IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

# Learning from Demonstrations via Multi-Level and Multi-Attention Domain-Adaptive Meta-Learning

Ziye Hu[1], Zhongxue Gan[1,2], Wei Li[1,2], Weikun Guo[1], Xiang Gao[2], Jiwei Zhu[1]

***Abstract*—Despite significant advances in few-shot classification, object detection, or speech recognition in recent years, training an effective robot to adapt to previously unseen environments in a small data regime is still a long-lasting problem for learning from demonstrations (LfD). A promising solution is meta-learning. However, we notice that simply constructing a model with a more complicated and deeper network via previous meta-learning methods does not perform well as we expected. One possible reason is that the shallow features are gradually lost as the network deepens, while these shallow features play an essential role in the adaptation process of meta-learning. Thus, we present a novel yet effective Multi-Level and Multi-Attention Domain-Adaptive Meta-Learning (MLMA-DAML) framework, which meta-learns multiple visual features via different attention heads to update the model policy. Once the model is updated, our MLMA-DAML predicts robot actions (e.g., positions of end-effectors) via fully connected layers (FCL). As we notice that directly converting visual signals to robot actions via FCL following prior methods is not robust to perform robot manipulation tasks, we further extend our MLMA-DAML to MLMA-DAML++. The proposed MLMA-DAML++ learns an effective representation of manipulation tasks via an extra goal prediction network with convolutional layers (CL) to predict more reliable robot actions (represented by feature pixels/grids).**

***Index Terms*—Learning from Demonstration, Meta-Learning, Deep Learning Methods.**

## I. Introduction

Recently, meta-learning has been widely explored in image classification [1], object detection [2], and automatic speech recognition [3]. However, training a robot to fast adapt to previously unencountered environments is still a long-lasting problem for learning from demonstrations (LfD) [4].

Manuscript received: March, 28, 2022; Revised June, 23, 2022; Accepted July, 31, 2022. This paper was recommended for publication by Editor Dana Kulic upon evaluating the Associate Editor and Reviewers' comments and was accepted as an oral presentation by ICRA 2023.

This work was supported in part by the Ji Hua Laboratory of Foshan under Grant X190021TB190, in part by the Science and Technology Commission of Shanghai Municipality under Grant 19511132000, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0103, in part by the Shanghai Engineering Research Center of Artificial Intelligence and Robotics, and in part by the Engineering Research Center of Artificial Intelligence and Robotics through the Ministry of Education, China.

Corresponding authors: Zhongxue Gan and Wei Li (email: ganzhongxue@fudan.edu.cn; fd_liwei@fudan.edu.cn).

[1]Ziye Hu, Zhongxue Gan, Wei Li, Weikun Guo, and Jiwei Zhu are with the Academy for Engineering and Technology, Fudan University, Shanghai, China. Zhongxue Gan and Wei Li are also with the Department of Engineering Research Center for Intelligent Robotics, Ji Hua Laboratory, Guangdong, China

[2]Xiang Gao is with the Department of Engineering Research Center for Intelligent Robotics, Ji Hua Laboratory, Guangdong, China

Digital Object Identifier (DOI): 10.1109/LRA.2022.3207558

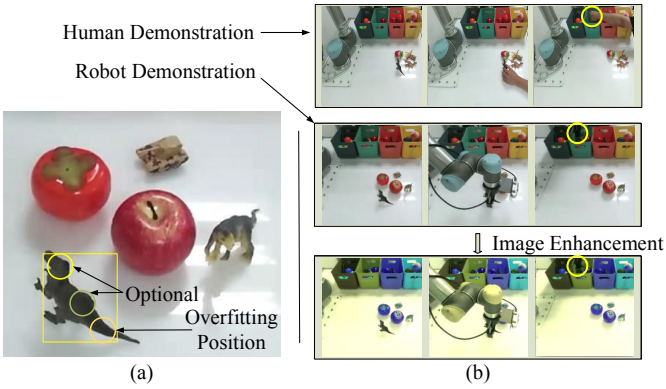



Fig. 1. Examples of predicting an overfitting position via FCL illustrated by (a) and a UR5 robot arm performing placing tasks using real-world data with image enhancement shown by (b).

Based on meta-learning, the preliminary work [5] proposes a meta-learning framework for one-shot imitation learning, which combines Long Short-Term Memory (LSTM) with the Soft-Attention Mechanism. Although the LSTM-based meta-learning method is proven to be effective in stacking tasks, it does not perform well on more challenging manipulation tasks, such as simulated reaching and pushing tasks [6]. Thus, [6] further proposes a One-Shot Visual Meta-Imitation Learning (MIL) framework based on Model-Agnostic Meta-Learning (MAML) [7], enabling the robot to learn from robot demonstrations in simulation experiments. Inspired by metric learning, [8] and [9] propose Task-Embedded Control Networks (TecNets) for imitation learning. However, the best success rate of TecNets on their real-world placing experiments is just 88.9%, leaving room for improvement. In order for the robot to be able to visually imitate from observing human demonstrations, domain-adaptive meta-learning (DAML) is proposed by [10]. To identify the target object from distractors, [11] extends the DAML to target recognition-based meta-imitation learning (TaR-MIL) by introducing a target recognition module. Inspired by the fact that humans could learn from various domains, Random Domain-Adaptive Meta-Learning (RDAML) [12] is explored to allow the robot to learn from human demonstrations and robot demonstrations. As few meta-learning methods teach the robot to distinguish tasks of what to do and what not to do, learning from demonstrations via Replayed Task-Contrastive Model-Agnostic Meta-Learning (RTMAML) [13] is proposed.

However, we notice that simply constructing a model with a more complicated and deeper network via previous meta-learning methods does not perform well as we expected, which

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

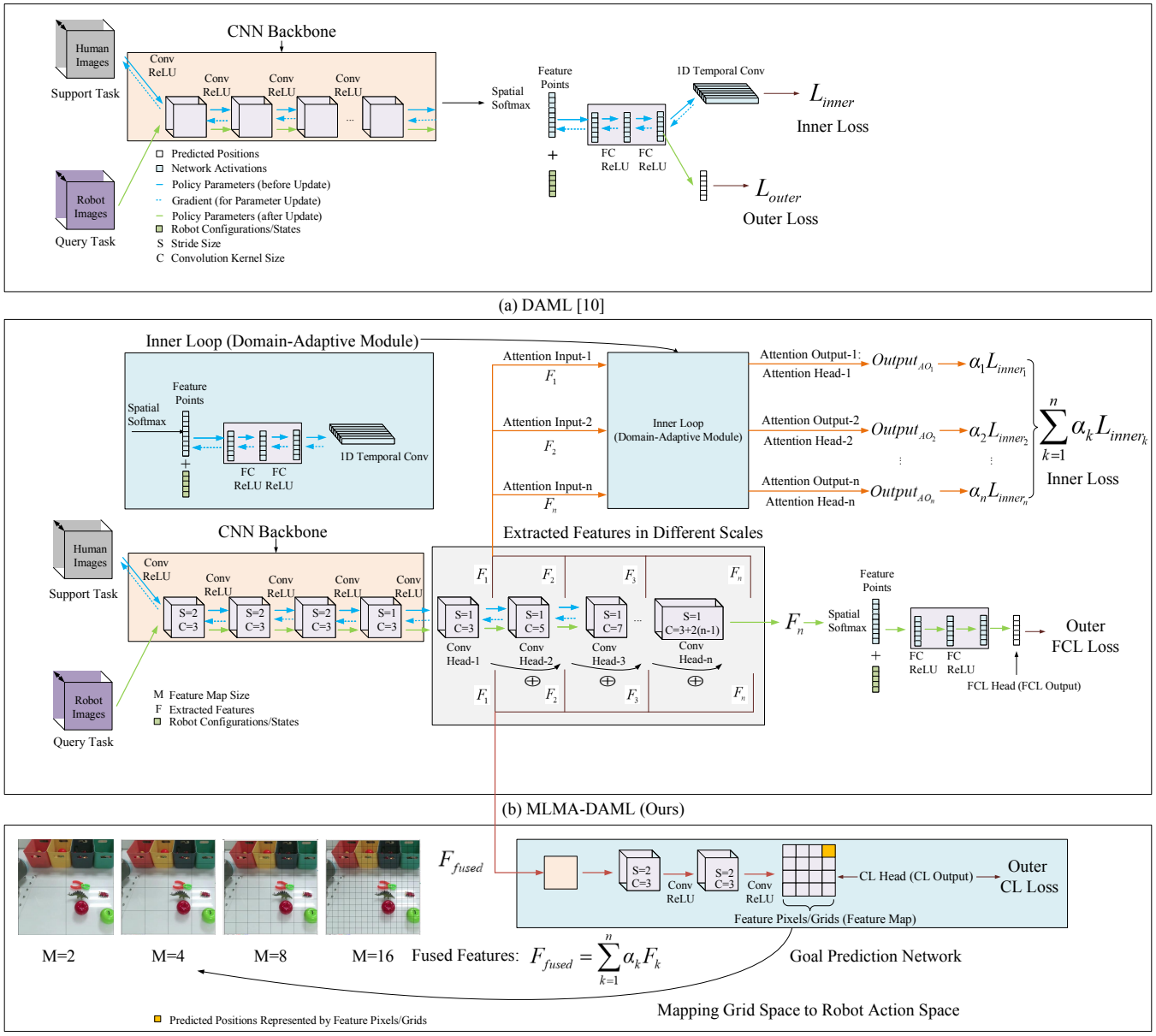



Fig. 2. Illustration of DAML [10], and our proposed MLMA-DAML/MLMA-DAML++. We implement the MLMA-DAML/MLMA-DAML++ framework based on DAML, where the MLMA-DAML consists of convolutional (Conv) layers, spatial softmax modules, and fully connected (FC) layers. First, multi-level features (e.g., $F_1 : F_n$) from the support task are extracted by MLMA-DAML with different Conv Heads. Then, these extracted features are fed into the domain-adaptive module to get the corresponding attention heads, where each attention head is assigned with an attention factor (e.g., $\alpha_1 : \alpha_n$) to indicate the importance of the attention head. Finally, we get a weighted inner loss according to these attention heads, and the model policy is updated based on the inner loss. After the model policy is updated, the FCL head predicts the robot actions (e.g., position of the end-effector) on the query task. Based on MLMA-DAML, a goal prediction network (the CL head) is introduced to MLMA-DAML++, which learns a representation (e.g., where to place the objects in the grid space) of a query task represented by feature pixels/grids. In MLMA-DAML++, the real-world space is divided into $M \times M$ parts/pixels/grids, where $M$ is the size of the feature map.

could not learn from visual demonstrations to adapt to new, unencountered environments reliably. We consider a possible reason is the shallow features are gradually lost as the network deepens, while these shallow features are essential for the fast adaptation of meta-learning. To prevent the model's shallow features from being forgotten, we have tried to implement the related meta-learning methods via a ResNet structure. However, this will result in an overfitting problem for meta-learning by simply using a ResNet-based model with a deeper network structure in our experiments, especially training a model in a small data regime for fast adaption to various environments. In this regard, it is demonstrated by [14] that training a model with shallow structures could alleviate the overfitting problem, while the shallow convolution and nonlinear operations help preserve the generalization ability of the model. Inspired by the fact that humans pay more attention to certain features

than others, a novel yet effective Multi-Level and Multi-Attention Domain-Adaptive Meta-Learning (MLMA-DAML) framework is presented in this paper. The proposed MLMA-DAML meta-learns the convolutional features at different levels via different attention heads with different weights.

Furthermore, as shown in Fig. 1, we find that directly converting visual signals to robot actions via FCL following prior methods tends to predict an overfitting position of the target object since it shows limited robustness to position shift. In this regard, we notice that predicting the position represented by a feature/pixel space is more robust to position shift. Therefore, we further extend our MLMA-DAML to MLMA-DAML++. Our MLMA-DAML++ aims to learn an effective representation of manipulation tasks (e.g., predicting the representation of where to place the object via a goal prediction network in Fig. 2). In MLMA-DAML++, the learned representations could be transformed to the corresponding real-world positions via a transformation matrix.

Considering that garbage or express sorting is a common issue in daily life, we aim to enable the robot to approach the corresponding containers with a picked object demonstrated by the demonstrators (e.g., a video recording a human or robot performing a task) via meta-learning. Thus, a wide range of real-world placing experiments is explored in this paper, which is of significant research value in the industrial and robotics communities.

## II. Related Work

In recent years, collecting demonstration data via teleoperation or kinesthetic teaching to teach robots various skills has been widely explored by imitation learning or learning from demonstrations (LfD) [4]. Based on inverse reinforcement learning (IRL), many approaches have been successfully applied to imitation learning or LfD [15]. Accounting for imperfect demonstrations, [16] and [17] consider that training a model with suboptimal or failed demonstrations based on IRL could generalize better and learn faster than conventional IRL approaches. However, designing a general reward function used to learn from new and unspecified tasks is hard for IRL [18], which requires a large amount of expert knowledge.

Considering that evaluating the learned costs in terms of robustness to various spatial perturbations only and deploying a robot with a fixed execution speed is impractical, [19] proposes time-invariant solutions through model-based IRL to learn temporally invariant rewards from misaligned demonstrations. As there is an open problem for scaling model-based IRL to real robotic manipulation tasks, [20] proposes a gradient-based IRL framework by considering three key challenges: 1) learning good dynamics models, 2) developing algorithms that could scale to high-dimensional state spaces and 3) the ability to learn from both visual and proprioceptive demonstrations. To teach robots via capturing human preference, active learning is explored by [21], [22], [23]. However, querying the users for input with a frustrating user-in-the-loop learning process makes active learning not easy to train a model [24]. Even though comprehensive approaches are explored for LfD, few of them allow for robots to adapt to new, unseen scenarios by learning from visual demonstrations.

To fast adapt to new, unseen environments, meta-learning [7], [25] is explored for LfD or imitation learning, such as One-Shot Imitation Learning [5], and Task-Embedded Control Networks for Imitation Learning (TecNets) [8]. Since Model-Agnostic Meta-Learning (MAML) [7] is proven to be effective in image classification, object detection, and reinforcement learning, [6] further extends MAML to MIL (One-Shot Visual Imitation Learning via Meta-Learning). To enable robots to learn from human demonstrations, a Domain-Adaptive Meta-Learning (DAML) is proposed based on MAML and MIL. Considering demonstrations alone may not provide enough information in the cases of task ambiguity or unobserved dynamics, [26] proposes an approach that can learn from both demonstrations and trial-and-error experience with sparse reward feedback. Compared to prior works, [27] presents a method that can translate human videos into robot demos and train the meta-police based on the translated data. In order to discriminate targets between tasks, [11] proposes a target recognition-based meta-imitation learning (TaR-MIL) framework by introducing a target recognition module to DAML. Although many other methods exist for meta-learning, enabling robots to learn from visual demonstrations (e.g., a video recording a human or robot performing a task) is still challenging so far.

## III. Proposed Methods

This section briefly introduces the related MAML-based framework DAML. In this section, a model $f$ with parameters $\theta$ is assumed, and we represent the model function by $f_\theta$. An overview of the entire DAML [10] and our proposed MLMA-DAML/MLMA-DAML++ frameworks are illustrated in Fig. 2. Following previous related work, we group task instances by their shared objectives. For example, we regard placing objects into the yellow box and placing objects into the blue box as two different tasks. In our experiments, we assume that the training set $D = \{D_h^s, D_r^s, D_r^q\}$ over tasks $p(\mathscr{T})$ is provided, where $D_h^s$ and $D_r^s$ denote the support tasks which contain human demonstrations and robot demonstrations, $D_r^q$ denotes the query tasks which contain robot demonstrations. As illustrated in Fig. 2, a weighted inner loss with multiple attention heads in MLMA-DAML is used to adapt to new tasks compared to DAML. Then, the MLMA-DAML++ further extends MLMA-DAML with a goal prediction network (the CL head). Please note that our approaches are based on one-shot meta-learning, which trains the model on a set of demonstration tasks and then gives a single example task on the robot that will be employed during testing.

### A. Preliminaries

The objective of MAML is to train a model with parameters $\theta$ that could quickly adapt to new unseen scenarios via one or more gradient updates. Given a support task, the task-adaptive model $\phi$ in MAML can be obtained by performing several gradient updates on $\theta$. For convenience, only one gradient update on $\theta$ is considered in this paper. As MAML is mainly applied to tasks such as regression, text classification, and image classification in previous work, DAML proposes a

two-head meta-learning framework (see Fig. 2 (a)) based on MAML for robots to learn from visual demonstrations: 1) the inner loss for the inner head and 2) the outer loss for the outer head.

In the inner head, the DAML learns from the support task $D^s$ to get a task-adaptive model $\phi$:

$$\phi = \theta - \delta \nabla_\theta L_{inner}(f_\theta, D^s), \tag{1}$$

where the loss function $L$ could be a cross-entropy loss (CEL) for discrete actions or mean squared error (MSE) for behavior cloning (BC), and $\delta$ is the step size for stochastic gradient descent (SGD).

As illustrated in Fig. 2 (a), if we define the output of the inner head as $O$, then we could compute the inner loss $L_{inner}(f_\theta, D^s)$ of DAML by:

$$L_{inner}(f_\theta, D^s) = \|O\|_2^2. \tag{2}$$

In the outer head, the DAML optimizes the meta-performance of the task-adaptive model on the query task $D^q$:

$$\min_\theta L_{outer}(f_\phi, D^q) = \min_\theta L_{outer}(f_{\theta - \alpha \nabla_\theta L_{inner}(f_\theta, D^s)}, D^q). \tag{3}$$

### B. MLMA-DAML

Then we introduce the MLMA-DAML shown in Fig. 2 (b). The meta-training process of MLMA-DAML is divided into two phases: 1) inner update (inner loop) with multiple attention heads and 2) outer update (outer loop) with the FCL head (equivalent to the outer head of DAML).

*1) Inner Update of MLMA-DAML:* To adapt to each support task $D^s$, $n$ domain-adaptive modules are constructed for each attention head, corresponding to each attention output $Output_{AO_k}$ (see Fig. 2 (b)).

Then, the domain-adapted inner loss is defined as follows:

$$\begin{aligned} L_{inner}(f_\theta, D^s) &= \sum_{k=1}^{n} \alpha_k \left\| Output_{AO_k} \right\|_2^2 \\ &= \sum_{k=1}^{n} \alpha_k \left\| W_k f_k + b_k \right\|_2^2 \end{aligned}, \tag{4}$$

where $D^s$ could be a human demo $D^s_{h_i}$ or a robot demo $D^s_{r_i}$, $\alpha_k$ is the k-th attention factor for the k-th attention head, and $n$ denotes the num of attention heads. We use $W_k$ and $b_k$ to indicate the weights and biases of the last layer of the k-th attention head, and $f_k$ is the corresponding input features of $W_k$ and $b_k$. Then we could denote each attention output $Output_{AO_k}$ by $W_k f_k + b_k$ for simplicity. The larger the $\alpha_k$, the more attention is paid to the k-th attention head, and vice versa.

In this paper, the $\alpha_k$ is define as follows:

$$\alpha_k = \begin{cases} (\frac{1}{d})^{\lfloor (\frac{n}{2}) - k + 1 \rfloor} & \text{if } k \leq \lfloor (\frac{n}{2}) \rfloor, \\ d^{\lfloor k - (\frac{n}{2}) \rfloor} & \text{if } k > \lfloor (\frac{n}{2}) \rfloor, \end{cases} \tag{5}$$

where $d$ is the decay factor used to generate different $\alpha_k$. The smaller $d$ is, the more attention is paid to the previous attention head, and vice versa. For example, if we set $n = 3$ and $d = 0.5$, then $\alpha_{1:n}$ will be $[2.0, 1.0, 0.5]$. If we set $n = 3$ and $d = 1$, then $\alpha_{1:n}$ will be $[1.0, 1.0, 1.0]$. If we set $n = 5$ and $d = 2$, then $\alpha_{1:n}$ will be $[0.25, 0.5, 1.0, 2.0, 4.0]$.

Therefore, our MLMA-DAML meta-learns multi-level features from visual demonstrations with a multi-attention mechanism.

Once the inner loss $L_{inner}(f_\theta, D^s)$ is computed, we could obtained the model parameters $\phi$ based on Equation (1):

$$\phi = \theta - \eta \nabla_\theta \sum L_{inner}(f_\theta, D^s), \tag{6}$$

where $\eta$ is the step size (learning rate for the inner loop) used for SGD. If not specified, we set $\eta$ to 0.0001 by default in this paper. Please note that although we both use $L_{inner}(f_\theta, D^s)$ to indicate the inner loss in DAML and MLMA-DAML for consistency, we compute them via different equations (e.g., Equation (2) for DAML while Equation (4) and Equation (5) for MLMA-DAML.

*2) Outer Update of MLMA-DAML:* After obtaining the model parameters $\phi$ in the inner loop, we optimize the meta-performance on the corresponding query task $D^q_{r_i}$ in the outer loop based on Equation (3):

$$\begin{aligned} \min_\theta L_{FCL}(f_\phi, D^q_{r_i}) &= \min_\theta L_{FCL}(f_{\theta - \eta \nabla_\theta \sum L_{inner}(f_\theta, D^s)}, D^q_{r_i}) \\ &= \min_\theta \left\| O_{FCL}(f_\phi, D^q_{r_i}) - a \right\|_2^2, \end{aligned} \tag{7}$$

where we use $O_{FCL}(f_\phi, D^q_{r_i})$ to indicate the output of the outer FCL head based on the model parameters $\phi$ and the query task $D^q_{r_i}$, and we use $a$ to indicate the corresponding supervised robot action.

During the outer update process, we use the widely used Adam (adaptive moment estimation) Optimizer to optimize the model parameters based on the $L_{FCL}(f_\phi, D^q_{r_i})$, where we set the learning rate for the outer loop as 0.0001 by default in this paper. Since we use a UR5 robot arm to conduct experiments, the robot actions defined in our paper are the UR5's end-effector positions.

### C. MLMA-DAML++

As shown in Fig. 2 (c), we further extend MLMA-DAML to MLMA-DAML++ by introducing a goal prediction network (the CL head) in the outer loop of MLMA-DAML. Thus, the inner loop of MLMA-DAML++ is the same as MLMA-DAML, while the outer loss of MLMA-DAML++ becomes:

$$\begin{aligned} L_{MLMA-DAML++}(f_{(\phi, F_{fused})}, D^q_{r_i}) = & L_{FCL}(f_\phi, D^q_{r_i}) + \\ & L_{CL}(f_{(\phi, F_{fused})}, D^q_{r_i}), \end{aligned} \tag{8}$$

where we use $F_{fused}$ to denote the fused extracted features that are used to compute the feature map shown in Fig. 2 (c).

In Equation (8), we compute $F_{fused}$ by:

$$F_{fused} = \sum_{k=1}^{n} \alpha_k F_k, \tag{9}$$

where $\alpha_k$ is the k-th attention factor for each extracted feature $F_k$ shown in Fig. 2.

Similar to $L_{FCL}(f_\phi, D^q_{r_i})$ defined in MLMA-DAML, we define $L_{CL}(f_{(\phi,F_{fused})}, D^q_{r_i})$ as follows:

$$L_{CL}(f_{(\phi,F_{fused})}, D^q_{r_i}) = \left\| O_{CL}(f_{(\phi,F_{fused})}, D^q_{r_i}) - m \right\|_2^2, \quad (10)$$

where we use $O_{CL}(f_{(\phi,F_{fused})}, D^q_{r_i})$ to indicate the output of the outer CL head based on the query task $D^q_{r_i}$ and the fused extracted features $F_{fused}$ illustrated in Fig. 2 (c), and $m$ is the corresponding supervised feature maps sampled from $D^q_{r_i}$. In our paper, we assume a feature map of size $M \times M$ is available, which consists of $M \times M$ grids/pixels to represent a task. During our exploration, we have noticed that too small (e.g., 2 or 4) or too large (e.g., 32 or larger) size of the feature map does not represent a task well, so we set $M = 8$ by default in this paper.

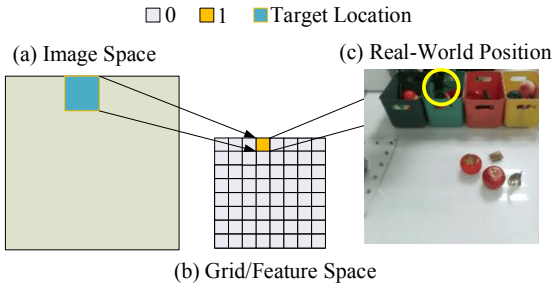



Fig. 3. Illustration of mapping image space to grid/feature space and mapping grid/feature space to real-world position.

An $8 \times 8$ feature map is illustrated in Fig. 3. In this feature map, we set a pixel to 1 (indicated by the orange pixel) to represent where the object should be placed in the grid/feature space, while we set the other pixels to 0 by default.

### D. Testing for MLMA-DAML and MLMA-DAML++

Following previous work (e.g., MIL, DAML, and TaR-MIL), supervised data (e.g., end-effector position) is necessary to train the model. Therefore, we need to define the target goal during training for the goal prediction network (the CL head) of MLMA-DAML++. However, we only need to provide the supervised data during the training phase. Once training is finished, the outer update of MLMAL-DAML/MLMA-DAML++ is no longer needed. Therefore, the supervised data is no longer needed during testing. In other words, we only need to provide the visual demos to teach the robot during testing.

Please note that we mainly train the model by learning from human demos using Equation (4) during training. After training, we test the model performance in three settings: 1) learning from a human demo (see Table I); 2) learning from a human demo with image noise (see Table II); and 3) learning from a previous unseen robot demo (see Table III). Therefore, only one type of video (a human video or a robot video) is used to teach the robot under different testing settings. For each test, we use one video per task to teach the robot and only update one step using Equation (4).

During testing, we could use either the FCL head or the goal prediction network (the CL head) to predict robot actions for MLMA-DAML++. In contrast to the FCL head that directly predicts the end-effector positions that need to be approached in the robot space, the goal prediction network (the CL head) learns a representation of a task via mapping the visual inputs to feature grids $G_{CL}$ in grid/feature space. Thus, if we use the goal prediction network (the CL head) to get the end-effector positions for the robot, we need to convert the $G_{CL}$ to the robot space via a transformation function $T$:

$$P_{CL} = T(G_{CL}), \quad (11)$$

where $P_{CL}$ denotes the transformed end-effector positions according to the predicted $G_{CL}$.

## IV. Experiments

Considering that garbage or express sorting is a common issue in daily life, enabling the robot to approach the corresponding containers with a picked object via meta-learning is explored in this paper. In our experiments, we assume that the UR5 robot arm can pick up an object by using the ROS interface or a trained neural model [28].

### A. Datasets

As shown in Fig. 4, we use a UR5 robot arm, a RealSense D435 camera, and a Kinect V2 camera to collect real-world data. Then, we extend our data via image enhancement [29] (e.g., image noise, color transformation with random RGB, and random cropped borders). In our experiments, the image noise consists of 1) Gaussian noise, b) salt and pepper noise, and c) random noise. The input images are resized to $256 \times 256$ before feeding into the model. We train our model on a training set that consists of 9600 human demonstrations/videos and 9600 robot demonstrations/videos. The 9600 human/robot videos are divided into 1920 tasks with different goals and five videos per task. In TecNets [8], we notice that only two frames are needed to teach the robot to perform a placing task: initial state and final state (placing position). Therefore, we have tried to collect videos of 3, 50, and 100 frames in our early experiments inspired by TecNets. Then we found that three frames per video are enough for each video to teach the robot about placing tasks and are convenient for collecting data. Therefore, for convenience, each video is three frames in our final experiments. Then, we test the trained model under three different settings: 1) A test set (Test Set1) consists of real-world tasks with 160 support tasks (human samples/demonstrations) and 160 query tasks; 2) An augmented test set (Test Set2) with 3200 support tasks (human samples/demonstrations) and 3200 query tasks, which is augmented based on Test Set1; 3) A test set (Test Set3) with 160 support tasks (robot samples/demonstrations) and 160 query tasks. Please note that the goal of Test Set2 is to test the robustness of the model under different new object and background colors, and even image noise. Test Set3 aims to test the model's generalization capability, where we train a model by learning from human demonstrations and test it by learning from robot demonstrations. We randomly place objects/bins on the table in our experiments, containing six to ten distractors, such as different color bins and objects.

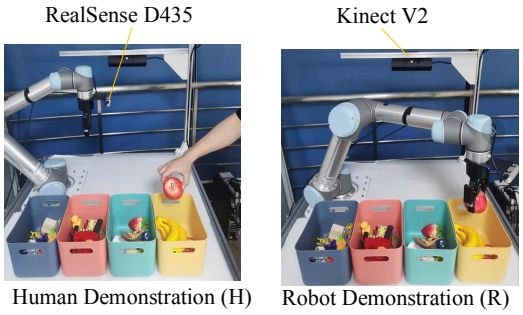
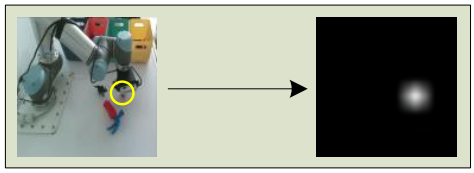
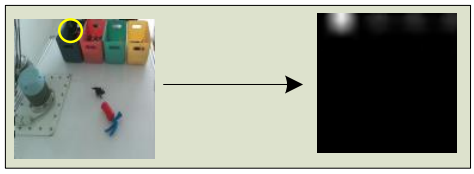



Fig. 4. The experimental platform for the placing tasks. We use this platform to collect human demos (left) and robot demos (right) for training and testing.

### B. Task Definition



(a) Where to Pick



(b) Where to Place

Fig. 5. Exploration of the goal prediction network (the CL head): predicting where to pick and where to place with two feature maps.

As shown in Fig. 4, we consider a placing task is successful if the robot could place the picked objects into the corresponding yellow bin designated by the human demonstration. For convenience, we mainly define the task by the bins' colors, while the bins' shapes are not necessarily the same. In our experiments, we found that our method can distinguish not only different colored bins but also objects of different sizes, textures, and shapes in different positions (e.g., Fig. 5 (a)). As bins are about $25cm \times 17cm$ in size, we consider it is better to constrain the target location to be a certain distance from the center of the target bin. Therefore, if the predicted end-effector position of the FCL head or the transformed end-effector position ($P_{CL}$) of the goal prediction network (the CL head) is within the range ($8cm$) of the designated bin, we consider it a success.

Please note that our training data are collected for placing tasks, so we use the goal prediction network (the CL head) to predict one feature map to indicate the target position. However, in our exploration, we found that we could predict different feature maps through the goal prediction network (the CL head). As shown in Fig. 5, we could predict two feature maps to indicate 1) where to pick and 2) where to place. In the future, we plan to collect a suitable dataset for pick-and-place tasks with further exploration with the goal prediction network (the CL head).

### C. Model Architecture and Hyper-parameters

We have illustrated our model frameworks in Fig. 2. Compared to previous works, we implement the methods (e.g., DAML, TaR-MIL, and MLMA-DAML) without using the robot configurations/states in this paper. The number of filters for each convolutional layer is set to 64 by default, except the number of filters for the last convolutional layer in the goal prediction network (the CL head) is 1. The size of fully connected layers is set to 200 by default, and the domain-adaptive module consists of two 1D-convolutional layers with 32 $10 \times 1$ filters and one 1D-convolutional layer with 32 $1 \times 1$ filters. To normalize the features extracted by the network, we use layer normalization by default. The input batch size is 16 for training, and we train our model with $30K$ iterations. Given a corresponding visual demonstration, the trained model predicts the end-effector positions under new environments during testing. To reduce the randomness of our results, we test the model with three random seeds (seed 0, seed 1, and seed 2) and report the averaged results to indicate the model performance.

### D. Experimental Results and Analysis

In this section, we mainly compare our MLMA-DAML/MLMA-DAML++ with the most related state-of-the-art meta-learning methods: DAML and TaR-MIL. To explore the effectiveness of DAML on our experiments, we implement DAML under three settings: 1) a shallow neural network with five convolutional layers (64 $3 \times 3$ kernels), denoted by DAML; 2) a ResNet-based network with 12 convolutional layers (64 $3 \times 3$ kernels), denoted by DAML$^{\dagger}$; 3) a ResNet-based network with 18 convolutional layers (64 $3 \times 3$ kernels), denoted by DAML$^{*}$. However, to our surprise, the DAML$^{\dagger}$ and DAML$^{*}$ perform worse than DAML on Test Set1, which indicates that simply constructing a model with a more complicated and deeper network based on meta-learning does not work well as we expected. In this regard, we consider that a deeper network is prone to overfitting, especially when training a model in a small data regime based on meta-learning. Another reason is that shallow features are gradually lost as the network deepens, while these shallow features play an essential role in the adaptation process of meta-learning.

In Table I, extensive ablation study is conducted on MLMA-DAML/MLMA-DAML++. First, we explore MLMA-DAML under different attention head sizes $n$ and different decay factors $d$. Experimental results show that MLMA-DAML achieves its best success rate of 93.33% on Test Set1 when $n = 3$ and $d = 0.5$, 2.91% higher than DAML and 12.91% higher than TaR-MIL. This result shows that 1) choosing a proper attention head size (e.g., $n = 3$) for MLMA-DAML is important and 2) too-small or too large decay factors (e.g., 0.1 and 10.0) are not suitable. In addition, we could notice that in the case of $n = 3$ and $d = 0.5$, the attention factors ($\alpha_1$ to $\alpha_3$) of MLMA-DAML are $[2, 1.0, 0.5]$, which has verified that 1) shallow features play an important role in the

| Methods | Hyper-Parameters $n$ | $d$ | Success Rates |
|---|---|---|---|
| DAML | - | - | $90.42 \pm 0.78\%$ |
| TaR-MIL | - | - | $80.42 \pm 1.64\%$ |
| DAML$^{\dagger}$ | - | - | $84.17 \pm 1.18\%$ |
| DAML$^{*}$ | - | - | $82.70 \pm 0.58\%$ |
| MLMA-DAML | 3 | 0.1 | $91.67 \pm 0.29\%$ |
| MLMA-DAML | 3 | 0.5 | $\mathbf{93.33 \pm 1.29}\%$ |
| MLMA-DAML | 3 | 1.0 | $91.67 \pm 0.59\%$ |
| MLMA-DAML | 3 | 2.0 | $92.30 \pm 0.59\%$ |
| MLMA-DAML | 3 | 10.0 | $90.00 \pm 0.51\%$ |
| MLMA-DAML | 5 | 0.1 | $90.83 \pm 0.29\%$ |
| MLMA-DAML | 5 | 0.5 | $92.29 \pm 1.18\%$ |
| MLMA-DAML | 5 | 1.0 | $89.38 \pm 0.88\%$ |
| MLMA-DAML | 5 | 2.0 | $90.21 \pm 0.29\%$ |
| MLMA-DAML | 5 | 10.0 | $87.46 \pm 0.83\%$ |
| MLMA-DAML$^{\dagger}$ | 3 | 0.5 | $87.46 \pm 1.31\%$ |
| MLMA-DAML$^{\dagger}$ | 5 | 0.5 | $91.04 \pm 0.78\%$ |
| MLMA-DAML++$_F$ | 3 | 0.5 | $90.42 \pm 0.29\%$ |
| MLMA-DAML++$_C$ | 3 | 0.5 | $93.96 \pm 0.59\%$ |
| MLMA-DAML++$_F$ | 5 | 0.5 | $90.83 \pm 0.29\%$ |
| MLMA-DAML++$_C$ | 5 | 0.5 | $95.00 \pm 0.51\%$ |
| MLMA-DAML++$_F$ | 7 | 0.5 | $92.08 \pm 0.29\%$ |
| MLMA-DAML++$_C$ | 7 | 0.5 | $\mathbf{96.46 \pm 0.78}\%$ |
| MLMA-DAML++$_F$ | 9 | 0.5 | $91.25 \pm 0.51\%$ |
| MLMA-DAML++$_C$ | 9 | 0.5 | $95.41 \pm 0.29\%$ |
| MLMA-DAML++$_C^{\dagger}$ | 3 | 0.5 | $93.13 \pm 0.51\%$ |
| MLMA-DAML++$_C^{\dagger}$ | 5 | 0.5 | $89.58 \pm 0.29\%$ |

TABLE I
THE EVALUATION SUCCESS RATES ON TEST SET1 (LEARNING FROM HUMAN DEMOS/VIDEOS). '† AND '$*$' DENOTE THE ABLATION STUDY MENTIONED IN SECTION IV-D. 'MLMA-DAML++$_F$' AND 'MLMA-DAML++$_C$' DENOTE THE EVALUATED RESULTS ON THE FCL HEAD AND THE CL HEAD, RESPECTIVELY.

| Methods | Hyper-Parameters $n$ | $d$ | Success Rates |
|---|---|---|---|
| DAML | - | - | $85.99 \pm 0.19\%$ |
| TaR-MIL | - | - | $73.59 \pm 0.21\%$ |
| MLMA-DAML | 3 | 0.5 | $88.58 \pm 0.44\%$ |
| MLMA-DAML | 5 | 0.5 | $91.67 \pm 1.28\%$ |
| MLMA-DAML++$_F$ | 5 | 0.5 | $87.79 \pm 0.14\%$ |
| MLMA-DAML++$_C$ | 5 | 0.5 | $92.16 \pm 0.16\%$ |
| MLMA-DAML++$_F$ | 7 | 0.5 | $88.69 \pm 0.25\%$ |
| MLMA-DAML++$_C$ | 7 | 0.5 | $\mathbf{93.32 \pm 0.24}\%$ |
| MLMA-DAML++$_F$ | 9 | 0.5 | $88.24 \pm 0.21\%$ |
| MLMA-DAML++$_C$ | 9 | 0.5 | $93.06 \pm 0.23\%$ |

TABLE II
THE EVALUATION SUCCESS RATES ON TEST SET2 (LEARNING FROM HUMAN DEMOS/VIDEOS WITH IMAGE NOISE).

| Methods | Hyper-Parameters $n$ | $d$ | Success Rates |
|---|---|---|---|
| DAML | - | - | $64.17 \pm 2.12\%$ |
| TaR-MIL | - | - | $54.38 \pm 1.02\%$ |
| MLMA-DAML | 3 | 0.5 | $68.34 \pm 1.25\%$ |
| MLMA-DAML | 5 | 0.5 | $69.17 \pm 0.78\%$ |
| MLMA-DAML++$_F$ | 5 | 0.5 | $63.13 \pm 0.88\%$ |
| MLMA-DAML++$_C$ | 5 | 0.5 | $67.92 \pm 1.06\%$ |
| MLMA-DAML++$_F$ | 7 | 0.5 | $63.34 \pm 1.06\%$ |
| MLMA-DAML++$_C$ | 7 | 0.5 | $\mathbf{69.59 \pm 1.28}\%$ |
| MLMA-DAML++$_F$ | 9 | 0.5 | $66.46 \pm 0.51\%$ |
| MLMA-DAML++$_C$ | 9 | 0.5 | $69.17 \pm 1.47\%$ |

TABLE III
THE EVALUATION SUCCESS RATES ON TEST SET3 (LEARNING FROM ROBOT DEMOS/VIDEOS).

adaptation process of meta-learning and 2) we need to pay more attention to the front attention heads (e.g., the feature $F_1$ extracted by the attention head-1). As we notice that each feature/attention input ($F_{1:n}$ shown in Fig. 2) to the corresponding attention head in MLMA-DAML has a different receptive field (e.g., convolution kernel sizes of $3,5,7$ for $n=3$), we further conduct an ablation study on MLMA-DAML: MLMA-DAML$^{\dagger}$ with a fixed receptive field (convolution kernel size of 3) in the case of 1) $n=3$ and $d=0.5$ and 2) $n=5$ and $d=0.5$. Experimental results show that MLMA-DAML$^{\dagger}$ perfroms worse than MLMA-DAML, indicating that learning multi-level features under different receptive fields is beneficial to improve the meta-learning performance.

Based on MLMA-DAML, we further test MLMA-DAML++ with a fixed decay factor ($d=0.5$) and different size of $n$ in Table I. As MLMA-DAML++ has two output heads (the FCL head and the CL head), we test the success rates of these two output heads, denoted by MLMA-DAML++$_F$ and MLMA-DAML++$_C$, respectively. We could observe that both MLMA-DAML++$_C$ outperfroms MLMA-DAML++$_F$, and the MLMA-DAML++$_C$ produces the highest success rate of 96.46% when $n=7$, 6.04% higher than DAML and 3.13% higher than MLMA-DAML. This result shows that a relatively large receptive field (e.g., $3,5,7,9,11,13,15$ when $n=7$) is suitable for the CL head, while the model performance no longer increases if we continue to increase the size of the receptive field with more attention heads (e.g, $n=9$). To explore the individual influence of MLMA-DAML++$_C$, we further remove the FCL head (MLMA-DAML++$_F$) in MLMA-DAML++ and train MLMA-DAML++ with a single CL head (denoted by MLMA-DAML++$_C^{\dagger}$). However, the performance of MLMA-DAML++$_C^{\dagger}$ drops significantly. This result indicates that it is beneficial to meta-learn tasks with the FCL head and the CL head simultaneously.

Considering that if we want to deploy a robot into the industrial pipeline, the robot should be robust enough to environmental noise. Thus, we further test our MLMA-DAML/MLMA-DAML++ on the more challenging Test Set2. As shown in Table II, the success rates of DAML and TaR-MIL are just 85.99% and 73.59%, respectively. On the contrary, our MLMA-DAML produces a better result with a success rate of 91.67% when $n=5$, 5.68% higher than DAML and 18.08% higher than TaR-MIL, respectively. Moreover, we achieve the highest success rate of 93.32% when $n=7$ and $d=0.5$, with a further increase of 1.65% over MLMA-DAML. These results indicate that our MLMA-DAML and MLMA-DAML++ are robust to environmental noise and could fast adapt to new previous unseen environments.

As we know, humans could acquire skills from a learning

domain, and then they could generalize the learned knowledge to another domain. So what about the performance of training by learning from human demonstrations and then testing by learning from previous unencountered robot demonstrations for the robot? We have reported the results in Table III. We could notice that both the performance of DAML and TaR-MIL drop significantly, with a success rate of 64.17% and a success rate of 54.38%, respectively. The reason is that the learning domain has been changed considerably, and there is a huge gap between human demonstrations and robot demonstrations since humans and robots are intrinsically different in morphology and dynamics. To our knowledge, few of meta-learning methods have investigated this problem, and it is still challenging for current meta-learning approaches so far. Nevertheless, our proposed MLMA-DAML++ still outperforms DAML by 5.42% and outperforms TaR-MIL by 15.21%, indicating that our MLMA-DAML++ has a better generalization capacity to previously unseen learning domain.

## V. Conclusion and Future Directions

In this paper, we have presented a novel yet practical MLMA-DAML framework to enable the robot to learn from visual demonstrations, which is proven to be helpful to meta-learn the visual demonstrations with multiple attention heads in different feature levels. We have also extended our MLMA-DAML to MLMA-DAML++ by introducing a feature mapping branch (the goal prediction network with a CL head), which could further improve the model performance by a large margin. Experimental results show that our MLMA-DAML and MLMA-DAML++ outperform current related state-of-the-art methods under three different testing settings. Although we mainly verify the effectiveness of our methods on robotic tasks, we believe they can be generalized to more meta-learning-related tasks, such as image classification, image detection or image segmentation, etc. In this regard, we leave some room for the future.

## References


[1] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9062–9071.

[2] J. Kj, J. Rajasegaran, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Incremental object detection via meta-learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[3] F. Lux and N. T. Vu, "Meta-learning for improving rare word recognition in end-to-end asr," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5974–5978.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[5] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," in *Advances in neural information processing systems*, 2017, pp. 1087–1098.

[6] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on Robot Learning*, 2017, pp. 357–368.

[7] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[8] S. James, M. Bloesch, and A. J. Davison, "Task-embedded control networks for few-shot imitation learning," in *CoRL 2018*, 2018.

[9] A. Bonardi, S. James, and A. J. Davison, "Learning one-shot imitation from humans without humans," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3533–3539, 2020.

[10] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, "One-shot imitation from observing humans via domain-adaptive meta-learning," in *Robotics: Science and Systems*, 2018.

[11] X. Yang, Y. Peng, W. Li, J. Z. Wen, and D. Zhou, "Vision-based one-shot imitation learning supplemented with target recognition via meta learning," in *Proceedings of the 2021 International Conference on Mechatronics and Automation (ICMA)*, 2021.

[12] Z. Hu, W. Li, Z. Gan, W. Guo, J. Zhu, X. Gao, X. Yang, Y. Peng, Z. Zuo, J. Z. Wen, *et al.*, "Learning with dual demonstration domains: Random domain-adaptive meta-learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3523–3530, 2022.

[13] Z. Hu, W. Li, Z. Gan, W. Guo, J. Zhu, J. Z. Wen, and D. Zhou, "Learning from visual demonstrations via replayed task-contrastive model-agnostic meta-learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8756–8767, 2022.

[14] B. Zhang, J. Su, D. Xiong, Y. Lu, H. Duan, and J. Yao, "Shallow convolutional neural network for implicit discourse relation recognition," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2230–2235.

[15] J. Hua, L. Zeng, G. Li, and Z. Ju, "Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning," *Sensors*, vol. 21, no. 4, p. 1278, 2021.

[16] K. Shiarlis, J. Messias, and S. Whiteson, "Inverse reinforcement learning from failure," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016, pp. 1060–1068.

[17] A. Kalinowska, A. Prabhakar, K. Fitzsimons, and T. Murphey, "Ergodic imitation: Learning from what to do and what not to do," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[18] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*, 2016, pp. 49–58.

[19] T. Davchev, S. Bechtle, S. Ramamoorthy, and F. Meier, "Learning time-invariant reward functions through model-based inverse reinforcement learning," *arXiv preprint arXiv:2107.03186*, 2021.

[20] N. Das, S. Bechtle, T. Davchev, D. Jayaraman, A. Rai, and F. Meier, "Model-based inverse reinforcement learning from visual demonstrations," in *Conference on Robot Learning*. PMLR, 2021, pp. 1930–1942.

[21] C. Basu, M. Singhal, and A. D. Dragan, "Learning from richer human guidance: Augmenting comparison-based learning with feature queries," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 132–140.

[22] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback," in *Robotics: Science and Systems (RSS)*. MIT Press Journals, 2020.

[23] N. Mourad, A. Ezzeddine, B. Nadjar Araabi, and M. Nili Ahmadabadi, "Learning from demonstrations and human evaluative feedbacks: Handling sparsity and imperfection using inverse reinforcement learning approach," *Journal of Robotics*, vol. 2020, pp. 3 849 309:1–3 849 309:18, 2020. [Online]. Available: https://doi.org/10.1155/2020/3849309

[24] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *Ai Magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[25] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[26] A. Zhou, E. Jang, D. Kappler, A. Herzog, M. Khansari, P. Wohlhart, Y. Bai, M. Kalakrishnan, S. Levine, and C. Finn, "Watch, try, learn: Meta-learning from demonstrations and rewards," in *International Conference on Learning Representations*, 2020.

[27] J. Li, T. Lu, X. Cao, Y. Cai, and S. Wang, "Meta-imitation learning by watching video demonstrations," in *International Conference on Learning Representations*, 2022.

[28] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9626–9633.

[29] R. Verma and J. Ali, "A comparative study of various types of image noise and efficient noise removal techniques," *International Journal of advanced research in computer science and software engineering*, vol. 3, no. 10, 2013.