Replication study of MAML in Reinforcement Learning

Paper: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks by Chelsea Finn et. al

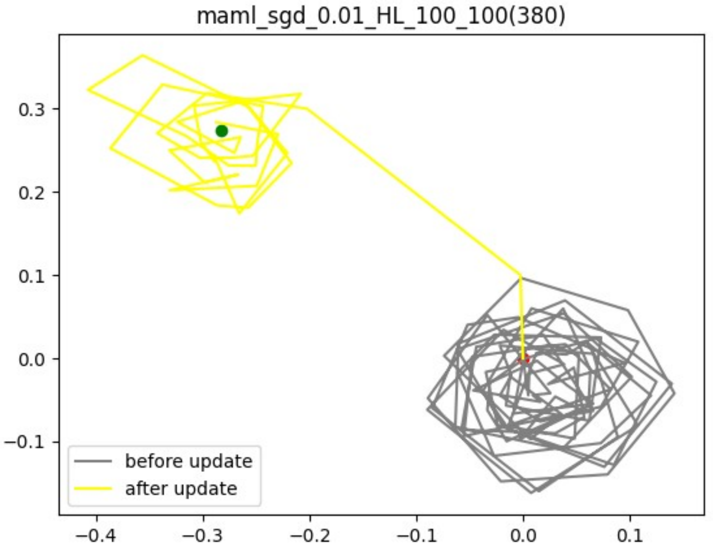
By André Henkel
Sommerterm 2020
University of Ulm
Institut für Neuroinformatik
Due Date: 18.08.2020

Scope of study

- Replication study of MAML in Reinforcement Learning
- Using 2D Navigation task
- Replicating the results of the paper
- Testing different setups
- Discuss found results and differences

Why Meta-learning?

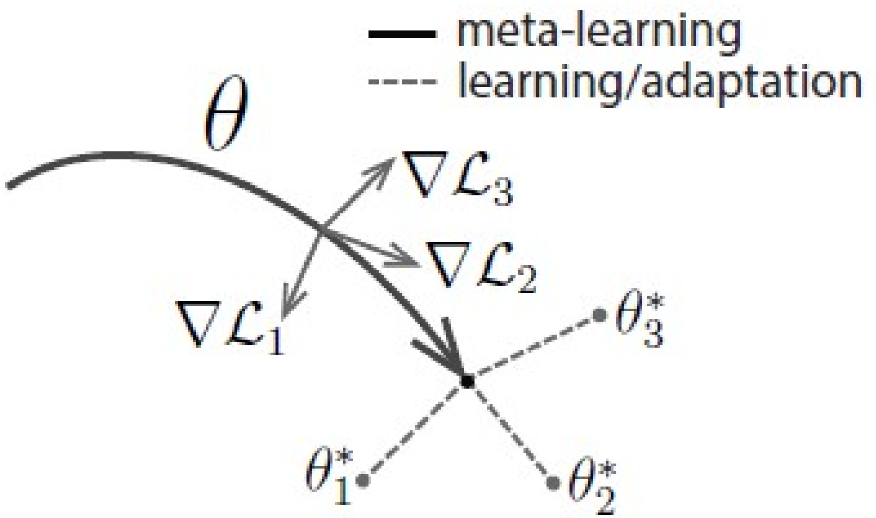
- Adapt to a new task quickly
- Few experience and updates
- For 2D Navigation:
  - Target perception through negative distance as reward
  - Observation includes the current position



MAML

- Model-agnostic meta-learning

- Train the model towards an initialization which adapts quickly to new tasks



MAML

- Initialization
- Inner-update
- Meta-update
- Loss of multiple tasks

- The paper askes three questions:
  - Quick adaptation
  - Multiple updates
  - Many domains

Algorithm 3 MAML for Reinforcement Learning

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

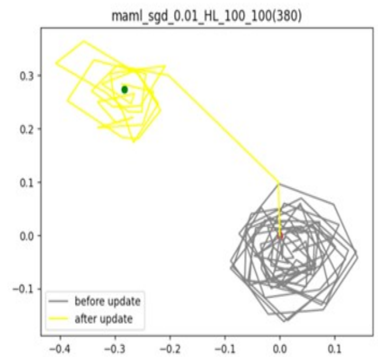
```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Sample K trajectories D = {(x_1, a_1, ...x_H)} using f_θ in T_i
        Evaluate ∇_θ L_{T_i}(f_θ) using D and L_{T_i} in Equation 4
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
        Sample trajectories D'_i = {(x_1, a_1, ...x_H)} using f_{θ'_i} in T_i
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i}) using each D'_i and L_{T_i} in Equation 4
end while
```

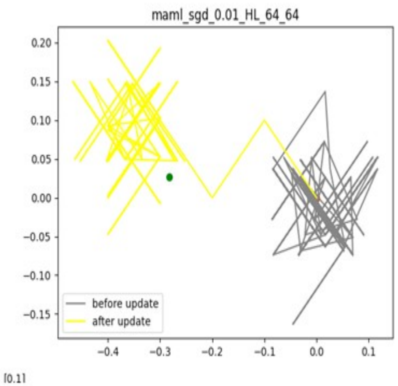
Implementation

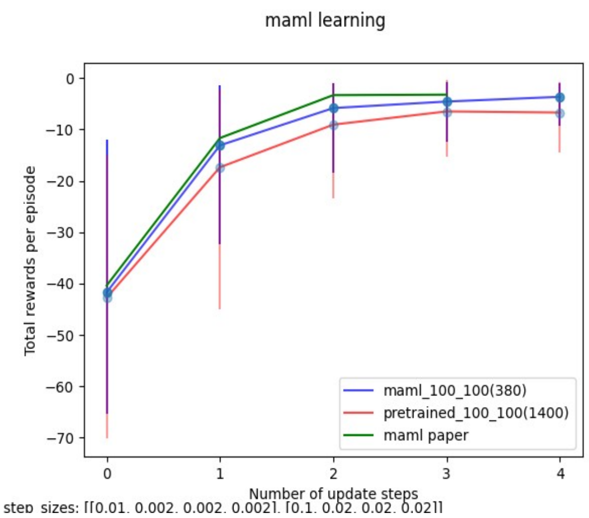
- Python, Pytorch
- Using TRPO for the meta-optimization
- Using most of the parameters as the authors
- Difference in step-sizes, policy variance and Neural Network output

Testing and evaluation

- Display and test script
- Task distribution
- Setup for equal evaluation
- Tested setups:
  - Hidden layer sizes
  - Variance(exploration)
  - Learning rate
  - Meta-updates
  - Normal training
  - Out of meta range adaptation

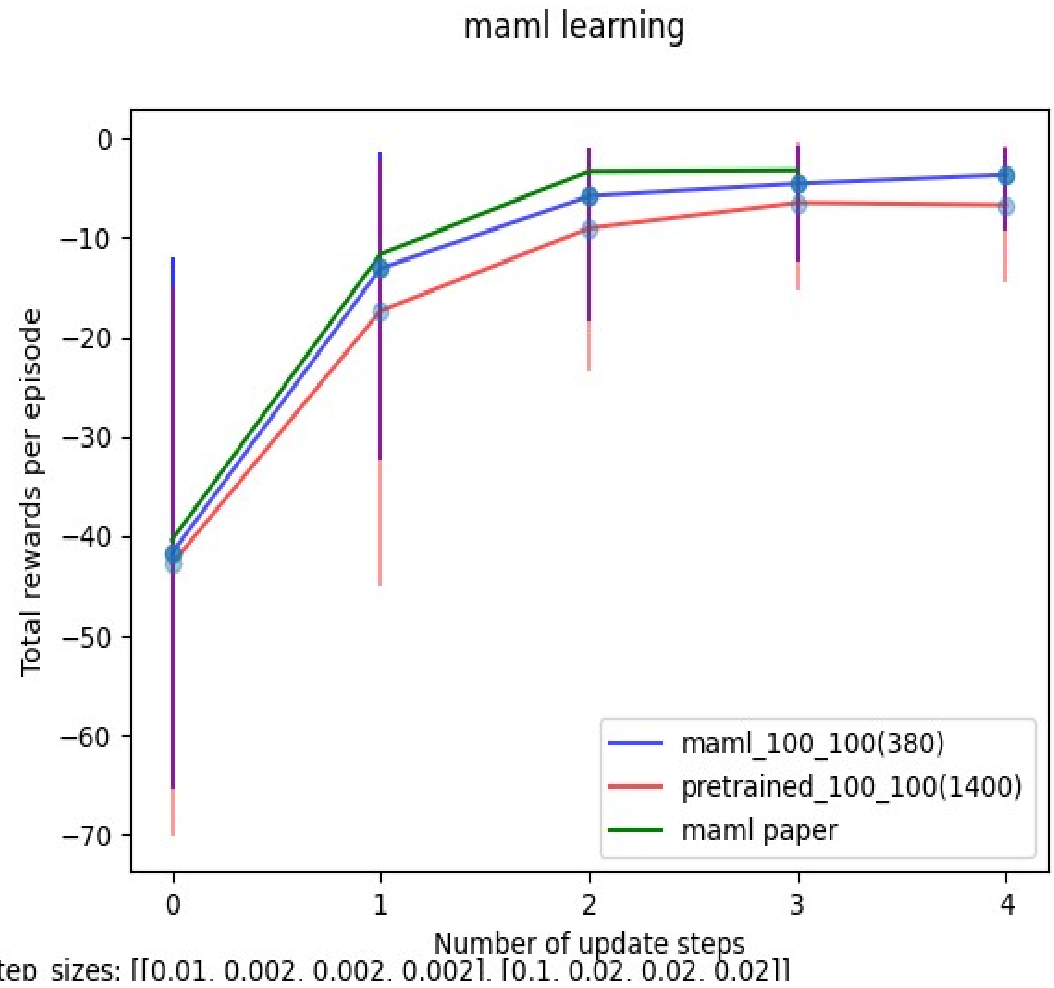






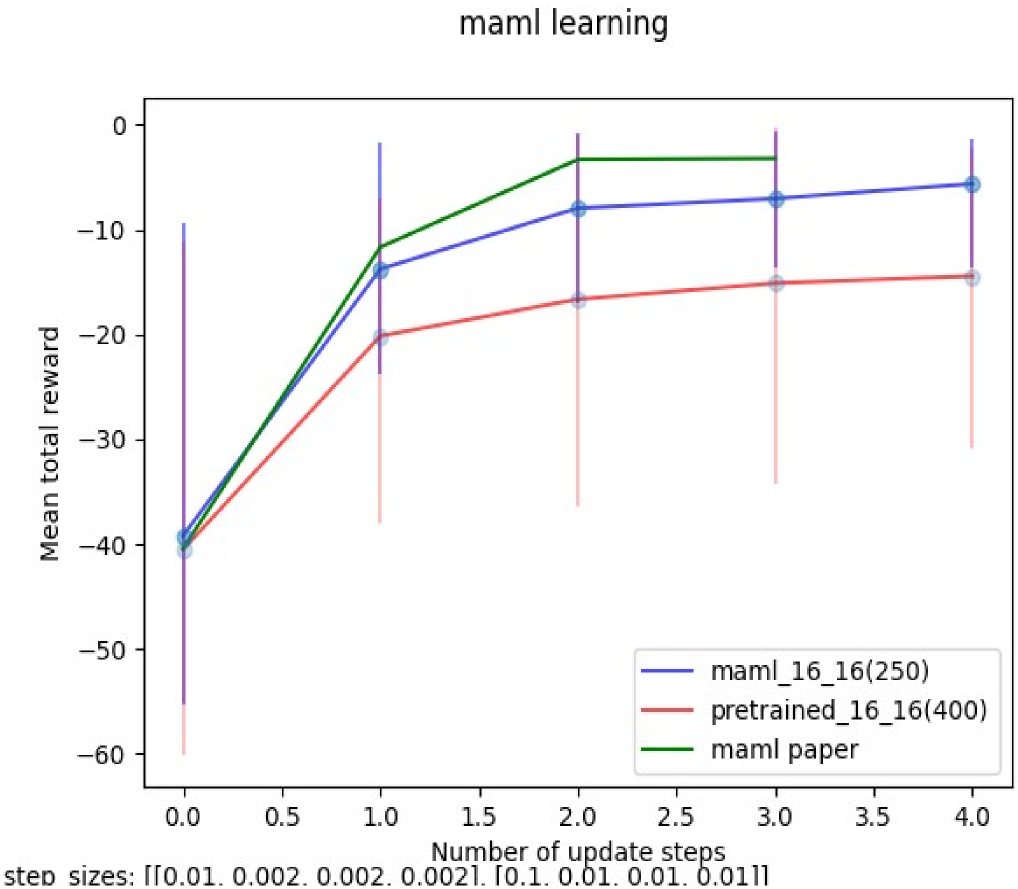
Adaptation comparison

- 100 units per HL
- 20 Tasks
- Errorbar(min/max)
- MAML paper comparison



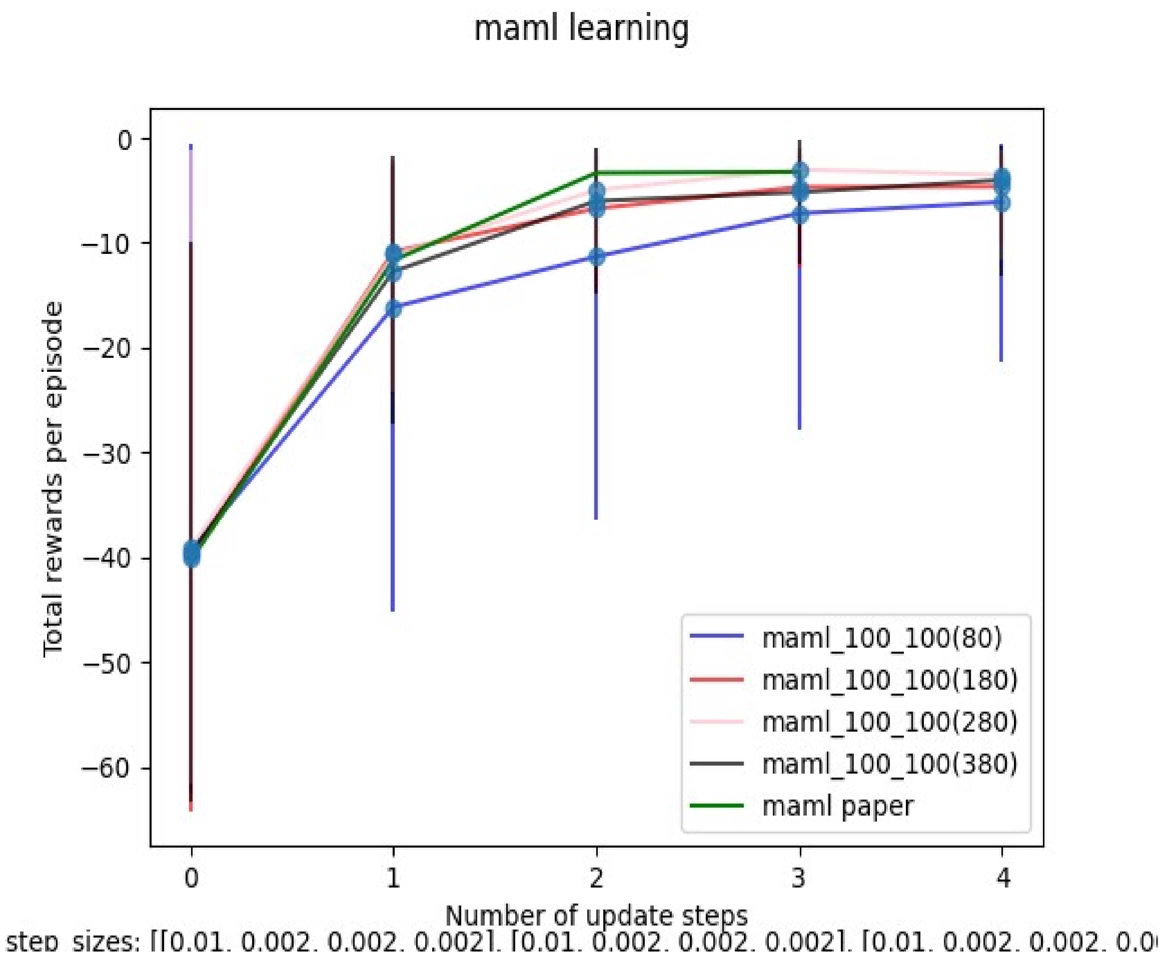
16 Units

- 16 Units per HL
- MAML still good
- Pretrained slightly worse



Effect of Meta-Updates

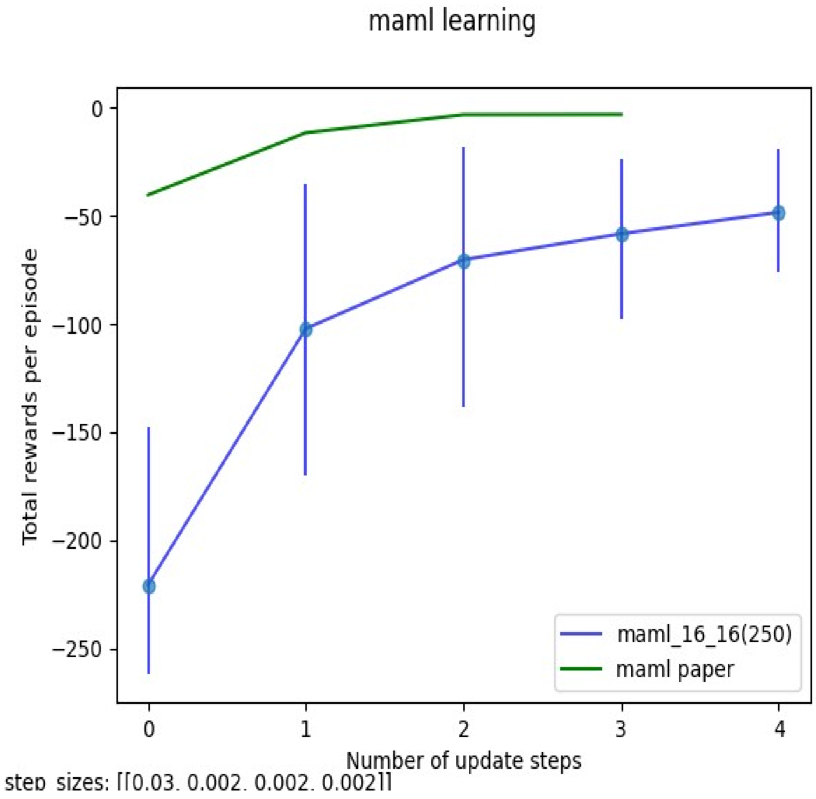
- 280 meta-updates in light orange performs best
- Same tasks comparison
- Just 80 updates has big min/max span and performs worse
- Empirically no more learning after ~380 updates



Out of range task performance

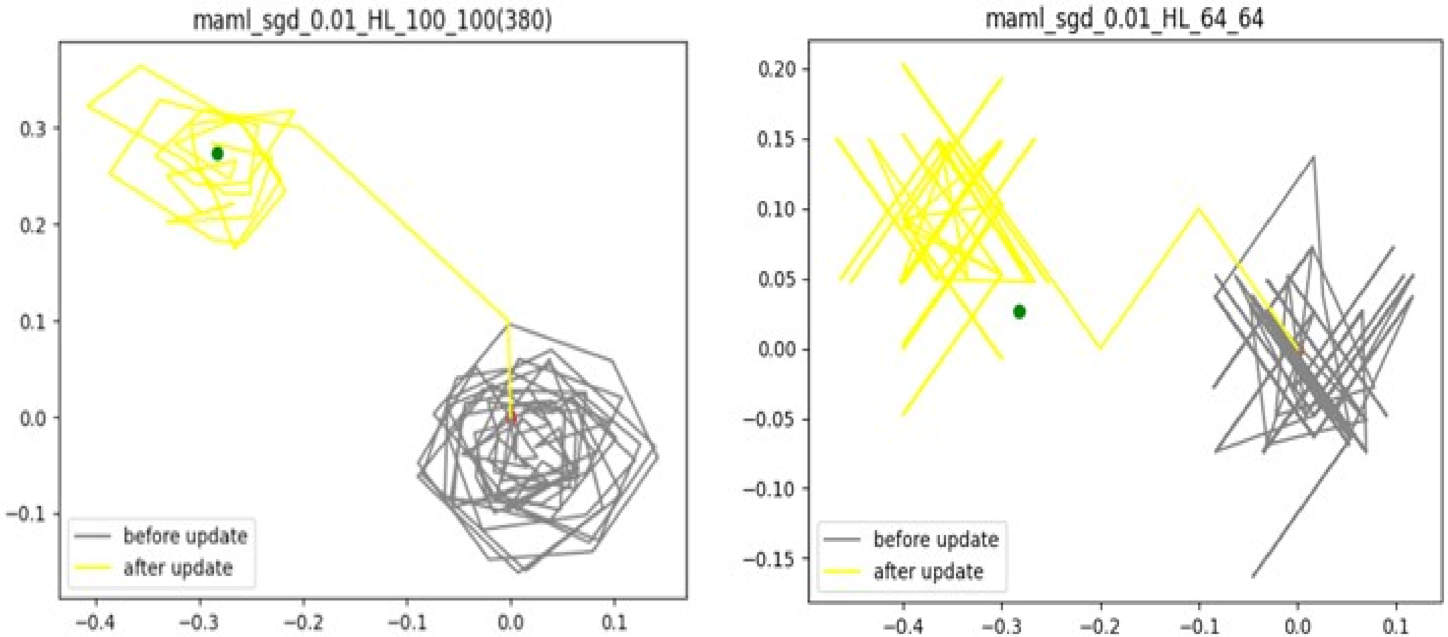
- Showing how a MAML pretrained model performs on tasks out of the meta-learned range

- Goal position |1|-|2| each direction. Usually 0.5



Variance

- In official Github repo a variance with 1.0 and a FC output is taken
- In the replication study a sigma value of 0.1 and an tanh for the NN output is used.



[0.1]

Discussion

- Replicated results
- Initial paper questions could be answered
- Notable differences in implementation
- Other aspects and performances were tested