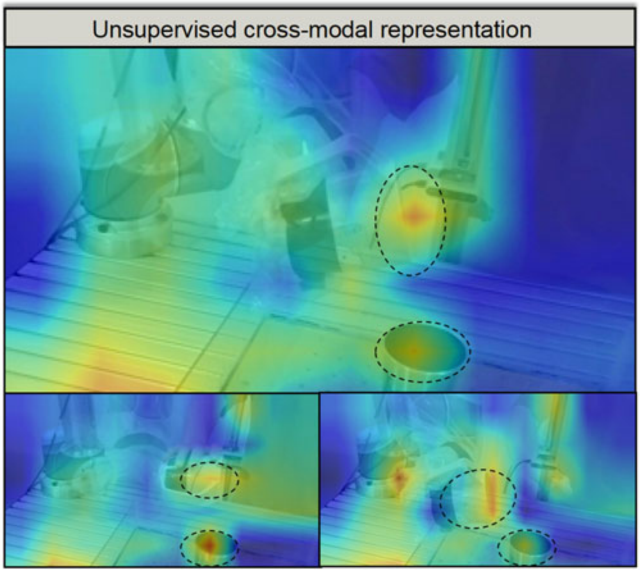


Figure 4. *The feature heatmaps of images from the pouring water task are extracted by the UCMR model.*

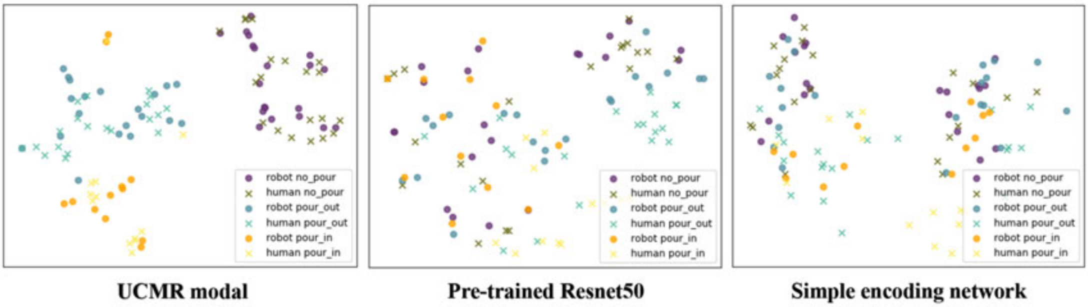


Figure 5. *This shows the feature space distribution of image representations, which are extracted by the UCMR model, pre-trained ResNet50, and simple encoding network. For the representations extracted by the UCMR model, the inter-class distance is large and the intra-class distance is small.*

epochs. As for Net3, it's trained with ConLoss for 100 epochs, then it's trained with CroMLoss for 100 epochs. The proxy task accuracy of Net2 and that of Net3 are 95.41% and 98.07%, respectively. This shows that the CroMLoss impels the UCMR model to focus on the task-relevant representations and the UCMR model understands what kind of audio exists when displayed by the image. In other words, the UCMR model can focus on the task-relevant representations. The accuracy of Net3 is 2.66% higher than that of Net2. The ConLoss trains the model through maximizing the similarity between the two augmentations of one image, which needs the UCMR model to focus on the invariant features. This improves the ability of the UCMR model to extract representations.

Figure 4 visualizes the attention of the UCMR model in the feature extraction process [29]. We can see that the UCMR model notices the orientation of the cup rim in the robot's hand and the center of the cup rim on the table. There is no doubt that for the water pouring task, these are the two most important task-relevant representations in image. Moreover, Figure 5 illustrates the feature space distribution of images. Symbols "×" and "∘" represent that the performer in the images are human and robot, respectively. We employ t-SNE [30] to reduce the dimensionality of image representations which are extracted by the

Table II. *Task learning experiments.*

Methods		Ours		ACGD		GAIfO	
		with Self-BC	w/o Self-BC	with Self-BC	w/o Self-BC	with Self-BC	w/o Self-BC
encoder	UCMR	86.73%	69.39%	66.32%	43.87%	19.39%	16.22%
	Resnet50	0%	0%	0%	0%	0%	0%
generalization		85.71%	-	65.98%	-	18.56%	-

UCMR model, pre-trained ResNet50, and simple encoding network. For the representations extracted by the UCMR model, the inter-class distance is large and the intra-class (e.g., robot performs and human performs) distance is small. However, the representations extracted by the other two methods cannot distinguish the categories.

4.3. Pouring water task

In this section, we compare our method with two other methods: ACGD [3] and GAIfO [4], in pouring water task. We illustrate the performances of the three methods when they use different extraction models. After that, we illustrate the ablation and generalization experiments of our methods.

4.3.1. Comparative

Table II shows the best success rate of the three methods in the pouring water task. In Table II, the symbol "UCMR" denotes that the input representations are extracted by the UCMR model, and the symbol "Resnet50" represents those input representations are extracted by the pre-trained Resnet50. Specifically, all three methods are trained for 10,000 epochs. Moreover, in the paper of Lukas Hermann et al. [3], ACGD needs the actions of the robot and that of the demonstrator. Since we do not have the actions in this experiment, the input of ACGD is only the extracted image representations.

As shown in Table II, when the representations are extracted by the UCMR model, the best success rate of our method is 86.73%. For ACGD and GAIfO, the best success rate is 66.32% and 19.39%. When the representations are extracted by the pre-trained Resnet50, no method has successfully completed the pouring water task since the pre-trained Resnet50 focuses on the objects such as cup, camera, microphone, and so on. These things appear in each step of the task. The representations of each step are almost the same which confuses the policy. These results prove that the UCMR model can well extract the task-relevant representations.

The success rate of the UCMR model is 20.41% higher than that of ACGD and 67.34% higher than that of GAIfO. The major advantage of our method is the multimodal reward. Our reward computes the reward based on the representations of images and the formants of audios. Image representations characterize the spatial position of the kettle, the rotation angle of the kettle, and the position of the cup rim. The formants of audio interpretably represent whether water has been poured into the cup. The comprehensive use of these two modalities can clearly indicate the task process and task results. This allows our method to have a better performance than the ACGD and the GAIfO (the rewards of ACGD and GAIfO only contain image representations). Figure 6 shows the change i in success rate during training. The success rate curves in Figure 6 are the average result of five experiments. GAIfO does not complete the task until about 9500 epoch because the reward of GAIfO is calculated by the discriminator and the discriminator should be trained at the same time as the policy. In the initial training, its reward is wrong, resulting in low training efficiency at the initial stage of training.

Table III. *Reward ablation experiment.*

	Our method with multimodal reward	**Our method with single modal reward**
success rate	86.73%	64.24%

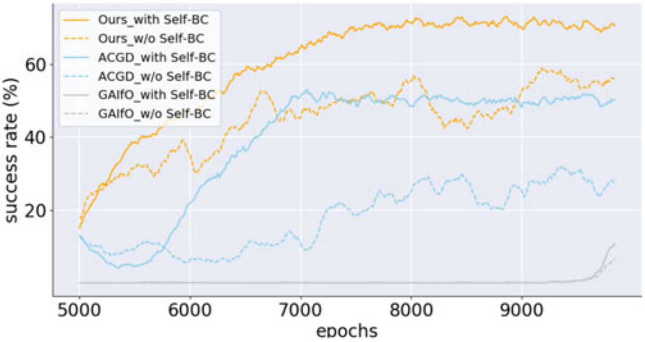


Figure 6. *This figure shows the change of success rate during training. This is the average result of five experiments.*

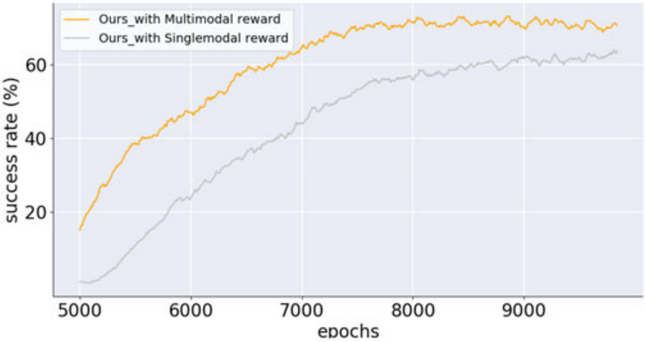


Figure 7. *This figure shows the change in success rate of the reward ablation experiment. This is the average result of five experiments.*

4.3.2. Ablation and generalization

Table II illustrates the results of the self-BC ablation experiment and generalization experiment. Symbol "w/o Self-BC" denotes that we do not use the self-BC during the RL training. As for the generalization experiment, these methods are trained with one cup and tested with other two cups. All image representations used in these experiments are extracted by the UCMR model. Without Self-BC module, the best success rate of our method, ACGD, and GAIfO is 69.39%, 43.87%, and 16.22%, respectively, which prove the effectiveness of self-BC. Self-BC allows RL algorithms to learn from their own successful experience. Moreover, as shown in Figure 6, self-BC not only improves the best success rate of the methods but also reduces the fluctuation of the success rate in training process.

Table III shows the reward ablation experiment. This experiment compares the effects of multimodal reward function and single reward function on the effectiveness of our method. The best success rate of our method with multimodal reward is 86.73%. For our method with single modal reward, the best success rate is 64.24%. Moreover, as shown in Figure 7, the success rate of our method with multimodal reward is about 20% higher than that of single modal reward on 5000 epochs. These confirm that the multimodal reward function can improve the effect of RL algorithm.

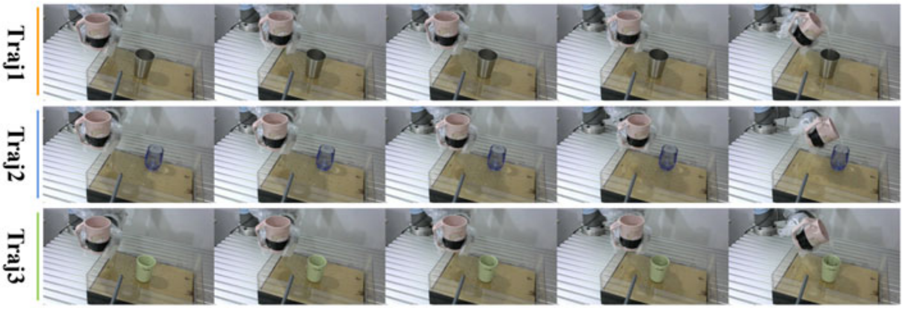


Figure 8. *The successful trajectories of generalization experiment.*

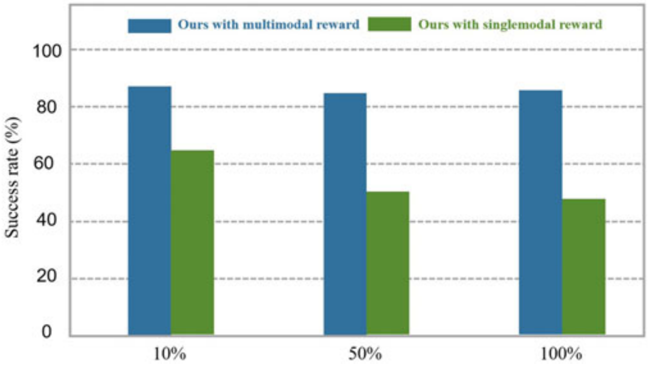


Figure 9. *The successful rate of the quantitative pouring task experiment. There are three types of target water volume: 10%, 50%, and 100%. The volume of the target cup is 200 ml. Blue represents our method with multimodal reward. Green represents our method with single modal reward.*

As for the generalization experiment (all methods use UCMR), when our method is tested with two new cups, the best success rate is 85.71%, and the best success rate of ACGD and GAIfO is 65.98% and 18.56%, respectively. This represents that the UCMR model focuses on the task-relevant representations and ignores the task-irrelevant representations such as the texture of the cup, change of illumination, and so on. We believe that this generalization ability is brought about by the UCMR model. Since RL algorithm itself does not have such generalization ability, we have not designed it to strengthen its generalization ability. Figure 8 shows three different successful trajectories of generalization experiment. Our method can adapt to different cups without fine-tuning. As for different positions of cups, it depends on whether this position is included in the demonstration.

4.4. Quantitative pouring task

We evaluate our method in the quantitative pouring task. The robot is expected to learn to pour the same volume of water as the demonstrator. There are three types of target water volume: 10%, 50%, and 100%. The volume of the target cup is 200 ml. Our method uses UCMR and self-BC. Moreover, we compare the performance of our method using multimodal reward and single modal reward. The difference between the volume of the robot pouring water and the target volume is within 5 ml, and we define it as a successful pour.

As shown in Figure 9, when the target water volume is 10%, the success rate is 86.13%. When the target water volume is 50% and 100%, the success rate is 84.16% and 85.15%, respectively. The average success rate of three target water volume is 85.15% which is only 1.58% less than the success rate of the

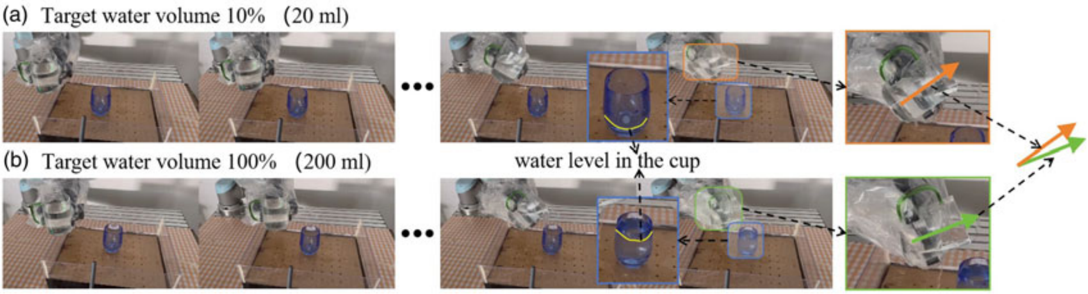

(a) Target water volume 10% (20 ml)
(b) Target water volume 100% (200 ml)
water level in the cup

Figure 10. *(a) and (b) show two real-world quantitative pouring tasks. The target water volumes of (a) and (b) are 10 % and 100 %, respectively. The yellow and green arrows demonstrate the orientation of the normal vector of the cup held by the robot.*

nonquantitative pouring water task (Section 4.3). Our method performs well in the quantitative pouring task, which indicates that our method has good generalization.

To validate the generality of the UCMR model, we compare the performance of our method using multimodal reward and single modal reward. In single modal reward method, the UCMR model needs to focus on water volume which is a new task-relevant representation [31]. As for the single modal reward experiment, when the target water volume is 10%, 50%, and 100%, the success rate is 55.45%, 50.50%, and 47.52%, respectively. The average success rate is 51.16%. While the single modal reward method's success rates are lower than those of the multimodal reward method, they are notably higher than the success rate of ACGD, which does not utilize the UCMR model (with a success rate of 0% as shown in Table II) in non-quantitative water pouring experiments. These experiments validate that the UCMR model can focus on the water volumes and is generalized. Figure 10 shows two real-world quantitative pouring tasks. The target water volumes of (a) and (b) are 10% and 100%, respectively. The yellow and green arrows demonstrate the orientation of the normal vector of the cup held by the robot. As shown in Figure 10, the inclination angle of the cup in (b) is greater than in (a) which indicates that the robot controls the inclination angle of the cup to achieve different target water volumes. Moreover, the success rate decreases as the volume of target water increases. When the target water volume is large, the water in the kettle will be insufficient because the robot is prone to pouring water out of the target cup during the pouring process.

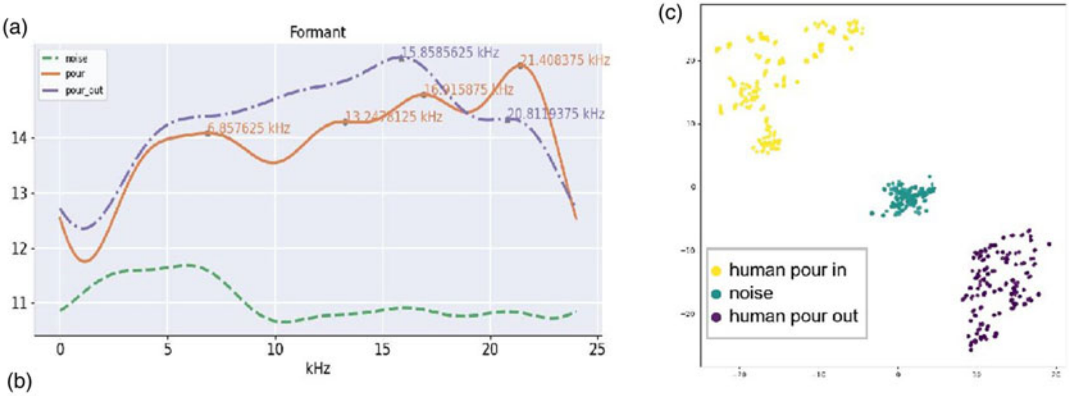
4.5. Pouring sand task

To further validate the generality of our method, we use our method to learn the pouring sand task. Sand as a solid makes a distinct sound from that of water when poured. Figure 11(a) shows the formants of sound in three cases: "noise" (no sand pouring occurred), "pour" (sand poured into the cup), and "pour_out" (sand poured out of the cup), which are differences from that of water. Although the amplitudes of the frequency spectrum of "pour" and "pour_out" are similar, their formants are significantly different. "pour" has four formants, two of which are at 5–15 KHz and the other two at 15–25 KHz. However, "pour_out" only have two formants, both at 15–25 KHz. "noise" is markedly different in amplitude from "pour" and "pour_out". Therefore, it is easy to distinguish "noise", "pour", and "pour_out" from frequency spectrum. Labels for audios can be used as classification labels for their corresponding images. Therefore, our multimodal unsupervised proxy task can be used in pouring sand task which is a pouring solids task. To sum up, our multimodal unsupervised proxy task is generalized and can be used both in pouring liquids and solids tasks.

Furthermore, sand and water flow differently, and a large inclination angle is required to pour the sand because there is friction between the sand particles. This challenges the generality of the Self-BC-based RL model. To train the UCMR model and the Self-BC-based RL algorithm, we collected 100

Table IV. *Pouring sand task best success rate.*

	Our method with Self-BC	ACGD without Self-BC
success rate	81.19%	45.54%



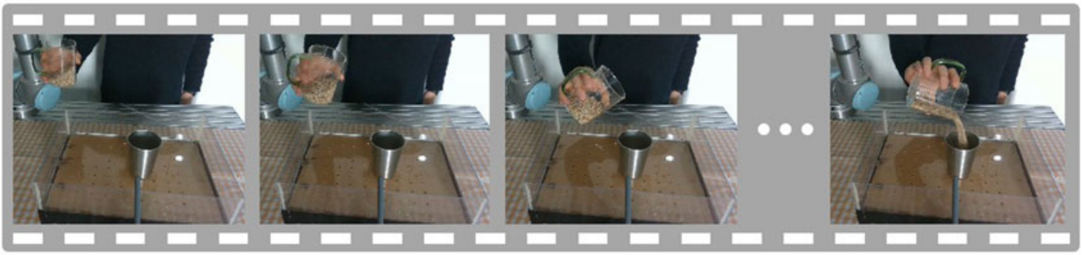


Figure 11. *(a) This shows the formants of audio in three cases: noise (no sand pouring occurred), pour (sand poured into the cup), and pour out (sand poured out of the cup). (b) Images that are sampled from a human pouring sand demonstration. (c) We use the UCMR model to extract representations of the demonstration images. Then, we employ t-SNE to embed the image representations in the Euclidean space.*

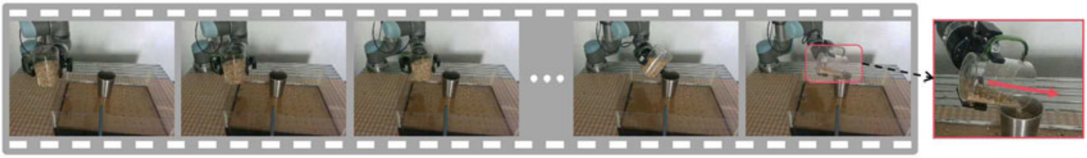


Figure 12. *Real-world robot pouring sand. Pouring sand tasks require a greater inclination angle than pouring water tasks.*

multimodal demonstrations of human pouring sand, including 50 successful and 50 failed. Figure 11(b) shows images that are sampled from a human pouring sand demonstration. In Figure 11, we employ t-SNE to embed the image representations, which are extracted by the trained UCMR model, in the Euclidean space. There are distinct distances between different categories of image representations, which prove that our multimodal unsupervised proxy task can be used in pouring sand task and is generalized.

Table IV shows success rate of our method and ACGD in pouring sand task. Our method is trained with the Self-BC model and image representations obtained from the UCMR model. Similarly, ACGD utilizes image representations extracted by the UCMR model. The success rate of our method and ACGD is 81.19% and 45.54%, respectively. Our method demonstrates a significantly superior success rate compared to ACGD. Figure 12 illustrates a real-world robot pouring sand task. To address the issue of friction among the sand particles, the robot learns to increase the inclination angle of the cup. The notable success rate and the adjustment in the inclination angle serve as evidence for the generalization of our method.

5. Conclusion

In this article, we propose a novel approach for robot imitation from observation via multimodal observations. Our approach contains two modules: the UCMRmodel and a Self-BC-based RL model. We design a multimodal unsupervised proxy task to pre-train the UCMR model and let the model focus on the task-relevant representations. This not only improves the learning efficiency of the policy but also reduces the difficulty of designing the reward function. Experiments show that Self-BC can accelerate the convergence of RL algorithm and improve its success rate. Moreover, the Self-BC-based RL model is optimized by a multimodal reward function.

However, we have not yet achieved quantitative water pouring. In other words, the robot cannot determine the amount of water to pour. Quantitative pouring means that the audio changes from discrete to continuous. In this case, the classification task will no longer be suitable to be a proxy task and we need to design another appropriate proxy task.

Author contributions. Xuanhui Xu and Mingyu You conceived and designed the study. Xuanhui Xu conducted data gathering. Hongjun Zhou and Bin He guided the study and provided critical insights. Xuanhui Xu, Mingyu You, and Hongjun Zhou wrote the article.

Financial support. This work was supported in part by the National Natural Science Foundation of China under Grant No. 62073244, 61825303, and 62088101.

Competing interests. The authors declare no conflicts of interest exist.

Ethical standards. Not applicable.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/S0263574724000626.

References

[1] Y. Chen, C. Zeng, Z. Wang, P. Lu and C. Yang, "Zero-shot sim-to-real transfer of reinforcement learning framework for robotics manipulation with demonstration and force feedback," *Robotica* **41**(3), 1015–1024 (2023).

[2] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou and B. Boots, "Imitation learning for agile autonomous driving," *Int. J. Robot. Res.* **39**(2-3), 286–302 (2019).

[3] L. Hermann, M. Argus, A. Eitel, A. Amiranashvili, W. Burgard and T. Brox. "Adaptive Curriculum Generation from Demonstrations for Sim-to-Real Visuomotor Control," *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France (2020) pp. 6498–6505.

[4] F. Torabi, G. Warnell and P. Stone, "*Generative adversarial imitation from observation*," arXiv preprint arXiv: 1807.06158 (2018).

[5] H. Karnan, G. Warnell, X. S. Xiao and P. Stone, "VOILA: Visual-Observation-Only Imitation Learning for Autonomous Navigation," *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA (2022) pp. 2497–2503.

[6] R. Shah and V. Kumar, "RRL: Resnet as Representation for Reinforcement Learning," *2021 In International Conference on Machine Learning (ICML)*, (2021) pp. 9465–9476.

[7] E. Cole, X. Yang, K. Wilber, O. M. Aodha and S. Belongie, "When Does Contrastive Visual Representation Learning Work?," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA (2022) pp.14755–14764.

[8] N. Saito, T. Ogata, S. Funabashi, H. Mori and S. Sugano, "How to select and use tools? : Active perception of target objects using multimodal deep learning," *IEEE Robot. Autom. Lett.* **6**(2), 2517–2524 (2021).

[9] D. Zhang, R. Ju and Z. Cao, “Reinforcement learning-based motion control for snake robots in complex environments,” *Robotica* **42**(4), 947–961 (2024).

[10] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal and S. Levine, “Time-Contrastive Networks: Self-Supervised Learning from Video,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia (2018) pp.1134–1141.

[11] F. Torabi, G. Warnell and P. Stone, “Behavioral Cloning from Observation,” *27th International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden (2018), pp. 4950–4957.

[12] K. Cobbe, O. Klimov, C. Hesse, T. Kim and J. Schulman, “Quantifying Generalization in Reinforcement Learning,” *International Conference on Machine Learning (ICML)*, Long Beach, CA, USA (1289) pp. 1282–1289.

[13] Y. Liu, A. Gupta, P. Abbeel and S. Levine, “Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, *Brisbane, QLD, Australia*, (1125) pp. 1118–1125.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial networks,” *Commun. Acm.* **63**(11), 139–144 (2020).

[15] E. Torabi, S. Geiger, G. Warnell and P. Stone, “Sample-efficient adversarial imitation learning from observation,” *J. Mach. Learn. Res.* **25**(31), 1–32 (2024).

[16] M. Lee, M. Tan, Y. Zhu and J. Bohg, “Detect, Reject, Correct: Crossmodal Compensation of Corrupted Sensors,” *IEEE International Conference on Robotics and Automation (ICRA)*, *Xian, China* (2021) pp. 909–916.

[17] J. F. Tremblay, T. Manderson, A. Noca, G. Dudek and D. Meger, “Multimodal dynamics modeling for off-road autonomous vehicles,” *2021 IEEE International Conference on Robotics and Automation (ICRA)*, *Xian, China* (1802) pp. 1796–1802.

[18] Q. M. Marwan, S. C. Chua and L. C. Kwek, “Comprehensive review on reaching and grasping of objects in robotics,” *Robotica* **39**(10), 1849–1882 (2021).

[19] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon and I. Havoutis, “Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control,” *IEEE Trans. Robot.* **38**(5), 2908–2927 (2022).

[20] L. Brunke, M. Greeff, A. W. Hall, Z. C. Yuan, S. Q. Zhou, J. Panerati and A. P. Schoellig, “Safe learning in robotics: From learning-based control to safe reinforcement learning,” *Annu. Rev. Contr. Robot. Auton. Sys.* **5**(1), 411–444 (2022).

[21] P. Saha, Y. Liu, B. Gick and S. Fels, “Ultra2Speech – A Deep Learning Framework for Formant Frequency Estimation and Tracking from Ultrasound Tongue Images,” In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference*, Lima, Peru, October 4–8, Proceedings, Part III 23, Springer International Publishing (2020) pp. 473–482.

[22] X. Chen and K. He. “Exploring Simple Siamese Representation Learning,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021) 15750–15758.

[23] D. Yi, J. Ahn and S. Ji, “An effective optimization method for machine learning based on ADAM,” *Appl. Sci.* **10**(3), 1073 (2020).

[24] Y. Gu, Y. H. Cheng, C. L. P. Chen and X. S. Wang, “Proximal policy optimization with policy feedback,” *IEEE Trans. Sys. Man. Cyber Syst.* **52**(7), 4600–4610 (2021).

[25] J. Schulman, S. Levine, P. Moritz, M. Jordan and P. Abbeel, “Trust Region Policy Optimization,” *International Conference on Machine Learning (ICML)*, *Lille, France* (1897) pp. 1889–1897.

[26] T. Li, Y. Wu, X. Cui, H. Dong, F. Fang and S. Russell, “Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient,” *Proc. Sym. Edu. Adva. Artifi. Intel. (AAAI)* **33**(01), 4213–4220 (2019).

[27] R. Yuan, F. Zhang, Y. Wang, Y. Fu and S. Wang, “A Q-learning approach based on human reasoning for navigation in a dynamic environment,” *Robotica* **37**(3), 445–468 (2019).

[28] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv preprint arXiv: 1609.04747, (2016).

[29] B. Yu and D. Tao, “Heatmap regression via randomized rounding,” *IEEE Trans. Pattern Anal.* **44**(11), 8276–8289 (2021).

[30] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008).

[31] C. Wang, H. Duan and L. Li, “Design, simulation, control of a hybrid pouring robot: Enhancing automation level in the foundry industry,” *Robotica* **42**(4), 1018–1038 (2024).

Cite this article: X. Xu, M. You, H. Zhou and B. He (2024). “Robot imitation from multimodal observation with unsupervised cross-modal representation”, Robotica **42**, 3247–3262. https://doi.org/10.1017/S0263574724000626