Deep Lagrangian Networks Applied to Underactuated Systems

Matt Philippi, Tom Power, Craig Knuth, Kaustav Chakraborty
Robotics Institute
University of Michigan - Ann Arbor
Ann Arbor, MI 48109

Abstract

This project investigates how to specialize deep learning to learning the dynamics of a mechanical system by incorporating physics as a prior in a general way. We take an existing method from the literature which injects Lagrangian mechanics into the structure of a neural network. We first implement the method and verify it on a two-bar linkage mechanism which was presented in the original paper. We are able to reproduce the results of the original paper on the fully actuated system and see data efficiency and generalization benefits. However, in the under-actuated case we find that the proposed method is outperformed in terms of data efficiency and generalization by a standard feed-forward neural network.

Introduction

Deep learning has been one of the most promising areas of research in recent years. In this paper we contribute three items:

1) An implementation of the DeLaN,
2) An adaption for application to an underactuated cartpole system, and
3) Empirical results of application to fitting to cartpole trajectories.

Method

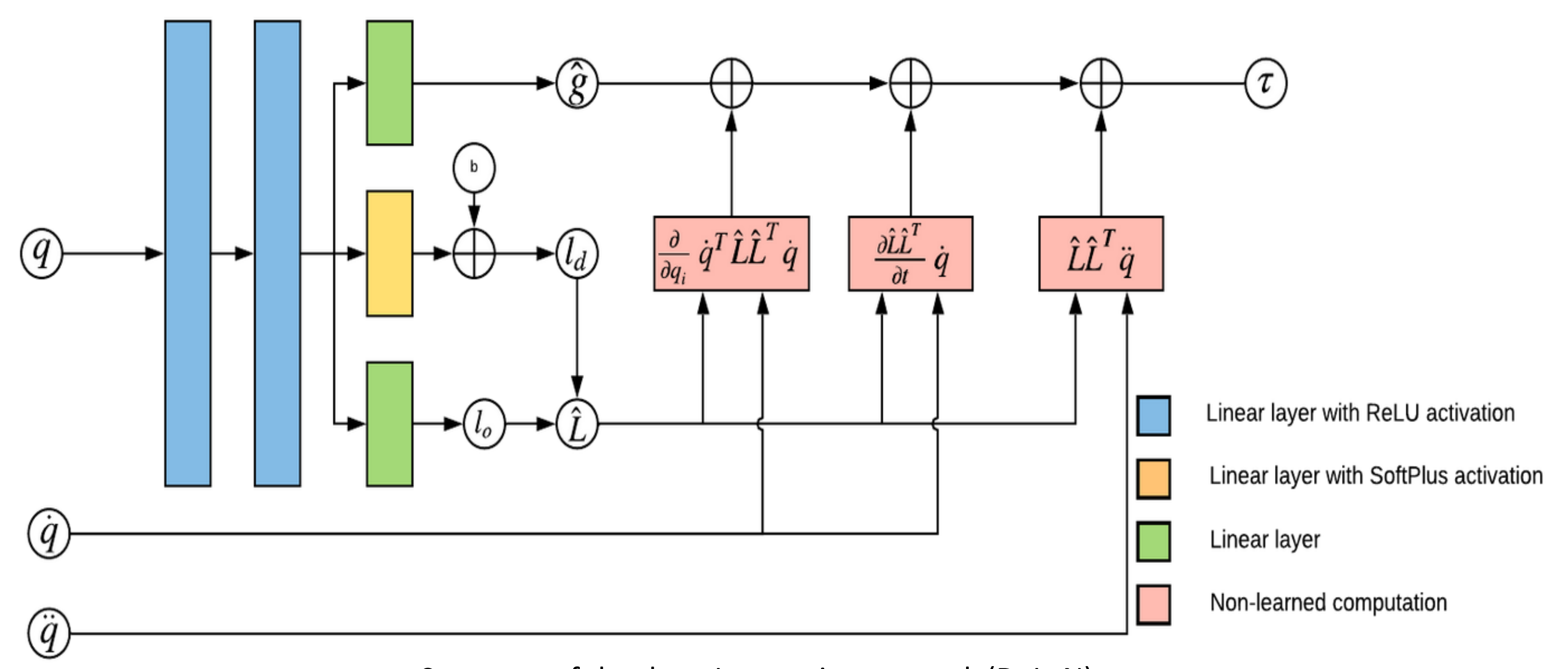
Lagrangian: $\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{c}(\mathbf{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$

Mass Matrix: $\boldsymbol{H}(\boldsymbol{q}) = \boldsymbol{L}(\boldsymbol{q})\boldsymbol{L}(\boldsymbol{q})^{\mathbf{T}}$

Coriolis and Centripetal Torques: $\mathbf{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \dot{\boldsymbol{H}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^{T}\boldsymbol{H}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right)^{T}$

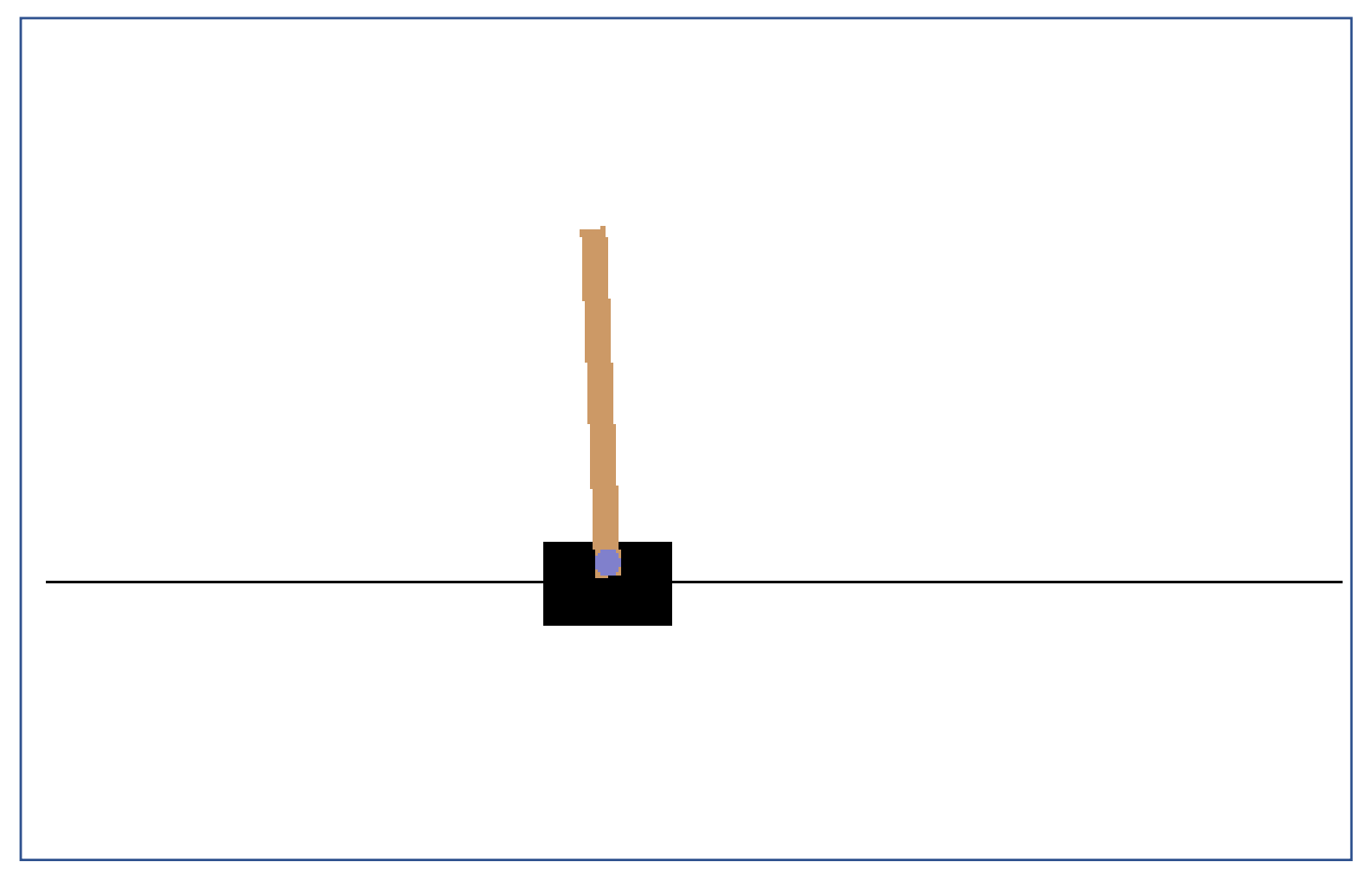
Gravitational Torque: $\mathbf{g}(\boldsymbol{q})$

Goal: Predict control torque τ from state $(q, \dot{q}, \ddot{q})$ using deep network

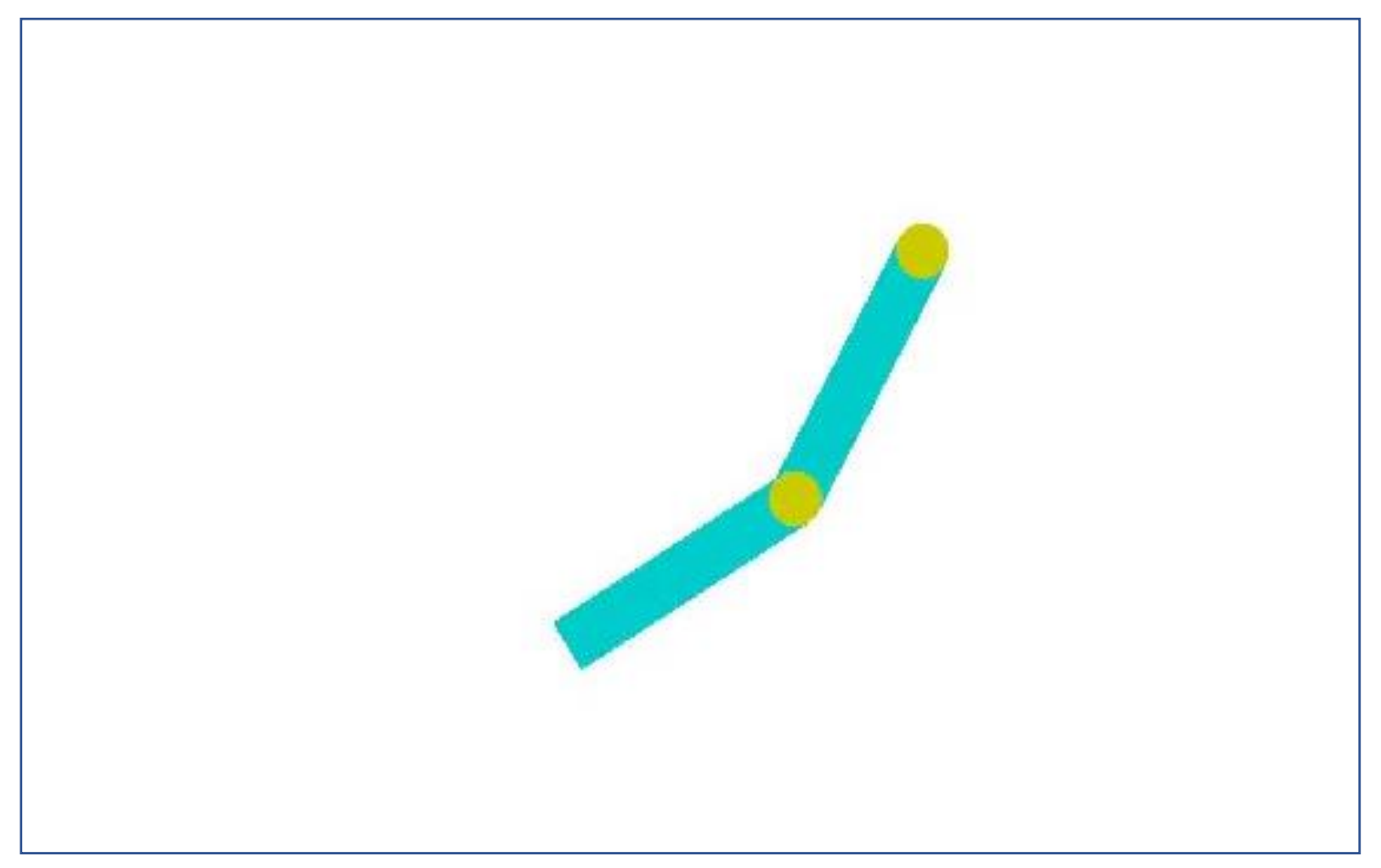

q
$\dot{q}$
$\ddot{q}$
$\hat{g}$
b
l_d
l_o
$\hat{L}$
τ
$\frac{\partial}{\partial q_i}\dot{q}^T\hat{L}\hat{L}^T\dot{q}$
$\frac{\partial \hat{L}\hat{L}^T}{\partial t}\dot{q}$
$\hat{L}\hat{L}^T\ddot{q}$
Linear layer with ReLU activation
Linear layer with SoftPlus activation
Linear layer
Non-learned computation

Structure of the deep Lagrangian network (DeLaN)

Systems

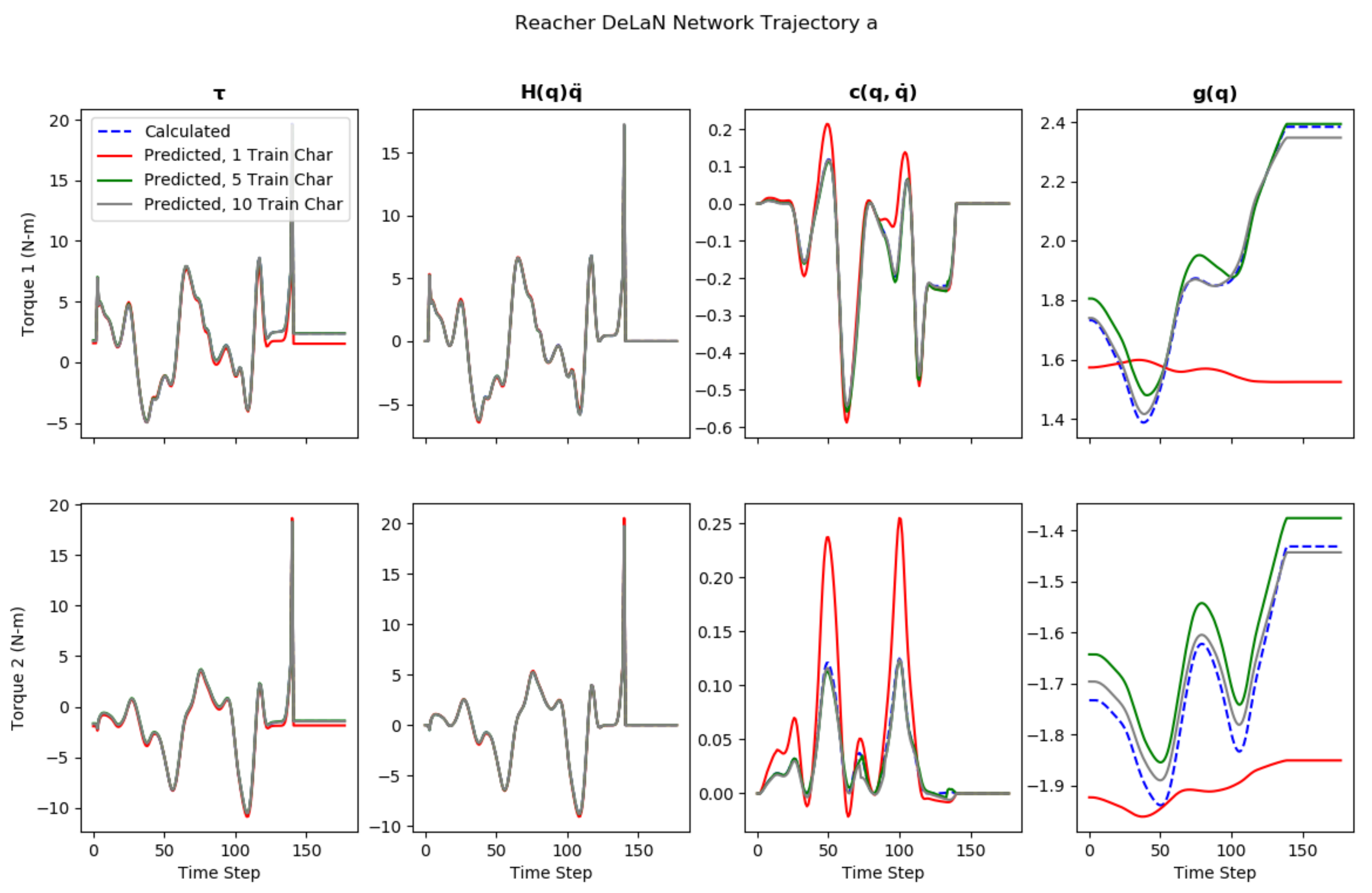


Cartpole Environment on OpenAI

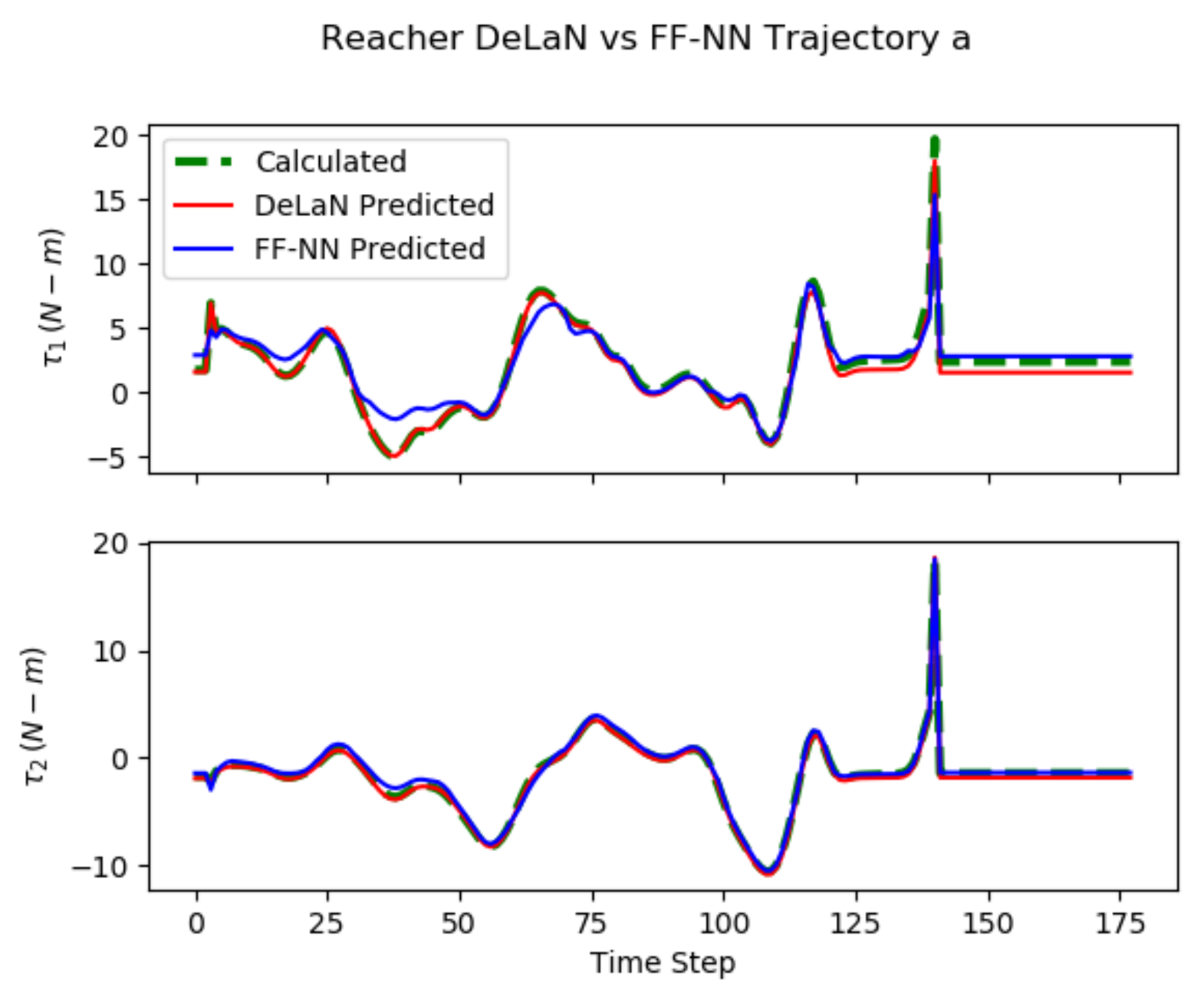


Reacher Environment on OpenAI

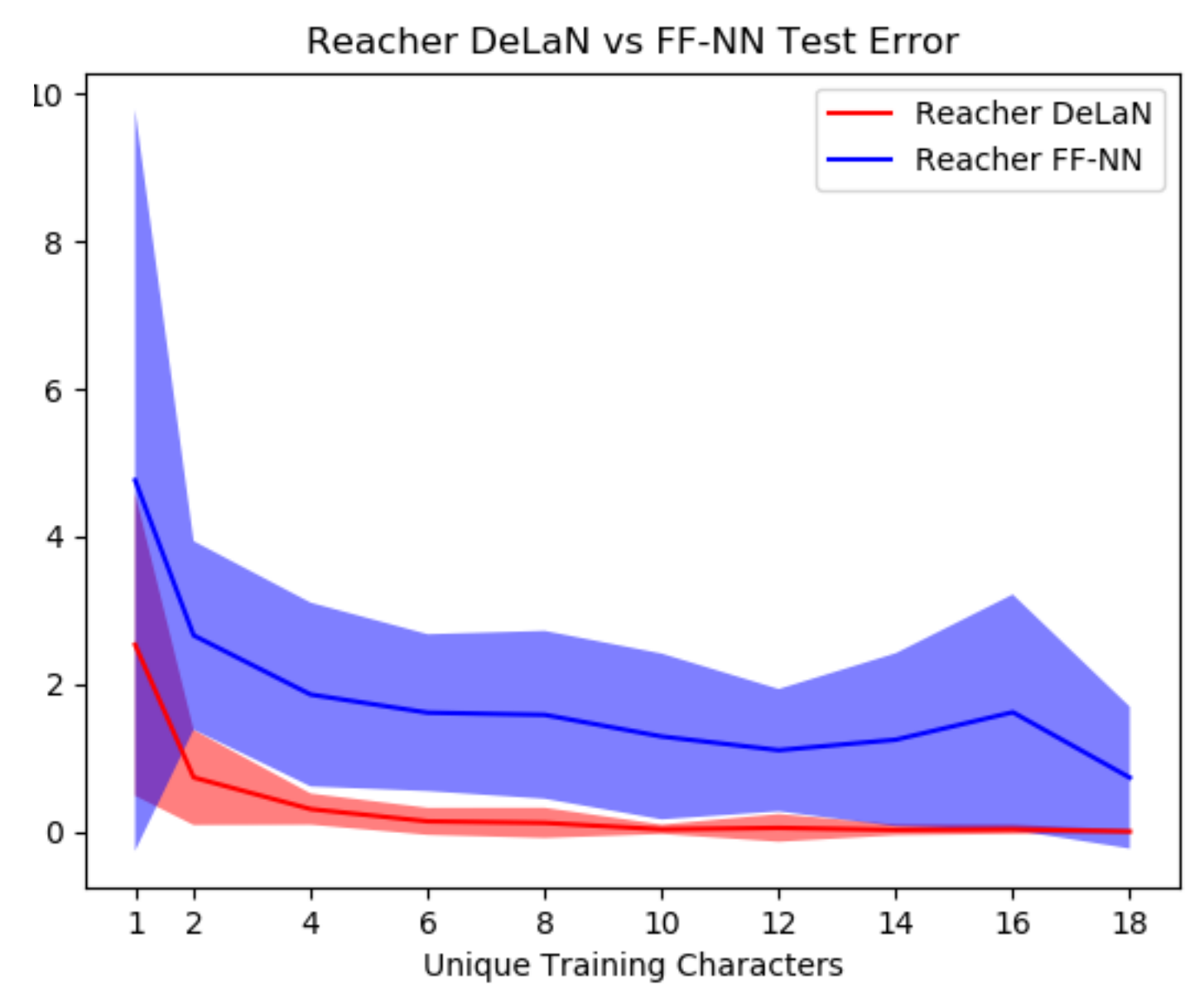
Experiments and Results



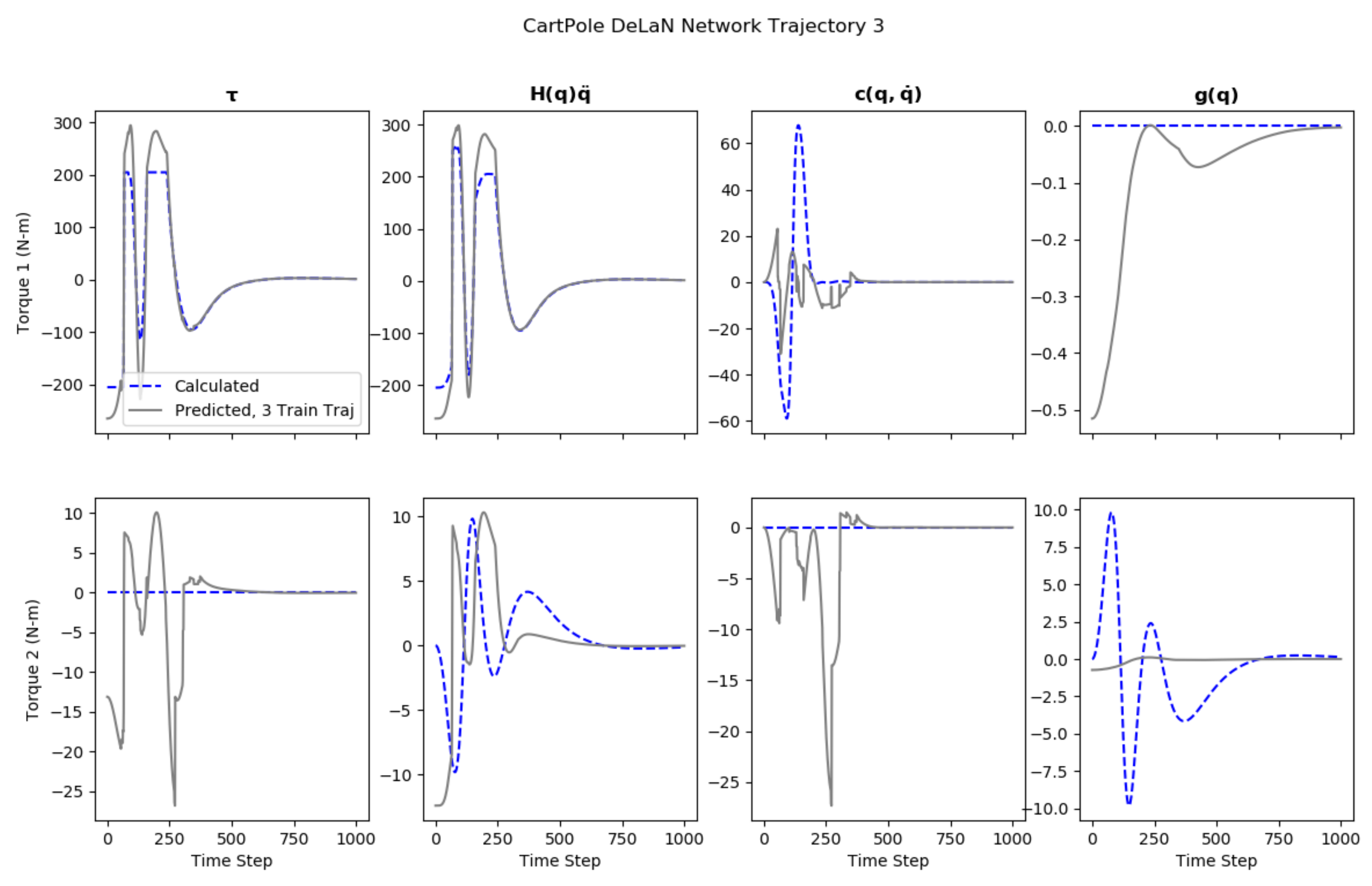
Reacher DeLaN predicted vs. calculated torques for character a DeLaN was trained on 1 sample of 1, 5 and 10 random characters excluding a for 200 epochs



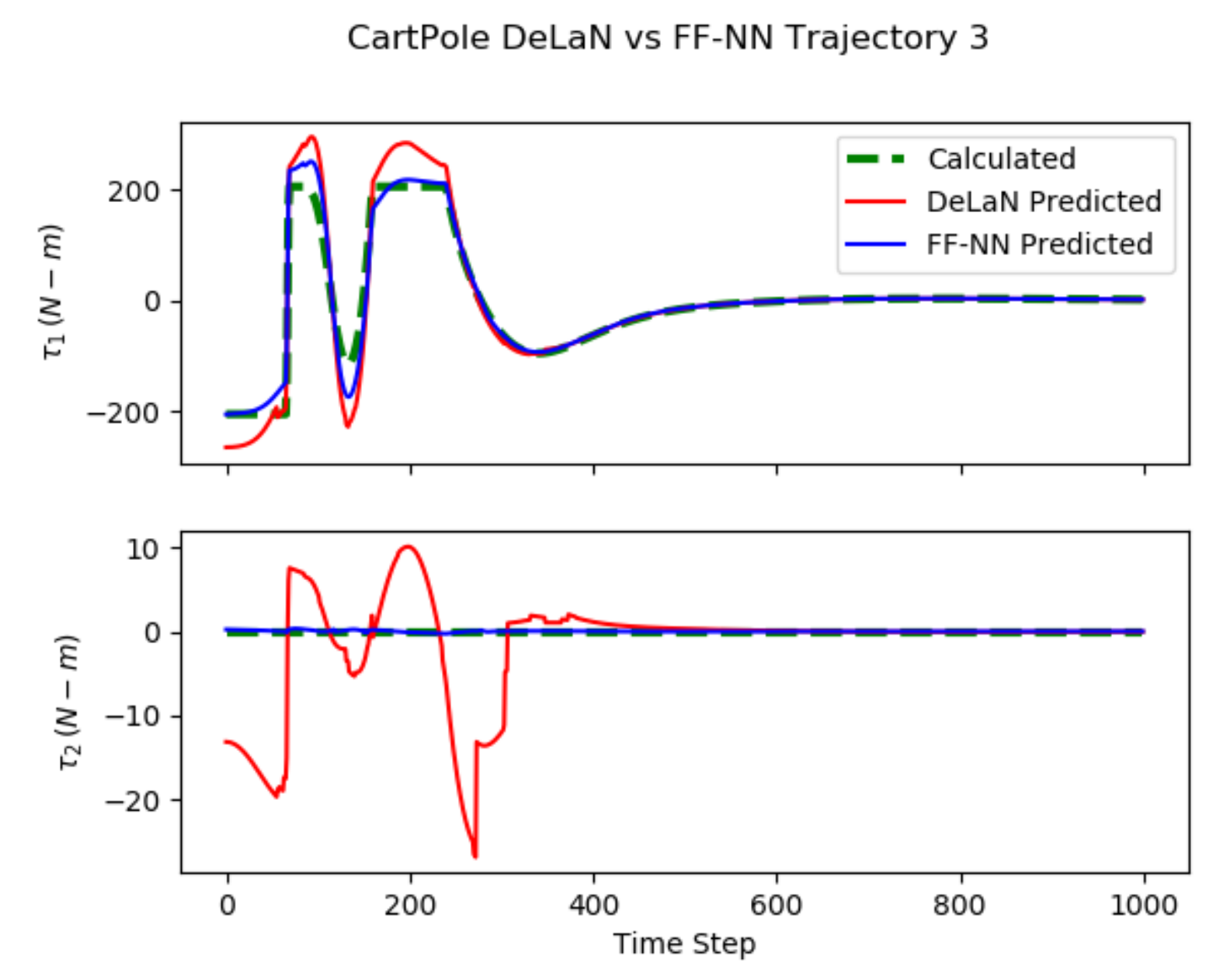
CartPole DeLaN and a FF-NN predicted vs. calculated torques for the revolution trajectory. Trained on 5 samples of the each of the other 3 trajectory types excluding revolution for 200 epochs



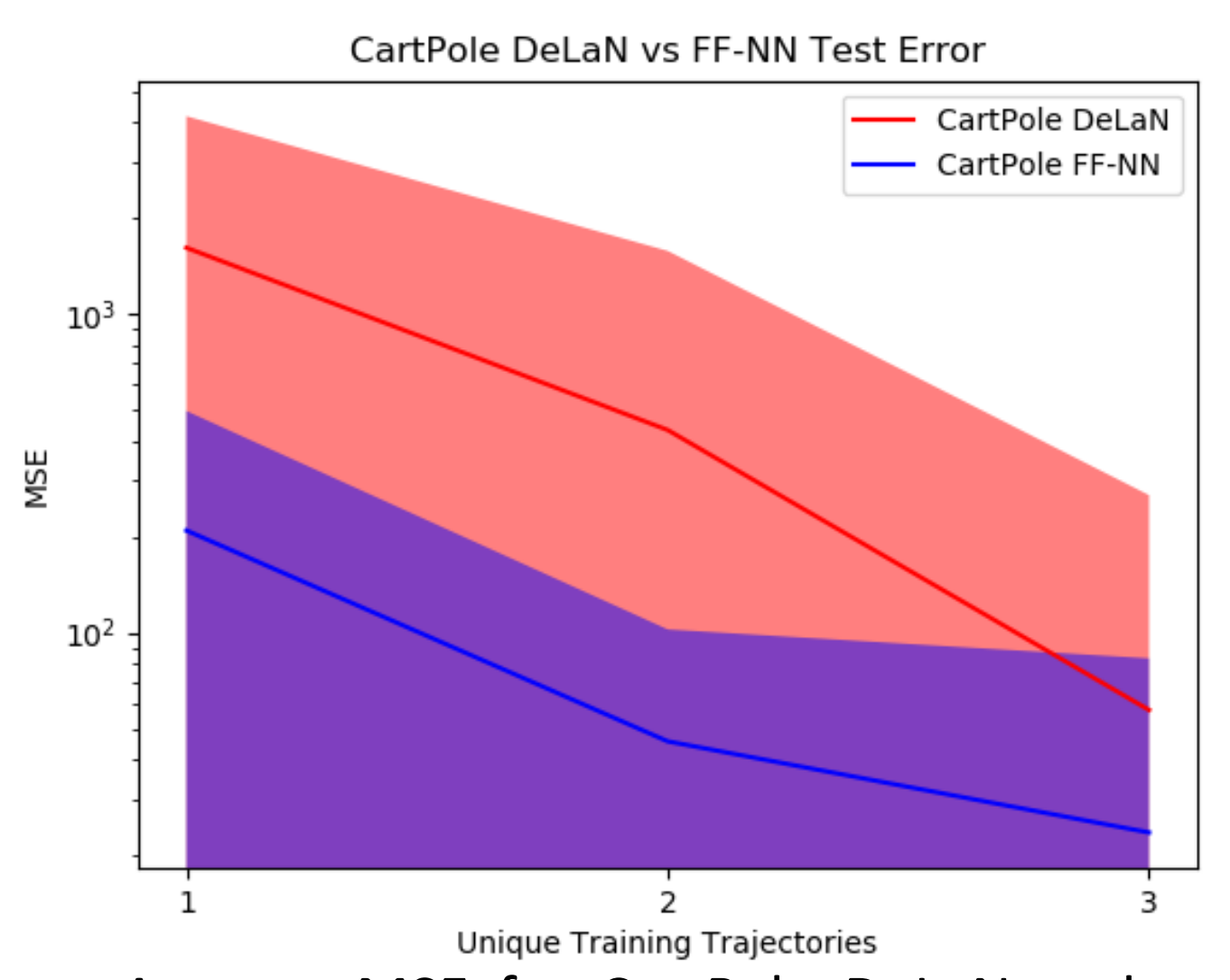
Average MSE for CartPole DeLaN and a Feed Forward Neural Network when trained on 5 samples of various numbers of unique trajectory types for 200 epochs averaged out over 10 different seeds. The shaded region is the 95% confidence interval.



CartPole DeLaN predicted vs. calculated torques for the revolution trajectory. DeLaN was trained 5 samples of the each of the other 3 trajectory types excluding revolution for 200 epochs



CartPole DeLaN and a FF-NN predicted vs. calculated torques for the revolution trajectory. Both networks were trained on 5 samples of the each of the other 3 trajectory types excluding revolution for 200 epochs.



Average MSE for CartPole DeLaN and a Feed Forward Neural Network when trained on 5 samples of various numbers of unique trajectory types for 200 epochs averaged out over 10 different seeds. The shaded region is the 95% confidence interval

Train Traj.	Test Traj.	DeLaN		DeLaN ($\tau_2 := 0$)		FF-NN	
		μ	σ	μ	σ	μ	σ
2,3,4	1	0.18	0.04	0.06	0.04	1.1	1.3
1,3,4	2	85.8	58.3	93.5	63.4	55.3	13.4
1,2,4	3	299	130	589	458	118	6.5
1,2,3	4	0.40	0.18	0.23	0.13	6.2	2.8

MSE mean and standard deviation of Networks for Testing on a Novel Trajectory

Conclusions

In this paper we presented an adapted DeLaN. We were able to replicate the results of the reacher environment, but the data efficiency and generalization did not hold for the CartPole environment. One theory for this result is due to too few training trajectories types.

Possible Solutions:
Adapt the problem formulation to explicitly handle underactuated systems by removing the positive definite requirement of H

Final Reflection:
Our project team considers our effort to be a success as we implemented an existing work, replicated the results of the paper and applied the method to a novel and challenging dataset. to produce a forward model of the system in order to produce new trajectories instead of following a given trajectory.