A Novel Teacher-Assistance-Based Method to Detect and Handle Bad Training Demonstrations in Learning from Demonstration

Qin Li and Yong Wang, *Senior Member*, *IEEE*

***Abstract*—Learning from demonstration (LfD) assists robots to derive a policy from a training set to execute a task. The training set consists of training demonstrations collected from a human who executes the same task. However, due to the human's varied skill level, the quality of the training set may be bad, which will affect the accuracy of the derived policy. To solve this problem, this paper proposes a novel method to improve the quality of the training set. This method includes two steps, namely detecting and handling bad training demonstrations in the training set. In the detecting step, a reference set containing reference demonstrations is provided by a human teacher. Based on the reference set, we calculate the influence of each training demonstration on the policy derivation. If the influence is negative, the corresponding training demonstration is bad. Afterward, in the handling step, we calculate the proportion of the negative influence with respect to the overall influence and reduce the proportion by iteratively removing bad training demonstrations until it is less than a threshold. The results show that the accuracy of a policy derived from the improved training set increases with up to 19.30%, which verifies the effectiveness of our method.**

***Index Terms*—Learning from demonstration, robot learning, teacher assistance, bad training demonstration.**

I. Introduction

Robots assist or replace human work by completing human-given tasks, which can effectively reduce the human labor. To complete a human-given task, a robot needs to acquire the corresponding policy. A policy is a type of mapping model from environmental states to actions. Environmental states refer to the task-related environmental information obtained by a robot through sensors. Actions are performed by the robot according to the environmental states. Based on the policy, the robot can perform appropriate actions and complete the task. To acquire a policy, two aspects should be considered: 1) designing a mapping model as the policy; and 2) calculating the values of parameters in this policy. Note that the second aspect is also regarded as policy derivation. In general, the first aspect is related to the task characteristics. In other words, if a task is given, the mapping model is deterministic. Therefore, the first aspect is easy to achieve. However, the mapping model always has a complex structure and a large number of parameters, which makes it difficult to derive the policy. To address this issue, a new research field is proposed, which is called robot learning. Robot learning is an intersection of statistic learning theory and robotics, which designs a policy based on a statistic learning model and derives the policy by using an optimization algorithm [1], [2]. Within this research field, learning from demonstration (LfD) is an effective method, which requires a human to imitate the way that a robot performs a task (i.e., performs corresponding actions according to environmental states) [3]–[7]. This process will generate a series of state-action pairs. Each state-action pair is regarded as a demonstration. All demonstrations build a training set. We refer to these demonstrations as training demonstrations. The training set is used for policy derivation.

The quality of a training set is determined by the quality of training demonstrations. The quality of a training demonstration is related to the accuracy of the action corresponding to the environmental state. If the action is not accurate, then the training demonstration is considered bad. For example, in autonomous vehicle control, a training demonstration contains an environment state describing the road information such as whether there are obstacles, pedestrians, etc., and an autonomous vehicle's action such as turning, moving, or stopping. Assuming that there are obstacles or pedestrians on a road, if the corresponding action is moving, then the training demonstration will be of poor quality. Bad training demonstrations will decrease the quality of the training set and the performance of LfD, resulting in poor accuracy of the derived policy.

To solve this problem, many methods have been proposed, which can be divided into two types. The first type utilizes LfD to derive a policy from a training set with unknown quality, and improves the policy accuracy based on reinforcement learning (RL) [8]–[12]. Specifically, RL helps a robot to explore new environmental states, and the robot performs actions corresponding to the new environmental states based on the policy. Then, RL provides a reward feedback for each action to indicate if the action is correct or not. Afterward, the reward feedbacks are used to update the parameters in the policy, thus increasing the policy accuracy. The representative work in this type is [8], in which Abbeel *et al.* used LfD to derive a policy for an autonomous helicopter, and then increased the policy accuracy based on a RL algorithm named finite-state Markov decision process. However, in typical RL algorithms, reward feedbacks are calculated by reward functions. Designing a reward function is always difficult and requires considerable

This work was supported in part by the National Natural Science Foundation of China under Grant 61976225, and in part by the Fundamental Research Funds for the Central Universities of Central South University. (*Corresponding author: Yong Wang*)

The authors are with the School of Automation, Central South University, Changsha 410083, China, and also with the Hunan Xiangjiang Artificial Intelligence Academy, Changsha 410083, China. (Email: liqin6@csu.edu.cn, ywang@csu.edu.cn)

RL-type expertise. To overcome this shortcoming, some methods recruit a human teacher to judge the performance of a robot completing a task and regard the judgements as reward feedbacks [13]–[20]. Note that the human teacher has a high skill level and the judgements from the human teacher are always correct. For example, Argall *et al.* [13] used LfD to derive a policy of a task, i.e., intercepting a moving ball, and recruited a human teacher to give reward feedbacks by judging whether a robot intercepts the ball. These reward feedbacks are then used to increase the policy accuracy. MacGlashan *et al.* [16] proposed a RL algorithm named COACH, which provides better representations of reward feedbacks from a human teacher. Overall, the first type of method needs robots to explore new states in the environment. However, during this process, dangerous environmental states may be explored, which would cause fatal accidents in safety-sensitive tasks like autonomous vehicle control.

The second type focuses on improving the quality of training sets. In this type, some methods use either multiple repeated training demonstrations [21], [22] or training demonstrations from multiple humans [23], [24]. By doing this, the number of training demonstrations increases and the influence of bad training demonstrations on the quality of the training set can be reduced. However, in these methods, the quality of repeated training demonstrations may be bad or the skill levels of multiple humans may be low, which will actually degrade the quality of the training set. Unlike these methods, some other methods detect and handle bad training demonstrations relying on the assistance from the human teacher [25]–[27]. Such methods are always called teacher-assistance-based methods. For example, Liu *et al.* [26] recruited a human teacher to record the attributes of each training demonstration. Then, they computed the matching degree between the recorded attributes and the desired ones of each training demonstration, and considered the training demonstrations with low matching degree to be bad. By removing bad training demonstrations, the quality of the training set can be improved. Beck *et al.* [27] introduced a framework in autonomous vehicle control. They recruited a human teacher to provide continuous scalar feedbacks for all training demonstrations, in which bad or dangerous training demonstrations are associated with low feedbacks. Then, all training demonstrations are weighted based on their feedbacks; thus, bad or dangerous training demonstrations could be neglected due to their low weights. Teacher-assistance-based methods improve the quality of a training set by detecting and handling bad training demonstrations, which can obtain a better quality improvement than methods based on multiple repeated training demonstrations or multiple humans. Moreover, compared with the first type of method, teacher-assistance-based methods do not explore new environmental states for policy improvement, therefore avoiding the attacks from dangerous environments.

However, current teacher-assistance-based methods still have some problems. First, they establish standards based on task attributes to judge the quality of training demonstrations. Unfortunately, different tasks have different attributes; thus, the corresponding standards may differ a lot. As a consequence, there is no general standard which can be used in different tasks. Second, they always recruit human teachers to analyze the characteristics of each training demonstration to detect bad ones, leading to a high labor cost.

Based on the above considerations, a novel teacher-assistance-based method is proposed, which consists of two steps: detecting and handling bad training demonstrations in a training set. The main contributions of this paper can be summarized as follows:

- In the detecting step, we calculate the influence of a training demonstration on the policy derivation as the standard to judge its quality. Specifically, the policy of a task is first designed and derived from the training set. Then, a human teacher is recruited to execute the task to provide a reference set which includes a small number of reference demonstrations. Based on the reference set, we calculate the reference loss of the policy. Afterward, we estimate the change of the reference loss when removing a training demonstration from the training set. This change can reflect the influence of the training demonstration on the policy derivation. If the influence is negative, it means that the training demonstration has bad quality. The advantage of using the influence as the standard is that it is not related to task attributes and can be used in different tasks. Furthermore, unlike current teacher-assistance-based methods that recruit human teachers to analyze all training demonstrations, our method only requires the human teacher to provide a small reference set, thus reducing the labor cost.
- In the handling step, we remove bad training demonstrations from the training set. Considering that directly removing all bad training demonstrations will reduce the generality of the training set, an easy and effective handling framework is proposed to selectively removing bad training demonstrations. In this framework, the proportion of the negative influence with respect to the overall influence is firstly calculated. Then, the proportion is reduced by iteratively removing bad training demonstrations. When the proportion drops below a threshold, the iterative removal operation stops, and the remaining training demonstrations constitute the improved training set. Based on this framework, most bad training demonstrations are removed, which improves the quality of the training set. Meanwhile, a small portion of bad training demonstrations with small negative influence is kept, which enhances the generality of the training set.
- The effectiveness of our method is validated by a classical LfD-based task: behavior imitation, in which a performer shows a behavior, and a Nao robot imitates the behavior. A series of experiments validates that our method has the capability to improve the quality of the training set.

II. Proposed Method

Our method consists of two steps: detecting and handling bad training demonstrations in a training set. In what follows, we will introduce these two steps in detail.

A. Detecting Bad Training Demonstrations

Training demonstrations are used to derive a policy, so the quality of training demonstrations greatly affects the policy derivation. Therefore, we regard the influence on the policy derivation as the standard to judge the quality of training demonstrations. Next, we introduce how to calculate the influence.

Given a task, a human is recruited to provide a training set: $\boldsymbol{Z}_{\text{tr}} = \{\boldsymbol{z}_{\text{tr}}^1, \ldots, \boldsymbol{z}_{\text{tr}}^n, \ldots, \boldsymbol{z}_{\text{tr}}^N\}$, where N indicates the size of $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{z}_{\text{tr}}^n = (s_{\text{tr}}^n, a_{\text{tr}}^n)$ is the nth training demonstration, and s_{tr}^n and a_{tr}^n are the nth environmental state and the corresponding human's action, respectively. Meanwhile, a policy of the task is designed, denoted as $\pi_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_k, \ldots, \theta_K\}$ indicates the parameter set and K is the number of parameters.

Based on $\boldsymbol{Z}_{\text{tr}}$, we firstly implement the policy derivation, i.e., minimize the training loss of $\pi_{\boldsymbol{\theta}}$ to obtain the optimal parameter set $\hat{\boldsymbol{\theta}} = \{\hat{\theta}_1, \ldots, \hat{\theta}_k, \ldots, \hat{\theta}_K\}$, where

$$\hat{\theta}_k = \arg\min_{\theta_k \in \boldsymbol{\Theta}_k} L_{\text{tr}} \tag{1}$$

In (1), $\boldsymbol{\Theta}_k$ is a value set of θ_k, which includes all possible values of θ_k. L_{tr} is the training loss obtained by calculating the mean of losses of all training demonstrations:

$$L_{\text{tr}} = \frac{1}{N}\sum_{n=1}^{N} L(\boldsymbol{z}_{\text{tr}}^n, \boldsymbol{\theta}) \tag{2}$$

where $L(\boldsymbol{z}_{\text{tr}}^n, \boldsymbol{\theta})$ is the loss of $\boldsymbol{z}_{\text{tr}}^n$, i.e, $L(\boldsymbol{z}_{\text{tr}}^n, \boldsymbol{\theta}) = f_{\text{loss}}(\pi_{\boldsymbol{\theta}}(s_{\text{tr}}^n), a_{\text{tr}}^n)$. $\pi_{\boldsymbol{\theta}}(s_{\text{tr}}^n)$ indicates the action obtained by $\pi_{\boldsymbol{\theta}}$ with respect to s_{tr}^n. f_{loss} is a loss function such as the cross-entropy loss function [28] or the hinge loss function [29], aiming to calculate the difference between the obtained action and the corresponding human's action. The bigger the difference, the higher the loss. After implementing the above process, we get the derived policy denoted as $\pi_{\hat{\boldsymbol{\theta}}}$.

Then, we calculate the reference loss of $\pi_{\hat{\boldsymbol{\theta}}}$. Specifically, a human teacher is recruited to execute the task and provide a reference set $\boldsymbol{Z}_{\text{ref}}$. $\boldsymbol{Z}_{\text{ref}}$ includes M demonstrations, i.e., $\boldsymbol{Z}_{\text{ref}} = \{\boldsymbol{z}_{\text{ref}}^1, \ldots, \boldsymbol{z}_{\text{ref}}^m, \ldots, \boldsymbol{z}_{\text{ref}}^M\}$. $\boldsymbol{z}_{\text{ref}}^m$ is composed of the mth environmental state (denoted as s_{ref}^m) and the corresponding human teacher's action (denoted as a_{ref}^m). We call such demonstrations as reference demonstrations. Note that M is approximately one-third of N. After obtaining $\boldsymbol{Z}_{\text{ref}}$, we calculate the reference loss of $\pi_{\hat{\boldsymbol{\theta}}}$ as follows:

$$L_{\text{ref}} = \frac{1}{M}\sum_{m=1}^{M} L(\boldsymbol{z}_{\text{ref}}^m, \hat{\boldsymbol{\theta}}) \tag{3}$$

where $L(\boldsymbol{z}_{\text{ref}}^m, \hat{\boldsymbol{\theta}})$ is the loss of $\boldsymbol{z}_{\text{ref}}^m$, i.e., $L(\boldsymbol{z}_{\text{ref}}^m, \hat{\boldsymbol{\theta}}) = f_{\text{loss}}(\pi_{\hat{\boldsymbol{\theta}}}(s_{\text{ref}}^m), a_{\text{ref}}^m)$. Obviously, L_{ref} can indicate the policy accuracy. The smaller the value of L_{ref}, the higher the accuracy of $\pi_{\hat{\boldsymbol{\theta}}}$.

Afterward, to calculate the influence of a training demonstration (denoted as $\hat{\boldsymbol{z}}_{\text{tr}}$), we remove $\hat{\boldsymbol{z}}_{\text{tr}}$ from $\boldsymbol{Z}_{\text{tr}}$ and the optimal parameter set becomes $\hat{\boldsymbol{\theta}}_{-\hat{\boldsymbol{z}}_{\text{tr}}} = \{\hat{\theta}_{-\hat{\boldsymbol{z}}_{\text{tr}},1}, \ldots, \hat{\theta}_{-\hat{\boldsymbol{z}}_{\text{tr}},k}, \ldots, \hat{\theta}_{-\hat{\boldsymbol{z}}_{\text{tr}},K}\}$, where

$$\hat{\theta}_{-\hat{\boldsymbol{z}}_{\text{tr}},k} = \arg\min_{\theta_k \in \boldsymbol{\Theta}_k} \frac{1}{N-1}\sum_{n=1,\boldsymbol{z}_{\text{tr}}^n \neq \hat{\boldsymbol{z}}_{\text{tr}}}^{N} L(\boldsymbol{z}_{\text{tr}}^n, \boldsymbol{\theta}) \tag{4}$$

As a result, the reference loss becomes:

$$L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} = \frac{1}{M}\sum_{m=1}^{M} L(\boldsymbol{z}_{\text{ref}}^m, \hat{\boldsymbol{\theta}}_{-\hat{\boldsymbol{z}}_{\text{tr}}}) \tag{5}$$

Based on L_{ref} and $L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$, the change of the reference loss after removing $\hat{\boldsymbol{z}}_{\text{tr}}$ from $\boldsymbol{Z}_{\text{tr}}$ can be calculated:

$$\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} = L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} - L_{\text{ref}} \tag{6}$$

It is clear that $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ also indicates the change of the policy accuracy after removing $\hat{\boldsymbol{z}}_{\text{tr}}$ from $\boldsymbol{Z}_{\text{tr}}$, which can reflect the influence of $\hat{\boldsymbol{z}}_{\text{tr}}$ on the policy derivation. However, calculating $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ based on (6) needs to derive both $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}_{-\hat{\boldsymbol{z}}_{\text{tr}}}$, which will lead to a high time cost. To solve this problem, inspired by [30], we weight $\hat{\boldsymbol{z}}_{\text{tr}}$ by a parameter ϵ and the optimal parameter set becomes $\hat{\boldsymbol{\theta}}_{\epsilon\hat{\boldsymbol{z}}_{\text{tr}}} = \{\hat{\theta}_{\epsilon\hat{\boldsymbol{z}}_{\text{tr}},1}, \ldots, \hat{\theta}_{\epsilon\hat{\boldsymbol{z}}_{\text{tr}},k}, \ldots, \hat{\theta}_{\epsilon\hat{\boldsymbol{z}}_{\text{tr}},K}\}$, where

$$\hat{\theta}_{\epsilon\hat{\boldsymbol{z}}_{\text{tr}},k} = \arg\min_{\theta_k \in \boldsymbol{\Theta}_k} \frac{1}{N}[\sum_{n=1,\boldsymbol{z}_{\text{tr}}^n \neq \hat{\boldsymbol{z}}_{\text{tr}}}^{N} L(\boldsymbol{z}_{\text{tr}}^n, \boldsymbol{\theta}) + \epsilon L(\hat{\boldsymbol{z}}_{\text{tr}}, \boldsymbol{\theta})] \tag{7}$$

Then, we consider $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ as the derivative of L_{ref} with respect to ϵ when ϵ equals 0:

$$\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} \approx \frac{\mathrm{d}L_{\text{ref}}}{\mathrm{d}\epsilon}|_{\epsilon=0} \tag{8}$$

According to [30], $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ can be calculated by:

$$\begin{aligned}\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} &\approx \frac{\mathrm{d}L_{\text{ref}}}{\mathrm{d}\epsilon}|_{\epsilon=0} = \frac{1}{M}\sum_{m=1}^{M} \nabla_{\hat{\boldsymbol{\theta}}} L(\boldsymbol{z}_{\text{ref}}^m, \hat{\boldsymbol{\theta}})^\top \frac{\hat{\boldsymbol{\theta}}_{\epsilon\hat{\boldsymbol{z}}_{\text{tr}}}}{\mathrm{d}\epsilon}|_{\epsilon=0} \\ &= -\frac{1}{M}\sum_{m=1}^{M} \nabla_{\hat{\boldsymbol{\theta}}} L(\boldsymbol{z}_{\text{ref}}^m, \hat{\boldsymbol{\theta}})^\top \boldsymbol{H}_{\hat{\boldsymbol{\theta}}}^{-1} \nabla_{\hat{\boldsymbol{\theta}}} L(\hat{\boldsymbol{z}}_{\text{tr}}, \hat{\boldsymbol{\theta}})\end{aligned} \tag{9}$$

where

$$\boldsymbol{H}_{\hat{\boldsymbol{\theta}}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\mathrm{d}^2 L(\boldsymbol{z}_{\text{tr}}^n, \hat{\boldsymbol{\theta}})}{\mathrm{d}\hat{\boldsymbol{\theta}}^2} \tag{10}$$

Calculating $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ based on (9) only needs $\hat{\boldsymbol{\theta}}$, which means that the policy derivation is implemented only once, so the time cost is effectively reduced. It is necessary to note that calculating $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ based on [30] is only applicable to the differentiable policy.

After obtaining $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$, we use $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ to indicate the influence of $\hat{\boldsymbol{z}}_{\text{tr}}$ on the policy derivation. Specifically, if $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} > 0$, it means that the accuracy of $\pi_{\hat{\boldsymbol{\theta}}}$ decreases after removing $\hat{\boldsymbol{z}}_{\text{tr}}$ from $\boldsymbol{Z}_{\text{tr}}$. Therefore, $\hat{\boldsymbol{z}}_{\text{tr}}$ has the positive influence on the policy derivation. The value of $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ indicates the positive influence level of $\hat{\boldsymbol{z}}_{\text{tr}}$. If $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} = 0$, it means that $\hat{\boldsymbol{z}}_{\text{tr}}$ has no influence on the policy derivation. If $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}} < 0$, it means that the accuracy of $\pi_{\hat{\boldsymbol{\theta}}}$ increases after removing $\hat{\boldsymbol{z}}_{\text{tr}}$ from $\boldsymbol{Z}_{\text{tr}}$. Thus, $\hat{\boldsymbol{z}}_{\text{tr}}$ has the negative influence on the policy derivation. The absolute value of $\Delta L_{\text{ref},-\hat{\boldsymbol{z}}_{\text{tr}}}$ indicates the negative influence level of $\hat{\boldsymbol{z}}_{\text{tr}}$.

Based on the above process, the influence of all training demonstrations can be calculated. We use I_{tr}^n to indicate the influence of the nth training demonstration $\boldsymbol{z}_{\text{tr}}^n$, i.e.,

$$I_{\text{tr}}^n \overset{\text{def}}{=} \Delta L_{\text{ref},-\boldsymbol{z}_{\text{tr}}^n} \tag{11}$$

Algorithm 1 *Handling Bad Training Demonstrations*

1: **Input:** $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{I}_{\text{tr,neg}}$, and $\boldsymbol{I}_{\text{tr,pos}}$
2: Calculate r by (12);
3: Set a threshold α;
4: **if** $0 < r \leq \alpha$ **then**
5: $\quad \hat{\boldsymbol{Z}}_{\text{tr}} \leftarrow \boldsymbol{Z}_{\text{tr}}$
6: **else**
7: $\quad$ Sort the elements in $\boldsymbol{I}_{\text{tr,neg}}$ in the ascending order and obtain a new sequence $\hat{\boldsymbol{I}}_{\text{tr,neg}}$;
8: $\quad$ **while** $r > \alpha$ **do**
9: $\quad\quad$ Remove the first element in $\hat{\boldsymbol{I}}_{\text{tr,neg}}$;
10: $\quad\quad$ Calculate r by using the remaining elements in $\hat{\boldsymbol{I}}_{\text{tr,neg}}$ and all elements in $\boldsymbol{I}_{\text{tr,pos}}$ based on (12);
11: $\quad$ **end while**
12: $\quad$ Build $\hat{\boldsymbol{Z}}_{\text{tr}}$ by integrating the training demonstrations corresponding to the remaining elements in $\hat{\boldsymbol{I}}_{\text{tr,neg}}$ and all elements in $\boldsymbol{I}_{\text{tr,pos}}$;
13: **end if**
14: **Output:** $\hat{\boldsymbol{Z}}_{\text{tr}}$

Let $\boldsymbol{I}_{\text{tr}} = [I_{\text{tr}}^1, \ldots, I_{\text{tr}}^n, \ldots, I_{\text{tr}}^N]$ be the influence sequence. We consider training demonstrations with the negative influence as bad training demonstrations.

B. Handling Bad Training Demonstrations

Having obtained bad training demonstrations, we then handle them to improve the quality of $\boldsymbol{Z}_{\text{tr}}$. A straightforward idea is to remove all bad training demonstrations from $\boldsymbol{Z}_{\text{tr}}$. However, in this way, the generality of $\boldsymbol{Z}_{\text{tr}}$ will decrease. The reasons are the following. On the one hand, detecting bad training demonstrations relies on the change of the reference loss, which is calculated based on $\boldsymbol{Z}_{\text{ref}}$. $\boldsymbol{Z}_{\text{ref}}$ consists of reference demonstrations generated by a human teacher who performs suitable actions with respect to the environmental states encountered in a task. However, it is impossible for the human teacher to encounter all possible environmental states in the task, so the generated reference demonstrations are not fully diverse, which will limit the generality of $\boldsymbol{Z}_{\text{ref}}$. On the other hand, the training demonstrations with the positive influence are similar to the reference demonstrations while the training demonstrations with the negative influence are different from the reference demonstrations. If all bad training demonstrations are removed, the improved $\boldsymbol{Z}_{\text{tr}}$ will only include the training demonstrations with the positive influence, thus resulting in the improved $\boldsymbol{Z}_{\text{tr}}$ and $\boldsymbol{Z}_{\text{ref}}$ being similar. Since $\boldsymbol{Z}_{\text{ref}}$ has limited generality, the generality of the improved $\boldsymbol{Z}_{\text{tr}}$ will also be limited.

To solve this problem, an easy and effective framework is proposed to enhance the generality of $\boldsymbol{Z}_{\text{tr}}$ while improving its quality. **Algorithm 1** shows the proposed framework. Let $\hat{\boldsymbol{Z}}_{\text{tr}}$ be the improved $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{I}_{\text{tr,pos}} = [I_{\text{tr,pos}}^1, \ldots I_{\text{tr,pos}}^n, \ldots, I_{\text{tr,pos}}^{N_{\text{pos}}}]$ be the positive influence sequence, and $\boldsymbol{I}_{\text{tr,neg}} = [I_{\text{tr,neg}}^1, \ldots, I_{\text{tr,neg}}^n, \ldots, I_{\text{tr,neg}}^{N_{\text{neg}}}]$ be the negative influence sequence. N_{pos} and N_{neg} are the sizes of $\boldsymbol{I}_{\text{tr,pos}}$ and $\boldsymbol{I}_{\text{tr,neg}}$, respectively, and $N_{\text{pos}} + N_{\text{neg}} = N$. First, we calculate the proportion of the negative influence with respect to the overall influence (line 2):

$$r = \frac{\sum_{n=1}^{N_{\text{neg}}} |I_{\text{tr,neg}}^n|}{\sum_{n=1}^{N_{\text{neg}}} |I_{\text{tr,neg}}^n| + \sum_{n=1}^{N_{\text{pos}}} I_{\text{tr,pos}}^n} \tag{12}$$

Then, we define a negative influence threshold α, which is the upper limit of r (line 3). Section III-A will give an analysis of α. Afterward, we compare r with α. If $0 < r \leq \alpha$, it means that bad training demonstrations in $\boldsymbol{Z}_{\text{tr}}$ have small negative influence. Moreover, the existence of such bad training demonstrations can reduce the similarity between $\boldsymbol{Z}_{\text{tr}}$ and $\boldsymbol{Z}_{\text{ref}}$, thus enhancing the generality of $\boldsymbol{Z}_{\text{tr}}$. Therefore, we do not remove any bad training demonstration. Under this condition, $\hat{\boldsymbol{Z}}_{\text{tr}}$ is equal to $\boldsymbol{Z}_{\text{tr}}$ (lines 4 and 5). It should be noted that in general, there is no training set with $r = 0$, so we do not consider this situation. If $r > \alpha$, it means that there are some bad training demonstrations that have high negative influence and should be removed. To find and remove such bad training demonstrations, we execute iterative removal operation (lines 8 to 11). Specifically, all elements in $\boldsymbol{I}_{\text{tr,neg}}$ are sorted in the ascending order and a new sequence $\hat{\boldsymbol{I}}_{\text{tr,neg}}$ is obtained. Then, each element in $\hat{\boldsymbol{I}}_{\text{tr,neg}}$ is removed in turn. After each removal, we calculate r based on the remaining elements in $\hat{\boldsymbol{I}}_{\text{tr,neg}}$ and all elements in $\boldsymbol{I}_{\text{tr,pos}}$ according to (12). When r drops below α, the iterative removal operation stops, and $\hat{\boldsymbol{Z}}_{\text{tr}}$ is built by integrating training demonstrations corresponding to the remaining elements in $\hat{\boldsymbol{I}}_{\text{tr,neg}}$ and all elements in $\boldsymbol{I}_{\text{tr,pos}}$ (line 12).

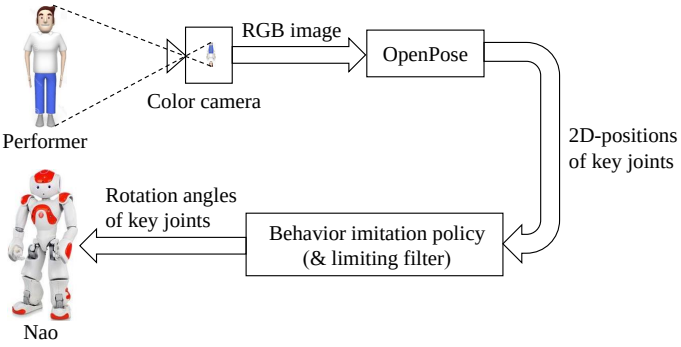


Fig. 1. Illustration of behavior imitation.

III. Experimental Studies

We validated our method in a classical LfD-based task: behavior imitation, that is, a performer shows a behavior and a Nao robot imitates the behavior. The specific process of the task is shown in Fig. 1. First, a performer was recruited to show a behavior which was recorded as an RGB image by a color camera. Then, the RGB image was fed into OpenPose[1] to calculate the 2D-positions of the performer's key joints in the RGB image. The key joints of the performer refer to the joints that both the performer and the Nao robot have. Note that to reduce the difficulty of the task, the performer only showed the upper body behaviors. As a result, the involved key joints were 'LShoulder', 'RShoulder', 'LElbow', and 'RElbow', where 'L'

[1]https://github.com/CMU-Perceptual-Computing-Lab/openpose

TABLE I
ROTATION ANGLES OF 'LSHOULDER', 'RSHOULDER', 'LELBOW', AND 'RELBOW'

Human joint	LShoulder	RShoulder	LElbow	RElbow
Rotation angle	LShoulderPitch LShoulderRoll	RShoulderPitch RShoulderRoll	LElbowYaw LElbowRoll	RElbowYaw RElbowRoll

TABLE II
UPPER AND LOWER LIMITS OF ALL ROTATION ANGLES

Rotation angle*	LShoulderPitch	LShoulderRoll	RShoulderPitch	RShoulderRoll
Upper limit	2.0875	1.6580	2.0875	0
Lower limit	-2.0875	0	-2.0875	-1.6580
Rotation angle	LElbowYaw	LElbowRoll	RElbowYaw	RElbowRoll
Upper limit	2.0875	0	2.0875	1.5707
Lower limit	-2.0875	-1.5707	-2.0875	0

* The values of rotation angles are in radians.

and 'R' represent 'left' and 'right', respectively. Afterward, a behavior imitation policy was designed based on ConvNet proposed in [31], which estimates the values of the key joints' rotation angles based on their 2D-positions. Table I shows the rotation angles of the key joints, which were used to control the rotation of the same key joints of the Nao robot. Meanwhile, to stabilize the estimation of the rotation angles, a limiting filter was designed, which sets a safe value range (i.e., a upper limit and a lower limit) for each rotation angle. If the calculated rotation angle is not within its safe value range, it will be set to the closest safe value. Table II shows the safe value ranges of all rotation angles. After implementing the limiting filtering, the values of all rotation angles were obtained to control the key joints of the Nao robot to rotate, thus making the Nao robot imitate the behavior shown by the performer.

In this task, the hardware platform consisted of two parts: a host computer and the Nao robot. The host computer, including Windows 10 operating system, Intel Core i7-7500 CPU @2.70 GHz, and 8 GB of RAM, captured RGB images based on its own camera and realized software programming. The host computer established a TCP connection with the Nao robot and transmitted the program instructions to the Nao robot for control. The software platform was NAOqi which programmed the Nao robot in the host computer's Windows 10 operating system. The programming language was python.

To validate our method, we first generated training demonstrations to build $\boldsymbol{Z}_{\text{tr}}$. The generation process of a training demonstration is as follows:

- Recruit the performer to show a behavior and record the behavior as an RGB image based on the color camera;
- Calculate the 2D-positions of the performer's key joints in the RGB image with OpenPose;
- Generate the rotation angles of the key joints during kinesthetic teaching (see Fig. 2(a)-(c)), in which a human controls the Nao robot to imitate the behavior shown in the RGB image to generate the rotation angles of the key joints;
- Combine the 2D-positions with the rotation angles of the key joints to form a training demonstration.

Based on the above process, we generated 7200 training demonstrations to build $\boldsymbol{Z}_{\text{tr}}$. Meanwhile, to fully validate our method, we also generated a number of misleading training

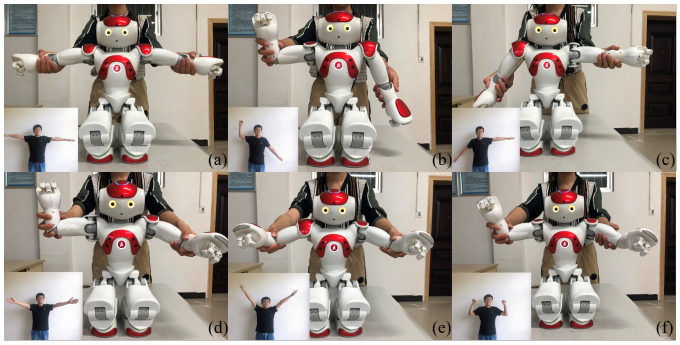


Fig. 2. Process of kinesthetic teaching, in which a human controls the Nao robot to imitate the behavior shown in an RGB image to generate the rotation angles of the key joints. Note that in (d)-(f), the human controls the Nao robot to perform wrong behaviors. As a result, wrong rotation angles are obtained and used to generate misleading training demonstrations.

TABLE III
PCR CORRESPONDING TO $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$ WHEN α VARIES FROM 0 TO 0.5.

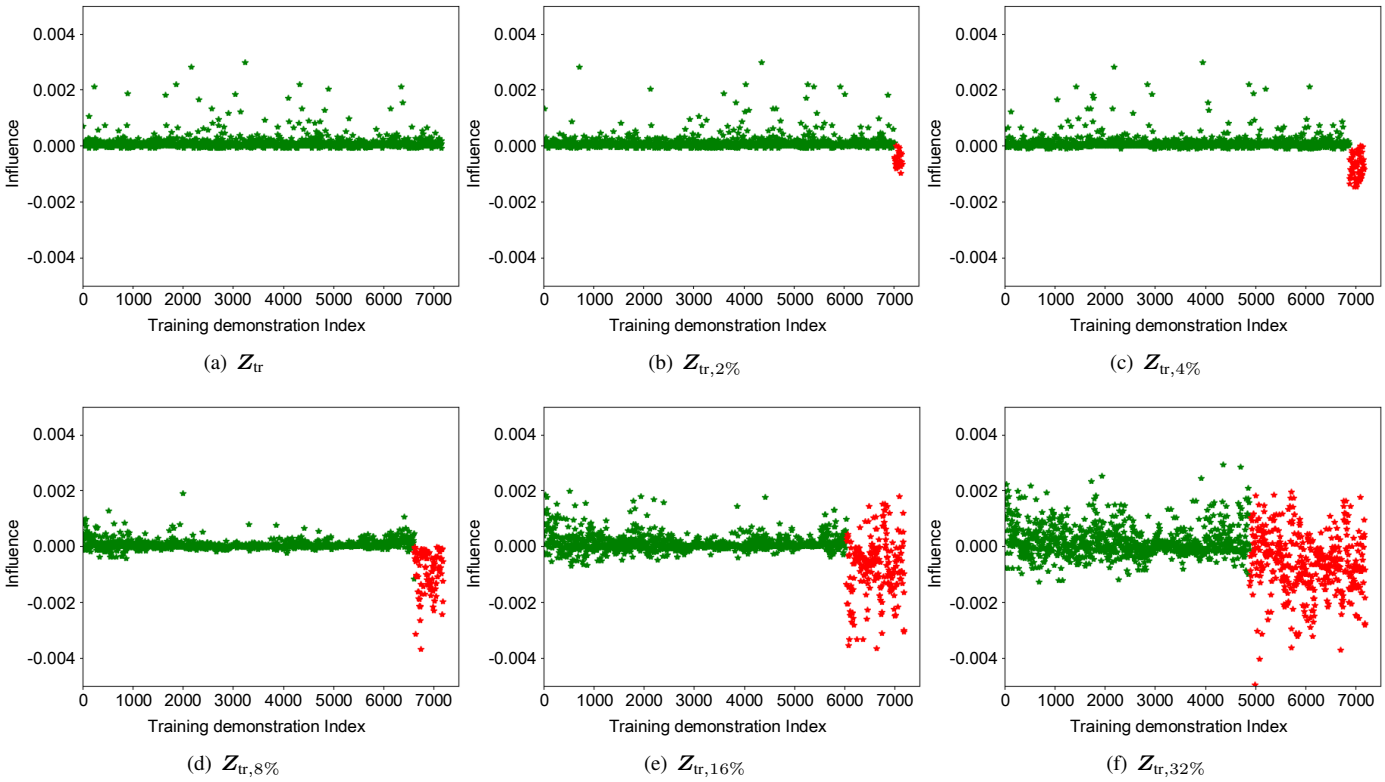
α	0	0.05	0.10	0.15	0.20	0.25
PCR[%]	82.13	83.84	**84.09**	83.02	81.92	75.24
α	0.30	0.35	0.40	0.45	0.50	
PCR[%]	75.24	75.24	75.24	75.24	75.24	

demonstrations. Specifically, the performer showed a number of behaviors which were recorded as RGB images. For each RGB image, we calculated the 2D-positions of the key joints based on OpenPose. Meanwhile, we recruited a human to control the Nao robot to perform a wrong behaviour which should be obviously different from the behaviour shown in the RGB image (see Fig. 2(d)-(f)). By doing this, the wrong rotation angles of the key joints could be obtained, which were then combined with the 2D-positions of the key joints to generate a misleading training demonstration. After generating all misleading training demonstrations, we built five new training sets, i.e., $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$. Each training set consists of a certain number of normal training demonstrations and misleading training demonstrations. The percentage represents the proportion of misleading training demonstrations in a training set. Note that these new training sets have the same size with $\boldsymbol{Z}_{\text{tr}}$.

Afterward, we built a reference set $\boldsymbol{Z}_{\text{ref}}$ and a test set $\boldsymbol{Z}_{\text{te}}$. These two building processes are similar with the process of building $\boldsymbol{Z}_{\text{tr}}$, except that during the kinesthetic teaching, a human teacher was recruited instead of a human to control the Nao robot. The size of $\boldsymbol{Z}_{\text{ref}}$ and $\boldsymbol{Z}_{\text{te}}$ were 2292 and 2214, respectively. Based on $\boldsymbol{Z}_{\text{te}}$, we calculated the Percentage of Correct Rotation-angles (PCR) to indicate the policy accuracy. PCR represents the percentage of correctly estimated test demonstrations. Note that a test demonstration is regarded to be correctly estimated only if the Euclidean distance between the estimated rotation angles and the ground-truth is less than a threshold. In this paper, the threshold was set to 0.5.

A. Analysis of α

In this section, we carried out an analysis of α based on $\boldsymbol{Z}_{\text{tr},8\%}$. First, α was set to a series of values from 0 to 0.5



(a) $\boldsymbol{Z}_{\text{tr}}$ (b) $\boldsymbol{Z}_{\text{tr},2\%}$ (c) $\boldsymbol{Z}_{\text{tr},4\%}$

(d) $\boldsymbol{Z}_{\text{tr},8\%}$ (e) $\boldsymbol{Z}_{\text{tr},16\%}$ (f) $\boldsymbol{Z}_{\text{tr},32\%}$

Fig. 3. Influence of training demonstrations in $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$. A green dot and a red dot represent the influence of a normal training demonstration and a misleading training demonstration, respectively.

with the step of 0.05. Note that $\alpha > 0.5$ means that the negative influence proportion of a training set is higher than 0.5, which is obviously unreasonable. Therefore, we did not consider this situation. Then, based on each value of α, the quality of $\boldsymbol{Z}_{\text{tr},8\%}$ was improved based on our method. Let $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$ be the improved $\boldsymbol{Z}_{\text{tr},8\%}$ based on α. To evaluate the quality of $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$, we derived the behavior imitation policy and calculated its PCR. Note that the bigger the value of PCR, the better the quality of $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$.

Table III summarizes the results. From Table III, when $0.05 \leq \alpha \leq 0.15$, PCRs corresponding to the improved training sets (i.e., $\hat{\boldsymbol{Z}}^{0.05}_{\text{tr},8\%}$, $\hat{\boldsymbol{Z}}^{0.10}_{\text{tr},8\%}$, and $\hat{\boldsymbol{Z}}^{0.15}_{\text{tr},8\%}$) are 83.84%, 84.09%, and 83.02% respectively, which are bigger than PCR corresponding to $\hat{\boldsymbol{Z}}^{0}_{\text{tr},8\%}$ (i.e., 82.13%). This is because when $0.05 \leq \alpha \leq 0.15$, a certain number of bad training demonstrations with small negative influence are kept in $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$. Thus, the generality is enhanced and better prediction accuracy is achieved. On the other hand, when $\alpha > 0.15$, more and more bad training demonstrations are retained. As a result, the quality of $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$ becomes worse and the corresponding PCR decreases gradually. Note that r of $\boldsymbol{Z}_{\text{tr},8\%}$ is 0.21. When $\alpha \geq 0.25$, no bad training demonstration is removed since $\alpha > r$ and $\hat{\boldsymbol{Z}}^{\alpha}_{\text{tr},8\%}$ is equivalent to $\boldsymbol{Z}_{\text{tr},8\%}$. Under this condition, PCR decreases to the minimum, i.e., 75.24%, and stays unchanged. It can be seen that $\alpha = 0.10$ achieves the highest performance improvement. Therefore, in the following experiments, α was set to 0.10.

B. Validation of Detecting Bad Training Demonstrations

In this section, we validated the effectiveness of our method for detecting bad training demonstrations. We used $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$ in this experiment. For each training set, we calculated the influence of each training demonstration based on our method. Fig. 3 shows the influence of all training demonstrations in the six training sets. In Fig. 3(a)–Fig. 3(d), the influence of all misleading training demonstrations is negative (as shown in the red dots), which means that all misleading training demonstrations can be detected. In Fig. 3(e) and Fig. 3(f), the influence of some misleading training demonstrations is positive. The reason may be that the large proportions of the misleading training demonstrations in $\boldsymbol{Z}_{\text{tr},16\%}$ and $\boldsymbol{Z}_{\text{tr},32\%}$ have misled the task-related statistical characteristics. Therefore, the behavior imitation policy derived from these two training sets is bad, thus resulting in incorrect calculation of the influence of some training demonstrations. However, even in this situation, the influence of most misleading training demonstrations is negative. The above results suggest that our method has the capability to detect bad training demonstrations.

We also implemented an experiment based on $\hat{\boldsymbol{Z}}_{tr,8\%}$ to ascertain whether the absolute value of the negative influence can indicate the negative influence level of a bad training demonstration. Specifically, for each misleading training demonstration in $\hat{\boldsymbol{Z}}_{tr,8\%}$, we calculated the Euclidean distance between its recorded rotation angles of the key joints and the real values. Then, we analyzed the relationship between the

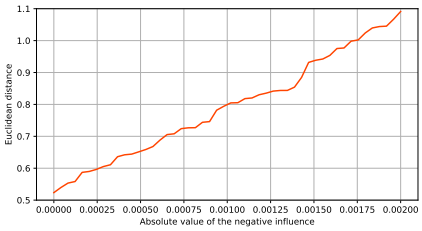


Fig. 4. Relationship between the absolute value of the negative influence and the Euclidean distance of each misleading training demonstration.

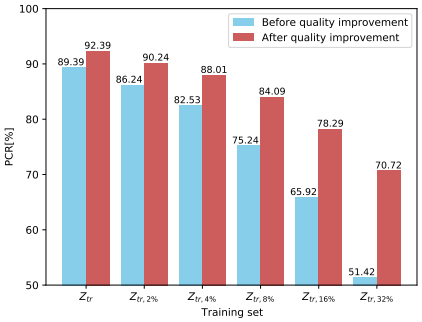


Fig. 5. PCRs corresponding to $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$ before and after the quality improvement.

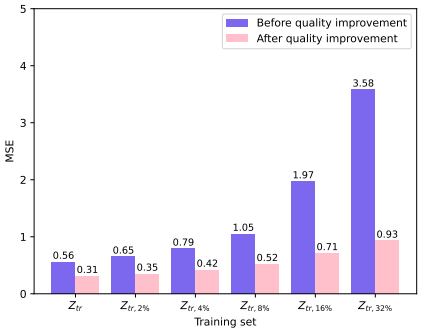
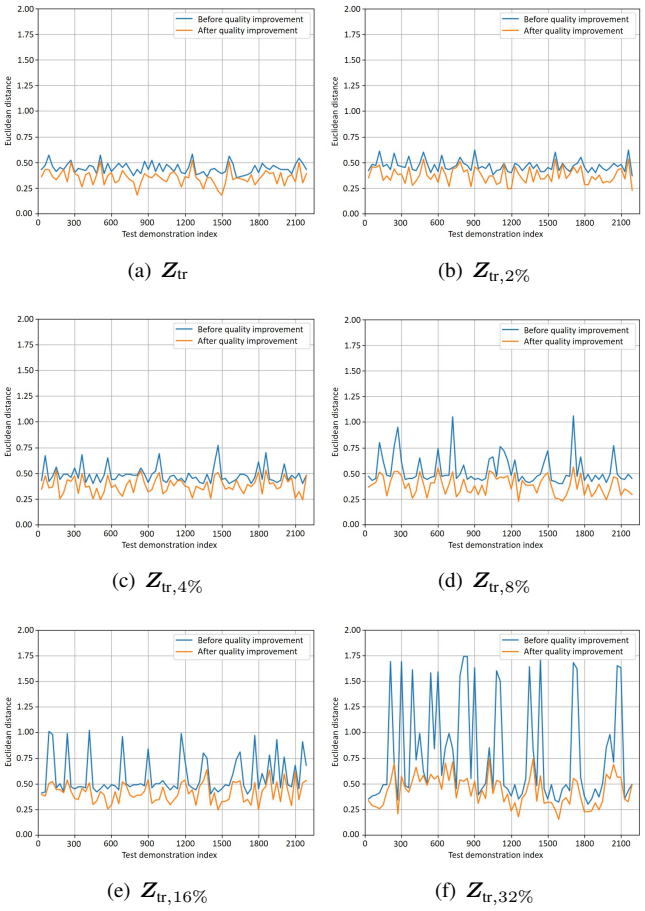


Fig. 6. MSEs corresponding to $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$ before and after the quality improvement.

absolute value of the negative influence and the Euclidean distance in Fig. 4. As shown in Fig. 4, as the Euclidean distance increases, the absolute value of the negative influence also increases, which demonstrates that the absolute value can effectively reflect the negative influence level of a bad training demonstration.

C. Validation of Improving the Quality of the Training Set

In this section, we validated the effectiveness of our method for improving the quality of a training set. $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$ were involved in this experiment. For each training set, we firstly derived the behavior imitation policy and calculated its PCR. Then, we improved the training set based on our method, implemented the policy derivation again based on the improved training set, and calculated the corresponding PCR. Fig. 5 presents the results. As can be seen, for each training set, PCR significantly increases after the quality improvement. In particular, PCR corresponding to $\boldsymbol{Z}_{\text{tr},32\%}$ increases by 19.30%. Meanwhile, we introduced the mean



(a) $\boldsymbol{Z}_{\text{tr}}$ (b) $\boldsymbol{Z}_{\text{tr},2\%}$

(c) $\boldsymbol{Z}_{\text{tr},4\%}$ (d) $\boldsymbol{Z}_{\text{tr},8\%}$

(e) $\boldsymbol{Z}_{\text{tr},16\%}$ (f) $\boldsymbol{Z}_{\text{tr},32\%}$

Fig. 7. Euclidean distance of all test demonstrations obtained based on the behavior imitation policy derived from $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$ respectively before and after the quality improvement.

squared error (MSE) between the estimated results and the ground-truth as another metric to evaluate the policy accuracy. Fig. 6 shows MSEs corresponding to $\hat{\boldsymbol{Z}}$, $\hat{\boldsymbol{Z}}_{\text{tr},2\%}$, $\hat{\boldsymbol{Z}}_{\text{tr},4\%}$, $\hat{\boldsymbol{Z}}_{\text{tr},8\%}$, $\hat{\boldsymbol{Z}}_{\text{tr},16\%}$, and $\hat{\boldsymbol{Z}}_{\text{tr},32\%}$ before and after the quality improvement. From Fig. 6, for each training set, the corresponding MSE drops after the quality improvement, which further verifies the effectiveness of our method.

Moreover, to analyze the performance of our method in depth, we presented the Euclidean distance between the estimated result and the ground-truth of each test demonstration, in which the estimated result is obtained based on the behavior imitation policies derived from a training set before and after the quality improvement. Fig. 7 shows the results of $\boldsymbol{Z}_{\text{tr}}$, $\boldsymbol{Z}_{\text{tr},2\%}$, $\boldsymbol{Z}_{\text{tr},4\%}$, $\boldsymbol{Z}_{\text{tr},8\%}$, $\boldsymbol{Z}_{\text{tr},16\%}$, and $\boldsymbol{Z}_{\text{tr},32\%}$. From Fig. 7, for each training set, the Euclidean distances of almost all test demonstrations decrease after the quality improvement. It should be noted that as the proportion of misleading training demonstrations increases, the obtained Euclidean distances of more and more test demonstrations are abnormally high (as shown in blue lines in Fig. 7(a)–Fig. 7(f)). It may be because a large proportion of misleading training demonstrations impairs the performance of the derived behavior imitation policy, thus leading to unreasonable estimations.

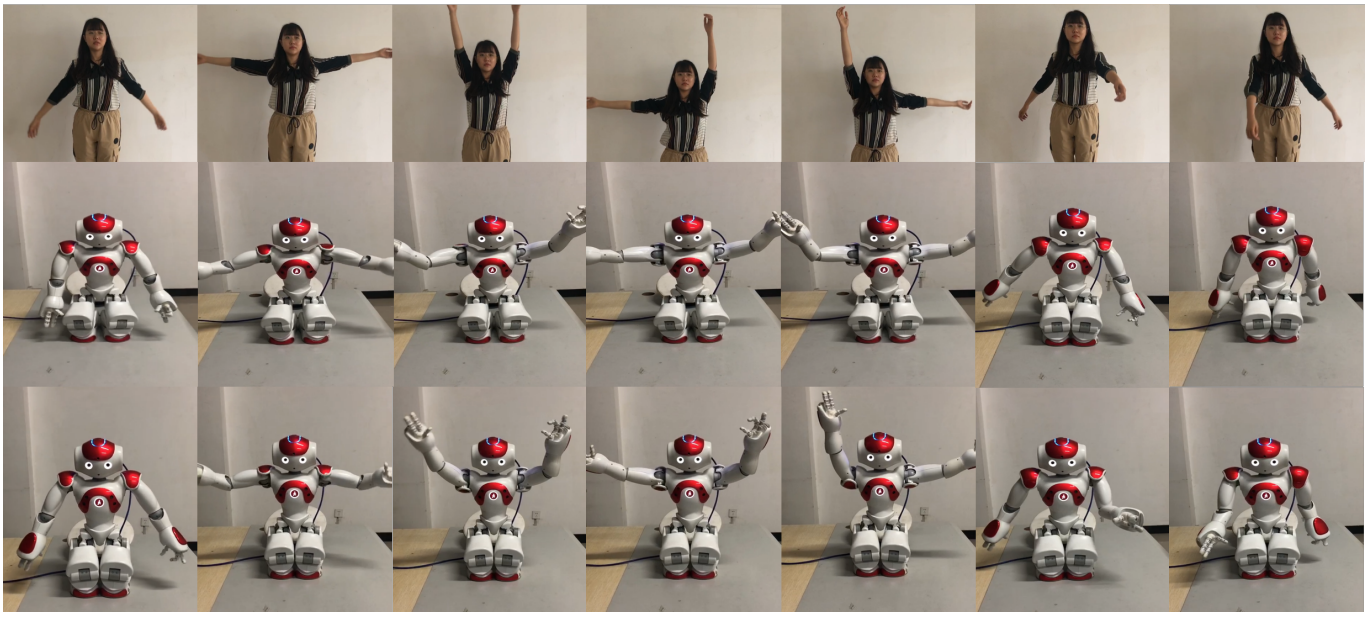


Fig. 8. Imitation results corresponding to some human behaviors. The first row shows human behaviors. The second and third rows show imitation results obtained based on $\pi_{\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}_{\mathrm{tr},8\%}}}$ and $\pi_{\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}}}$, respectively.

TABLE IV
AUC-ROCs OF THE CLASSIFIERS TRAINED BY $\hat{\boldsymbol{Z}}_{\mathrm{tr}}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},2\%}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},4\%}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},16\%}$, AND $\hat{\boldsymbol{Z}}_{\mathrm{tr},32\%}$.

Training set	$\hat{\boldsymbol{Z}}_{\mathrm{tr}}$	$\hat{\boldsymbol{Z}}_{\mathrm{tr},2\%}$	$\hat{\boldsymbol{Z}}_{\mathrm{tr},4\%}$	$\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}$	$\hat{\boldsymbol{Z}}_{\mathrm{tr},16\%}$	$\hat{\boldsymbol{Z}}_{\mathrm{tr},32\%}$
AUC-ROC	0.63	0.64	0.68	0.71	0.75	0.77

In a LfD task, there may be a covariate shift between the training set and the test set. The covariate shift implies that the training set and the test set have different distributions, which will cause that the policy derived from the training set has a poor accuracy on the test set and cannot be used in real scenes. To this end, we utilized a simple method to judge whether the training set improved based on our method encounters the covariate shift. First, we selected training demonstrations in an improved training set and test demonstrations in the test set. The numbers of the selected training and test demonstrations are the same. Then, we labeled the training demonstrations as 1 and the test demonstrations as 0. These labeled training and test demonstrations were combined to form a new dataset. In this dataset, 80% of demonstrations were selected to train a classifier (i.e., a support vector machine in this paper) and the remaining 20% demonstrations were used to calculate the area under the receiver operating characteristics (AUC-ROC) of the classifier. AUC-ROC is an indicator to measure the classification performance of the classifier. If AUC-ROC is higher than a threshold, it means that the classifier can distinguish the training and test demonstrations well. The threshold is usually set to 0.80. Based on the above method, we recorded AUC-ROCs of the classifier trained by $\hat{\boldsymbol{Z}}_{\mathrm{tr}}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},2\%}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},4\%}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}$, $\hat{\boldsymbol{Z}}_{\mathrm{tr},16\%}$, and $\hat{\boldsymbol{Z}}_{\mathrm{tr},32\%}$ in Table IV. As can be seen, AUC-ROCs of all improved training sets are lower than 0.80, which reflects that all improved training sets have similar distributions with $\hat{\boldsymbol{Z}}_{\mathrm{te}}$. In other words, these improved training sets do not encounter the covariate shift.

Furthermore, to visualize the quality improvement of the training set, we used the behavior imitation policies derived from $\boldsymbol{Z}_{\mathrm{tr},8\%}$ and $\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}$ (denoted as $\pi_{\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}_{\mathrm{tr},8\%}}}$ and $\pi_{\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}}}$) to control the Nao robot, with the aim of imitating human behaviors. Fig. 8 depicts the imitation results. The first, second, and third rows correspond to the human behaviors, the imitation results based on $\pi_{\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}_{\mathrm{tr},8\%}}}$, and the imitation results based on $\pi_{\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}}}$, respectively. As shown in Fig. 8, the imitation results in the third row are more similar to the human behaviors, which means that $\pi_{\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{Z}}_{\mathrm{tr},8\%}}}$ performs better than $\pi_{\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}_{\mathrm{tr},8\%}}}$.

IV. CONCLUSION

This paper proposed a novel teacher-assistance-based method to improve the quality of the training set used for policy derivation in LfD. This method included two steps, i.e., detecting and handling bad training demonstrations in a training set. In the detecting step, we calculated the influence of each training demonstration on the policy derivation, and selected the training demonstrations with the negative influence as bad training demonstrations. The influence is not related to task attributes and can be used in different tasks for demonstration quality evaluation. Then, in the handling step, we calculated the proportion of the negative influence with respect to the overall influence, and reduced the proportion by iteratively removing bad training demonstrations until it was less than a threshold. In this way, most bad training demonstrations are removed, which improves the quality of the training set. At the same time, a small portion of bad training demonstrations with small negative influence is kept, which enhances the generality of the improved training set. The results showed that our method could not only detect bad training demonstrations, but also improves the quality of the

training set. As mentioned before, in this paper our method is only applied to the task in which the policy is differentiable. Therefore, in the future, we will try to design methods for the tasks with non-differentiable policies.

The videos of the experiments can be downloaded from: https://intleo.csu.edu.cn/publication.html

REFERENCES

[1] J. Li, Z. Li, X. Li, Y. Feng, Y. Hu, and B. Xu, "Skill learning strategy based on dynamic motion primitives for humancrobot cooperative manipulation," *IEEE Trans. Cogn. Develop. Syst.*, vol. 13, no. 1, pp. 105–117, 2021.

[2] C. Delgrange, J. Dussoux, and P. F. Dominey, "Usage-based learning in human interaction with an adaptive virtual assistant," *IEEE Trans. Cogn. Develop. Syst.*, vol. 12, no. 1, pp. 109–123, 2020.

[3] C. Yang, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang, "Biologically inspired motion modeling and neural control for robot learning from demonstrations," *IEEE Trans. Cogn. Develop. Syst.*, vol. 11, no. 2, pp. 281–291, 2019.

[4] C. Eteke, D. Kebüde, and B. Akgün, "Reward learning from very few demonstrations," *IEEE Transactions on Robot.*, vol. 37, no. 3, pp. 893–904, 2020.

[5] Y. Ma, D. Xu, and F. Qin, "Efficient insertion control for precision assembly based on demonstration learning and reinforcement learning," *IEEE Trans. Ind. Inform.*, vol. 17, no. 7, pp. 4492–4502, 2021.

[6] W. Wang, R. Li, Y. Chen, Z. M. Diekel, and Y. Jia, "Facilitating humancrobot collaborative tasks by teaching-learning-collaboration from human demonstrations," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 640–653, 2019.

[7] S. Chen, Y. Cao, Y. Kang, P. Li, and B. Sun, "Deep feature representation based imitation learning for autonomous helicopter aerobatics," *IEEE Trans. Artif. Intell.*, in press, doi: 10.1109/TAI.2021.3053511.

[8] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *ACM Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 1–8.

[9] J. Peters and S. Schaal, "Natural actor-critic," in *Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 280–291.

[10] K. Shiarlis, J. Messias, and S. Whiteson, "Inverse reinforcement learning from failure," in *ACM Int. Joint Conf. Auto. Agents Multi. Syst. (AAMAS)*, 2016, pp. 1060–1068.

[11] Y. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama, "Imitation learning from imperfect demonstration," in *Proc. Springer Int. Conf. Mach. Learn. (PMLR)*, 2019, pp. 6818–6827.

[12] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Prez, "Deep reinforcement learning for autonomous driving: A survey," *arXiv preprint arXiv:2002.00444*, 2021.

[13] B. Argall, B. Browning, and M. M. Veloso, "Learning by demonstration with critique from a human teacher," in *ACM/IEEE Int. Conf. Human-Robot Inter. (HRI)*, 2007, pp. 57–64.

[14] S. Chernova and M. M. Veloso, "Learning equivalent action choices from demonstration," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2008, pp. 1216–1221.

[15] B. Argall, B. Browning, and M. M. Veloso, "Learning robot motion control with demonstration and advice-operators," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2008, pp. 399–404.

[16] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, G. Wang, D. L. Roberts, M. E. Taylor, and M. L. Littman, "Interactive learning from policy-dependent human feedback," in *ACM Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 2285–2294.

[17] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *arXiv preprint arXiv:1706.03741*, 2017.

[18] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in atari," *arXiv preprint arXiv:1811.06521*, 2018.

[19] D. S. Brown and S. Niekum, "Deep bayesian reward learning from preferences," *arXiv preprint arXiv:1912.04472*, 2019.

[20] E. Biyik, N. Huynh, M. J. Kochenderfer, and D. Sadigh, "Active preference-based gaussian process regression for reward learning," *arXiv preprint arXiv:2005.02575*, 2020.

[21] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," *Robot. Auton. Syst.*, vol. 54, no. 5, pp. 409–413, 2006.

[22] M. Yeasin and S. Chaudhuri, "Toward automatic robot programming: learning human skill from visual data," *IEEE Trans. Syst. Man Cyber. Part B Cyber.*, vol. 30, no. 1, pp. 180–185, 2000.

[23] P. K. Pook and D. H. Ballard, "Recognizing teleoperated manipulations," in *IEEE Int. Conf. Robot. Auto. (ICRA)*, 1993, pp. 578–585.

[24] M. Laskey, J. Lee, C. Chuck, D. Gealy, W. Hsieh, F. T. Pokorny, A. D. Dragan, and K. Goldberg, "Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations," in *IEEE Int. Conf. Auto. Science Eng. (ICASE)*, 2016, pp. 827–834.

[25] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. L. Thomaz, "Using perspective taking to learn from ambiguous demonstrations," *Robot. Auton. Syst.*, vol. 54, no. 5, pp. 385–393, 2006.

[26] R. Liu, X. Zhang, and H. Zhang, "Web-video-mining-supported workflow modeling for laparoscopic surgeries," *Artif. Intell. Med.*, vol. 74, no. 1, pp. 9–20, 2016.

[27] J. Beck, Z. Papakipos, and M. Littman, "Reneg and backseat driver: Learning from demonstration with continuous human feedback," *arXiv preprint arXiv:1912.04472*, 2019.

[28] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *arXiv preprint arXiv:1805.07836*, 2018.

[29] Y. Wu and Y. Liu, "Robust truncated hinge loss support vector machines," *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 974–983, 2007.

[30] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *ACM Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1885–1894.

[31] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 7025–7034.

Qin Li received the B.S. degree in intelligent science and technology and the M.S. degree in control science and engineering from the Central South University, Changsha, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D degree at the Central South University. Her current research interests include human motion analysis and human-robot interaction.

Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include intelligent learning and optimization and their interdisciplinary applications.

Dr. Wang is an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *Swarm and Evolutionary Computation*. He was a recipient of Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science highly cited researcher in Computer Science in 2017 and 2018.