One-Shot Visual Imitation Learning

Chelsea Finn*, Tianhe Yu*, Tianhao Zhang, Pieter Abbeel, and Sergey Levine
University of California, Berkeley

I. Introduction

Learning behavior from high-dimensional perceptual inputs presents a challenge for all types of learning algorithms, particularly when constrained by the amount of supervision and experience that can realistically be collected on a real robotic system. Typical visuomotor learning approaches start from scratch for every new task, throwing away old experience and disregarding supervision received for other tasks [4]. By reusing experience and supervision across tasks, robots should be able to amortize and significantly improve data efficiency, requiring minimal supervision for new tasks. However, it is not readily obvious how information should be shared across tasks for faster learning.

In this work, we propose to combine meta-learning with imitation, allowing the robot to reuse demonstration data across tasks in order to learn a new, related task from a single demonstration that specifies the task. Unlike prior one-shot imitation learning methods [2], our approach learns a parameterized policy that can be adapted to different tasks rather than a single policy with different input per task. As a result, our model is more flexible while having fewer overall parameters. Furthermore, unlike Duan et al. [2], we demonstrate one-shot imitation from raw pixels. Other approaches to visual imitation learning require large demonstration datasets [6, 1] or experience to be collected by the robot [7, 5].

The primary contribution of this abstract is to demonstrate an approach for one-shot imitation learning from raw pixels. We evaluate our approach on a simulated planar reaching task which entails reaching a target of a particular color, amid distractor objects of different colors. Our approach is able to learn a policy that can adapt to new task variants using only one video demonstration. Existing methods for one-shot imitation learning have required tens of thousands of demonstrations for meta-learning and only been applied to tasks with low-dimensional state information [2]. By employing a parameter-efficient method for meta-learning, our approach both requires many fewer demonstrations for meta-learning while also being able to learn a new task from raw pixel inputs. With the ability to learn from much smaller datasets, our method can feasibly be applied to real robotic systems, which we plan to explore in future work.

II. Background: Model-Agnostic Meta-Learning

Our goal is to learn a policy that can quickly adapt to new tasks from a single demonstration of that task. To do so, we will use meta-learning to train for quick adaptation across a number of tasks, enabling generalization to new tasks. Because we would like to learn visual policies without requiring extreme amounts of demonstration data per task, we will use a recently-proposed meta-learning algorithm that is parameter-efficient: model-agnostic meta-learning (MAML) [3]. In this section, we will provide an overview of MAML, which we will combine with imitation learning in the next section.

MAML aims to learn the weights θ of a model f_θ such that standard gradient descent can make rapid progress on new tasks $\mathcal{T}$ drawn from $p(\mathcal{T})$, without overfitting to a small number of examples. Because the method uses gradient descent as the optimizer, it does not introduce any additional parameters, making it more parameter-efficient than other meta-learning methods. When adapting to a new task $\mathcal{T}_i$, the model's parameters θ become θ_i'. In MAML, the updated parameter vector θ_i' is computed using one or more gradient descent updates on task $\mathcal{T}_i$, i.e. $\theta_i' = \theta - \alpha\nabla_\theta\mathcal{L}_{\mathcal{T}_i}(f_\theta)$. For simplicity of notation, we will consider one gradient update for the rest of this section, but using multiple gradient updates is a straightforward extension.

The model parameters are trained by optimizing for the performance of $f_{\theta_i'}$ with respect to θ across tasks sampled from $p(\mathcal{T})$, corresponding to the following problem:

$$\min_\theta \sum_{\mathcal{T}_i\sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) = \sum_{\mathcal{T}_i\sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta-\alpha\nabla_\theta\mathcal{L}_{\mathcal{T}_i}(f_\theta)}) \qquad (1)$$

Note that the meta-optimization is performed over the model parameters θ, whereas the objective is computed using the updated model parameters θ'. In effect, MAML aims to optimize the model parameters such that one or a small number of gradient steps on a new task will produce maximally effective behavior on that task. The meta-optimization across tasks is performed via stochastic gradient descent (SGD) with meta step size β.

III. One-shot visual imitation learning

To use MAML with imitation learning, we collect a dataset of demonstrations with at least two demonstrations per task. In our experiments, a task will be defined as reaching a target of a particular color, where the positions of the target and distractors are randomized within a task. The input, $\mathbf{x}_t$, is the agent's observation at time t, e.g. an image, whereas the output $\mathbf{y}_t$ is the action taken at time t, e.g. torques applied to the robot's joints. We will denote a demonstration trajectory as $\tau := \{\mathbf{x}_1, \mathbf{y}_1, ...\mathbf{x}_T, \mathbf{y}_T\}$. We use a mean squared error loss of the form:

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\tau^{(j)}\sim\mathcal{T}_i} \sum_t \|f_\phi(\mathbf{x}_t^{(j)}) - \mathbf{y}_t^{(j)}\|_2^2, \qquad (2)$$

*Denotes equal contribution.

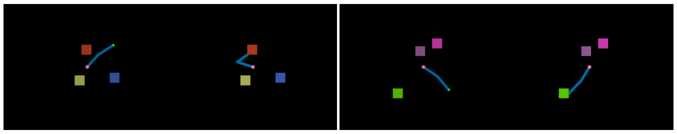

Fig. 1. Two example tasks. As shown on the right of each image pair, the goal is to reach the red and green target on the left and right tasks respectively.

We will primarily consider the one-shot case, where only a single demonstration $\tau^{(j)}$ is used for the gradient update.

During meta-training time, each meta-optimization step entails the following: A batch of tasks is sampled and two demonstrations are sampled per task. Using one of the demonstrations, θ_i' is computed for each task $\mathcal{T}_i$ using Equation 2. Then, the other demonstration per task is used to compute the gradient of the meta-objective by using Equation 1 with the loss function 2. Finally, θ is updated according to the gradient.

The result of meta-training is a policy that can be adapted to new tasks using a single demonstration. Thus, at meta-test time, a new task $\mathcal{T}$ is sampled, one demonstration for that task is provided, and the model is updated to acquire a policy for that task.

IV. Experiments

Our goals with the experimental evaluation is to determine (1) can we learn to learn a policy which maps from image pixels to motor torques using a single demonstration of the task? (2) how does the proposed approach perform compared to existing methods and using varying dataset sizes?

We evaluate our method on family of planar reaching tasks, as illustrated in Figure 1, where the goal of a particular task is to reach a target of a particular color, amid distractors with different colors. A policy roll-out is considered a success if it comes within 0.05 distance of the goal. The input to the policy includes the image observation, the arm joint angles, and end-effector position. The policy output is the torques applied to the two joints of the arm.

We optimized an expert policy using iLQG and collected roll-outs from that expert as demonstrations, collecting several demonstrations per task. At meta-test time, we evaluate the policy on held-out colors and target positions. Note that the one provided demonstration typically involves a different target position than the trial.

We compare the proposed method to the following two baselines (which were both proposed by Duan et al. [2]):

- **random policy**: A policy that outputs random actions from a normal distribution with zero mean. For normal distributions, we pick the variance that produces the best results in each domain.
- **contextual policy**: A feedforward policy, which takes as input the final image of the provided demonstration to indicate the goal of the task. The current image is also part of the input, while the action to take is the output.
- **LSTM**: A recurrent neural network which ingests the provided demonstration and the current observation, and outputs the current action, as proposed by Duan et al. [2].

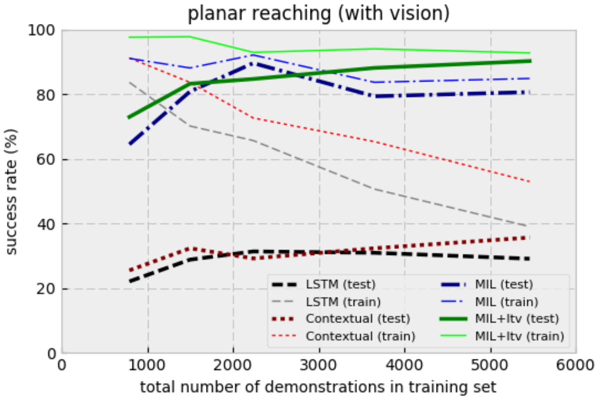

Fig. 2. One-shot success rate as a function of the meta-learning dataset size.

The baselines and the proposed approach are all trained using the same dataset, with equal supervision. All three methods use a convolutional neural network policy with 3 convolution layers of 40 3×3 filters each followed by 4 fully-connected layers with hidden dimension 100. The recurrent policy additionally has an LSTM with 2048 units that takes as input the features from the final layer. All methods are trained via a behavioral cloning objective (mean-squared error) to the expert torque values, and using the Adam optimizer with default hyperparameters and meta batch-size of 5 tasks. Our policy with MAML uses 3 meta-gradient updates each with step size 0.001. We also find it helpful to clip the meta-gradient to lie in the interval $[-10, 10]$ before applying it.

As shown in Figure 2, we find that our approach is able to effectively learn how to adapt a visuomotor policy to new tasks using only one demonstration. Furthermore, we confirm that MAML provides significantly better data-efficiency than unconstrained meta-learning methods that learn an optimization strategy from scratch rather than utilizing gradient descent.

V. Conclusion

In this work, we proposed a method for one-shot visual imitation learning using an efficient meta-learning method, demonstrating the ability to learn new tasks from a single video demonstration. Our approach compares favorably to existing state-of-the-art approaches while using significantly fewer demonstrations for meta-learning. Our experiments suggest that the proposed approach can feasibly be applied to a real robotic platform, which we plan to explore in future work.

References

[1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence Jackel, Matthew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[2] Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*, 2017.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning (ICML)*, 2017.

[4] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end learning of deep visuomotor policies. *Journal of Machine Learning Research (JMLR)*, 2016.

[5] Ashvin Nair, Pulkit Agarwal, Dian Chen, Phillip Isola, Pieter Abbeel, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. *International Conference on Robotics and Automation (ICRA)*, 2017.

[6] D. Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NIPS)*, 1989.

[7] Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699*, 2016.