Learning Transferability Metrics Across Tasks

Riley Edmunds, Jianlan Luo*

August 14, 2018

Abstract

For successful transfer and meta-learning, it's critical to have a method for choosing relevant source and target tasks. Existing metrics require empirically evaluating transferability by quantifying time required to achieve benchmark performance on the target task. We would like a metric that can quantify potential for success after transfer, without empirically performing said transfer.

In this work, we (1) describe relevant related work, highlighting lack of effective metrics for quantifying transfer in reinforcement learning, (2) introduce AEDist, a modification on the RBMDist transferability metric, (3) introduce a way to train and evaluate the metric, (4) introduce ways to incorporate the metric with Model-Agnostic Meta-Learning (MAML) by: guiding the learning of the metric using existing empirical metrics in MAML, guiding the learning of the MAML meta-learner using AEDist, and mutually training the AEDist metric and the MAML meta-model, (5) evaluate all above models, (6) pose relevant research questions for directions of future work in expanding upon the AEDist transferability metric, and (7) create the first MuJoCo tensegrity environment, capable of locomotion with model-free learning, and (8) motivate research into transferability of tensegrity locomotion policies across environments with varying degrees of terrain incline.

Our main takeaways are that (1) low AEDist is in fact predictive of tasks to which we can successfully transfer, in terms of immediate performance, asymptotic performance, and in terms of total return, and (2) incorporation of AEDist into the MAML meta-objective is not conducive to better MAML few-shot learning (neither MAML-return guided AEDist, nor AEDist-guided training of the MAML meta-learner, nor mutual guidance by both AEDist and empirical metrics).

Our main contributions are as follows:

- We introduce a "task-transferability" metric and evaluate its effectiveness as a stand-alone, and in combination with Model-Agnostic Meta-Learning.
- We provide insight into transferability of Tensegrity locomotion policies across terrains of varying inclines, and motivate further work both in the "task-transferability" metric, and its use in the domain of Tensegrity robotics.

1 Related Work

1.1 Current Transferability Metrics

There is no well-established metric for performance of transfer learning for deep reinforcement learning. Rather, empirical performance metrics (sometimes task-reward dependant) are used. Taylor et al. detail the following metrics [7]:

1. **Jumpstart**: "The initial performance of an agent in a target task may be improved by transfer from a source task."
2. **Asymptotic Performance**: "The final learned performance of an agent in the target task may be improved via transfer."
3. **Total Reward**: "The total reward accumulated by an agent (i.e., the area under the learning curve) may be improved if it uses transfer, compared to learning without transfer"
4. **Transfer Ratio**: "The ratio of the total reward accumulated by the transfer learner and the total reward accumulated by the non-transfer learner."
5. **Time to Threshold**: "The learning time needed by the agent to achieve a pre-specified performance level may be reduced via knowledge transfer."

Note that in order to evaluate potential for transferability between two tasks using the above metrics, it is necessary to perform such transfer. In other words, these metrics do not provide a transferability function parametrized by the task definitions, but rather a metric parametrized by transfer performance – a very expensive requirement when we are dealing with choosing between multiple possible tasks.

1.2 RBMDist

Ammar et al. [1] propose a measure of similarity between MDPs called RBMDist. To measure the similarity between tasks $\mathcal{T}_1$ and $\mathcal{T}_2$, they sample from each task $\mathcal{T}_1$ and $\mathcal{T}_2$ to generate datasets $\mathcal{D}_1 = \{\langle s_1^{(j)}, a_1^{(j)}, s_1^{'(j)} \rangle\}_{j=1}^m$ and $\mathcal{D}_2 = \{\langle s_2^{(j)}, a_2^{(j)}, s_2^{'(j)} \rangle\}_{j=1}^n$, where $s_1' \sim P_1(s_1^{(j)}), a_1^{(j)}$, $s_2' \sim P_2(s_2^{(j)}, a_2^{(j)})$.

Then, a Restricted Boltzmann Machine (RBM) [2] is trained on source data samples $\mathcal{D}_1$ from source task $\mathcal{T}_1$. This RBM is then fed target data samples $\mathcal{D}_2$. They define the mean of the reconstruction errors of all points in $\mathcal{D}_2$ to be the RBMDist between task $\mathcal{T}_1$ and $\mathcal{T}_2$.

RBMDist is independent of learning algorithm, as it relies exclusively on state transition tuples from the learned policy on each task.

Further, Ammar et al. show that clustering of tasks based on RBMDist resulted in clusters of tasks with similar dynamics, suggesting ties between transferability, similarity of MDP representations, and dynamics of learned task policies.

1.3 Model-Agnostic Meta-Learning

In a meta-learning scenario, we consider the problem of learning a model that can adapt to multiple tasks. Tasks are distributed according to $p(\mathcal{T})$. For a task $\mathcal{T}_i = \{\mathcal{L}_i, P_i\} \sim p(\mathcal{T})$, the K-shot learning problem requires finding a good model using only K labeled samples drawn from P_i and the feedback generated by $\mathcal{L}_i$. In Meta-Agnostic Meta-Learning (MAML) [5], this task is accomplished by explicitly optimizing a model with information about the general distribution of tasks $p(\mathcal{T})$. This general model can be

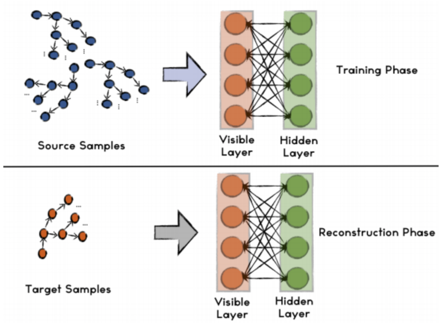

Figure 1: The training and evaluation phases of the AEDist transferability metric. Notice that source-task samples are used for training, and that target-task samples are used for reconstruction. Figure from [1].

quickly trained to work for any $\mathcal{T}_i \sim p(\mathcal{T})$ using only K samples drawn from P_i. To compute parameters θ_i' that work well on a particular task $\mathcal{T}_i$, we perform the gradient descent update on the meta-model parameters θ given a step size α on P_i:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_i(f_\theta) \tag{1}$$

We would like to learn a good θ on a set of training tasks, so that we can later perform this update using K samples from a particular task $\mathcal{T}_i$ and achieve a low loss. This leads to the MAML meta-objective:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_i(f_{\theta_i'}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_i(f_{\theta - \alpha \nabla_\theta \mathcal{L}_i(f_\theta)}) \tag{2}$$

That is, we optimize over the meta-model's parameters θ based on how each set of updated task-specific parameters θ_i' perform on their respective tasks $\mathcal{T}_i$. Once optimized, the meta-model is not likely to perform optimally on any of the individual tasks, but rather is close (in parameter space) to the optimal models for a variety of the individual tasks. Thus, going from the meta-model to a model optimized (*fine-tuned*) for a particular task consists of standard SGD optimization with respect to the task-specific loss, taking the meta-model parameters as the initial starting point.

There are many meta-learning models that can perform this process, however MAML is one of the simplest formulations and is also one of the most effective. MAML differs from other techniques [9, 4] in that it does not require extra parameters for the meta-objective. Despite this, it still achieves state-of-the-art empirical results [5]

2 Method

2.1 Overview

We take inspiration from Ammar et al.'s RMBDist to define a measure of similarity between tasks. We use AutoEncoders in place of Restricted Boltzmann Machines in order to avoid the need for careful model design [6], and possibility for straightforward extension to the many variants of the AutoEncoder [3] [8].

We train an autoencoder on the state transition tuples from task A to learn task A's most salient features, hypothesizing that the autoencoder will not only be capable of reconstructing samples from this source task, but also from similar tasks to which transfer may prove successful. To evaluate the "transferability" from source task A to target task B, we compute the autoencoder's reconstruction loss on state transition tuples from task B.

We incorporate our AEDist metric into the Model-Agnostic Meta-Learning training regime. The MAML framework naturally lends itself as a testbed for quantification of transferability, as it is explicitly optimizing for this criterion. Further, we can use MAML to evaluate whether AEDist reconstruction errors align with empirical returns from MAML transfer.

We hypothesize that there are two chief potential advantages to incorporation of AEDist into the MAML training regime: first, that we can guide the learning of an accurate AEDist metric with task-specific empirical returns; second, that we can guide the learning of the MAML meta-model by task-specific AEDist scores.

We highlight that current empirical measures of transfer largely focus on evaluating performance of transfer in the first k gradient updates (for a typically small k), and at the same time, that AEDist can evaluate task similarity for a considerable fraction of the distribution of possible actions (it considers a large distribution of state transition tuples for each task). We thus hypothesize that current empirical measures of transfer are likely predictive of few-shot transfer success, and that the AEDist metric is likely a more robust predictor of longer-term transfer success.

2.2 Notation

2.2.1 MAML & MDPs

f_θ := MAML meta-model
θ := MAML meta-model parameters
$f_{\theta'_i}$:= MAML fine-tuned model for task $\mathcal{T}_i$
$\pi_{\mathcal{T}_i}$:= expert policy on task $\mathcal{T}_i$
$R^{\mathcal{T}_i}_{\pi_{\mathcal{T}_i}}$:= reward accrued expert policy $\pi_{\mathcal{T}_i}$ on task $\mathcal{T}_i$
$R^{\mathcal{T}_i}_{f}$:= reward accrued running policy f on task $\mathcal{T}_i$ (after fine-tuning)
$\mathcal{L}_{\mathcal{T}_i}$:= task specific loss for task $\mathcal{T}_i$
$\mathcal{D}_{\mathcal{T}_i}$:= rollout data of meta-model on task $\mathcal{T}_i$
$\mathcal{D}$:= rollout data of meta-model on all tasks

For the remainder of the standard MAML notation, see MAML [5].

2.2.2 AutoEncoder

AE := AutoEncoder network
$\mathcal{L}_{AE}(\cdot)$:= AutoEncoder AE's loss
$AE_{\mathcal{D}}$:= AutoEncoder network trained to reconstruct $\mathcal{D}$
$\mathcal{L}_{AE_{\mathcal{D}}}(\cdot)$:= Loss of AutoEncoder network trained to reconstruct $\mathcal{D}$
$\mathcal{L}_{AE_{\mathcal{D}_{\mathcal{T}_i}}}(\cdot)$:= Task-weighted loss for task $\mathcal{T}_i$
For more on AutoEncoders, please see Chapter 14 of the Deep Learning Book [6].

2.2.3 Transfer Metrics

$t_{\mathcal{T}_A \rightarrow \mathcal{T}_B}$:= $R^{\mathcal{T}_B}_{\pi_{\mathcal{T}_B}} - R^{\mathcal{T}_B}_{f_A}$ ("Return-based Transferability Loss")
AEDist$_{\mathcal{T}_A \rightarrow \mathcal{T}_B}$:= $\mathcal{L}_{AE_{\mathcal{D}_{\mathcal{T}_A}}}(\mathcal{D}_{\mathcal{T}_B})$ ("AutoEncoder-based Transferability" between $\mathcal{T}_A$ and $\mathcal{T}_B$)
c := constant hyper-parameter $\in [0, 1]$

$d :=$ constant hyper-parameter $\in [0, 1]$

2.3 Theory

We explore four ways to incorporate AEDist into the MAML framework. The pseudocode for the full implementation is detailed in algorithm 1, with our additions to canonical MAML in green. In each of the following sections, we include a subset of these green lines, appropriately indicating which lines from algorithm 1 should be included.

2.4 Vanilla AEDist after training MAML

We use the distribution of tasks defined by a MAML instance to quantify the measure of transferability from the meta-model to each subtask $\mathcal{T}_i$.

Here, we train the meta-model to convergence. Then, we sample from f_θ and from $\pi_{\mathcal{T}_i}$ in their respective environments, generating $\mathcal{D}$ and $\mathcal{D}_i$. We train an autoencoder on $\mathcal{D}$, and evaluate it on $\mathcal{D}_i$ in order to quantify the AEDist between the meta-model "task" and the subtask $\mathcal{T}_i$.

Again, MAML serves as a natural testbed for defining relevant tasks, as their MAML empirical transfer metrics can be compared with AEDist.

2.5 Empirical Transfer Metrics as a Guide to AEDist

Next, we introduce a training scheme for guiding the learning of the AEDist metric using existing empirical metrics within MAML.

By the aforementioned hypothesis that empirical transferability metrics predict short-term transferability and that AEDist predicts longer-term transferability, we motivate the use of empirical transferability metrics to help guide the learning of an AEDist autoencoder, with the objective of learning a transferability metric predictive of both long-term transferability and short-term transferability.

Between each MAML training loop, we use the gathered state transition tuples from rollouts of $f_{\theta_i'}$ on $\mathcal{T}_i$ to quantify the "Return Based Transferability Loss" $t_{\mathcal{T}\to\mathcal{T}_i}$ between the meta-model "task," and each subtask $\mathcal{T}_i$. We then train our autoencoder, scaling the autoencoder's loss $\mathcal{L}_{AE}$ for each sample by a scaling factor c corresponding to the "Return Based Transferability Loss" $t_{\mathcal{T}\to\mathcal{T}_i}$ between the meta-model, and the task from which the sample was drawn. Include lines 4, 11, 13 and 14 from algorithm 1.

In this way, we increase the loss for samples for which the "Task Based Transferability Loss" is high, and decrease the loss for samples for which the "Task Based Transferability Loss" is low. We hypothesize that "guiding" the transfer metric's learning in this way may allow it to learn to predict few-shot transferability.

2.6 AEDist as a Guide to Empirical Transfer Metrics

Next, we introduce a training scheme for guiding the learning of the MAML meta-learner using the AEDist metric.

Assuming AEDist's ability to predict longer-term transferability, we hypothesize that penalizing high AEDist may facilitate the learning of more robust meta-learners, effective in longer-term transfer scenarios. Additionally, given AEDist encodes measures of the similarity between the dynamics of two tasks [1], we hope to use MAML to learn initialization of policies whose dynamics align with target-task dynamics.

Between each MAML training loop, we use the gathered state transition tuples to train our autoencoder, with traditional MSE autoencoder's loss $\mathcal{L}_{AE}$ on samples across all tasks. Then, when updating the meta-model's parameters, we scale each adapted-parameter (one-shot fine-tuned parameter) task loss by the corresponding task's AEDist. In other words, we scale the meta-model gradient update by an AEDist scaling factor on a task-specific basis. Include lines 4, 11, 14 and 16 from algorithm 1.

In this way, we increase meta-model loss for tasks whose AEDist is high, and decrease loss for tasks for which AEDist is low. We thus incentivize learning a meta-model for which both empirical transfer metrics and AEDist are low.

2.7 Mutually Guiding AEDist and Empirical Transfer Metrics

Finally, we introduce a training scheme for guiding both the learning of the MAML meta-learner using the AEDist metric, and the learning of the AEDist metric using the existing empirical metrics in MAML (combining the two aforementioned modes of guidance).

Assuming optimality of empirical transferability metrics for predicting short-term transferability and optimality of AEDist for predicting longer-term transferability, we hypothesize that jointly optimizing the two metrics will yield an AEDist metric that is robust and accurate for both short and long-term transfer scenarios, and a MAML initialization that can effectively fine-tune to subtask objectives in both short and long-term transfer scenarios.

In implementing this approach, we combine all aforementioned algorithms. Specifically, we gather state transition tuples, quantify the "Return Based Transferability Loss" $t_{\mathcal{T}\rightarrow\mathcal{T}_i}$ between the meta-model "task," and each subtask $\mathcal{T}_i$, train our autoencoder (scaling its loss $\mathcal{L}_{AE}$ for each sample by a scaling factor $c \cdot t_{\mathcal{T}\rightarrow\mathcal{T}_i}$), and, finally, incorporate each task's AEDist into the meta-model update. See the full Algorithm 1, including lines 4, 11, 13, 14 and 16.

Assuming optimality of both approaches, we hypothesize potential mutual improvement and convergence to an optima favorable to both short and long-term transfer objectives.

3 Results

Our main takeaways are that (1) low AEDist is in fact predictive of tasks to which we can successfully transfer, in terms of immediate performance, asymptotic performance, and in terms of total return, and (2) incorporation of AEDist into the MAML meta-objective is not conducive to better MAML few-shot learning (neither MAML-return guided AEDist, nor AEDist-guided training of the MAML meta-learner).

3.1 Experimental Setup

All experiments use Finn et al's MAML framework [5]. We use Rllab, the MuJoCo physics engine, and the MuJoCo Ant environment. Our tasks are defined by uniformly sampling from the Ant objectives: directional speed goal, target velocity goal, and target pose goal, with randomly sampled hyper-parameters to each task, as in original MAML. The fine-tuned tasks are Ant environments with directional goals of -1 and 1.

3.2 Vanilla AEDist after training MAML

We train MAML for 1000 iterations with task distribution $p(\mathcal{T})$, and then fine-tune on a directional velocity task $\mathcal{T}_i$ of target direction 1 (forward locomotion) drawn from $p(\mathcal{T})$ (see the magenta plot in Figure 2, and a directional velocity task $\mathcal{T}_j$ of target velocity 0 (backward locomotion) drawn from $p(\mathcal{T})$ (see the yellow plot in Figure 2. The listed

Algorithm 1 Mutually Guiding AEDist and MAML Meta-Model for learning MDP transferability metrics (additions to MAML are lines in green).

```
Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha, \beta$: step size hyperparameters
Require: Expert policy rewards $R^{\mathcal{T}_i}_{\pi}$ on each task $\mathcal{T}_i$
randomly initialize $\theta$
while not done do
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    $\mathcal{D} \leftarrow (\ )$
    for all $\mathcal{T}_i$ do:
        Sample K trajectories $\mathcal{D}_{\mathcal{T}_i} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_\theta$ in $\mathcal{T}_i$
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Eq. 1
        Compute adapted parameters with GD: $\theta'_i = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
        Sample K trajectories $\mathcal{D}'_i = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta'_i}$ in $\mathcal{T}_i$
    end for
    $\mathcal{D} \leftarrow \bigcup_{\mathcal{T}_i \sim p\mathcal{T}} \mathcal{D}_{\mathcal{T}_i}$
    for all $\mathcal{T}_i$ do:
        Compute $t_{\mathcal{T}\to\mathcal{T}_i}$ using $R^{\mathcal{T}_i}_{f_{\theta'_i}}$ by roll-out of $f_{\theta'_i}$ in $\mathcal{T}_i$
        Train AE on $\mathcal{D}$ using $\mathcal{L}_{AE} = c(t_{\mathcal{T}\to\mathcal{T}_i}) \cdot \mathcal{L}_{AE}$
    end for
    Update $\theta \leftarrow \theta - \beta\nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})}( b(1-$ AEDist$_{\mathcal{T}_i})\cdot\mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}))$ using each $D'_i$ and $\mathcal{L}_{\mathcal{T}_i}$
end while
```

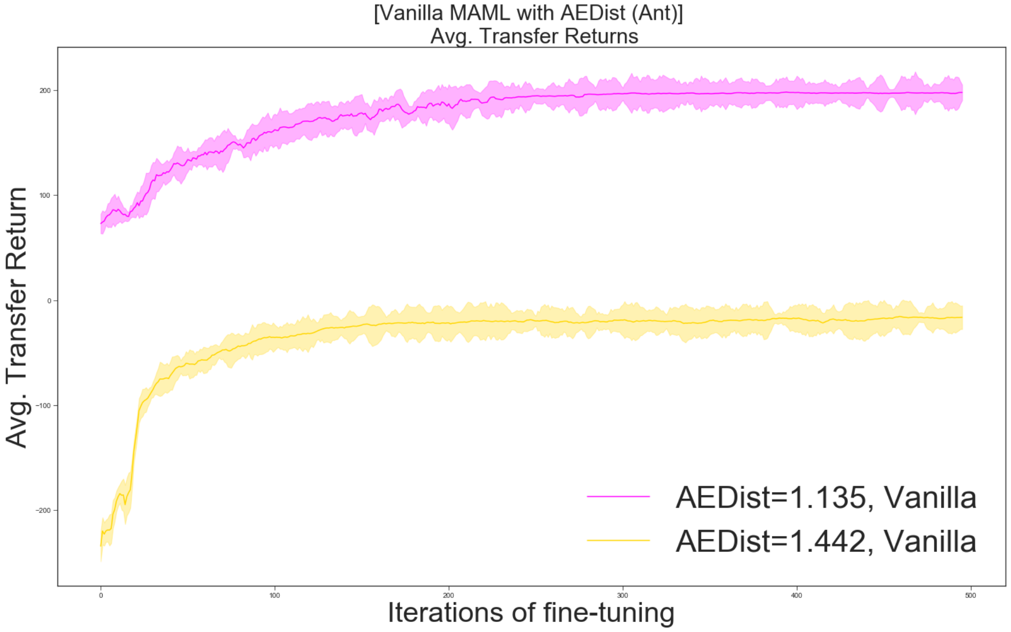

Figure 2: Average transfer returns from MAML meta-model to tasks with varying AEDists, using Vanilla AEDist and Vanilla MAML.

AEDist	Jumpstart	Asymptotic Performance	Total Reward
1.135	14.72	82.86	11067.9
1.442	-6.76	93.87	10655

Table 1: Empirical transfer metrics corresponding to vanilla AEDist.

AEDist Order	Jumpstart ∇	Asymptotic Performance ∇	Total Reward ∇	Transfer Ratio
1.135 over 1.442	345.96	213.86	106107	5.03

Table 2: Comparison of empirical performance between guided AEDist metrics with varying AEDists.

AEDists correspond to tasks $\mathcal{T}_i$ and $\mathcal{T}_j$ after the last iteration of the MAML meta-model training. Our measure of empirical transfer is $t_{\mathcal{T}_A \rightarrow \mathcal{T}_B}$, the difference between expert performance and few-shot fine-tuned performance on the target task.

We recorded empirical transferability metrics between the meta-model and $\mathcal{T}_i$, and between the meta-model and $\mathcal{T}_j$. These are reported above in the table above 3. The differences between the empirical metrics for these two transfers is shown in the table below 4.

In this case, we saw that, according to our hypotheses, tasks with low AEDists had significantly lower empirical transferability metrics. These results suggest that low AEDist is in fact predictive of successful transferability. The asymptotic results suggest that AEDist is predictive not only of few-shot transfer success, but also of longer-term transfer success.

3.3 Empirical Transfer Metrics as a Guide to AEDist

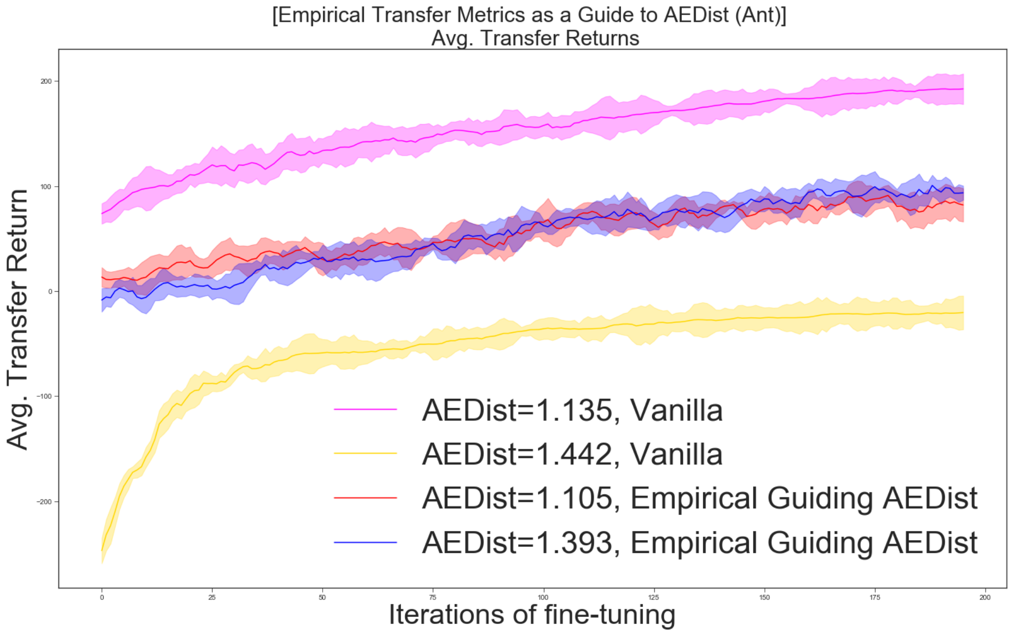

Figure 3: Average transfer returns from MAML meta-model to tasks with varying AEDists, using AEDist guided by empirical transfer metrics.

Using empirical transfer metrics to guide AEDist learning proved unsuccessful. In

AEDist	Jumpstart	Asymptotic Performance	Total Reward
1.135	72.58	197.33	88522.4
1.442	-232.27	-16.53	-17584.2

Table 3: Empirical transfer metrics corresponding to guided AEDist.

AEDist Order	Jumpstart ∇	Asymptotic Performance ∇	Total Reward ∇	Transfer Ratio
1.135 over 1.442	22.86	-11.00	412.93	1.04

Table 4: Comparison of empirical performance between guided AEDist metrics with varying AEDists.

fact, in this case, the AEDist transferability metric (visible in Figure 3) proved to be far less indicative of transfer performance than when trained after running MAML to convergence, as in the previous experiment (visible in Figure 2).

As can be seen by the empirical performance deltas, there is no longer a significant distinction in transfer performance between the task with low AEDist and the task with high AEDist.

3.4 AEDist as a Guide to Empirical Transfer Metrics

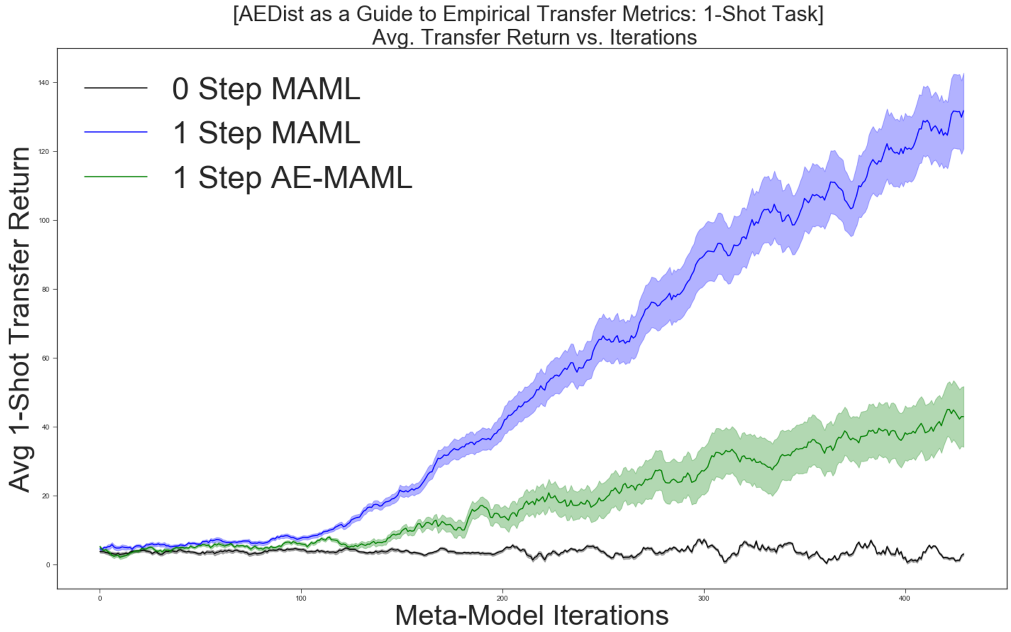

Figure 4: Average one-shot transfer returns to a single subtask $\mathcal{T}_i$, from a Vanilla MAML meta-model (blue) and from a MAML model guided by AEDist (green).

Using AEDist to guide meta-model learning proved unsuccessful in learning an AEDist model predictive of transferability. The "AE-MAML" one-step transfer returns can be seen in Figure 4 in green, and, although monotonically increasing, are less successful that their vanilla MAML counterpart. Additional experimentation with guiding meta-model learning using AEDist is detailed in the next section.

3.5 Mutually Guiding AEDist and Empirical Transfer Metrics

Incorporating mutual guidance, we saw unsuccessful one-shot learning for the MAML subtasks, as can be seen in Figure 5.

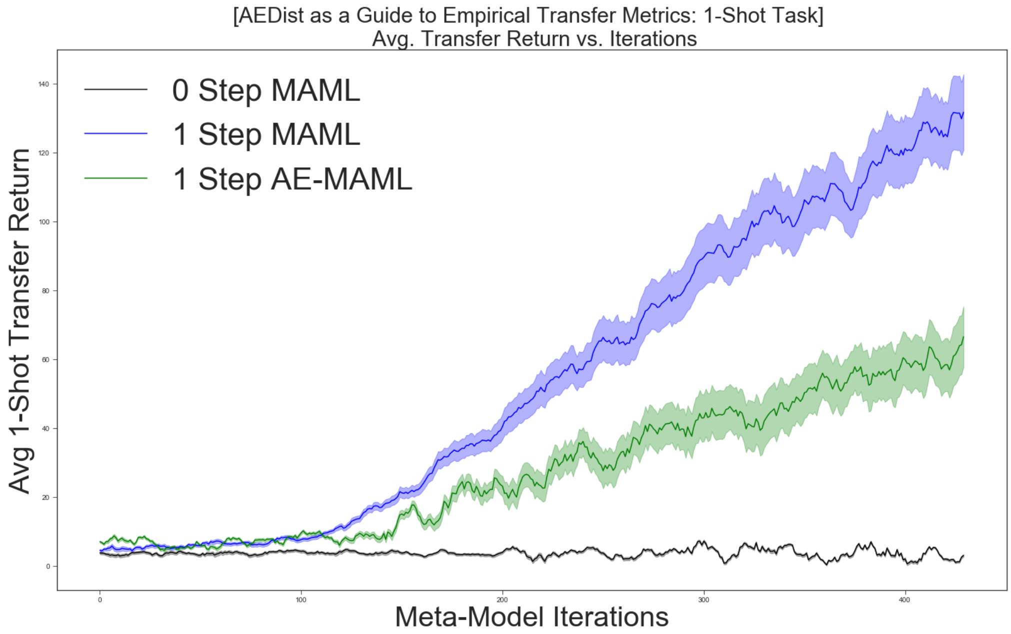

Figure 5: Average one-shot transfer returns to a single subtask $\mathcal{T}_i$, from a Vanilla MAML meta-model (blue) and from a MAML model learned with mutual guidance (green).

Reflecting on the success of AEDist, and the failure of both single and mutual guidance, we suspect that the incorporation of the distance metric requires tweaking and tuning before it can achieve good performance, or that Finn et al.'s [5] approach of directly using gradients from one-shot fine-tuned models may lead to a more robust meta-model initialization.

4 Future Work & Tensegrity Robotics

Finally, we motivate future research into transferability across both agents and tasks. Namely, we motivate study of transfer of tensegrity robot locomotion policies across terrains on varying inclines, and present preliminary work towards this end, as described below.

4.1 Tensegrity Work

Finally, our previous results show that there is striking simplicity and periodicity to the gait of tensegrity robots. We created a MuJoCo tensegrity environment, and aim to further investigate transfer of tensegrity locomotion between terrains of varying inclines, using AEDist.

- Previous Work: Our previous work on control policies for 6-bar Tensegrity robots using Guided Policy Search (MDGPS and PIGPS) showed that both the distance traveled and learned rolling gait of tensegrity locomotion policies was largely unchanged after a reduction in observation space from 12 dimensions to 3 dimensions.

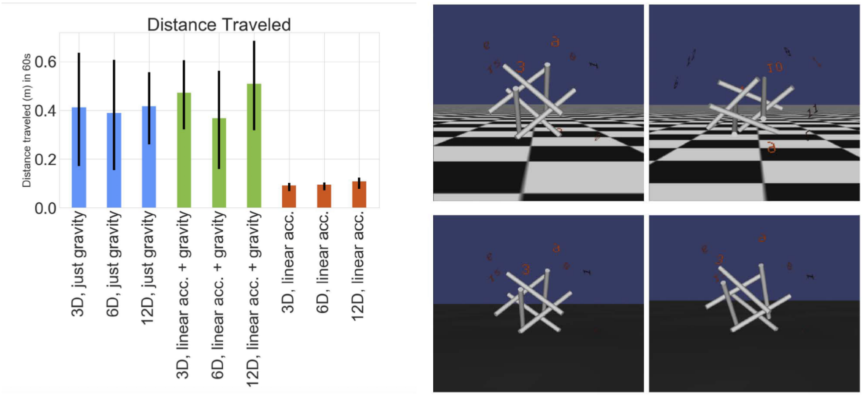

Figure 6: Comparison of tensegrity locomotion performance under varied observation spaces (left), and tensegrity robots rolling on flat ground in the NASA Tensegrity Robotics Toolkit (NTRT).

- Modeling: Over the last couple months, we crafted a MuJoCo Model of the tensegrity 6-bar robot for locomotion (objective is velocity along a line in the x-y plane).

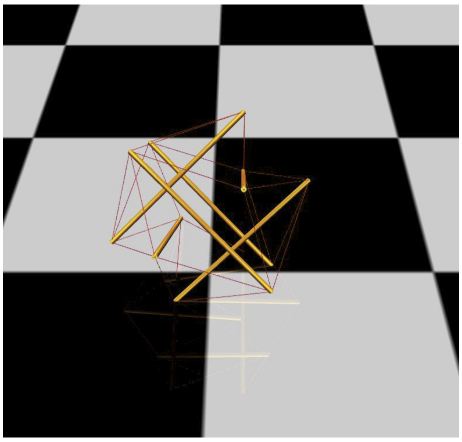

Figure 7: The MuJoCo tensegrity model we built to study transferability of locomotion policies across terrains of varying inclines.

- Method: We plan to use AEDist to explore transfer across locomotion tasks on terrains of varying inclines (MAML subtasks). This requires looking at transfer between tasks characterized by different environments, not simply different objectives for the agent (to our knowledge, the only definition of tasks in reinforcement learning that has been studied so far with MAML).

4.2 Modifications on AEDist

- We plan to use AEDist to explore transferability, not only between the MAML meta-model and subtasks, but also between subtasks themselves.

- We plan to attempt to cluster tasks by AEDist, and investigate transferability between tasks lying in the same cluster, as well as transferability between tasks lying in different clusters (in similar fashion to the experiments of Ammar et al. [1]).
- We would like to explore the use of the latent space of a VAEDist transferability metric (using Variational Autoencoders) to attempt generation of state transition tuples for synthetic tasks with minimal VAEDist [3], and possible reconstruction of such tasks from their state transition tuples. We believe this approach could allow for discovery of optimal source and target tasks for meta-learning.
- We would like to explore the use of recurrent autoencoders for non-Markovian encoding of MDP state transition tuples and, as suggested by Ammar et al. [1], characterization of transferability metrics across MDPs with domains of various sizes (transfer between tasks with different cardinality of observation space or action space).

5 Conclusion

In conclusion, we proposed and evaluated a metric to quantify "transferability" of reinforcement learning policies between tasks. The metric is model-agnostic, as it exclusively relies on state transition tuples. More importantly, the metric allows quantification of transferability between tasks without requiring expensive empirical evaluation of said transfer. We envision AEDist (or variants thereof) being used for automated selection of source and target tasks in meta-learning scenarios. We propose and evaluate methods for incorporating the AEDist transferability metric into the Model-Agnostic Meta-Learning framework. Further, we present a realistic MuJoCo model and environment for a Tensegrity 6-bar robot – a step towards model-free learning approaches in the tensegrity community. Lastly, we motivate research into transferability of tensegrity locomotion policies between terrains of various inclines, and promising extensions to AEDist.

References

[1] Haitham Bou Ammar, Eric Eaton, Matthew E. Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tulys. An Automated Measure of MDP Similarity for Transfer in Reinforcement Learning. *Machine Learning for Interactive Systems: AAAI-14 Workshop*, 2014.

[2] George Dahl, Marc aurelio Ranzato, Abdel rahman Mohamed, and Geoffrey E Hinton. Phone Recognition with the Mean-Covariance Restricted Boltzmann Machine. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 469–477. Curran Associates, Inc., 2010.

[3] C. Doersch. Tutorial on Variational Autoencoders. *ArXiv e-prints*, June 2016.

[4] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[7] Peter Stone Matthew E. Taylor. Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research 10*, 2009.

[8] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. *ICML*, 2011.

[9] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.