Transformer-based Meta-Imitation Learning for Robotic Manipulation

Théo Cachet, Julien Perez and Seungsu Kim
firstname.lastname@naverlabs.com

Abstract

Imitation learning has been considered as one of the promising approaches to enable a robot to acquire competencies. Recently, one-shot imitation learning has shown encouraging results for executing variations of initial conditions of a given task without requiring task-specific engineering. However, it remains inefficient for generalizing in variations of tasks involving different reward or transition functions. In this work, we aim at improving the generalization ability of demonstration based learning to unseen tasks that are significantly different from the training tasks. First, we introduce the use of transformer-based sequence-to-sequence policy networks trained from limited sets of demonstrations. Then, we propose to meta-train our model from a set of training demonstrations by leveraging optimization-based meta-learning. Finally, we evaluate our approach and report encouraging results using the recently proposed framework Meta-World which is composed of a large set of robotic manipulation tasks organized in various categories.

1 Introduction

As robotic platforms are becoming affordable, end-user environments like personal houses are becoming a novel context of deployment for such systems [7]. However, the robotic manipulation platform has traditionally been deployed in a fully specified environment with predefined and fixed tasks to accomplish [6]. For this reason, we need to develop control paradigms where non-expert users will be able to specify new tasks, possibly complex and compositional. Recently, reinforcement learning has attracted a lot of interest to tackle this problem [8, 5]. Nonetheless, safe and efficient exploration in a real environment can be difficult [4, 1] and a reward function can be challenging to set up in a real physical environment. As an alternative, a collection of demonstrations have often been proposed as a convenient way to define a new task [13, 10].

In this paper, we propose an approach that associates metric-based and optimization-based meta-learning to perform transfer across robotic manipulation tasks beyond the variation of the same task using a limited amount of demonstrations. First, we introduce a transformer-based model of imitation learning. Second, we propose to leverage optimization-based meta-learning to meta-train our few-shot meta-imitation learning model. This approach allows us to efficiently use a small number of demonstrations while fine-tuning the model to the target task. Finally, we evaluate our approach to the recently proposed framework Meta-World [15] that regroups a large set of manipulation tasks which are organized in several categories. We show significant improvement compared to the one-shot-imitation framework in various settings. As an example, our approach can acquire 100% success on 100 occurences of a completely new manipulation tasks with less than 15 demonstrations.

2 Preliminaries

The goal of imitation learning [11] is to train a policy π that can imitate the expert behavior expressed in the demonstrations. In this context, one-shot imitation learning setting [2, 3] aims at learning

NeurIPS 2019 Workshop on Robot Learning: Control and Interaction in the Real World, Vancouver, Canada

a meta-policy that can adapt to new, unseen tasks from a limited amount of demonstrations. The approach has originally been proposed to learn from a single trajectory of the target task. However, this setting can be extended to few-shot if several demonstrations of the target task are available. Here, we assume an unknown distribution of tasks $p(\tau)$ and a set of meta-training tasks $\{\tau_i\}$ sampled from it. Then, for each meta-training task τ_i, a set of demonstrations $\mathcal{D}_i = \left\{d_1^i, d_2^i, ...d_N^i\right\}$ is available. Each demonstration d is a temporal sequence of { observations ; actions } tuples of successful behavior for that task $d_n = [(o_1^n, a_1^n), \cdots, (o_T^n, a_T^n)]$. This meta-training demonstration can have been produced by human or heuristic policies if tasks are simple enough. In a simulated environment, it is even possible to use reinforcement learning to create a policy from which trajectories can be sampled. Then, each task can contain different objects and require different skills from the policy. In the case of robotic manipulation tasks, these tasks can be for example Reaching, Pushing, Sliding, Grasping, or Placing. Each task will be defined by a unique combination of required skills and the nature and positions of objects define a task.

Overall, one-shot imitation learning techniques learn a meta-policy π_θ, which takes as input both the current observation o_t and a demonstration d corresponding to the task to be performed, and outputs an action. Conditioning on different demos can lead to different tasks being performed for the same observation. At training time, the algorithm first sample a task τ_i, and then sample two demonstrations d_m and d_n corresponding to this task. The meta-policy is conditioned on one of these two demonstrations d_n and optimize the following loss on the expert observation-action pairs from the other demonstration, d_m: $\mathcal{L}_{\text{bc}}(\theta, d_m, d_n) = \sum_{t=1}^{T} \mathcal{L}(a_t^m, \pi_\theta(o_t^m, d_n))$ where the $\mathcal{L}$ is an action estimation loss function; in this paper we simply use L^2 norm for this.

The one-shot imitation learning loss consists in summing across all tasks and all possible corresponding demonstration pairs: $\mathcal{L}_{\text{osi}}(\theta, \{\mathcal{D}_i\}) = \sum_{i=1}^{M} \sum_{d_m, d_n \sim \mathcal{D}_i} \mathcal{L}_{\text{bc}}(\theta, d_m, d_n)$, where M is the total number of training tasks.

3 Transformer-based Meta-Imitation

First, we propose to improve the policy network of [2] by using a transformer-based neural architecture [14]. This model allows to better capture correspondences between the input demonstration and the current episode demonstrations using the multi-headed attention layers introduced in the transformer architecture. To adapt this model to demonstration-based learning, the encoder takes as input the demonstration of the task to accomplish and the decoder takes as input all the observations of the current test episode. So, we add a mixture of sinusoïdes with different periods and phases to each dimension of the input sequences to encoder positioning informations. As in one-shot imitation model, the next action to perform is the output of the model. Second, we propose to leverage optimization-based meta-learning [9] to pre-train our policy network. Algorithm 1 describes the three consecutive steps of our meta-learning and finetuning algorithm. First, our policy is meta-trained using Reptile over the set of training tasks with an early-stopping over validation tasks. Finally, we test the model by fine-tuning on the corresponding demonstrations. In this last step, the fine-tuned policy is evaluated in terms of accumulated reward and success rate by simulated episodes in the Meta-World environment.

4 Experiments

Evaluation framework: All the evaluations are done in the Meta-World framework [15] which is composed of 50 manipulation tasks. To sample our demonstrations, we train a neural network policy for each task using clipped Proximal Policy Optimization [12]. The network used for RL-trained policy is a stack of fully connected layers as policy with the current state as input and action as output. Following the results reported in [15], we manage to converge a successful policy for a total of 46 out of 50 tasks with the same hyperparameters. This success signal provided by the framework allows us to evaluate our meta-train policies. We gather 5K demonstrations per task for meta-training by sampling the PPO-trained policies. Then, The user tasks are defined using demonstrations as we do not assume having access to a reward signal or having the possibility to explore the environment at test-time. Using the shape of the task reward function, we group each task of Meta-World in 3 categories which are (1) Push (2) Reach and (3) Pick-Place as defined in [15].

Algorithm 1 Meta-Learning and Testing algorithm with Reptile

Input Set of demonstrations $D_{\mathcal{T}_r}, D_{\mathcal{T}_e}, D_{\mathcal{T}_s}$ of train, validation and test tasks $\mathcal{T}_r$
Input Meta learning rate β

Initialize policy π_θ
while not Earlystop **do** ▷ Meta-Train
  for all task τ in $\mathcal{T}_r$ **do**
    Sample batchs of pairs of demonstrations $\{d_i^\tau, d_j^\tau\}$ from $D_{\mathcal{T}_r}(\tau)$
    Compute $W_i = Adam(L_{bc}(\tau, \pi_\theta))$
    Update policy: $\theta \leftarrow \theta + \beta(W_i - \theta)$
  end for

  ValLoss = 0
  for all task τ in $\mathcal{T}_e$ **do** ▷ Validation
    Sample all pairs of demonstrations $\{d_i^\tau, d_j^\tau\}$ from $D_{\mathcal{T}_e}(\tau)$
    Compute $\theta' = Adam(L_{bc}(\tau, \pi_\theta))$
    $ValLoss+ = L_{bc}(\tau, \pi_{\theta'})$
  end for
  Earlystop(ValLoss)
end while

for all task τ in $\mathcal{T}_s$ **do** ▷ Test
  Sample all pairs of demonstrations $\{d_i^\tau, d_j^\tau\}$ from $D_{\mathcal{T}_s}(\tau)$
  Compute $\theta'' = Adam(L_{bc}(\tau, \pi_\theta))$
  Execute $\pi_{\theta''}$ on simulator for N episodes
  Mesure reward, success rate
end for

Architecture: we compare the performance of our Transformer architecture to the state-of-the-art recurrent architecture. Both models are meta-train using Reptile and individually fine-tuned on each test task. We note a dramatic improvement for transfer across tasks, the results are summarized in Figure 1. We believe this result is particularly valuable as Pick-Place tasks are known to be challenging and usually require a large number of trials in a reinforcement learning setting as reported in [15]. In the case of transfer within the same task category, the comparable performances confirm the successful results reported in [2]. Nonetheless, in this setting, our Transformer-based policy seems to better handle the case with the lowest amount of demonstrations. For transfer across task categories, our model shows clear superiority which can be explained at least in two ways.

Meta-learning and Finetuning: We compare all considered pre-training approaches and we focus on the proposed Transformer-based policy. Results are depicted in Figure 2. As a first baseline, we consider the case where no pre-training strategy is applied. We use Xavier uniform initialization then directly optimize the policy on the demonstration of the test tasks. Otherwise, we pre-train a policy with Reptile or with Multi-Task which is mixing all tasks in the batch gradient. The fact that random initialization is inferior assesses the complexity of the considered tasks and confirms that the meta-training approaches do capture regularities of the consider environment across tasks. In transfer within the same task category, Reptile seems to provide better results as the number of demonstration decreases as for transfer across different task categories. Finally, we evaluate the performance of the Multi-task approach without performing finetuning. This training approach is the closest to the one proposed in the one-shot imitation method. The model does capture regularities that allow it to have a success rate in transfers within task categories.

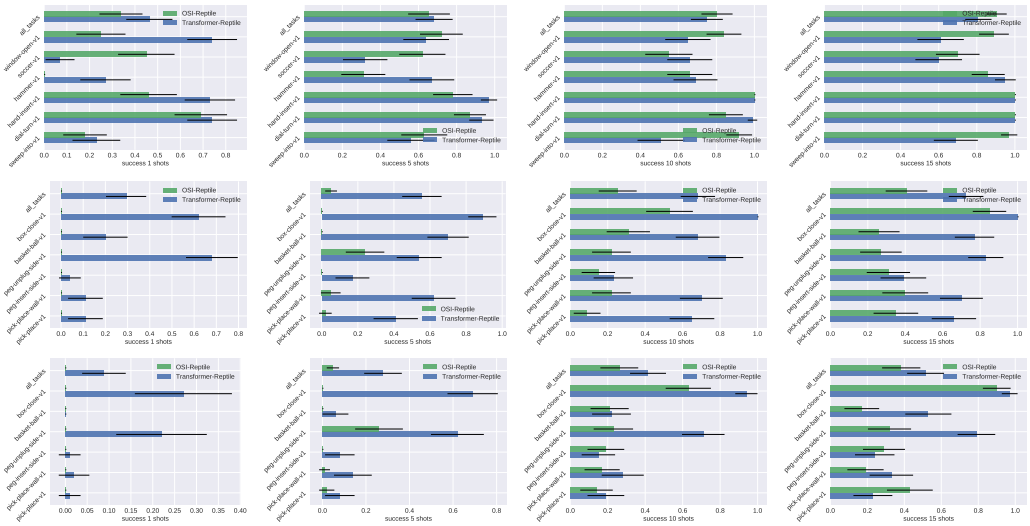

Figure 1: Comparison of policy architectures by success rate, (Top-line) Transfer within Push tasks; (Middle-line) Transfer from Push tasks to Pick-Place tasks; (Bottom-line) Transfer from Reach tasks to Pick-Place tasks. Error bars represent the standard deviation on 100 episodes with 4 seeds.

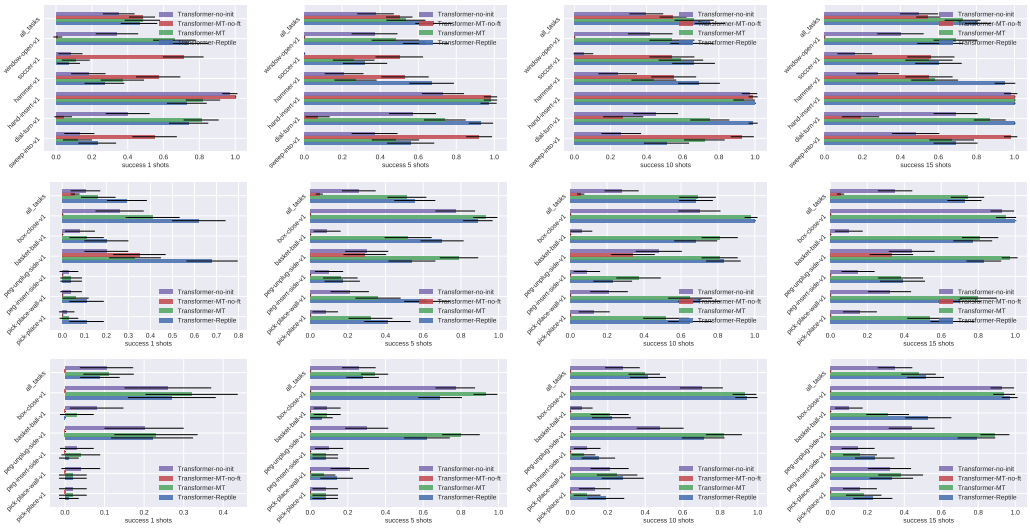

Figure 2: Comparison of initialization and finetuning approaches, (Top-line) Transfer within Push tasks; (Middle-line) Transfer from Push tasks to Pick-Place tasks; (Bottom-line) Transfer from Reach tasks to Pick-Place tasks. Error bars represent the standard deviation on 100 episodes with 4 seeds.

5 Conclusion

In this work, we propose a method to combine metric-based and optimization-based meta-learning for behavior cloning in robotic manipulations. We have introduce transformer-based architecture for meta-imitation learning and shown encouraging results. We demonstrated the effectiveness of this approach on the Meta-World environment in a large variety of tasks and show clear improvements to the original one-shot imitation learning method. Regarding further works, extending our approach to visual demonstrations is interesting for enhancing the genericity of our model. Another direction is to explore the interplay between reward-defined and demonstration-defined tasks.

References

[1] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerík, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *ArXiv*, abs/1801.08757, 2018.

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

[3] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *CoRL*, 2017.

[4] Juliette Garcia and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *J. Artif. Intell. Res.*, 45:515–564, 2012.

[5] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, 2018.

[6] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *ArXiv*, abs/1907.03146, 2019.

[7] Kayla Matheus and Aaron M. Dollar. Benchmarking grasping and manipulation: Properties of the objects of daily living. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5020–5027, 2010.

[8] Hai Nguyen and Hung M. La. Review of deep reinforcement learning for robot manipulation. *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 590–595, 2019.

[9] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018.

[10] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3758–3765, 2017.

[11] Stefan Schaal, Auke Jan Ijspeert, and Aude Billard. Computational approaches to motor learning by imitation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 358 1431:537–47, 2003.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[13] Gokhan Solak and Lorenzo Jamone. Learning by demonstration and robust control of dexterous in-hand robotic manipulation skills. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8246–8251, 2019.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[15] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2019.