
Domain Adaptation of Drag Reduction Policy to Partial Measurements

Anton Plaksin
Imperial College London
London, UK
aplaksin@imperial.ac.uk

Georgios Rigas
Imperial College London
London, UK
g.rigas@imperial.ac.uk

Abstract

Feedback control of fluid-based systems poses significant challenges due to their high-dimensional, nonlinear, and multiscale dynamics, which demand real-time, three-dimensional, multi-component measurements for sensing. While such measurements are feasible in digital simulations, they are often only partially accessible in the real world. In this paper, we propose a method to adapt feedback control policies obtained from full-state measurements to setups with only partial measurements. Our approach is demonstrated in a simulated environment by minimising the aerodynamic drag of a simplified road vehicle. Reinforcement learning algorithms can optimally solve this control task when trained on full-state measurements by placing sensors in the wake. However, in real-world applications, sensors are limited and typically only on the vehicle, providing only partial measurements. To address this, we propose to train a Domain Specific Feature Transfer (DSFT) map reconstructing the full measurements from the history of the partial measurements. By applying this map, we derive optimal policies based solely on partial data. Additionally, our method enables determination of the optimal history length and offers insights into the architecture of optimal control policies, facilitating their implementation in real-world environments with limited sensor information.

1 Introduction

Feedback control of fluid-based engineering systems offers significant potential to improve energy efficiency in key sectors such as energy and transport, driving progress toward net-zero emissions. Recent breakthroughs in machine learning have greatly advanced the modeling, optimization, and control of fluid flows [Duriez et al., 2016, Brunton et al., 2020, Rabault et al., 2020, Garnier et al., 2021, Sharma et al., 2023]. Despite these advancements, designing feedback control remains difficult due to the nonlinear, multiscale nature of fluid dynamics. Optimal control laws typically rely on a dense set of spatio-temporally resolved measurements—a requirement that is feasible in simulations but challenging to meet in real-world applications. In practice, sensor placement is often limited by technical constraints, resulting in only partial measurements being available. In this paper, we propose a framework to adapt feedback control policies, originally developed using full-state measurements, to realistic scenarios where only partial measurements are accessible due to real-world limitations.

We demonstrate our approach in a simulated environment aimed at reducing the aerodynamic drag of road vehicles [Sudin et al., 2014, Choi et al., 2014]. The simulations are conducted in two-dimensional direct numerical simulation (DNS) environments, where the flow dynamics are obtained by numerically solving the Navier-Stokes equations, and the vehicle is modeled as a simplified 2D bluff body with square geometry. Recent numerical studies [Rabault et al., 2019, Rabault and Kuhnle, 2019, Tang et al., 2020, Paris et al., 2021, Li and Zhang, 2022, Chen et al., 2023] have successfully employed Reinforcement Learning (RL) [Sutton and Barto, 2018] algorithms to derive feedback control policies in such environments, demonstrating that jet actuators can stabilize the unsteadiness

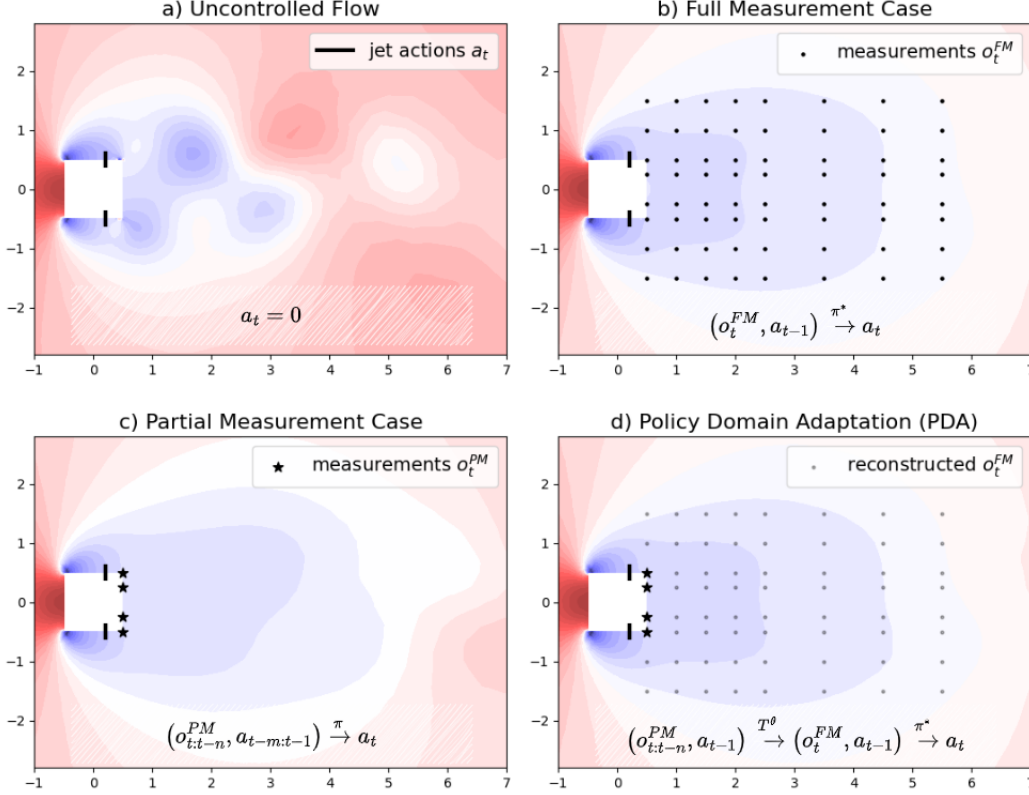


Figure 1: Flow pressure fields with the jet actuators and measurement locations for the following cases: uncontrolled flow (a); the flow stabilized by the policies trained by TQC algorithm using wake measurements (b) and the measurement history from the body base (c); the flow stabilized by PDA policy (2) with $n = 48$ and $m = 1$ (d).

in the wake behind the body when full-state measurements in the wake are available. However, in realistic scenarios, control policies rely on the history of measurements on the vehicle (partial measurements), which presents a more complex optimization challenge, as in Xia et al. [2024].

In our paper, we address the intermediate scenario where full measurements behind the vehicle are available during the training stage, but only partial measurements are accessible during deployment. To bridge this gap, we propose a supervised learning method of Policy Domain Adaptation (PDA), which leverages a Domain-Specific Feature Transfer (DSFT) map [Wei et al., 2018] to reconstruct full measurements from the history of partial measurements. This approach enables us to achieve the following results:

- Taking the composition of the trained DSFT map and the optimal policy for the full measurement case, we obtain the optimal policy for the partial measurement case (see Fig. 2a).
- Without training a new policy, we analyze the PDA performance depending on the measurement history length and find the minimum length giving the optimal result (see Fig. 2b).
- Our research provides insights into the architecture of optimal policies for the partial measurement case (see Discussion and Limitations).

2 Preliminaries and Baselines

The environment is a 2D direct numerical simulation (DNS) of a laminar flow past a square bluff body with $Re = 100$ (see Fig. 1a). The implementation is taken from Rabault et al. [2019], Rabault and Kuhnle [2019] and is based on FEniCS and the Dofin library Logg et al. [2012]. The flow dynamics is calculated as a solution $(v_t(\cdot), p_t(\cdot))$ of the Navier-Stokes equation, where $v_t(\cdot)$ is a velocity function and $p_t(\cdot)$ is a pressure function. The flow exhibits vortex shedding in the wake,

an instability which increases the drag. The body has two controllable jet actuators on the side edges which influence the vortex shedding. The task is to find a feedback policy that, taking certain measurements as input, reduces the drag of the body by stabilising the vortex shedding.

The discrete-time task can be considered within the RL paradigm [Sutton and Barto, 2018] as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where \mathcal{S} is the space of all possible states $s_t = (v_t(\cdot), p_t(\cdot), a_{t-1})$, \mathcal{A} is the space of two-dimensional actions $a_t \in [-a_*, a_*]^2$ defining a smooth change of the jet actuators' force, \mathcal{P} is a transition function giving the next state $s_{t+1} = (v_{t+1}(\cdot), p_{t+1}(\cdot), a_t)$ by the current state $s_t = (v_t(\cdot), p_t(\cdot), a_{t-1})$ and the current action a_t according to the Navier-Stokes equation, \mathcal{R} is a reward function giving the negative drag coefficient r_t , γ is a discount factor. Our aim is to find a policy maximizing the expectation of the total reward $\sum_{t=0}^{\infty} \gamma^t r_t$.

It is important to note that the state space of this task is infinite-dimensional, making a direct solution highly challenging. However, previous studies [Chen et al., 2023, Xia et al., 2024] have demonstrated that optimal performance can be achieved within the class of policies $\pi(o_t^{FM}, a_{t-1})$, where o_t^{FM} represents pressure measurements at specific discrete locations in the wake (see Fig. 1b). In this paper, we replicate these findings and use the optimal policy $\pi^*(o_t^{FM}, a_{t-1})$ as a baseline target for our approach (represented by the green line in Fig. 2a).

Although such policies solve the task optimally in simulations, learning them in real-world environments may be technically unfeasible due to the inability to access o_t^{FM} . A more realistic task is to find a policy that achieves optimal performance based only on pressure measurements obtained on the body base o_t^{PM} (see Fig. 1c). Such problem can be considered (see, e.g., Bertsekas [2012]) as a Partial Observed MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$, where Ω is a set of such measurements o_t^{PM} and \mathcal{O} is a function given o_t^{PM} by s_t . The paper of Xia et al. [2024] tackled this task by directly applying RL algorithms to find a policy $\pi(o_{t-n:t}^{PM}, a_{t-m:t-1})$, where $o_{t-n:t}^{PM}$ and $a_{t-m:t-1}$ denote the measurement history $(o_{t-n}^{PM}, o_{t-n+1}^{PM}, \dots, o_t^{PM})$ and the action history $(a_{t-m}, a_{t-m+1}, \dots, a_{t-1})$, respectively. We replicate this approach to establish a baseline policy, which we aim to match or surpass (see red line in Fig. 2a).

3 Policy Domain Adaptation

Similar to Xia et al. [2024], our goal is to identify the optimal policy $\pi^*(o_{t-n:t}^{PM}, a_{t-m:t-1})$, but we introduce an alternative approach. Specifically, we assume that the optimal policy $\pi^*(o_t^{FM}, a_{t-1})$ is already known—this could be obtained, for example, through simulations or using additional measurements that are only available during the training phase but need to be removed at deployment. Additionally, we assume that we have collected a trajectory dataset $D = \{o_{t,i}^{PM}, o_{t,i}^{FM}, a_{t,i}\}_{t \in \overline{0:T}, i \in \overline{1,k}}$, where t is a step number and i is a trajectory number. Following the DSFT approach [Wei et al., 2018], our aim is to establish a map T and select n, m such that

$$T(o_{t-n:t,i}^{PM}, a_{t-m:t-1,i}) \approx o_{t,i}^{FM}, \quad t \in \mathbb{T} = \{\max\{n, m\}, \dots, T\}, \quad i \in \mathbb{I} = \{1, \dots, k\}.$$

To achieve this, we approximate this map using a neural network T^θ , which is trained by minimising

$$Loss(\theta) = \mathbb{E}_{i \sim \mathbb{I}, t \sim \mathbb{T}} \|T^\theta(o_{t-n:t,i}^{PM}, a_{t-m:t-1,i}) - o_{t,i}^{FM}\|^2 \rightarrow \min_{\theta} \quad (1)$$

for any m and n without interacting with the environment. The result of our PDA is the policy

$$\pi^*(o_{t-n:t}^{PM}, a_{t-m:t-1}) = \pi^*(T^\theta(o_{t-n:t,i}^{PM}, a_{t-m:t-1,i}), a_{t-1}). \quad (2)$$

4 Experiments

In this section, we compare the performance of our PDA approach against the baseline cases and analyze its sensitivity to the lengths of the measurement and action histories. Before diving into the performance analysis, it is important to highlight the following key details that contribute to the improved performance of our approach:

- Following Xia et al. [2024], we apply the TQC algorithm Kuznetsov et al. [2020] to obtain an optimal policy $\pi^*(o_t^{FM}, a_{t-1})$, but we approximate it with a neural network without hidden layers, i.e. $\pi^{A,B}(o_t^{FM}, a_{t-1}) = a_* \tanh(Ao_t^{FM} + Ba_{t-1})$, where A and B are parameter matrices. We found this class of policies is sufficient to obtain optimal performance and more robust to approximations, improving the results of our PDA approach (Appendix A).

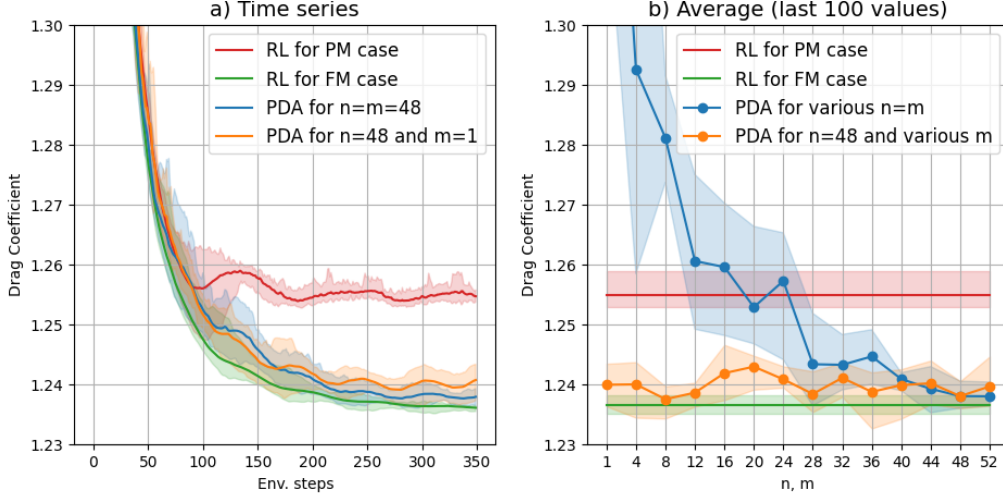


Figure 2: Time series and an average of the last 100 values of the Drag Coefficient (negative reward) for the four policies under consideration. Bold lines (left figure) and dots (right figure) indicate the average drag coefficient over 65 runs. Shaded regions illustrate the min-max areas.

- We use a neural network with one hidden layer for T^θ . We found a linear map performed worse, and the two hidden layer networks did not improve results (Appendix B).

Fig. 2a shows the drag coefficient over time for the trained policies we considered. The corresponding flow fields are shown in Fig. 1. All policies converge to a reduced drag state. However, the drag achieved by $\pi(o_{t-n:t}^{PM}, a_{t-m:t-1})$ trained by the RL algorithm directly (red line) is significantly suboptimal compared to the drag of the optimal policy $\pi^*(o_t^{FM}, a_{t-1})$ (green line) and of the policies obtained by our PDA approach (blue and green lines).

We also study the dependence of the drag on the measurement history length n and the action history length m in Fig. 2b. Our results indicate that, in general, increasing the measurement history length improves the performance of the PDA-derived policy (blue line), bringing it closer to the optimal value (green line), particularly when $n = m = 48$. Interestingly, the action history does not significantly impact the performance (orange line).

5 Discussion and Limitations

From the above results, the following valuable insights can be drawn for future studies:

- The studies of Mao et al. [2022], Xia et al. [2024] consider policies with action history which could indicate that the controlled process is associated with MDP with delay Katsikopoulos and Engelbrecht [2003], also discussed in Garnier et al. [2021]. However, our optimal policy without the action history (see the orange line in Fig. 2a) calls into question the necessity to incorporate delay in the formalization of similar drag reduction tasks.
- Recently, Partially Observed MDPs have been successfully solved using complex models of data sequence processing (see, e.g., Chen et al. [2021], Hafner et al. [2024]). Nonetheless, the use of such models for the problems under consideration seems redundant, taking into account the optimal policy we found (2), which is a network with only one hidden layer.
- We establish that the class of the simplest policies (without hidden layers) is not sufficient to obtain the optimal result, which challenges the effectiveness of linear controllers in partially observable environments.

The primary limitation of our PDA approach arises from the assumption that full measurements are available during the training stage, which is achievable in controlled settings such as digital twin environments or wind-tunnel laboratories. Further research could focus on assessing the robustness of sim-to-real or lab-to-real policy transfer, exploring the performance of policies trained in such environments when deployed in real-world conditions.

Acknowledgments and Disclosure of Funding

This research has been funded through the UKRI AI for Net Zero grant "Real-time digital optimisation and decision making for energy and transport systems" (EP/Y005619/1).

References

- D. Bertsekas. *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, Belmont, Massachusetts, 2012.
- S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.
- L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Proceedings of 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.
- W. Chen, Q. Wang, L. Yan, G. Hu, and B. R. Noack. Deep reinforcement learning-based active flow control of vortex-induced vibration of a square cylinder. *Physics of Fluids*, 35, 2023.
- H. Choi, J. Lee, and H. Park. Aerodynamics of heavy vehicles. *Annual Review of Fluid Mechanics*, 46:441–468, 2014.
- T. Duriez, S. L. Brunton, and B. R. Noack. *Machine Learning Control — Taming Nonlinear Dynamics and Turbulence*. Springer, 2016.
- P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem. A review on deep reinforcement learning for fluid mechanics: an update. *Computers and Fluids*, 225, 2021.
- D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *ArXiv:2301.04104*, 2024.
- K. V. Katsikopoulos and S. E. Engelbrecht. Markov decision processes with delays and asynchronous cost collection. *IEEE Transactions on Automatic Control*, 48(4):568–574, 2003.
- A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 119:5556–5566, 2020.
- J. Li and M. Zhang. Reinforcement-learning-based control of confined cylinder wakes with stability analyses. *Journal of Fluid Mechanics*, 932:A44, 2022.
- A. Logg, K.-A. Mardal, and G. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.
- Y. Mao, S. Zhong, and H. Yin. Active flow control using deep reinforcement learning with time-delays in markov decision process and autoregressive policy. *Physics of Fluids*, 34(5), 2022.
- R. Paris, S. Beneddine, and J. Dandois. Robust flow control and optimal sensor placement using deep reinforcement learning. *Journal of Fluid Mechanics*, 913:A25, 2021.
- J. Rabault and A. Kuhnle. Accelerating deep reinforcement learning strategies of flow control through a multi- environment approach. *Physics of Fluids*, 31(9), 2019.
- J. Rabault, M. Kuchta, A. Jensen, U. Reglade, and N. Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019.
- J. Rabault, F. Ren, W. Zhang, H. Tang, and H. Xu. Deep reinforcement learning in fluid mechanics: A promising method for both active flow control and shape optimization. *Journal of Hydrodynamics*, 32:234–246, 2020.
- A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL <http://jmlr.org/papers/v22/20-1364.html>.

- P. Sharma, W. T. Chung, B. Akoush, and M. Ihme. A review of physics-informed machine learning in fluid mechanics. *Energies*, 16(5), 2023.
- M. N. Sudin, M. A. Abdullah, S. A. Shamsuddin, F. R. Ramli, and M. M. Tahir. Review of research on vehicles aerodynamic drag reduction methods. *International Journal of Mechanical and Mechatronics Engineering*, 14(2):37–47, 2014.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning An Introduction (Second Edition)*. The MIT Press, Cambridge, Massachusetts, 2018.
- H. Tang, J. Rabault, A. Kuhnle, Y. Wang, and T. Wang. Robust active flow control over a range of reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Physics of Fluids*, 32(5), 2020.
- P. Wei, Y. Ke, and C. K. Goh. A general domain specific feature transfer framework for hybrid domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1440–1451, 2018.
- C. Xia, J. Zhang, E. C. Kerrigan, and G. Rigas. Active flow control for bluff body drag reduction using reinforcement learning with partial measurements. *Journal of Fluid Mechanics*, 981:A17, 2024.

A Appendix

In our research, we use TQC algorithm Kuznetsov et al. [2020] from Stable Baselines3 Raffin et al. [2021] with hyperparameters mostly taken from Xia et al. [2024]:

Parameters	TQC
environment steps	$\approx 3.5e6$
parallel environments	65
learning rate	1e-4
batch size	128
smooth param. τ	5e-3
gradient steps	40
discount factor γ	0.99
Q -model	[512,512,512]
π -model	[] or [512,512,512]

We consider two types of neural networks as a policy model:

- π^* is an optimal policy without hidden layers;
- π_+^* is an optimal policy with three hidden layer with 512 neurons as in Xia et al. [2024].

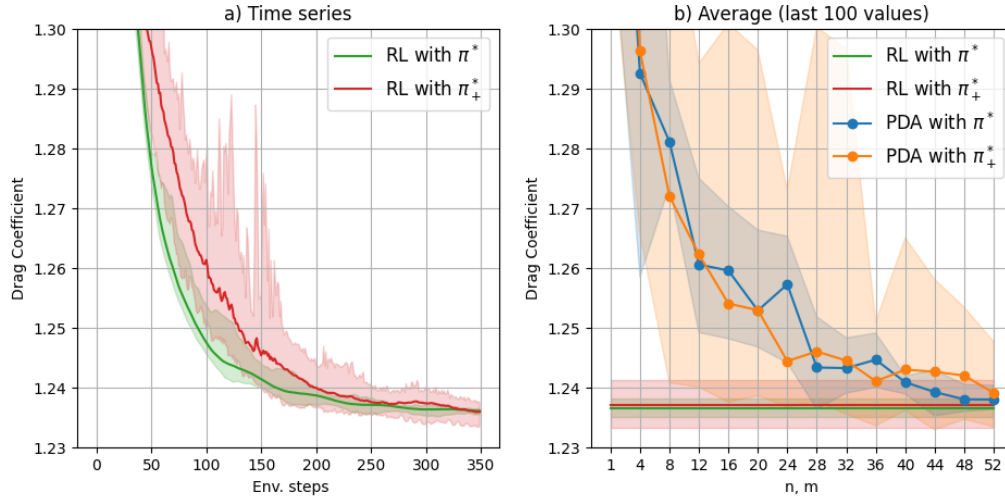


Figure 3: Time series and average of the last 100 values of the Drag Coefficient (negative reward) for the optimal policy without hidden layers π^* and the optimal policy with three hidden layers π_+^* . Bold lines (left figure) and dots (right figure) indicate the average drag coefficient over 65 runs.

We found both policies are capable of producing optimal performance, but the policy π^* appears to be more robust to input imprecision because its results (see Fig. 3a) and results of composition (2) with this policy (see Fig. 3b) are more stable with respect to runnings.

B Appendix

We solve optimization problem (1) with following hyperparameters:

Parameters	PDA
number of epochs	10000
optimization method	Adam
learning rate	1e-3
batch size	10000
dataset normalization	z-scale
T -model	[] or [128] or [64,64]

We consider tree type of neural networks to approximate the map T :

- the model T_-^θ without hidden layers (linear);
- the model T^θ with one hidden layer with 128 neurons;
- the model T_+^θ with two hidden layer with 64 neurons;

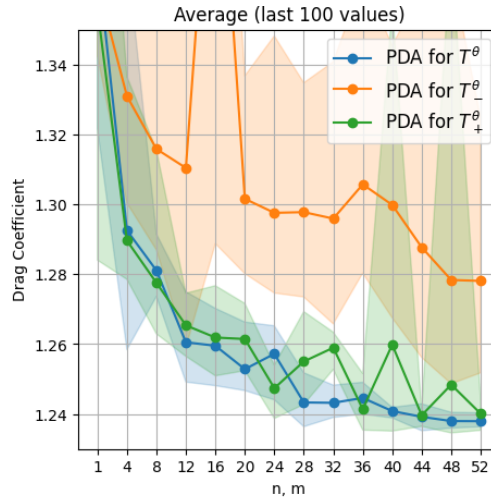


Figure 4: Average of the last 100 values of the Drag Coefficient for the policies obtained by PDA approach as composition (2), where the policy π^* is fixed, but the map T^θ is approximated by the models T_-^θ , T^θ , and T_+^θ . Dots indicate the average drag coefficient over 65 runs.

We found (see Fig. 4) the linear map T_-^θ gets results that are quite far from the optimal, while deeper models T^θ and T_+^θ achieve optimal performance and generally show similar results. Nonetheless, note that the model T^θ is more stable than T_+^θ over runnings, which may be due to overfitting of the deeper model T_+^θ on the training data and consequently reducing its generalization ability.