Learning to Learn from Flawed, Failed, and Figurative Demonstrations

Ge Yang
University of Chicago

Bradly C. Stadie
Vector Institute

Roberto Calandra
Facebook AI Research

Pieter Abbeel
UC Berkeley

Sergey Levine
UC Berkeley

Chelsea Finn
UC Berkeley

Abstract

While imitation learning has demonstrated success in a variety of domains, including one-shot imitation problems, learning from *flawed and imperfect behaviors* remains a challenge. This is because if we hope to accomplish the task by copying the demonstrations, the demonstration themselves must be able to complete the task. In this work, we consider the problem of learning from behavior that clearly communicates a particular intention, but does not optimally accomplish the goal. We view this problem from the perspective of meta-learning, aiming to solve this problem by leveraging prior experience coming from imperfect demonstrations on previous tasks in combination with a reinforcement learning meta-objective. We show that the agent can succeed at the intended task with both noisy or biased demonstrations, as well as gestures that merely communicate intent but do not actually perform the task. We demonstrate our approach on a variety of simulated robotic manipulation tasks. Critically, we show that we can learn successful policies from multiple types of imperfect demonstrations using a single meta-learned imitation learning procedure.

1 Introduction

When humans work along-side each other, we sometimes convey the objectives of the task by offering demonstrations. Learning from demonstrations as a means to accomplish the intended task is usually cast as an imitation learning problem (IL), inverse reinforcement learning (IRL), or few-shot imitation problem if only a few demonstrations are available. A number of prior methods in this domain provide algorithms that can infer task reward from *optimal* demonstrations [Atkeson and Schaal, 1997, Abbeel and Ng, 2004, Ziebart et al., 2008, Ross et al., 2011, Finn et al., 2016, Ho and Ermon, 2016]. However when humans watch others, we can infer the intent of the task even with sub-optimal or failed demonstrations that merely gesture towards a target. Some work in the past provided mechanisms to accomplish this with incomplete demonstrations [Kitani et al., 2012, Huang and Kitani, 2014, Muelling et al., 2015] and suboptimal demonstrations under a known model of suboptimality [Billings et al., 1998]. All of which, however, assumes that a particular type of flaw is present in the demonstrations and are designed around it. How to take the designer out of the loop, and finding a general (meta-)method that allows us to obtain an imitation learner for *any* type of flawed demonstration on *any* task at-scale, without the human-in-the-loop hand-designing the algorithm, is the problem we consider in this paper. Our key observation is that, even though a failed attempt at a task cannot be used to learn a policy or reward from scratch, with respect to relevant prior knowledge, they might be able to inform the objectives of the task. After all, language describing a task can be viewed as a form of demonstration. The sentences themselves can not carry the task to completion. But equipped with knowledge, we are able to execute a task according to the instruction.

Preprint. Under review.

To accomplish this, we take the learning-to-learn approach, and meta-learns an imitation-learning algorithm that is able to obtain an appropriate policy for a task from a few imperfect demonstrations. This learned imitation learning algorithm is composed of the model initialization parameter θ, the learning objective L_ϕ, as well as the learning rates α in the stochastic gradient descent (SGD) operator–all three components learned during the metra-traing phase, and then fixed ahead of test time. To learn these components, we assume that the broader meta-learning algorithm has access to a distribution of tasks $\{M_i\}$ at test time. For each task there are a few corresponding imperfect demonstrations $\mathcal{D}_i$ that are available. At test time, when demonstrations that contain similar flaws are provided, the learned learning algorithm would be able to produce a policy that completes the task after a few steps of gradient-based optimization. The primary contribution of this paper is that by meta-learning the algorithm, we avoid the need to manually specify a model of sub-optimality, enable learning from a variety of sources of failures.

Experimental results indicate that our method can effectively acquire prior experience that enables learning reaching tasks, pick-and-place tasks, and door opening tasks from "experts" that miss the target, fail to grasp, and point at the correct door rather than actually opening it. Further, we find that our approach is able to learn a single model that can adapt to multiple types of failures.

2 Related Work

Inverse reinforcement learning and imitation learning algorithms enable agents to model the intentions and imitate other agents, under the assumption that the agent is acting optimally [Pomerleau, 1989, Atkeson and Schaal, 1997, Abbeel and Ng, 2004]. Some works have enabled learning from partial or incomplete demonstrations [Kitani et al., 2012, Huang and Kitani, 2014, Rhinehart and Kitani, 2017, Muelling et al., 2015] or learning from experts that follow a particular model or form of suboptimality [Ziebart et al., 2008, Billings et al., 1998, Lopes et al., 2007]. In contrast to approaches that explicitly model the expert, our approach is model-free, which enables us to train a single agent that can learn from demonstrations with multiple types of suboptimality.

Our approach leverages meta-learning, or learning to learn [Thrun and Pratt, 1998, Schmidhuber, 1987, Bengio et al., 1991, Naik and Mammone, 1992] to acquire an implicit prior over the tasks that the expert may be performing. Our approach is most similar to works in meta-learning for one-shot imitation learning [Duan et al., 2017, Finn et al., 2017b, Yu et al., 2018, Xu et al., 2017], though these approaches require optimal or approximately optimal demonstrations. Further, while these works learn one-shot imitation via imitation learning, our meta-level optimization is instead performed via reinforcement learning, a design decision that is crucial for attaining a robust policy (as validated in Section 5). Borsa et al. [2017] considered a similar reinforcement learning optimization for learning policies that have the opportunity to observe demonstrations; however, it did not consider the problem of learning from one or a few failed demonstrations. A suboptimal demonstration can be viewed as a form of weak supervision. Hence, our meta-learning involves learning to learn from weak supervision, similar to some prior works [Grant et al., 2017, Yu et al., 2018].

A number of works have considered the problem of leveraging failed or suboptimal demonstrations during reinforcement learning [Taylor et al., 2011, Grollman and Billard, 2012, Kim et al., 2013, Gao et al., 2018], e.g. as a mechanism for exploration. Our goal is critically different; Our meta-learning procedure produces an algorithm that does not assume a reward function at test time, and instead aim to learn new tasks from suboptimal behavior alone.

3 Preliminaries

Our method extends previous work in meta-learning and reinforcement learning to learn an imitation learning algorithm that is able to effectively generalize across the type of imperfect demonstrations that it has seen during meta-training. In this section, we introduce notation and overview the imitation learning and reinforcement learning objectives as it pertains to our approach. We also review the model-agnostic meta-learning algorithm, which we build upon.

We will be considering sequential decision making problems that are formalized as finite-horizon Markov decision processes (MDPs). An MDP entails a tuple (S, A, r, T, H) consisting of continuous states $\mathbf{s} \in S$, continuous actions $\mathbf{a} \in A$, unknown transition dynamics $\mathbf{s}_{t+1} = T(\mathbf{s}_t, \mathbf{a}_t)$, a scalar reward function $r(\mathbf{s}, \mathbf{a})$, and horizon H.

Generally, **imitation learning** is concerned with attaining good performance on an MDP M_i according to an implicit reward r that the agent have no access to, after learning from a set of demonstrations

$\mathcal{D}$. The typical imitation learning problem assumes that the demonstration $\mathcal{D}$ consists of state-action trajectories rolled out from an expert policy π^*, providing a dataset of the form $\mathcal{D}_{\text{expert}} = \{\mathbf{s}_t^*, \mathbf{a}_t^*\}$. Algorithms like *behavior cloning* achieve this by maximizing the log-likelihood of the demonstrated actions $\mathbf{a}^*$. Hence, the objective is to find the policy parameters θ that minimize

$$\min_{\theta} \mathbb{E}_{[\mathbf{s}^*, \mathbf{a}^* \sim \mathcal{D}_{\text{expert}}]} \left[\mathcal{L}_{\text{BC}}(\theta, \{\mathbf{s}^*, \mathbf{a}^*\}) \right] \tag{1}$$

where $\mathcal{L}_{\text{BC}}(\theta, \{\mathbf{s}^*, \mathbf{a}^*\})$ denotes the negative log likelihood of the expert actions under the policy π_θ which is being learned,

$$\mathcal{L}_{\text{BC}}(\theta, \mathcal{D}_{\text{expert}}) = -\log \pi_\theta(\mathbf{a}^*|\mathbf{s}^*). \tag{2}$$

Unlike imitation learning, **reinforcement learning** aims to achieve good task performance by directly optimizing reward. In this work, we will use the proximal policy optimization (PPO) algorithm Schulman et al. [2017] as our meta-optimization objective Stadie et al. [2018]

$$\mathcal{L}_{\text{PPO}}\big(\theta, \{\mathbf{s}, \mathbf{a}\}_{\sim\pi_\theta}\big) = \mathbb{E}_{(\mathbf{s},\mathbf{a}\sim\pi_{\bar{\theta}})}\left[\min\left(\frac{\pi_\theta(\mathbf{s}|\mathbf{a})}{\pi_{\bar{\theta}}(\mathbf{a}|\mathbf{s})} A(\mathbf{s}_t), \text{clip}_\epsilon\left(\frac{\pi_\theta(\mathbf{a}|\mathbf{s})}{\pi_{\bar{\theta}}(\mathbf{a}|\mathbf{s})}, 1 \right) A(\mathbf{s}_t) \right) \right], \tag{3}$$

where π_θ is the policy used to sample the trajectories. $\bar{\theta}$ indicates that the gradient does not propagate through this term, and the action likelihood is treated as fixed at sampling. $\text{clip}_\epsilon(x, 1)$ is the clipping function that enforces the trust-region, by clipping the value of input x within the neighborhood $1 \pm \epsilon$. $A(s_t)$ is the *advantage*. Specifically for variance reduction we use the *generalized advantage estimator* (GAE), details for which can be found in Schulman et al. [2015]. During learning, the PPO algorithm alternates between sampling with π_θ from the environment and taking multiple gradient steps on the above objective, making it one of the more sample efficient policy gradient methods.

Our approach uses **meta-learning** to learn an imitation learning procedure. The goal of meta-learning is to replace what humans do in designing learning algorithms by data. Instead of having a human summarize domain knowledge on a set of learning problems and manually iterate through designs of a learning algorithm, meta-learning acquires a typically more efficient learning algorithm by meta-training over this distribution of tasks. In the context of sequential decision making, each task corresponds to a Markov decision process (MDP) M. Each task M_i that is drawn from this distribution $P(M)$ during meta-training phase becomes a single datapoint. A successful meta-learning algorithm produces a learning algorithm that is able to quickly master meta-test tasks M_j that are never seen before, but drawn from the same distribution.

The model-agnostic meta-learning (MAML) algorithm Finn et al. [2017a] approaches this problem by learning an initial parameter vector θ that enables few-shot learning. Formally, supposed we have a learning objective $\mathcal{L}(\theta, \mathcal{D})$ where θ is the initial parameter of the model that we want to meta-learn. $\mathcal{D}$ is the data, the MAML objective is

$$\min_{\theta} \sum_{M_i \sim P(M)} \mathcal{L}(\theta_i', \mathcal{D}_i^{\text{test}}) \quad \text{where} \quad \theta' = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{train}}). \tag{4}$$

A key insight is that we can retain the computation graph of Eq. (4), and back-propagate the gradient signal from the outer optimization objective in Eq. (5) through this θ' sub-graph to generate high-order gradient for learning θ. The data $\mathcal{D}_i^{\text{train}}$ and $\mathcal{D}_i^{\text{test}}$ are both datapoints sampled from meta-training task M_i. In the original MAML formulation the inner loss in Eq. (4) is identical to the outer one in Eq. (5). When different inner and outer losses are used, we can refer to the outer loss function in equation (5) as the *meta-objective* and the inner loss in (4) as the *adaptation loss* function. Notably in our approach, the meta-learning object is the *proximal policy optimization* (PPO) objective, which is an RL objective that requires interacting with the environment and a reward. This objective is only used to meta-learn the learned components of the inner learning algorithm (see Sec. 4), but not needed during meta-test time.

MAML has been extended to the **meta-imitation learning** problem , where the goal is to learn new tasks with only one or a few teleoperated demonstrations (see Finn et al. [2017b]) or video demonstrations of a human performing the task Yu et al. [2018]. These methods use a behavior cloning objective (Eq. (3)) for the meta-objective. Often times, when learning from demonstration videos, actions of a human demonstrator are not available, and can not be mapped easily to the robot joint torques. Hence, Yu et al. [2018] jointly meta-learns both the model initialization parameter θ and the parameters φ for the adaptation loss $\mathcal{L}_\varphi$

$$\min_{\theta,\varphi} \sum_{M_i} \mathcal{L}_{\text{BC}}(\theta - \alpha \nabla_\theta \mathcal{L}_\varphi(\theta, \mathcal{D}_i^{\text{train}}), \mathcal{D}_i^{\text{test}}) \tag{5}$$

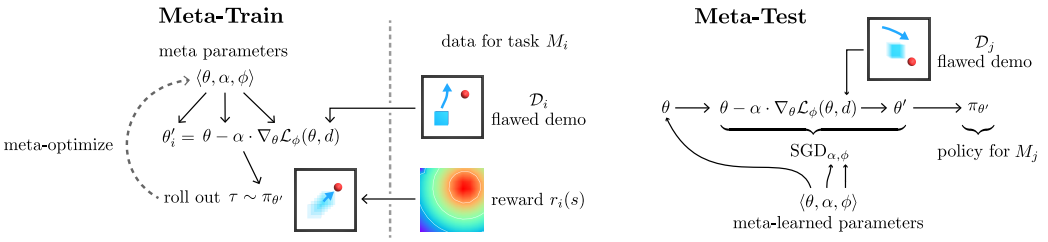


Figure 1: **Learning to learn from imperfect demonstrations**. *Left*: During **meta-training**, we assume we have access to demostration data $\mathcal{D}_i$ that is paired with the task M_i, parameterized by the task-reward $r_i(s)$. M_i is drawn from the distribution $P(M)$. For simplicity, figure only illustrates for 1 inner gradient update. In practice, we make multiple steps of inner gradient update, each with a different demonstration. *Right*: at **meta-test** time, our agent receives a previously-unseen, flawed demonstration that is similiar to those used during training. The meta-learned initialization parameters θ, learning rates α, and loss parameters ϕ allows it infer a policy π_θ' which succeeds at the task.

In Sec. 5, we show that the meta-imitation learning objective is insufficient for learning to learn from imperfect demonstrations. Next, we discuss how we can extend these approaches to handle this problem setting.

4 Learning to Learn from Imperfect Demonstrations

The goal of imitation learning (see Sec. 3) and the learning objective that *behavior cloning* optimizes for (see Eq. (3)), are notably distinct. Formulating imitation learning as supervised learning by demonstrations overlooks the underlying task M_i that the demonstrator might be acting under and the distribution $P(M)$ from which M_i is drawn. More generally, one can consider imitation learning as "instruction by demonstrations", where a human is trying to instruct the agent to accomplish a certain task. In this vein, we consider the problem of an agent learning from behaviors that convey what the task could be, but does not do so in a way that optimally solves the task.

Our goal in this work is to find a general meta-learning algorithm that can produce an agent that can learn from "demonstrations" that are flawed, that fail to accomplish the task, or those that barely carry enough information of what the intended task is with respect to some context. To take humans out of the design loop, we do not assume up-front what kind of imperfection is present. Instead we will assume that examples of such imperfect demonstrations is available, in addition to their intended tasks. Notably this tuple of $\langle$flawed demonstration D_i, intended task $M_i\rangle$ is also the intuitive evaluation metric we use as experimentalists to quantify how well the agent learns. Hence in this problem formulation, the (meta-)learning objective and the intuitive goal of the (meta-)algorithm, is now one.

Problem Overview We frame the problem of learning to learn from imperfect demonstrations as follows: The agent has access to a set of MDPs $\{M_i\}$ for meta-training sampled from the distribution $P(M)$ where each MDP is paired with a small amount of imperfect demonstrations in the form of a tuple $\langle M_i, \mathcal{D}_i\rangle$ where $\mathcal{D}$ is generated by an agent who has access to the task reward R_i. This agent does not have to show optimal trajectories nor complete the task, as long as the imperfect demonstration carries enough information of the task reward R_i to distinguish it from the other tasks in the distribution $P(M)$. We seek to learn an imitation learning procedure $\mathcal{I}(\mathcal{D}_i)$ that produces a parameterized policy $\pi_{\theta_i'}$, such that the inferred policy $\pi_{\theta_i'}$ attains high reward R_i for the task $\mathcal{M}_i$. The end-goal is to be able to learn new meta-test tasks M_j from imperfect demonstrations $\mathcal{D}_j$ using $\mathcal{I}$.

Meta-Learning from Imperfect Demonstrations We cast the problem of learning to learn from imperfect demonstrations as a meta learning problem, where we learn an imitation learning procedure $\mathcal{I}(\mathcal{D}_i)$ that produces a policy π_{θ_i}' to achieve good performance $\forall M_i = \langle R_i, \mathcal{D}_i\rangle$, where M_i is the underlying task with reward R_i and $\mathcal{D}_i$ is a small number of imperfect demonstrations for that task.

Specifically, the inference function $\mathcal{I}(\mathcal{D}_i)$ which consists of a tuple of meta-learnable components: $\langle\mathcal{L}_\phi, \pi_\theta, \text{SGD}_\alpha\rangle$. The parameters θ serve as the initialization of the policy parameters $\mathcal{L}_\phi$ is a learned adaptation loss that is akin to a critic. The stochastic gradient descent operator is parameterized by a vector learning rate $\hat{\alpha}$, and it receives the initial policy parameters θ as input, where $\hat{\alpha}$ and θ have the same dimensionality. At meta-test time, we take one or a few gradient steps on θ. For simplicity, the 1-step update takes the form $\theta_i' = \text{SGD}_{\hat{\alpha}} \circ \mathcal{L}_\phi(\theta, \mathcal{D}_i) = \theta - \hat{\alpha} \cdot \nabla_\theta \mathcal{L}_\phi(\theta, \mathcal{D}_i)$.

A key question that remains is: how should we go about learning this imitation learning operation $\mathcal{I}$? One option is to optimize it with respect to the inferred policy's ability to imitate expert demonstrations. While simple, we empirically find that this approach performs poorly because of the mismatch between behavior cloning objectives and actual policy performance. Instead, we choose to optimize the imitation learning procedure through reinforcement learning, with the objective of maximizing the reward under inferred policy $\pi_{\theta'_i}$. In particular, we consider $\mathcal{I}$ to be a differentiable computation graph with which we can directly optimize its learnable components via reinforcement on the post-update policy π'_θ over the task M_i. Hence during meta-training time, we collect trajectories from the inferred policy $\pi_{\theta'_i}$ in MDP M_i that contain rewards from the task M_i. Then we meta-optimize the learnable components of $\mathcal{I}$ using the proximal policy optimization (PPO) policy gradient objective in Eq. (3). This approach allows us to directly meta-optimize for a robust imitation learning procedure that can generate a policy $\pi_{\theta'_i}$ that attains good performance on $M_i \; \forall \; \langle M_i, D_i \rangle \sim \mathcal{M}, \mathcal{D}$.

Algorithm 1 Learning from Flawed Demonstrations

Require: task distribution: $P(\mathcal{M})$
Require: imperfect demonstrations $\mathcal{D}_i$ for each task M_i
Require: meta-learning step size β
Require: number of gradient updates for meta-optimization n_{meta}
Randomly initialize θ, ϕ, $\hat{\alpha}$
while not done **do**
  Sample batch of tasks M_i from $P(M)$
  for each M_i **do**
    Sample imperfect demonstrations $\{\tau\}_i \sim \mathcal{D}_i$
    $\theta'_i \leftarrow \theta - \hat{\alpha} \nabla_\theta \mathcal{L}_\phi(\theta, \{\tau\}_i)$
    Sample roll-outs $\{\tau\}_i^{\text{meta}}$ from policy $\pi_{\theta'_i}$ in MDP M_i
  end for
  for i in n_{meta} **do**
    $\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}_{\text{PPO}}(\theta'_i, \{\tau\}_i^{\text{meta}})$
    $\alpha \leftarrow \alpha - \beta \nabla_\alpha \sum_i \mathcal{L}_{\text{PPO}}(\theta'_i, \{\tau\}_i^{\text{meta}})$
    $\phi \leftarrow \phi - \beta \nabla_\phi \sum_i \mathcal{L}_{\text{PPO},\phi}(\theta'_i, \{\tau\}_i^{\text{meta}})$
  end for
end while

Incidentally, because the PPO policy gradient objective allows many gradient updates using the same rollout samples, we can run on the order of one-hundred meta-gradient updates on $\mathcal{I}_{\phi,\hat{\alpha},\theta}$ for each batch of roll-out. This reduces the sample complexity of the overall meta-learning algorithm and wall-time. The full meta-learning algorithm can be found in Algorithm 1.

5 Experiments

We design our experiments to answer the following question: can we learn an imitation algorithm, that is able to produce a successful policy for the task, given only a few noisy, incomplete, or failed demonstrations, that contain progressively less information of the policy that is needed to complete the task? Beyond this, we consider: (a) is using a return-based outer objective important for effective meta-learning? (b) can our method learn a single prior that can be used with a variety of types of failures and imperfections, forcing it to implicitly learn more than just a single model of sub-optimality? And (c), how well does our method generalize, when it is given demonstrations that fall outside of the distribution that it learns from?

To answer these questions, we run our experiments with a set of simulated MuJoCo robotics environments. We compare with the following imitation learning and meta-imitation baselines:

- **behavior cloning (BC)**: a feedforward network that takes the robot's observation in the state space, and outputs the predicted action $a = \pi(s)$. The learning objective (see Eq. (3)) maximizes the log-likelihood of the policy with respect to the demonstrations.
- **behavior cloning, few-shot**: behavior cloning but with the same amount of data as our method (4-rollouts)
- **domain-adaptive meta-learning (DAML)**: approach introduced in Yu et al. [2018] A learned linear loss is introduced because the human actions are not observable in the video demonstrations (see Sec.3). DAML uses a behavior cloning meta-objective.

All methods uses a Gaussian policy to represent the action space. The robot in the 2D navigation task is controlled by its 2-dimensional linear velocity. The 2-link arm is controlled by the rotational velocity of the two joints. The action space for the simulated Sawyer environment are the three dimensions of the linear velocity of the robot gripper. We train each policy using the weight-decoupled AdamW optimizer introduced in Loshchilov and Hutter [2017]. In the few-shot setting, the algorithm

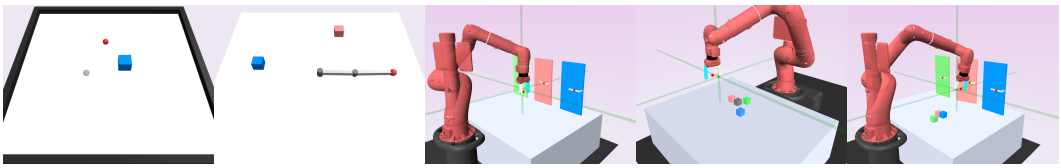


Figure 2: The environments used in the experimental evaluation: *2D Navigation*, *2-link Arm*, *Door Opening*, *Pick and Place*, and *Mixed*.

receives 4 demonstrations. We use a 5-fold held-out set for the demonstrations to evaluate all the methods. Detailed information of the multi-task environments see Fig. 2. Videos of the results can be found on the supplementary website [1]. We are planning on releasing the full experimental code upon publication.

5.1 Learning from Noisy, Biased, Incomplete, and Metaphorical Demonstrations

In this section, we start with perfect demonstrations, then progressively reduce the amount of information that the demonstrations carry. At the same time, we successively increase the complexity of the task we are trying to learn, which necessitates introducing more learned components in the imitation algorithm. With sufficient data and compute, we would hope to train a single method that is able to learn from different modes of imperfections. We consider the following tasks:

In the *2D navigation* domain, the robot observes two target positions. A task randomly samples one target to be the goal each time. The robot has no access to the information of which target is the *true goal* except through the reward signal. We illustrate the true *goal* by the red sphere, whereas the *distractor* is gray (see Fig.2a). similarly in the *2-link arm* environment (see Fig.2b), the blue block indicates the goal position for the end-effector of the arm, and the red box is the distractor. In the *3D manipulation* tasks (see Fig. 2c,d,e), the robot observes multiple boxes and a few doors. Each task require picking up one of the boxes, or opening one of the doors. The task is deemed "successful" when the robot reaches within 2 cm to the goal, with the object in-hand if it is a picking task. The door needs to be opened at least 20 centimeters from its closed potion.

The **perfect demonstrations** are collected from a set of expert policies that are trained from scratch for one of the tasks. The full dataset consists of 10000 timesteps of data. Each demonstration is 50 timesteps for the 2D environments and 100 timesteps for the 3D manipulation environments. All methods achieve 100% success rate on the 2D navigation tasks and the 2-link Arm. However the behavior-cloned agent fails with door opening, pick and place and the mixed sawyer environment. This occurs because picking up the object and grasping the handle require precise positioning of the robot gripper, and collision with the object moves the state of the support of the expert demonstrations.

Next, we **introduce bias** to the demonstrations by adding a random offset vector δ. The size of this vector is fixed as a parameter, whereas the direction is sampled randomly at the beginning of the experiment. Despite that this vector is held constant, both the behavior cloning baseline and the few-shot behavior cloning baseline fail to recover a bias-free policy. The policy consistently reaches a location offset from the true target due to this bias. In addition, this bias translates the support of the behavior data. When the environment is initialized at locations in the state space that is not covered by the biased demonstration, the learned agents behaves randomly.

The meta-imitation baseline (DAML), on the other hand, learns to compensate for the observation bias within 5 meta epochs using a learned bias transformation. This shows meta-training can in fact be used to learn algorithms that can interpret biased demonstrations, as long as perfect, and unbiased demonstrations are available during meta-learning. Nevertheless because the outer learning algorithm is behavior cloning, it suffers from the *distributional shift problem*:

Table 1: Success rate on 2D navigation and 2D reacher from imperfect demonstrations that navigate to a location offset from the target. A rollout is counted as successful if the agent ends the episode within 2 cm of the goal. The offset vector is 10 cm in length, with the direction randomly sampled at the beginning of training.

method	2D Navigation	Reacher
learning from scratch (few-shot)	0%	0%
learning from scratch (many)	0%	0%
meta-imitation	100%	90%
meta-imitation (masked-states)	100%	60%
LeaF (ours)	100%	90%

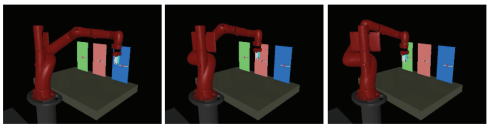
[1] https://sites.google.com/view/leaf-meta-learning

the distribution of states from the policy does not match that of the teacher exactly, leading to new states that are out side of the support of the training data Ross et al. [2011]. This problem is the most prominent with the 3D manipulation tasks, where contact causes the object to move. As a result the meta-imitation approach fails to complete the task in these environments.

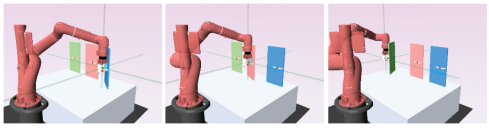
To demonstrate this problem more clearly, we sample only half of the state space during meta-training, exposing the agent to perfect demonstrations that are only sampled from the left half of the plane. The result of this experiment is presented in Table 1. In this case, the success rate on the point-mass environment remains at 100%. However, the success rate on the 2-link arm drops to 50%. Close inspection of the environment reveals that the robot in the 2D navigation task learned to use the distance vector from its current location to the goal as the main signal, which generalizes well. Whereas in the case of the 2-link arm reacher, it did not. In comparison, our approach – **LeaF** – succeeds on both the 2D navigation task and the 2-link arm reaching task, as well as the 3D manipulation tasks. Because our method uses a policy-gradient meta-objective, the learned imitation algorithm is exposed through the policy it produces to a wide variety of states. The support of the training distribution therefore extends beyond the few demonstration that are available. This illustrates the importance of using a reward-based reinforcement learning objective during meta-training, such that the post-update policy perform well under a wide variety of situations.

5.2 Learning A Single Agent from Multiple Modes of Imperfections

In practice, demonstrations can contain multiple modes of imperfections at the same time. In this section, we evaluate LeaF in this challenging scenario, where the demonstration can be both noisy, incomplete, or misleading biased. A human algorithm designer would have to learn to balance a single learning method across all three modes of imperfections. As we will show soon, LeaF is able to learn from data, and produce successful policy for any of the given type of imperfect demonstration.



a. gestures towards the door



b. updated policy

Figure 3: The demonstration gestures towards the door, but never opens it. Whereas the updated policy via **LeaF** opens the correct door in each case.

We evaluate LeaF in this mixed imperfection scenario on two manipulation domains: door opening and picking. The noisy end effector contains a random offset vector. The incomplete demonstrations consists of truncated rollouts collected from an optimal agent. In the door opening domain, the guesture/metaphorical demonstration would hover the end effector in-front of the door but fails to grasp the door handle. In the pick and place task it would hover the gripper 5 cm above the target object.

Meta-imitation performs poorly on this task, partially because picking involves contact, which often cause the block to be pushed out of the support of the demonstration data collected from experts. We found it helpful, however, to use the weights trained from the meta-imitation baseline to initialize our model, to reduce the time it takes to meta-learn via policy gradient. Notably, we found that it is necessary to use a learned imitation objective for this task. The L2 behavior cloning loss limits the extent to which the post-update policy π'_θ can adapt. In comparison, both the learned action target loss (learned L2 metric loss) and the learned critic are expressive enough to learn a successful policy π'_θ, as shown in Table 2. Indeed, inspection of the post-update policy adapted using the behavior cloning L2 loss shows that the agent does learn to pick up the block, but is limited to hover a few

Table 2: Imitating metaphorical demonstrations that point at the correct door, and mixed mode of demonstrations that contain both noise and gestures. Success rate on picking task from demonstrations that gestures towards the target block but do not pick up the object. Using a meta-learned loss, **LeaF** learns a success policy. Instead, using a fixed L2 loss that is not learned, LeaF succeeds in picking up the block, but fails to reach the goal.

method	door opening	picking
learning from scratch (few-shot)	0%	0%
learning from scratch (many demos)	0%	0%
meta-imitation	14%	0%
LeaF (L2 loss)	0%	0%
LeaF (learned L2 metric loss)	100%	100%
LeaF (learned critic)	100%	100%
learned critic w/ mixed demos	100%	-

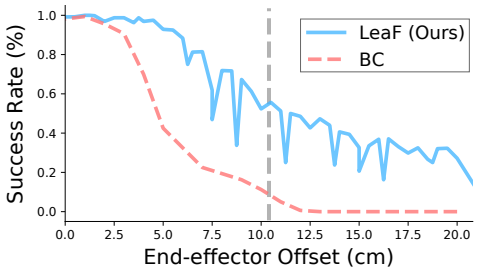


Figure 4: Generalization over demonstrations with different amounts of noise.We evaluate different fixed offset to the position of the goal ranging from 0 to 20.8 cm. When the demonstration given is perfect (i.e., zero offset), both BC and LeaF reaches the goal successfully. When the offset is increased to 10.4 cm (i.e., half the distance between the goal and the distractor), LeaF (our method) still achieves the task 50% (see vertical dashed line in gray), while BC trained with this offset only achieves the task 8% of the time.

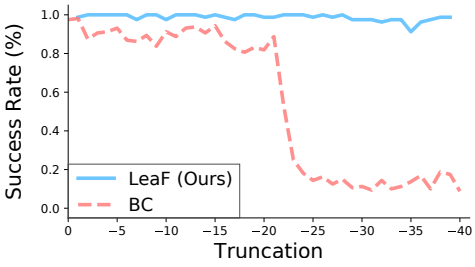


Figure 5: Generalization over truncation of demonstration. Each demonstration is 50 timesteps. Horizontal axis is the truncation index from the end of the demonstration, where the robot is closest to the goal. BC is able to reach the goal initially because the expert giving the demonstration (before truncation) hangs over the goal. When only 29/50 timesteps are left, performance drops. In comparison, LeaF (our method) is robust against truncation all the way till only 10 data-points are left.

centimeters above the table top, not
able to lift up further due to the limit
of the behavior cloning inner objective. The agent trained with mixed modes of imperfections is able to adapt to both noisy demonstrations, as well as gestures that do not pick up the object.

5.3 Generalization Over Imperfect Demonstrations

To further understand the applicability of our method, we study how well it generalizes over different amount of noise in the demonstrations, in comparison to BC baseline. In the 2D navigation domain, we meta-train using perfect demonstrations but test the learned imitation algorithm over demonstrations that have increasing amount of bias. To do so, we distribute the goal and the distractor in a 40 cm by 40 cm box, with an average distance of 20.8 centimeters. We sweep the length of the bias vector from 0 to 20.8 cm, and look at the changes in the success rate after 4 steps of imitation update. For the BC baseline we train the agent from scratch for each datapoint. As shown in Fig. 4, both method starts at 100% success rate when there is no offset, but as the offset increases, the BC agent quickly degradates, succeeding only 8% of the time when the noise approaches the mid-point between the two objects (on average). On the other hand, our approach still achieves about 50% success rate, and over all degrades more gracefully. Furthermore, in Fig. 5 we can see the effect of having incomplete demonstrations as we compare our approach with the behavior cloning baseline. Each demonstration is 50 timesteps. We start the truncation from the end of the demonstration, where the robot is closest to the goal. These timesteps carry substantial information of the task. For the BC baseline, the success rate is initially flat, because the demonstration contain redundancies as the robot halts at the goal position in the end. However, a sharp drop-off in success rate occurs when on average 29 timesteps are left in the demonstration. In comparison, our approach stays constant, unaffected by the incompleteness of the demonstration all the way.

6 Discussion

We have presented a method to learn a robust imitation algorithm that is able to learn from imperfect demonstrations that contain large biases or completely fail to accomplishing the task, purely from data. Under our proposed meta-learning framework, the agent takes advantage of its experience during the simulated reinforcement learning process, and learns how to learn a policy that is superior to the demonstration that it is shown. More broadly, this idea can be viewed as learning to learn from bits of information, where the information available at test time does not involve standard forms of supervision, i.e. it is not in the form of a supervised dataset or rewards in a few RL trials. Prior work has established connections between meta-learning and learning priors in hierarchical Bayesian models. But, thus far, meta-learning has largely been applied in a subset of the types of problems that hierarchical Bayes considers – in problems of learning priors over supervised datasets and reinforcement learning trials. The ability to apply meta-learning to more general Bayesian

modeling problems is important because, compared to their Bayesian modeling counterparts, modern meta-learning techniques are generally more applicable to large-scale problems. This work suggests that we can indeed apply meta-learning to a broader class of problems involving learning priors and performing inference under those learned priors, and suggests a mechanism for doing so with gradient-based meta-learning techniques.

References

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

C. G. Atkeson and S. Schaal. Robot learning from demonstration. In *ICML*, volume 97, pages 12–20. Citeseer, 1997.

Y. Bengio, S. Bengio, and J. Cloutier. Learning a synaptic learning rule. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 2, 1991.

D. Billings, D. Papp, J. Schaeffer, and D. Szafron. Opponent modeling in poker. *Aaai/iaai*, 493:499, 1998.

D. Borsa, B. Piot, R. Munos, and O. Pietquin. Observational learning by reinforcement learning. *arXiv preprint arXiv:1706.06617*, 2017.

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*, pages 1087–1098, 2017.

C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pages 49–58, 2016.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017a.

C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*, 2017b.

Y. Gao, J. Lin, F. Yu, S. Levine, T. Darrell, et al. Reinforcement learning from imperfect demonstrations. *arXiv preprint arXiv:1802.05313*, 2018.

E. Grant, C. Finn, J. Peterson, J. Abbott, S. Levine, T. Griffiths, and T. Darrell. Concept acquisition via meta-learning: Few-shot learning from positive examples. In *NIPS Workshop on Cognitively-Informed Artificial Intelligence*, 2017.

D. H. Grollman and A. G. Billard. Robot learning from failed demonstrations. *International Journal of Social Robotics*, 4(4):331–342, 2012.

J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

D.-A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *European Conference on Computer Vision*, pages 489–504. Springer, 2014.

B. Kim, A.-m. Farahmand, J. Pineau, and D. Precup. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems*, pages 2859–2867, 2013.

K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.

M. Lopes, F. S. Melo, and L. Montesano. Affordance-based imitation learning in robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1015–1021. IEEE, 2007.

I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 2017.

K. Muelling, A. Venkatraman, J.-S. Valois, J. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell. Autonomy infused teleoperation with application to bci manipulation. *arXiv preprint arXiv:1503.05451*, 2015.

D. K. Naik and R. Mammone. Meta-neural networks that learn by learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 437–442, 1992.

D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017.

S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

J. Schmidhuber. *Evolutionary principles in self-referential learning*. PhD thesis, Institut für Informatik, Technische Universität München, 1987.

J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

B. Stadie, G. Yang, R. Houthooft, P. Chen, Y. Duan, Y. Wu, P. Abbeel, and I. Sutskever. The importance of sampling inmeta-reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 9300–9310, 2018.

M. E. Taylor, H. B. Suay, and S. Chernova. Integrating reinforcement learning with human demonstrations of varying ability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 617–624. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

S. Thrun and L. Pratt. *Learning to learn*. Kluwer Academic Publishers, 1998.

D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese. Neural task programming: Learning to generalize across hierarchical tasks. *arXiv preprint arXiv:1710.01813*, 2017.

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. *arXiv preprint arXiv:1802.01557*, 2018.

B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.