1709.04905 - One-Shot Visual Imitation Learning via Meta-Learning

- **Yunqiu Xu**

- Other reference:
  - Presentation: https://www.youtube.com/watch?v=_9Ny2ghjwuY
  - Code: https://github.com/tianheyu927/mil

---

1. Introduction

- Challenge: learning each skill from a scratch is infeasible
- One-Shot Visual Imitation Learning via Meta-Learning
  - Reuse past experience and learn new skills from a single demonstration
  - Visual: use raw visual inputs
  - Meta-Learning: MAML C. Finn et.al. 2017
- Do not take task identity / demonstration as the input into a contextual policy: learn parameterized policy, adapt into new tasks through gradient updates

2. Related work

- In this work, the state of env is unknown $\rightarrow$ should be learned from raw sensory inputs
- Two challenges for learning from demonstrations:
  - Compounding errors: not in this work
  - **The need of a large number of demonstrations for each task**
- Inverse RL can reduce the number of demonstrations, but requires additional robot experience to optimize the reward C. Finn et.al. 2016
- How we tackle this: share data across tasks
  - First, use a dataset of demonstrations of many other tasks for meta learning,

then we can learn a new task from its single demonstration
  - Some existed work:
    - Contextual policies: provide the task as an input to the policy or value function
    - Train a variety of controllers, then learn a mapping from given task to controller parameters
  - What we use: **meta-learning**

3. Meta-Imitation Learning

Three requirements for what qualifies as a meta-learning system [2]:

- The system must include a learning sub-system, which adapts with experience
- Experience is gained by exploiting meta knowledge extracted
  - In a previous learning episode on a single dataset
  - From different domains or problems

- By training for adaptation across tasks, meta-learning effectively treats entire tasks as datapoints.
- MAML

Model-Agnostic Meta-Learning

Learn the weights ϴ of a model such that gradient descent can make rapid progress on new tasks.

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Evaluate ∇_θ L_{T_i}(f_θ) with respect to K examples
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i})
end while
```

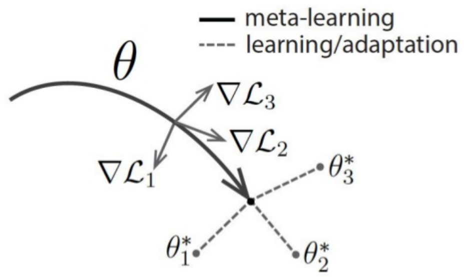


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

$$\mathcal{T}_i = \{\tau = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_T, \mathbf{a}_T\} \sim \pi_i^\star, \mathcal{L}(\mathbf{a}_{1:T}, \hat{\mathbf{a}}_{1:T}), T\}$$

Experts

- Extend MAML to MIL

Learn a policy that can quickly adapt to new tasks from a single demonstration of that task.

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Evaluate ∇_θ L_{T_i}(f_θ) with respect to K examples
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i})
end while
```


Algorithm 1 Meta-Imitation Learning with MAML

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Sample demonstration τ = {o_1, a_1, ...o_T, a_T} from T_i
        Evaluate ∇_θ L_{T_i}(f_θ) using τ and L_{T_i} in Equation (2)
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
        Sample demonstration τ'_i = {o'_1, a'_1, ...o'_T, a'_T} from T_i for the meta-update
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i}) using each τ'_i and L_{T_i} in Equation 2
end while
return parameters θ that can be quickly adapted to new tasks through imitation.
```

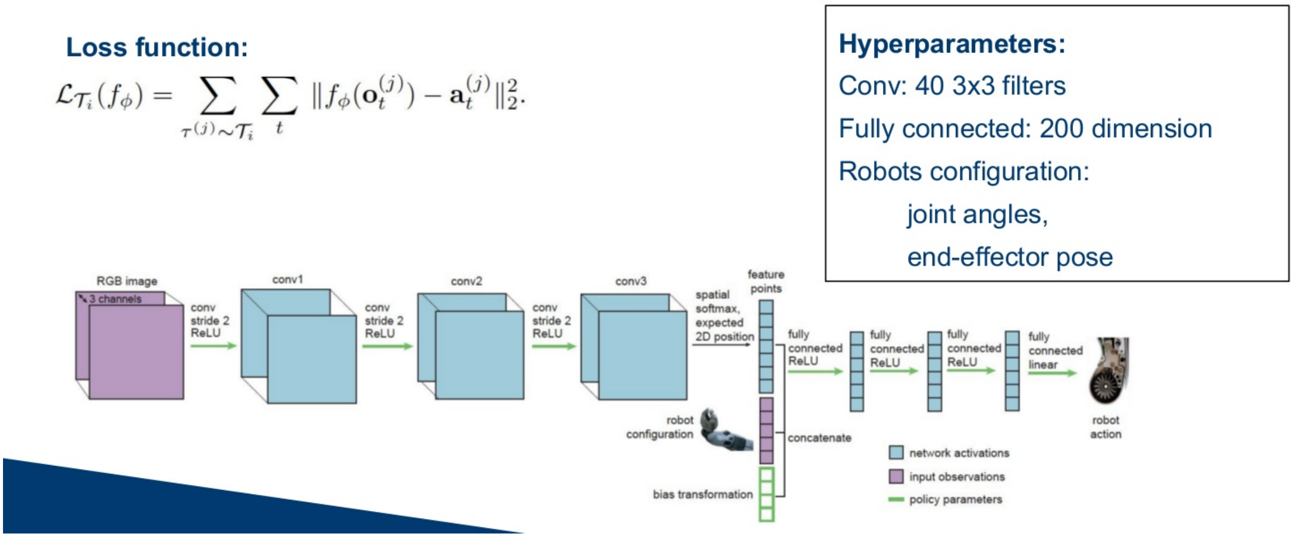

$$\mathcal{T}_i = \{\tau = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_T, \mathbf{a}_T\} \sim \pi_i^\star, \mathcal{L}(\mathbf{a}_{1:T}, \hat{\mathbf{a}}_{1:T}), T\}$$

Experts

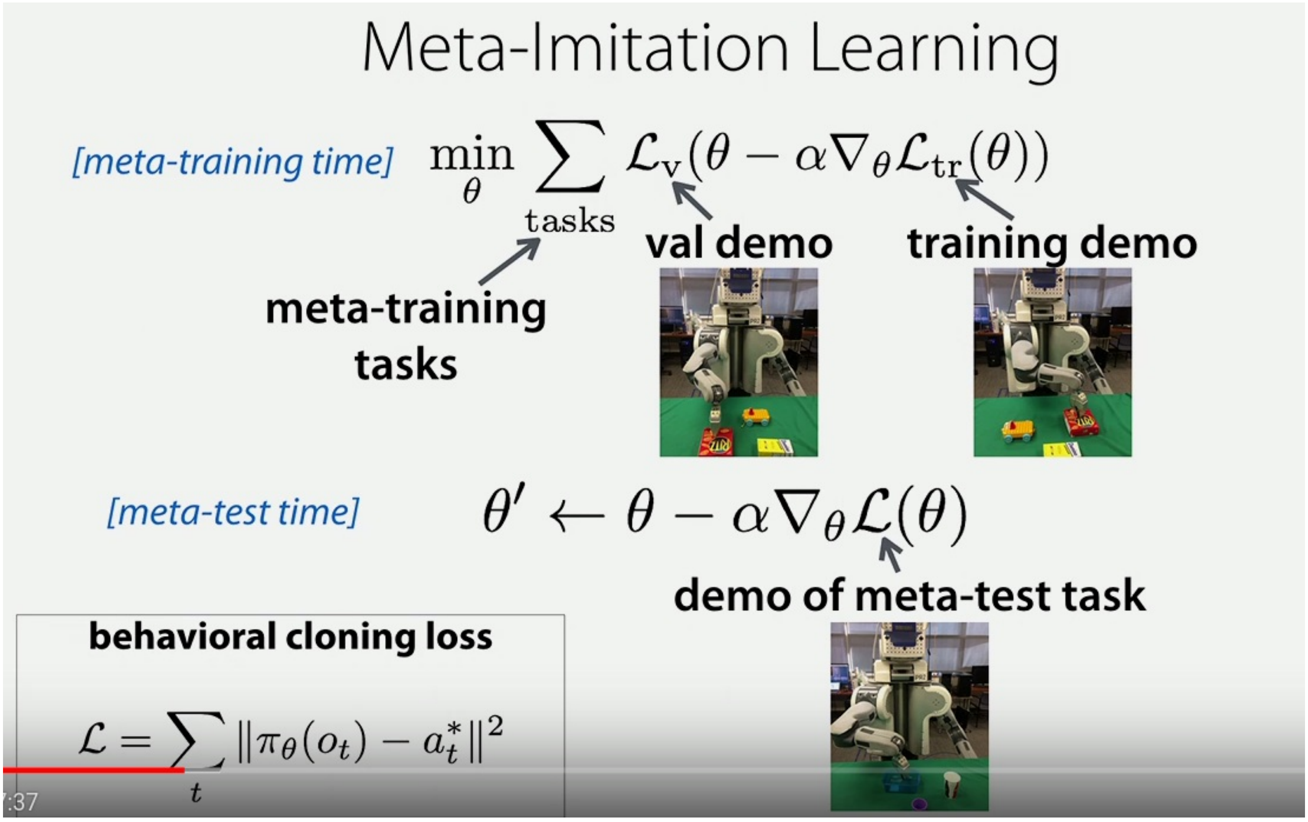
- Loss function (Equation 2 in pseudo code)

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\tau^{(j)} \sim \mathcal{T}_i} \sum_t \|f_\phi(\mathbf{o}_t^{(j)}) - \mathbf{a}_t^{(j)}\|_2^2. \quad (2)$$

- Network architecture



- Training and testing

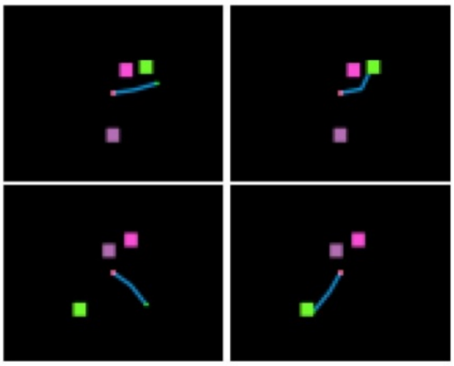


4. Experiment

Comparison:

- **Random policy:** A polity that outputs random actions from a standard Normal distribution
- **Contextual policy:** A feedforward policy, which takes as input the final image of the demonstration and the current image, and outputs the current action.
- **LSTM:** use a recurrent network to ingest the provided demonstration and current observation, and outputs the current action.
- **LSTM+attention:** use attention mechanism

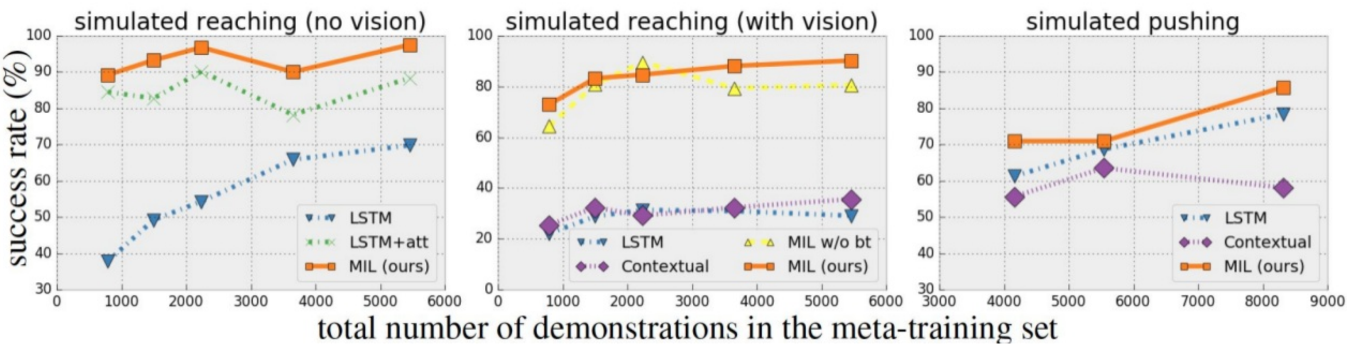
Simulated reaching



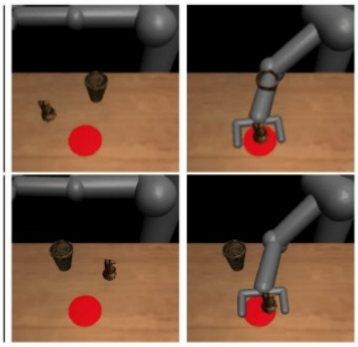
Task:

- reaching a target of a particular color
- Policy must learn to localize the target using the demonstration and generalize to new positions
- Meta-training must learn to handle different colors
- 150 tasks x 10 different trials per task

Simulated reaching



Simulated pushing

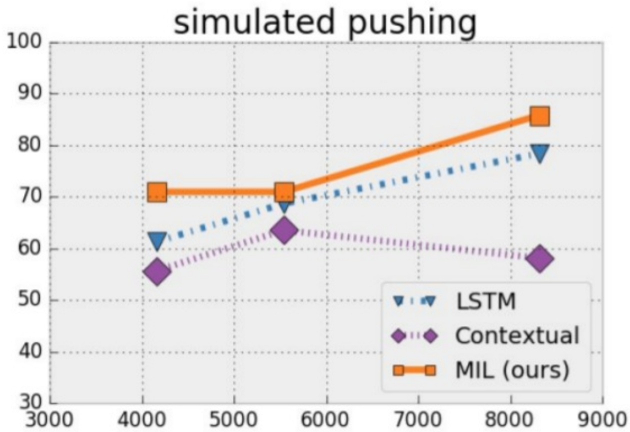


Task:

- A push is considered as success if the center of the target object lands on the red target circle for at least 10 timestamps.
- Each task is defined as pushing a particular objects
- 74 tasks x 6 different trials per task

method		video+state +action	video +state	video
LSTM	1-shot	78.38%	37.61%	34.23%
contextual	1-shot	n/a	58.11%	56.98%
MIL (ours)	1-shot	**85.81%**	**72.52%**	**66.44%**
LSTM	5-shot	83.11%	39.64%	31.98%
contextual	5-shot	n/a	64.64%	59.01%
MIL (ours)	5-shot	**88.75%**	**78.15%**	**70.50%**

Table 1: One-shot and 5-shot simulating pushing success rate with varying demonstration information provided at test-time. MIL can more successfully learn from a demonstration without actions and without robot state and actions than LSTM and contextual policies.



Real-World Placing

subset of training objects | test objects | subset of training objects | test objects

Task:

Evaluate how well a real robot (PR2) can learn to interact with new unknown objects from a single visual demonstration.

Success: the held object landed in or on the target container after the gripper is opened

method	test performance
LSTM	25%
contextual	25%
MIL	**90%**
MIL, video only	**68.33%**

Table 2: One-shot success rate of placing a held item into the correct container, with a real PR2 robot, using 29 held-out test objects. Meta-training used a dataset with ~ 100 objects. MIL, using video only receives the only video part of the demonstration and not the arm trajectory or actions.

- Demo: let robot put the ball in blue cup
- Test: shuffle the cups and let robot achieve the goal

5. Summary and Ongoing Work (On CoRL)

- Summary:
  - reuse prior experience when learning in new settings
  - learning-to-learn enables effective one-shot learning
- Ongoing work: one-shot imitation from human video $\rightarrow$ during demo, let human put the ball in cup