Analyses of Policy Gradient for Language Model Finetuning and Optimal Control

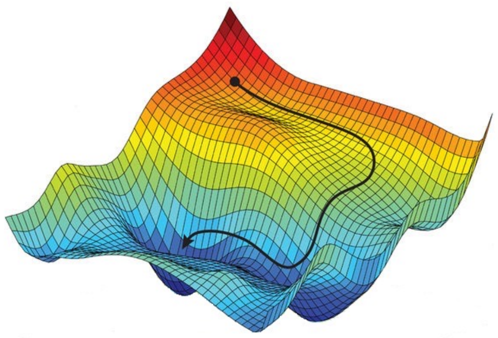
Noam Razin
Tel Aviv University

MML Seminar MPI MIS + UCLA 7 March 2024

Optimization and Generalization in Modern Machine Learning

Optimization and Generalization in Modern Machine Learning

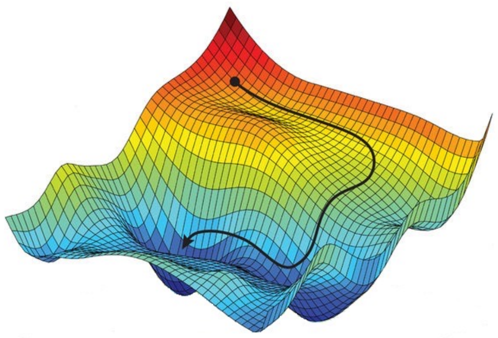
Optimization



Minimize a **non-convex** training objective

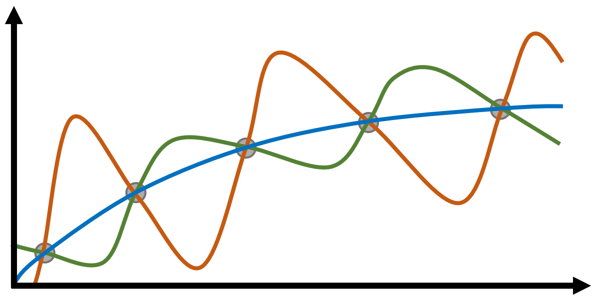
Optimization and Generalization in Modern Machine Learning

Optimization



Minimize a **non-convex** training objective

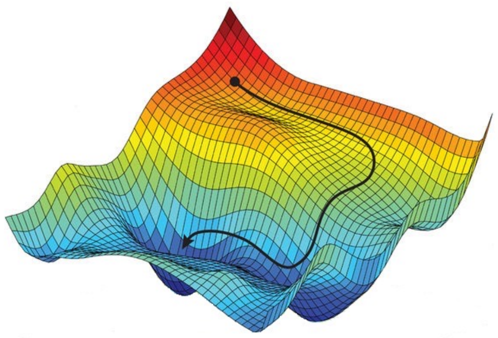
Generalization



Performance on **data unseen in training**

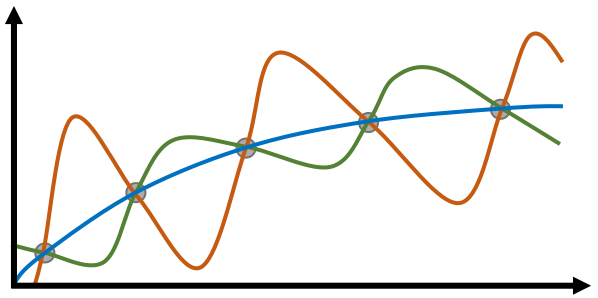
Optimization and Generalization in Modern Machine Learning

Optimization



Minimize a **non-convex** training objective

Generalization

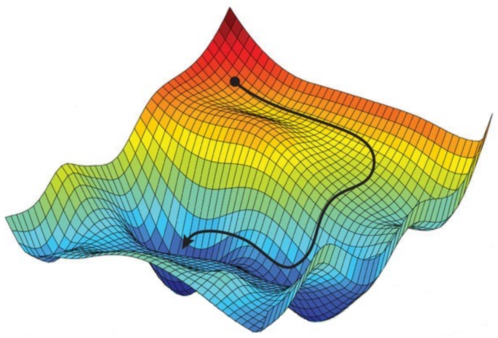


Performance on **data unseen in training**

Determined by **implicit bias** of training algorithm

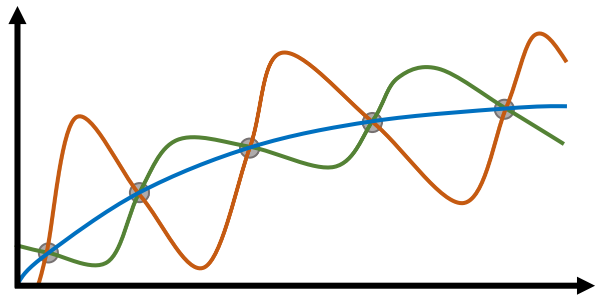
Optimization and Generalization in Modern Machine Learning

Optimization



Minimize a **non-convex** training objective

Generalization



Performance on **data unseen in training**

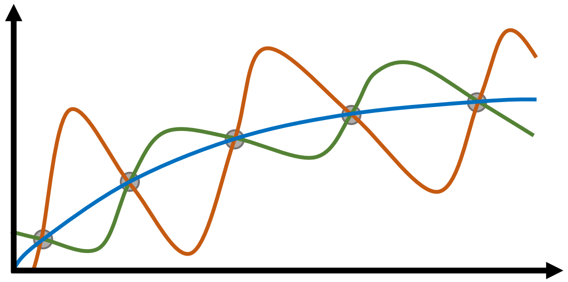
Determined by **implicit bias** of training algorithm

Gradient-based methods are the workhorse behind
optimization and generalization in modern machine learning

Supervised Learning vs Optimal Control/Reinforcement Learning

Supervised Learning vs Optimal Control/Reinforcement Learning

Supervised Learning

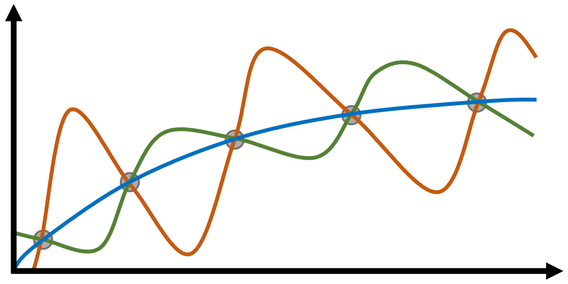


Task: Learn predictor minimizing loss over **labeled data**

Training Algorithm: Gradient descent

Supervised Learning vs Optimal Control/Reinforcement Learning

Supervised Learning



Task: Learn predictor minimizing loss over labeled data

Training Algorithm: Gradient descent

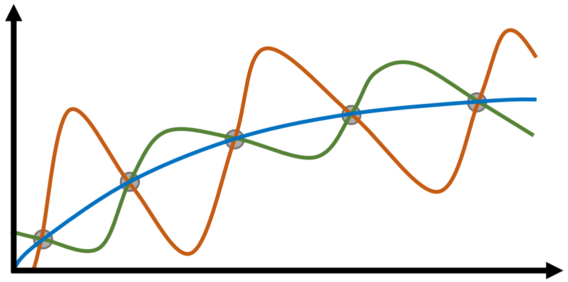
Optimization Dynamics and Implicit Bias

Extensively Studied

(e.g., Neyshabur et al. 2014, Gunasekar et al. 2017, Soudry et al. 2018, Arora et al. 2019, Ji & Telgarsky 2019; **R** et al. 2020/21/22, Pesme et al. 2021, Lyu et al. 2021, Boursier et al. 2022, Andriushchenko et al. 2023, Frei et al. 2023, Jin & Montúfar 2023, Abbe et al. 2023)

Supervised Learning vs Optimal Control/Reinforcement Learning

Supervised Learning



Task: Learn predictor minimizing loss over **labeled data**

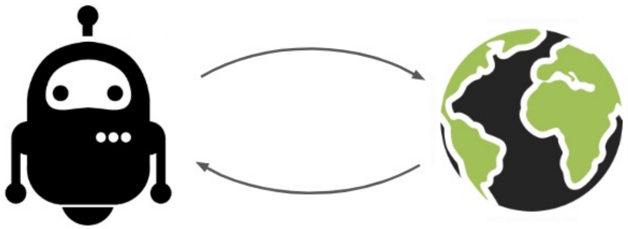
Training Algorithm: Gradient descent

Optimization Dynamics and Implicit Bias

Extensively Studied

(e.g., Neyshabur et al. 2014, Gunasekar et al. 2017, Soudry et al. 2018, Arora et al. 2019, Ji & Telgarsky 2019; **R** et al. 2020/21/22, Pesme et al. 2021, Lyu et al. 2021, Boursier et al. 2022, Andriushchenko et al. 2023, Frei et al. 2023, Jin & Montúfar 2023, Abbe et al. 2023)

Optimal Control/Reinforcement Learning

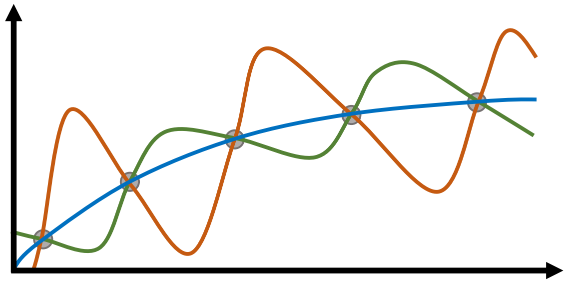


Task: Learn policy minimizing cost/maximizing reward over **dynamical system**

Training Algorithm: Policy gradient

Supervised Learning vs Optimal Control/Reinforcement Learning

Supervised Learning



Task: Learn predictor minimizing loss over **labeled data**

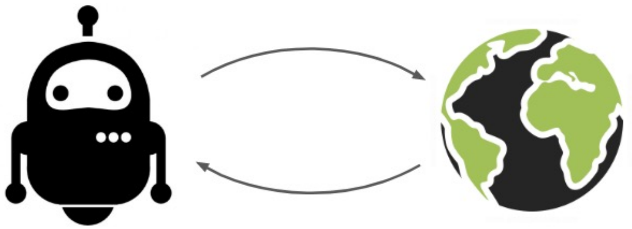
Training Algorithm: Gradient descent

Optimization Dynamics and Implicit Bias

Extensively Studied

(e.g., Neyshabur et al. 2014, Gunasekar et al. 2017, Soudry et al. 2018, Arora et al. 2019, Ji & Telgarsky 2019; **R** et al. 2020/21/22, Pesme et al. 2021, Lyu et al. 2021, Boursier et al. 2022, Andriushchenko et al. 2023, Frei et al. 2023, Jin & Montúfar 2023, Abbe et al. 2023)

Optimal Control/Reinforcement Learning



Task: Learn policy minimizing cost/maximizing reward over **dynamical system**

Training Algorithm: Policy gradient

Optimization Dynamics and Implicit Bias

Limited Understanding

(Fazel et al. 2018, Mei et al. 2020, Hu et al. 2021)

Sources

Optimization

Vanishing Gradients in Reinforcement Finetuning of Language Models

R + Zhou + Saremi + Thilak + Bradley + Nakkiran + Susskind + Littwin | *ICLR 2024*

Implicit Bias

Implicit Bias of Policy Gradient in Linear Quadratic Control: Extrapolation to Unseen Initial States

R + Alexander + Cohen-Karlik + Giryes + Globerson + Cohen | *arXiv 2024*

Vanishing Gradients in Reinforcement Finetuning of Language Models

R + Zhou + Saremi + Thilak + Bradley + Nakkiran + Susskind + Littwin | *ICLR 2024*

Language Models (LMs)

Language Models (LMs)

Language Model (LM): Neural network trained on large amounts of text data to produce a **distribution over text**

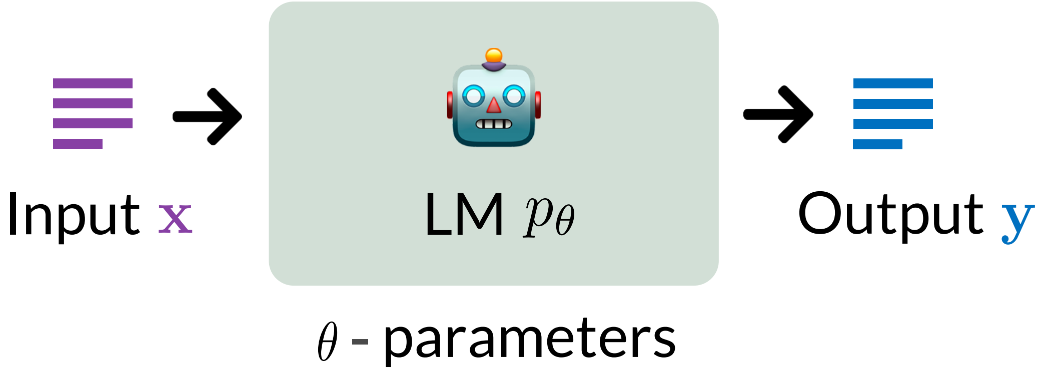
🤖

LM p_θ

θ - parameters

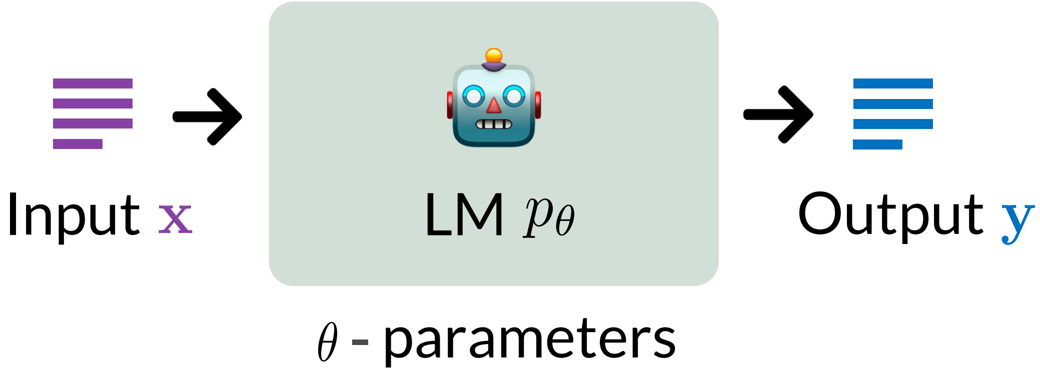
Language Models (LMs)

Language Model (LM): Neural network trained on large amounts of text data to produce a **distribution over text**



Language Models (LMs)

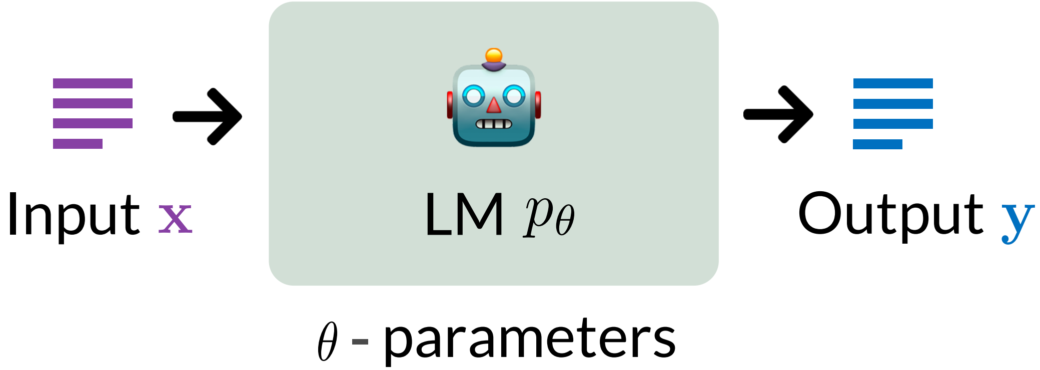
Language Model (LM): Neural network trained on large amounts of text data to produce a **distribution over text**



LMs are typically autoregressive

Language Models (LMs)

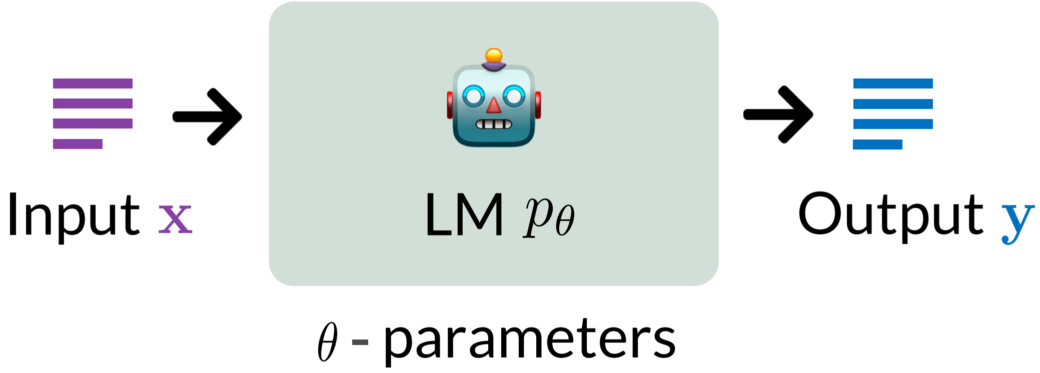
Language Model (LM): Neural network trained on large amounts of text data to produce a **distribution over text**



LMs are typically autoregressive $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{l=1}^{L} p_\theta(y_l|\mathbf{x}, \mathbf{y}_{\leq l-1})$

Language Models (LMs)

Language Model (LM): Neural network trained on large amounts of text data to produce a **distribution over text**



LMs are typically autoregressive $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{l=1}^{L} \underbrace{p_\theta(y_l|\mathbf{x}, \mathbf{y}_{\leq l-1})}$

softmax is used for producing next-token probabilities

Supervised Finetuning of LMs

LMs are adapted to human preferences and downstream tasks via **finetuning**

Supervised Finetuning of LMs

LMs are adapted to human preferences and downstream tasks via **finetuning**

Supervised Finetuning (SFT)

Minimize cross entropy loss over labeled inputs via **gradient-based methods**

Supervised Finetuning of LMs

LMs are adapted to human preferences and downstream tasks via **finetuning**

Supervised Finetuning (SFT)

Minimize cross entropy loss over labeled inputs via **gradient-based methods**

Limitations:

Supervised Finetuning of LMs

LMs are adapted to human preferences and downstream tasks via **finetuning**

Supervised Finetuning (SFT)

Minimize cross entropy loss over labeled inputs via **gradient-based methods**

Limitations:

Hard to formalize human preferences through labels

Supervised Finetuning of LMs

LMs are adapted to human preferences and downstream tasks via **finetuning**

Supervised Finetuning (SFT)

Minimize cross entropy loss over labeled inputs via **gradient-based methods**

Limitations:

- Hard to formalize human preferences through labels
- Labeled data is expensive

Reinforcement Finetuning of LMs

Limitations of SFT led to wide adoption of a **reinforcement learning**-based approach

(e.g. Ziegler et al. 2019, Stiennon et al. 2020, Ouyang et al. 2022, Bai et al. 2022, Dubois et al. 2023, Touvron et al. 2023)

Reinforcement Finetuning of LMs

Limitations of SFT led to wide adoption of a **reinforcement learning**-based approach

(e.g. Ziegler et al. 2019, Stiennon et al. 2020, Ouyang et al. 2022, Bai et al. 2022, Dubois et al. 2023, Touvron et al. 2023)

Reinforcement Finetuning (RFT)

Maximize reward over unlabeled inputs via **policy gradient algorithms**

reward function $r(\mathbf{x}, \mathbf{y})$

Reinforcement Finetuning of LMs

Limitations of SFT led to wide adoption of a **reinforcement learning**-based approach

(e.g. Ziegler et al. 2019, Stiennon et al. 2020, Ouyang et al. 2022, Bai et al. 2022, Dubois et al. 2023, Touvron et al. 2023)

Reinforcement Finetuning (RFT)

Maximize reward over unlabeled inputs via **policy gradient algorithms**

reward function $r(\mathbf{x}, \mathbf{y})$

Expected reward for input $\mathbf{x}$: $V_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{y} \sim p_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x}, \mathbf{y})\right]$

Reinforcement Finetuning of LMs

Limitations of SFT led to wide adoption of a **reinforcement learning**-based approach
(e.g. Ziegler et al. 2019, Stiennon et al. 2020, Ouyang et al. 2022, Bai et al. 2022, Dubois et al. 2023, Touvron et al. 2023)

Reinforcement Finetuning (RFT)

Maximize reward over unlabeled inputs via **policy gradient algorithms**

reward function $r(\mathbf{x}, \mathbf{y})$

Expected reward for input $\mathbf{x}$: $V_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{y} \sim p_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x}, \mathbf{y})\right]$

Reward function $r(\mathbf{x}, \mathbf{y})$ can be:

Reinforcement Finetuning of LMs

Limitations of SFT led to wide adoption of a **reinforcement learning**-based approach
(e.g. Ziegler et al. 2019, Stiennon et al. 2020, Ouyang et al. 2022, Bai et al. 2022, Dubois et al. 2023, Touvron et al. 2023)

Reinforcement Finetuning (RFT)

Maximize reward over unlabeled inputs via **policy gradient algorithms**

reward function $r(\mathbf{x}, \mathbf{y})$

Expected reward for input $\mathbf{x}$: $V_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{y} \sim p_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x}, \mathbf{y})\right]$

Reward function $r(\mathbf{x}, \mathbf{y})$ can be:

- Learned from human preferences

Reinforcement Finetuning of LMs

Limitations of SFT led to wide adoption of a **reinforcement learning**-based approach
(e.g. Ziegler et al. 2019, Stiennon et al. 2020, Ouyang et al. 2022, Bai et al. 2022, Dubois et al. 2023, Touvron et al. 2023)

Reinforcement Finetuning (RFT)

Maximize reward over unlabeled inputs via **policy gradient algorithms**

, , $\cdots$, reward function $r(\mathbf{x}, \mathbf{y})$

Expected reward for input $\mathbf{x}$: $V_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{y} \sim p_\theta(\cdot|\mathbf{x})} \left[r(\mathbf{x}, \mathbf{y}) \right]$

Reward function $r(\mathbf{x}, \mathbf{y})$ can be:

- Learned from human preferences
- Tailored to a downstream task

Main Contributions: Vanishing Gradients in RFT

Main Contributions: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ Fundamental vanishing gradients problem in RFT

Main Contributions: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Main Contributions: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Exploring ways to overcome vanishing gradients in RFT

Main Contributions: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Exploring ways to overcome vanishing gradients in RFT

Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

Theorem

$$\|\nabla_\theta V_\theta(\mathbf{x})\| = O\big(\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\big)$$

*Same holds for PPO gradient

Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

Theorem

$$\|\nabla_\theta V_\theta(\mathbf{x})\| = O\Big(\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\Big)$$

Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal

*Same holds for PPO gradient

Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

Theorem

$$\|\nabla_\theta V_\theta(\mathbf{x})\| = O\left(\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\right)$$

Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal

*Same holds for PPO gradient

Proof Idea: Stems from use of softmax + reward maximization objective

Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

Theorem

$$\|\nabla_\theta V_\theta(\mathbf{x})\| = O\big(\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\big)$$

Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal

*Same holds for PPO gradient

Proof Idea: Stems from use of softmax + reward maximization objective

Note: Bound applies to expected gradients of individual inputs (as opposed to of batch/population)

Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

Theorem

$$\|\nabla_\theta V_\theta(\mathbf{x})\| = O\left(\mathrm{STD}_{\mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\right)$$

Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal

*Same holds for PPO gradient

Proof Idea: Stems from use of softmax + reward maximization objective

Note: Bound applies to expected gradients of individual inputs (as opposed to of batch/population)

Can be problematic when finetuning text distribution differs from pretraining

Main Contributions: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Exploring ways to overcome vanishing gradients in RFT

Prevalence and Detrimental Effects of Vanishing Gradients

Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding I

3 of 7 datasets contain considerable # of train inputs with small reward std and low reward

Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding I

3 of 7 datasets contain considerable # of train inputs with $\overbrace{\text{small reward std}}^{\text{vanishing gradients}}$ and low reward

Prevalence and Detrimental Effects of Vanishing Gradients

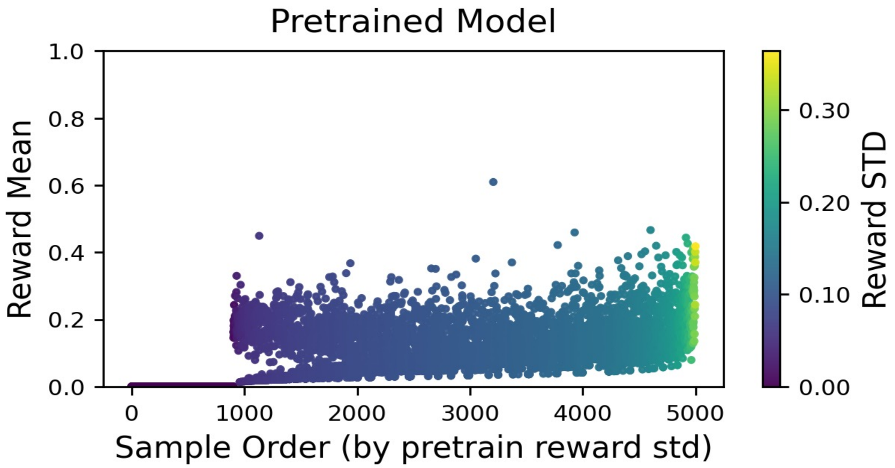
Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding I

3 of 7 datasets contain considerable # of train inputs with $\overbrace{\text{small reward std}}^{\text{vanishing gradients}}$ and low reward

NarrativeQA
(many inputs w/ small std)



Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

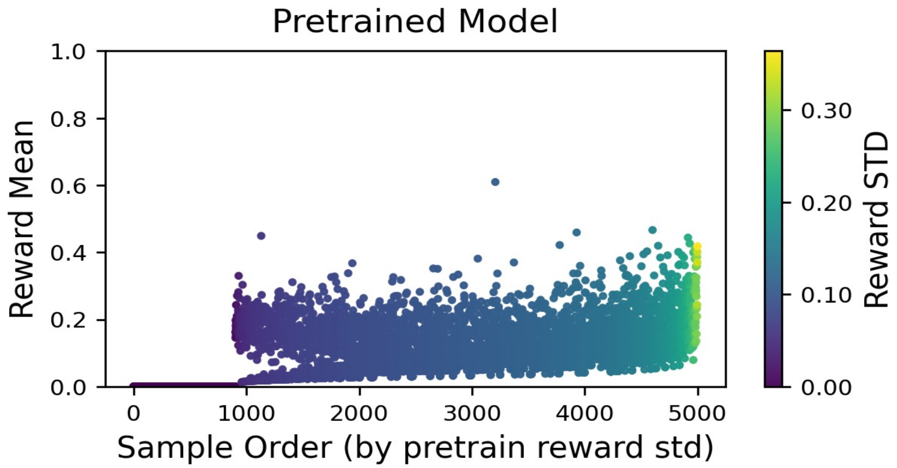
Models: GPT-2 and T5-base

vanishing gradients

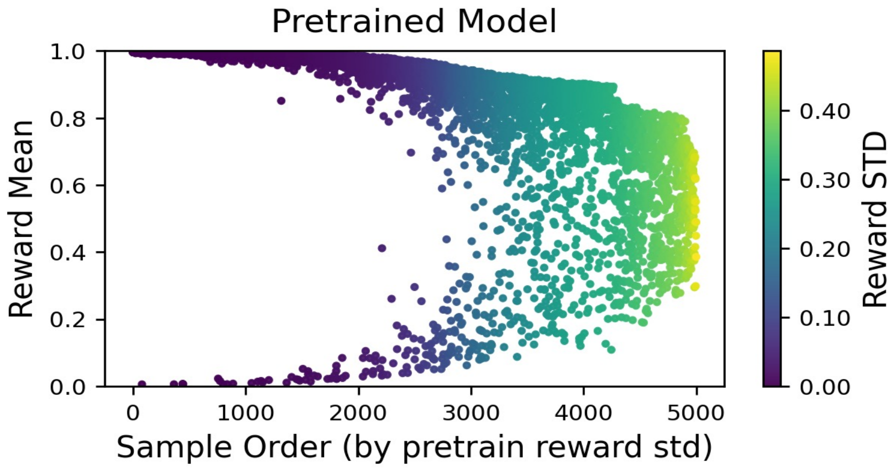
Finding I

3 of 7 datasets contain considerable # of train inputs with small reward std and low reward

NarrativeQA
(many inputs w/ small std)



IMDB
(few inputs w/ small std)



Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding II

As expected, RFT has limited impact on the reward of inputs with small reward std

Prevalence and Detrimental Effects of Vanishing Gradients

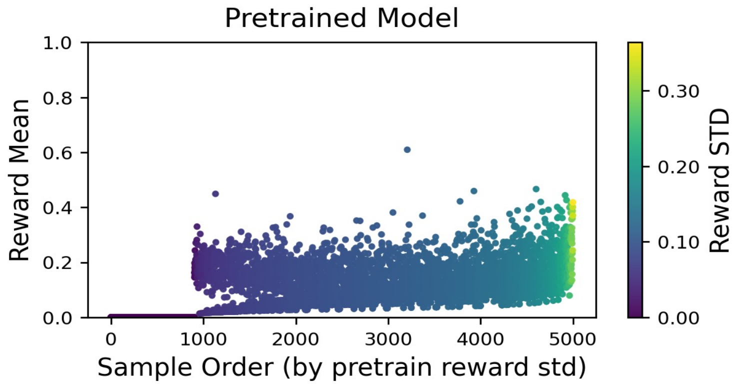
Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

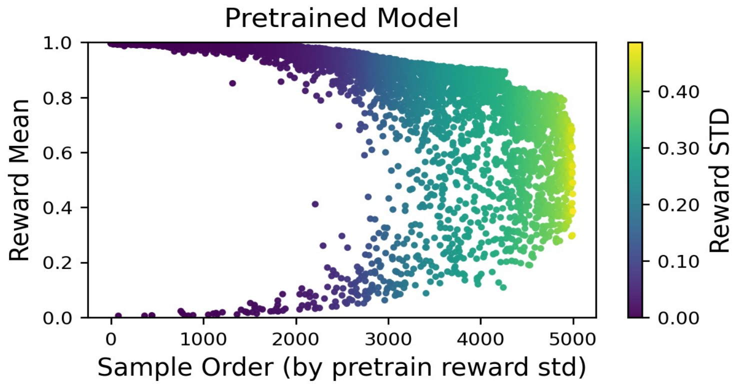
Finding II

As expected, RFT has limited impact on the reward of inputs with small reward std

NarrativeQA
(many inputs w/ small std)



IMDB
(few inputs w/ small std)



Prevalence and Detrimental Effects of Vanishing Gradients

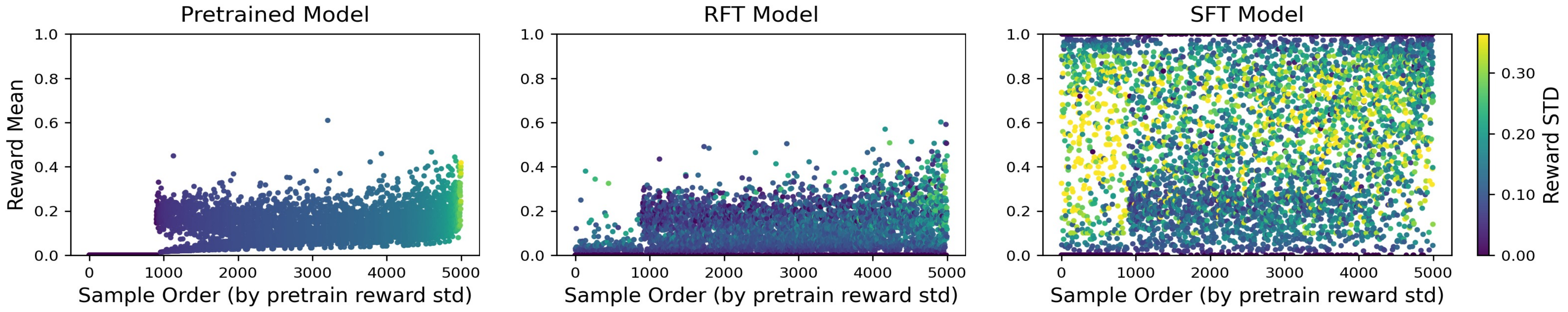
Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

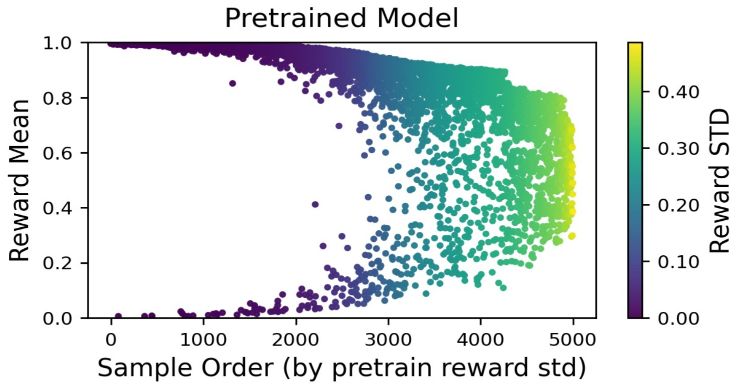
Finding II

As expected, RFT has limited impact on the reward of inputs with small reward std

NarrativeQA
(many inputs w/ small std)



IMDB
(few inputs w/ small std)



Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

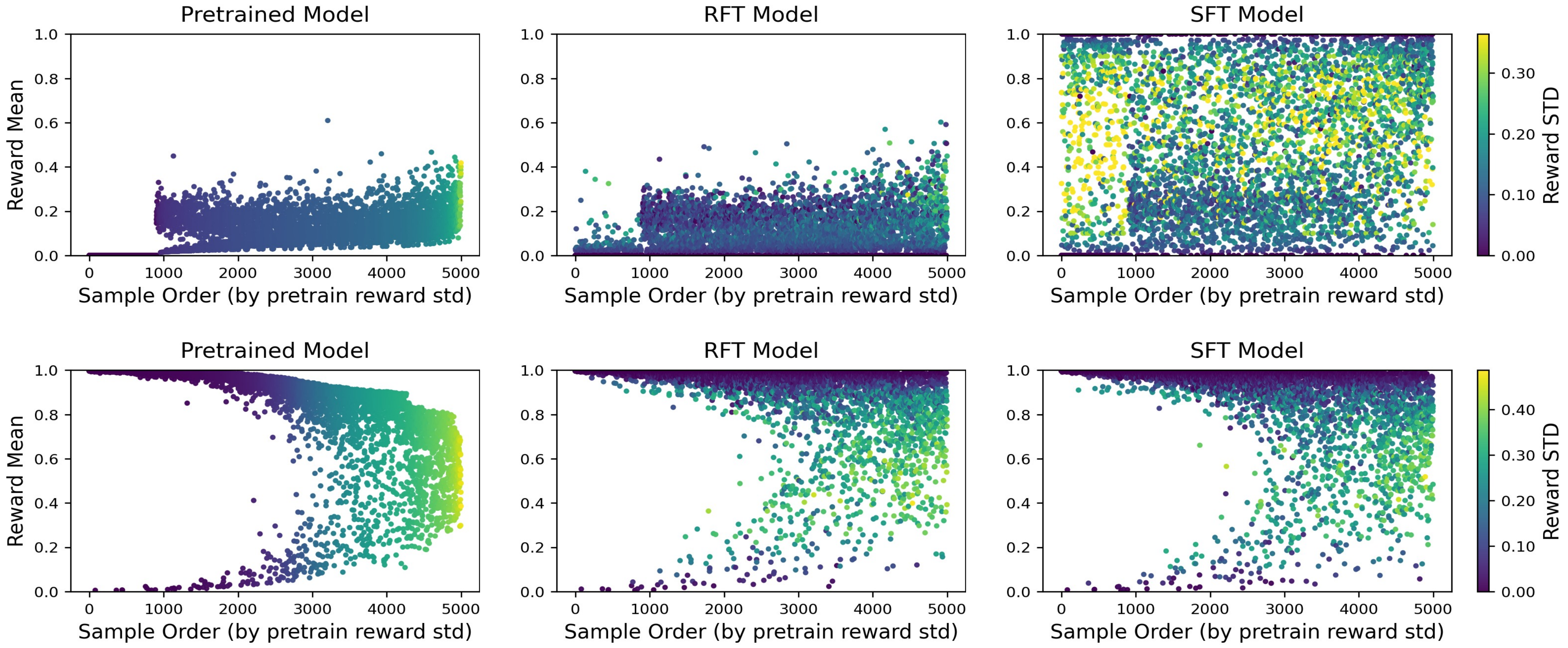
Models: GPT-2 and T5-base

Finding II

As expected, RFT has limited impact on the reward of inputs with small reward std

NarrativeQA
(many inputs w/ small std)

IMDB
(few inputs w/ small std)



Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding III

Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding III

RFT performance is worse when inputs with small reward std are prevalent

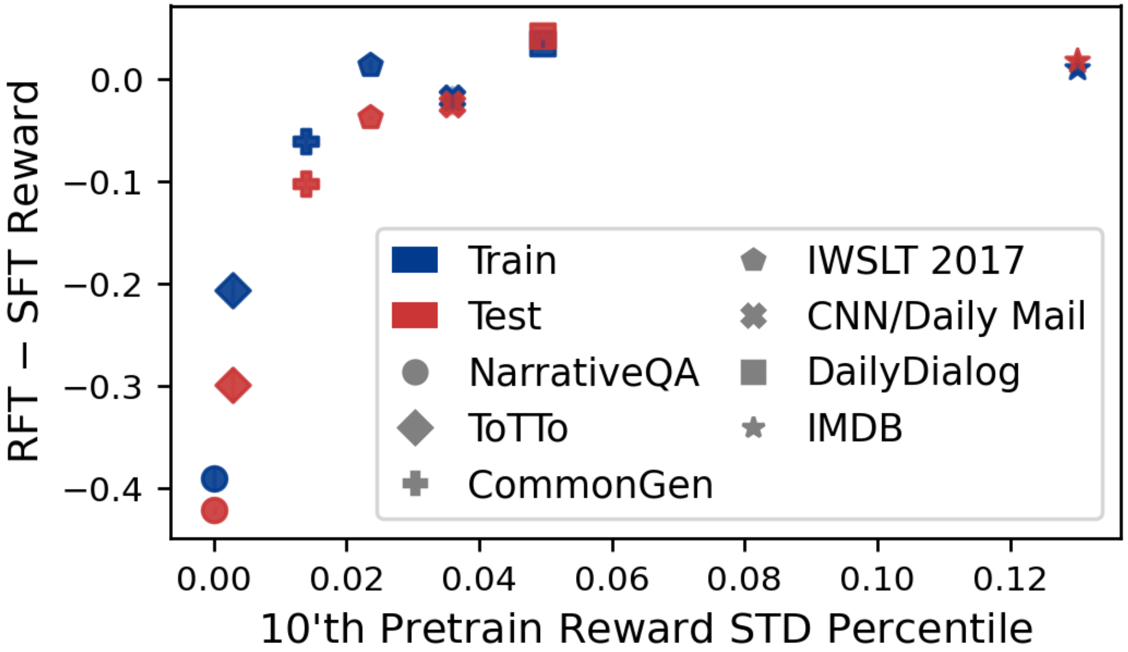
Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

Finding III

RFT performance is worse when inputs with small reward std are prevalent



Main Contributions: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Exploring ways to overcome vanishing gradients in RFT

Overcoming Vanishing Gradients in RFT

Overcoming Vanishing Gradients in RFT

Common Heuristics: Increasing learning rate, temperature, entropy regularization

Overcoming Vanishing Gradients in RFT

Common Heuristics: Increasing learning rate, temperature, entropy regularization ❌

Overcoming Vanishing Gradients in RFT

Common Heuristics: Increasing learning rate, temperature, entropy regularization ❌

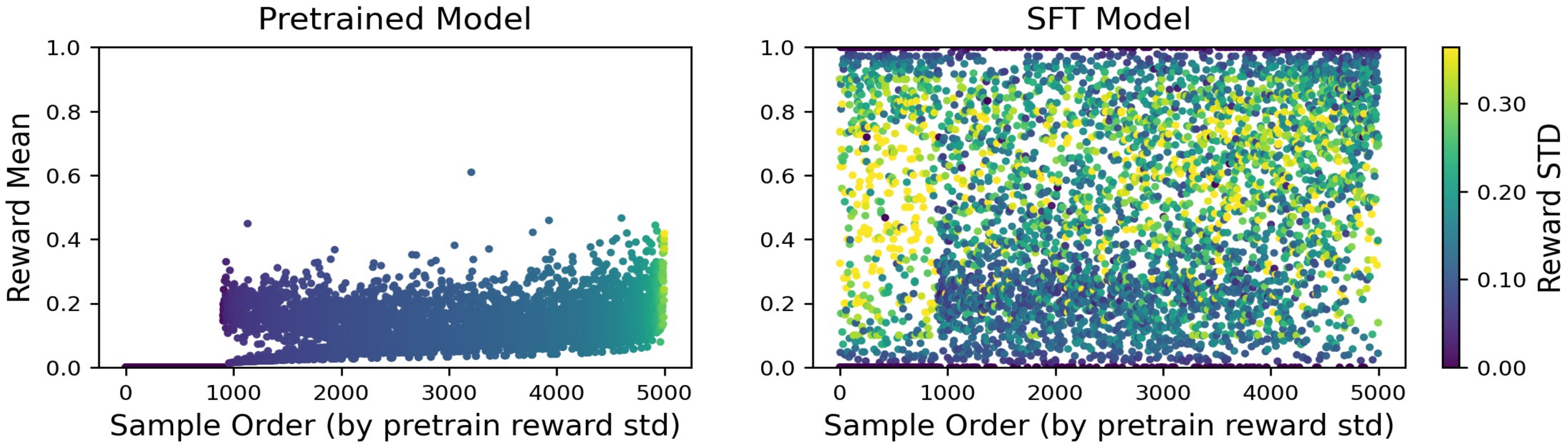
Observation: Initial SFT phase reduces number of inputs with small reward std

Overcoming Vanishing Gradients in RFT

Common Heuristics: Increasing learning rate, temperature, entropy regularization ❌

Observation: Initial SFT phase reduces number of inputs with small reward std

NarrativeQA (train)

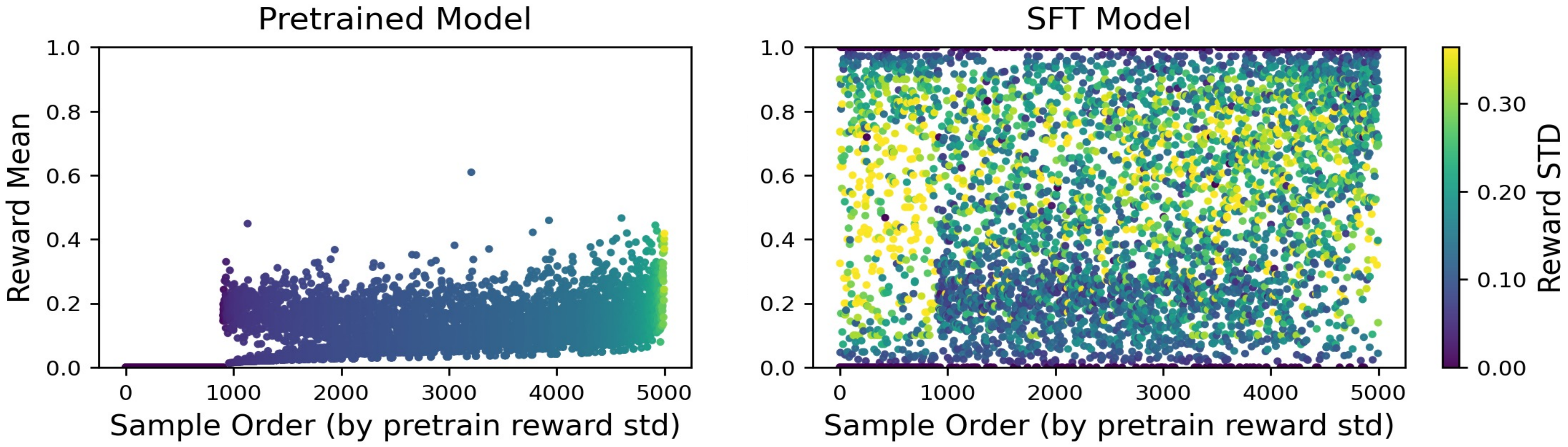


Overcoming Vanishing Gradients in RFT

Common Heuristics: Increasing learning rate, temperature, entropy regularization ❌

Observation: Initial SFT phase reduces number of inputs with small reward std

NarrativeQA (train)



ⓘ **Importance of SFT in RFT pipeline: mitigates vanishing gradients**

A Few SFT Steps on a Small Number of Samples Suffice

A Few SFT Steps on a Small Number of Samples Suffice

Limitation of Initial SFT Phase: Requires labeled data

A Few SFT Steps on a Small Number of Samples Suffice

Limitation of Initial SFT Phase: Requires labeled data

Expectation: If SFT phase is beneficial due to mitigating vanishing gradients for RFT

A Few SFT Steps on a Small Number of Samples Suffice

Limitation of Initial SFT Phase: Requires labeled data

Expectation: If SFT phase is beneficial due to mitigating vanishing gradients for RFT

→ A few steps of SFT on small # of labeled samples should suffice

A Few SFT Steps on a Small Number of Samples Suffice

Limitation of Initial SFT Phase: Requires labeled data

Expectation: If SFT phase is beneficial due to mitigating vanishing gradients for RFT

→ A few steps of SFT on small # of labeled samples should suffice

Result

Using **1% of labeled samples** and 40% of steps for initial SFT allows RFT to reach roughly same reward as with “full” initial SFT

A Few SFT Steps on a Small Number of Samples Suffice

Limitation of Initial SFT Phase: Requires labeled data

Expectation: If SFT phase is beneficial due to mitigating vanishing gradients for RFT

→ A few steps of SFT on small # of labeled samples should suffice

Result

Using **1% of labeled samples** and 40% of steps for initial SFT allows RFT to reach roughly same reward as with "full" initial SFT

The initial SFT phase does not need to be expensive!

Conclusion: Vanishing Gradients in RFT

Conclusion: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$

Expected gradient for an input vanishes in RFT
if the input's reward std is small

Conclusion: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$

Expected gradient for an input vanishes in RFT
if the input's reward std is small

Vanishing gradients in RFT are prevalent and detrimental to maximizing reward

Conclusion: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ **Expected gradient for an input vanishes in RFT** if the input's reward std is small

Vanishing gradients in RFT are prevalent and detrimental to maximizing reward

Initial SFT phase allows overcoming vanishing gradients in RFT, and **does not need to be expensive**

Conclusion: Vanishing Gradients in RFT

$\nabla_\theta \mathbf{V}_\theta(\mathbf{x}) \approx \mathbf{0}$ **Expected gradient for an input vanishes in RFT** if the input's reward std is small

Vanishing gradients in RFT are prevalent and detrimental to maximizing reward

Initial SFT phase allows overcoming vanishing gradients in RFT, and **does not need to be expensive**

Reward std is a key quantity to track for successful RFT

Implicit Bias of Policy Gradient in Linear Quadratic Control: Extrapolation to Unseen Initial States

R + Alexander + Cohen-Karlik + Giryes + Globerson + Cohen | *arXiv 2024*

Policy Gradient in Optimal Control

Optimal Control Problem

Policy Gradient in Optimal Control

Optimal Control Problem

System: Starting from an initial state $\mathbf{x}_0$

Policy Gradient in Optimal Control

Optimal Control Problem

System: Starting from an initial state $\mathbf{x}_0$

$$\mathbf{x}_{h+1} = f(\underbrace{\mathbf{x}_h}_{\text{state}}, \underbrace{\mathbf{u}_h}_{\text{control}}) \qquad h = 0, \ldots, \underbrace{H}_{\text{time horizon}} - 1$$

Policy Gradient in Optimal Control

Optimal Control Problem

System: Starting from an initial state $\mathbf{x}_0$

$$\mathbf{x}_{h+1} = f(\underbrace{\mathbf{x}_h}_{\text{state}}, \underbrace{\mathbf{u}_h}_{\text{control}}) \qquad \underbrace{h = 0, \ldots, H - 1}_{\text{time horizon}}$$

Goal: Choose controls that minimize the cost $\sum_{h=0}^{H} c(\mathbf{x}_h, \mathbf{u}_h)$

Policy Gradient in Optimal Control

Optimal Control Problem

System: Starting from an initial state $\mathbf{x}_0$

$$\mathbf{x}_{h+1} = f(\underbrace{\mathbf{x}_h}_{\text{state}}, \underbrace{\mathbf{u}_h}_{\text{control}}) \qquad h = 0, \dots, \underbrace{H}_{\text{time horizon}} - 1$$

Goal: Choose controls that minimize the cost $\sum_{h=0}^{H} c(\mathbf{x}_h, \mathbf{u}_h)$

Policy Gradient

Parameterize controller (e.g. as neural network)

Policy Gradient in Optimal Control

Optimal Control Problem

System: Starting from an initial state $\mathbf{x}_0$

$$\mathbf{x}_{h+1} = f(\mathbf{x}_h, \mathbf{u}_h) \qquad h = 0, \ldots, H-1$$

state control time horizon

Goal: Choose controls that minimize the cost $\sum_{h=0}^{H} c(\mathbf{x}_h, \mathbf{u}_h)$

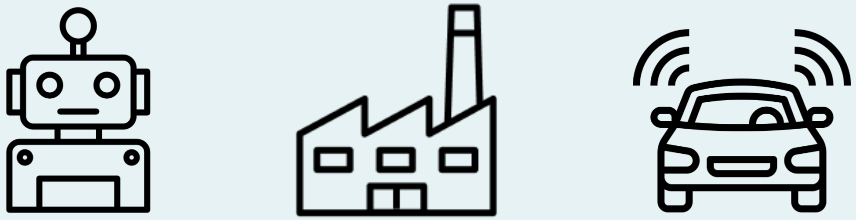
Policy Gradient

Parameterize controller (e.g. as neural network)

Minimize cost via **gradient descent** w.r.t. controller parameters

Extrapolation to Unseen Initial States

Issue of Prime Importance: Extrapolation to **initial states unseen in training**



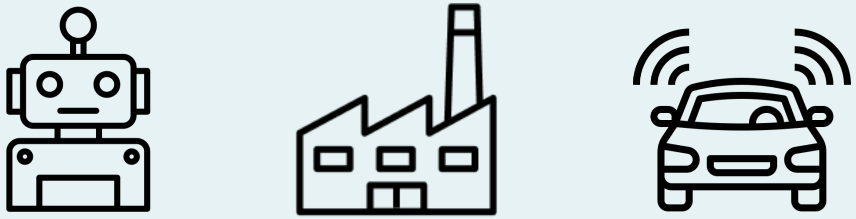
Extrapolation to Unseen Initial States

Issue of Prime Importance: Extrapolation to initial states unseen in training

Often multiple controllers minimize cost for initial states seen in training

Extrapolation to Unseen Initial States

Issue of Prime Importance: Extrapolation to **initial states unseen in training**

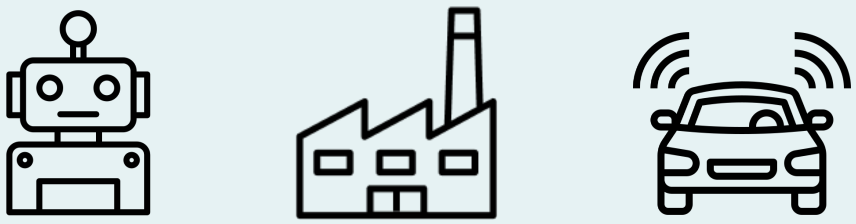


Often multiple controllers minimize cost for **initial states seen in training**

→ Extrapolation is determined by the **implicit bias** of policy gradient

Extrapolation to Unseen Initial States

Issue of Prime Importance: Extrapolation to **initial states unseen in training**



Often multiple controllers minimize cost for **initial states seen in training**

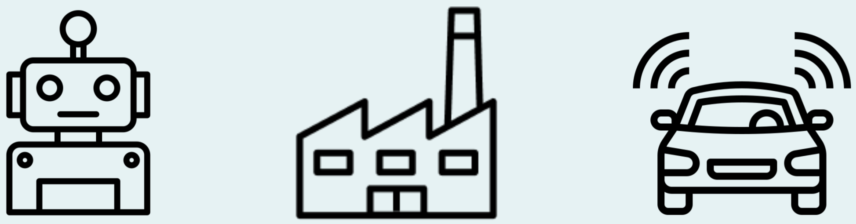
→ Extrapolation is determined by the **implicit bias** of policy gradient

Effect of implicit bias on extrapolation was theoretically studied in supervised learning

(Xu et al. 2021, Abbe et al. 2022/23, Cohen-Karlik et al. 2022/23)

Extrapolation to Unseen Initial States

Issue of Prime Importance: Extrapolation to **initial states unseen in training**



Often multiple controllers minimize cost for **initial states seen in training**

→ Extrapolation is determined by the **implicit bias** of policy gradient

Effect of implicit bias on extrapolation was theoretically studied in supervised learning

(Xu et al. 2021, Abbe et al. 2022/23, Cohen-Karlik et al. 2022/23)

not understood in optimal control

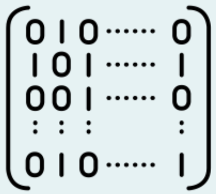
Main Contributions: Effect of Implicit Bias on Extrapolation

Main Contributions: Effect of Implicit Bias on Extrapolation

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?

Main Contributions: Effect of Implicit Bias on Extrapolation

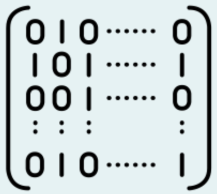
Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?



Theory for the Linear Quadratic Regulator (LQR) Problem:
Extrapolation depends on an **interplay between the system and initial states seen in training**

Main Contributions: Effect of Implicit Bias on Extrapolation

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?

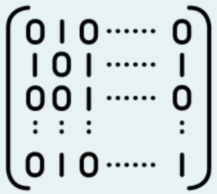


Theory for the Linear Quadratic Regulator (LQR) Problem:
Extrapolation depends on an **interplay between the system and initial states seen in training**

Experiments:
Support theory for LQR and demonstrate its conclusions apply to **non-linear systems and neural network controllers**

Main Contributions: Effect of Implicit Bias on Extrapolation

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?



Theory for the Linear Quadratic Regulator (LQR) Problem:
Extrapolation depends on an **interplay between the system and initial states seen in training**

Experiments:
Support theory for LQR and demonstrate its conclusions apply to **non-linear systems and neural network controllers**

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System

$$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$$

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System

$$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$$

Quadratic Costs

$$\sum_{h=0}^{H} \mathbf{x}_h^\top \mathbf{Q}\mathbf{x}_h + \mathbf{u}_h^\top \mathbf{R}\mathbf{u}_h$$

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System

$$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$$

Quadratic Costs

$$\sum_{h=0}^{H} \mathbf{x}_h^\top \mathbf{Q}\mathbf{x}_h + \mathbf{u}_h^\top \mathbf{R}\mathbf{u}_h$$

Linear Controller

$$\mathbf{u}_h = \mathbf{K}\mathbf{x}_h$$

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System	Quadratic Costs	Linear Controller
$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$	$\sum_{h=0}^{H} \mathbf{x}_h^\top \mathbf{Q}\mathbf{x}_h + \mathbf{u}_h^\top \mathbf{R}\mathbf{u}_h$	$\mathbf{u}_h = \mathbf{K}\mathbf{x}_h$

For **training set of initial states** $\mathcal{S} \subset \mathbb{R}^D$ the controller is learned by minimizing the **training cost:**

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System	Quadratic Costs	Linear Controller
$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$	$\sum_{h=0}^{H} \mathbf{x}_h^\top \mathbf{Q}\mathbf{x}_h + \mathbf{u}_h^\top \mathbf{R}\mathbf{u}_h$	$\mathbf{u}_h = \mathbf{K}\mathbf{x}_h$

For **training set of initial states** $\mathcal{S} \subset \mathbb{R}^D$ the controller is learned by minimizing the **training cost:**

$$cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top \big(\mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K}\big)\mathbf{x}_h$$

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System	Quadratic Costs	Linear Controller
$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$	$\sum_{h=0}^{H} \mathbf{x}_h^\top \mathbf{Q}\mathbf{x}_h + \mathbf{u}_h^\top \mathbf{R}\mathbf{u}_h$	$\mathbf{u}_h = \mathbf{K}\mathbf{x}_h$

For **training set of initial states** $\mathcal{S} \subset \mathbb{R}^D$ the controller is learned by minimizing the **training cost:**

$$cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K})\mathbf{x}_h$$

Learning the controller via **policy gradient** amounts to:

$$\mathbf{K}^{(t+1)} = \mathbf{K}^{(t)} - \eta \cdot \nabla cost_{\mathcal{S}}(\mathbf{K}^{(t)})$$

learning rate

The Linear Quadratic Regulator (LQR) Problem

LQR Problem (state $\mathbf{x}_h \in \mathbb{R}^D$, control $\mathbf{u}_h \in \mathbb{R}^M$)

Linear System	Quadratic Costs	Linear Controller
$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \mathbf{B}\mathbf{u}_h$	$\sum_{h=0}^{H} \mathbf{x}_h^\top \mathbf{Q}\mathbf{x}_h + \mathbf{u}_h^\top \mathbf{R}\mathbf{u}_h$	$\mathbf{u}_h = \mathbf{K}\mathbf{x}_h$

For **training set of initial states** $\mathcal{S} \subset \mathbb{R}^D$ the controller is learned by minimizing the **training cost:**

$$cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top \left(\mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K}\right)\mathbf{x}_h$$

Learning the controller via **policy gradient** amounts to:

$$\mathbf{K}^{(t+1)} = \mathbf{K}^{(t)} - \underbrace{\eta}_{\text{learning rate}} \cdot \nabla cost_{\mathcal{S}}(\mathbf{K}^{(t)})$$

initialization $\mathbf{K}^{(0)} = \mathbf{0}$

Existing Analyses of Policy Gradient in LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top \big(\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}\big) \mathbf{x}_h$

Existing Analyses of Policy Gradient in LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Existing Analyses of Policy Gradient in LQR (e.g. Fazel et al. 2018, Malik et al. 2019, Bu et al. 2019/20)

Typically assume that:

Existing Analyses of Policy Gradient in LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Existing Analyses of Policy Gradient in LQR (e.g. Fazel et al. 2018, Malik et al. 2019, Bu et al. 2019/20)

Typically assume that:

- Cost matrix $\mathbf{R}$ is positive definite — controls are regularized

Existing Analyses of Policy Gradient in LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Existing Analyses of Policy Gradient in LQR (e.g. Fazel et al. 2018, Malik et al. 2019, Bu et al. 2019/20)

Typically assume that:

- Cost matrix $\mathbf{R}$ is positive definite – **controls are regularized**
- Training set of initial states $\mathcal{S}$ **spans** $\mathbb{R}^D$

Existing Analyses of Policy Gradient in LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Existing Analyses of Policy Gradient in LQR (e.g. Fazel et al. 2018, Malik et al. 2019, Bu et al. 2019/20)

Typically assume that:

- Cost matrix $\mathbf{R}$ is positive definite – **controls are regularized**
- Training set of initial states $\mathcal{S}$ **spans** $\mathbb{R}^D$

→ Training cost has a **unique minimizer**

Existing Analyses of Policy Gradient in LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Existing Analyses of Policy Gradient in LQR (e.g. Fazel et al. 2018, Malik et al. 2019, Bu et al. 2019/20)

Typically assume that:

- Cost matrix $\mathbf{R}$ is positive definite – **controls are regularized**
- Training set of initial states $\mathcal{S}$ **spans** $\mathbb{R}^D$

→ Training cost has a **unique minimizer**

Under these assumptions implicit bias is irrelevant

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top \left(\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}\right) \mathbf{x}_h$

Underdetermined LQR

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**
- Training set of initial states $\mathcal{S}$ **does not span** $\mathbb{R}^D$

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**
- Training set of initial states $\mathcal{S}$ **does not span** $\mathbb{R}^D$
- $\mathbf{B}$ is full rank – controller's ability to affect the state is not limited

$$\boxed{\mathbf{x}_{h+1} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}_h}$$

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \mathbf{x}_h^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x}_h$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**
- Training set of initial states $\mathcal{S}$ **does not span** $\mathbb{R}^D$
- $\mathbf{B}$ is full rank – controller's ability to affect the state is not limited

$$\boxed{\mathbf{x}_{h+1} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}_h}$$

For simplicity:
$\mathbf{B} = \mathbf{Q} = \mathbf{I}$

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \|\mathbf{x}_h\|^2$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**
- Training set of initial states $\mathcal{S}$ **does not span** $\mathbb{R}^D$
- $\mathbf{B}$ is full rank – controller's ability to affect the state is not limited

$$\mathbf{x}_{h+1} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}_h$$

For simplicity:
$\mathbf{B} = \mathbf{Q} = \mathbf{I}$

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \|(\mathbf{A} + \mathbf{K})^h \mathbf{x}_0\|^2$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**
- Training set of initial states $\mathcal{S}$ **does not span** $\mathbb{R}^D$
- $\mathbf{B}$ is full rank – controller's ability to affect the state is not limited

$$\boxed{\mathbf{x}_{h+1} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}_h}$$

For simplicity:
$\mathbf{B} = \mathbf{Q} = \mathbf{I}$

Setting: Underdetermined LQR

Training cost: $cost_{\mathcal{S}}(\mathbf{K}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_0 \in \mathcal{S}} \sum_{h=0}^{H} \|(\mathbf{A} + \mathbf{K})^h \mathbf{x}_0\|^2$

Underdetermined LQR

- $\mathbf{R} = \mathbf{0}$ – controls are **not regularized**
- Training set of initial states $\mathcal{S}$ **does not span** $\mathbb{R}^D$
- $\mathbf{B}$ is full rank – controller's ability to affect the state is not limited

$\mathbf{x}_{h+1} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}_h$

For simplicity:
$\mathbf{B} = \mathbf{Q} = \mathbf{I}$

In this setting the training cost has multiple minimizers

Quantifying Extrapolation

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost **if and only if** $\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0$ for all $\mathbf{x}_0 \in \mathcal{S}$

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost **if and only if** $\underbrace{\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0}_{\mathbf{K} \text{ sends } \mathbf{x}_0 \text{ to zero}}$ for all $\mathbf{x}_0 \in \mathcal{S}$

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost **if and only if** $\underbrace{\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0}_{\mathbf{K} \text{ sends } \mathbf{x}_0 \text{ to zero}}$ for all $\mathbf{x}_0 \in \mathcal{S}$

Let $\mathcal{U}$ be an arbitrary orthonormal basis of $\mathcal{S}^\perp$

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost if and only if $\underbrace{\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0}_{\mathbf{K} \text{ sends } \mathbf{x}_0 \text{ to zero}}$ for all $\mathbf{x}_0 \in \mathcal{S}$

Let $\mathcal{U}$ be an arbitrary orthonormal basis of $\mathcal{S}^\perp$

Controllers minimizing the training cost

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost if and only if $\underbrace{\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0}_{\mathbf{K}\text{ sends }\mathbf{x}_0\text{ to zero}}$ for all $\mathbf{x}_0 \in \mathcal{S}$

Let $\mathcal{U}$ be an arbitrary orthonormal basis of $\mathcal{S}^\perp$

Controllers minimizing the training cost $\begin{cases} \text{produce identical controls for states in } \mathcal{S} \\ \text{differ arbitrarily in their controls for states in } \mathcal{U} \end{cases}$

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost if and only if $\underbrace{\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0}_{\mathbf{K} \text{ sends } \mathbf{x}_0 \text{ to zero}}$ for all $\mathbf{x}_0 \in \mathcal{S}$

Let $\mathcal{U}$ be an arbitrary orthonormal basis of $\mathcal{S}^\perp$

Controllers minimizing the training cost $\begin{cases} \text{produce identical controls for states in } \mathcal{S} \\ \text{differ arbitrarily in their controls for states in } \mathcal{U} \end{cases}$

We quantify extrapolation for a controller $\mathbf{K}$ by its performance on initial states in $\mathcal{U}$

Quantifying Extrapolation

Optimality Condition: $\mathbf{K}$ minimizes the training cost if and only if $\underbrace{\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2 = 0}_{\mathbf{K} \text{ sends } \mathbf{x}_0 \text{ to zero}}$ for all $\mathbf{x}_0 \in \mathcal{S}$

Let $\mathcal{U}$ be an arbitrary orthonormal basis of $\mathcal{S}^\perp$

Controllers minimizing the training cost $\begin{cases} \text{produce identical controls for states in } \mathcal{S} \\ \text{differ arbitrarily in their controls for states in } \mathcal{U} \end{cases}$

We quantify extrapolation for a controller $\mathbf{K}$ by its performance on initial states in $\mathcal{U}$

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_0 \in \mathcal{U}} \|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2$$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_0 \in \mathcal{U}} \|(\mathbf{A} + \mathbf{K})\mathbf{x}_0\|^2$$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_0 \in \mathcal{U}} \|(\mathbf{A} + \mathbf{K})\mathbf{x}_0\|^2$$

Perfectly Extrapolating $\mathrm{K}_{\mathrm{ext}}$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|}\sum_{\mathbf{x}_0\in\mathcal{U}}\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2$$

Perfectly Extrapolating K_{ext}

Satisfies $(\mathbf{A}+\mathbf{K}_{\text{ext}})\mathbf{x}_0 = \mathbf{0}$ for all $\mathbf{x}_0 \in \mathbb{R}^D$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|}\sum_{\mathbf{x}_0\in\mathcal{U}}\|(\mathbf{A}+\mathbf{K})\mathbf{x}_0\|^2$$

Perfectly Extrapolating $\mathrm{K}_{\mathrm{ext}}$

Satisfies $(\mathbf{A}+\mathbf{K}_{\mathrm{ext}})\mathbf{x}_0 = \mathbf{0}$ for all $\mathbf{x}_0 \in \mathbb{R}^D$

Minimizes the training cost and
$\mathcal{E}(\mathbf{K}_{\mathrm{ext}}) = 0$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_0 \in \mathcal{U}} \|(\mathbf{A} + \mathbf{K})\mathbf{x}_0\|^2$$

Perfectly Extrapolating $\mathrm{K}_{\mathrm{ext}}$

Satisfies $(\mathbf{A} + \mathbf{K}_{\mathrm{ext}})\mathbf{x}_0 = \mathbf{0}$ for all $\mathbf{x}_0 \in \mathbb{R}^D$

Minimizes the training cost and

$\mathcal{E}(\mathbf{K}_{\mathrm{ext}}) = 0$

Non-Extrapolating $\mathrm{K}_{\text{no-ext}}$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_0 \in \mathcal{U}} \|(\mathbf{A} + \mathbf{K})\mathbf{x}_0\|^2$$

Perfectly Extrapolating $\mathrm{K}_{\mathrm{ext}}$

Satisfies $(\mathbf{A} + \mathbf{K}_{\mathrm{ext}})\mathbf{x}_0 = \mathbf{0}$ for all $\mathbf{x}_0 \in \mathbb{R}^D$

Minimizes the training cost and

$\mathcal{E}(\mathbf{K}_{\mathrm{ext}}) = 0$

Non-Extrapolating $\mathrm{K}_{\text{no-ext}}$

Satisfies $(\mathbf{A} + \mathbf{K}_{\text{no-ext}})\mathbf{x}_0 = \begin{cases} \mathbf{0} & , \mathbf{x}_0 \in \mathcal{S} \\ \mathbf{A}\mathbf{x}_0 & , \mathbf{x}_0 \in \mathcal{U} \end{cases}$

Quantifying Extrapolation: Baseline Controllers

Extrapolation Error

$$\mathcal{E}(\mathbf{K}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_0 \in \mathcal{U}} \|(\mathbf{A} + \mathbf{K})\mathbf{x}_0\|^2$$

Perfectly Extrapolating K_{ext}

Satisfies $(\mathbf{A} + \mathbf{K}_{\text{ext}})\mathbf{x}_0 = \mathbf{0}$ for all $\mathbf{x}_0 \in \mathbb{R}^D$

Minimizes the training cost and

$\mathcal{E}(\mathbf{K}_{\text{ext}}) = 0$

Non-Extrapolating $\mathrm{K}_{\text{no-ext}}$

Satisfies $(\mathbf{A} + \mathbf{K}_{\text{no-ext}})\mathbf{x}_0 = \begin{cases} \mathbf{0} & , \mathbf{x}_0 \in \mathcal{S} \\ \mathbf{A}\mathbf{x}_0 & , \mathbf{x}_0 \in \mathcal{U} \end{cases}$

Minimizes the training cost but

$\mathcal{E}(\mathbf{K}_{\text{no-ext}})$ is **typically high**

Intuition: Extrapolation Depends on Exploration

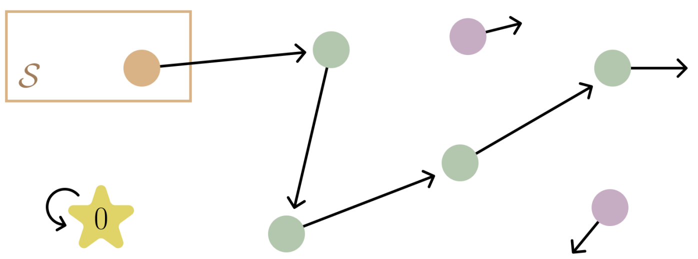
Intuition Behind Our Analysis: Extrapolation depends on **degree of exploration induced by the system** from training initial states

Intuition: Extrapolation Depends on Exploration

Intuition Behind Our Analysis: Extrapolation depends on **degree of exploration induced by the system** from training initial states

Policy Gradient Iterate t Controller

State Dynamics: $\mathbf{A} + \mathbf{K}^{(t)}$



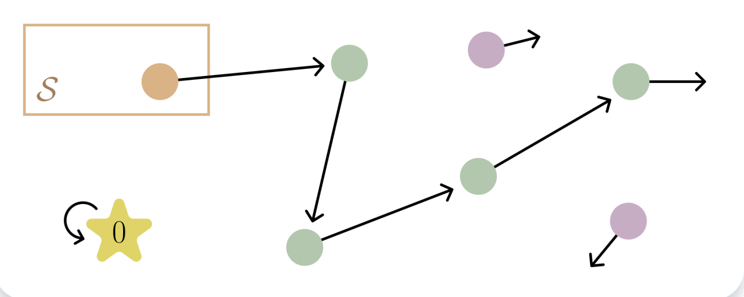
- initial state seen in training
- state explored during policy gradient
- state unexplored during policy gradient

Intuition: Extrapolation Depends on Exploration

Intuition Behind Our Analysis: Extrapolation depends on **degree of exploration induced by the system** from training initial states

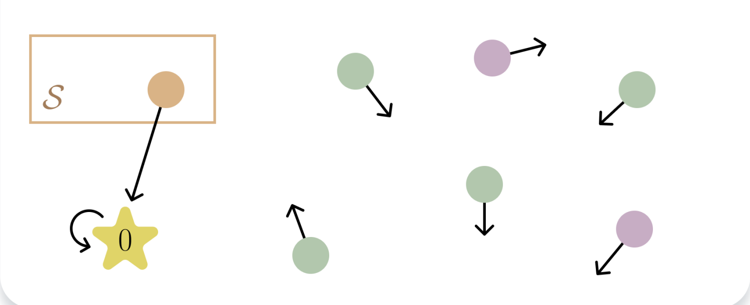
Policy Gradient Iterate t Controller

State Dynamics: $\mathbf{A} + \mathbf{K}^{(t)}$



Non-Extrapolating Controller

State Dynamics: $\mathbf{A} + \mathbf{K}_{\text{no-ext}}$



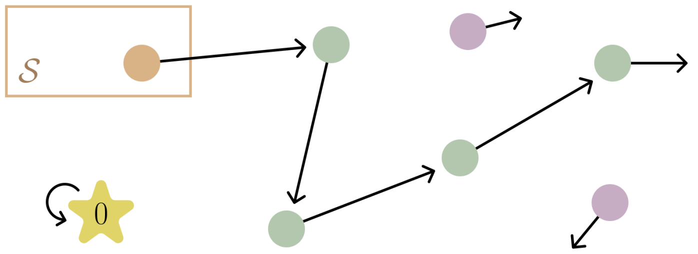
- initial state seen in training
- state explored during policy gradient
- state unexplored during policy gradient

Intuition: Extrapolation Depends on Exploration

Intuition Behind Our Analysis: Extrapolation depends on **degree of exploration induced by the system** from training initial states

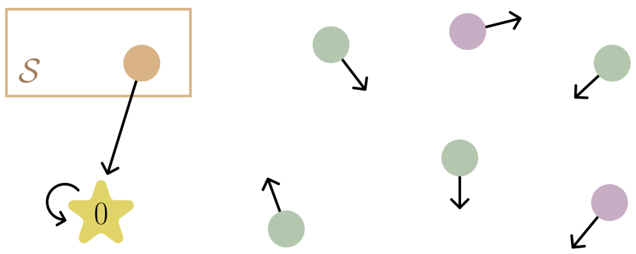
Policy Gradient Iterate t Controller

State Dynamics: $\mathbf{A} + \mathbf{K}^{(t)}$



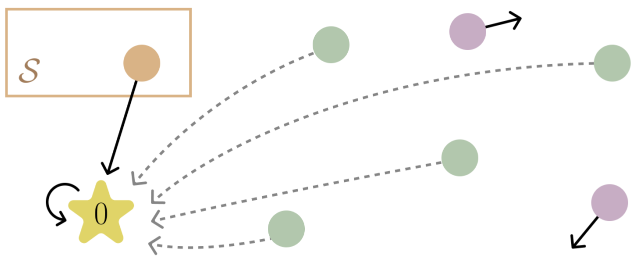
Non-Extrapolating Controller

State Dynamics: $\mathbf{A} + \mathbf{K}_{\text{no-ext}}$



Policy Gradient Final Controller

State Dynamics: $\mathbf{A} + \mathbf{K}_{\text{pg}}$



- initial state seen in training
- state explored during policy gradient
- state unexplored during policy gradient

Extrapolation Requires Exploration

Extrapolation Requires Exploration

$\mathbf{K}^{(t)}$ – the policy gradient controller at iteration t

$\mathcal{S}$ – initial states seen in training

Extrapolation Requires Exploration

$\mathbf{K}^{(t)}$ – the policy gradient controller at iteration t

$\mathcal{S}$ – initial states seen in training

$\mathcal{X}_{\text{pg}}$ – the set of **states encountered during policy gradient**

Extrapolation Requires Exploration

$\mathbf{K}^{(t)}$ – the policy gradient controller at iteration t

$\mathcal{S}$ – initial states seen in training

$\mathcal{X}_{\mathrm{pg}}$ – the set of **states encountered during policy gradient**

Proposition – *Exploration is Necessary for Extrapolation*

Extrapolation Requires Exploration

$\mathbf{K}^{(t)}$ – the policy gradient controller at iteration t

$\mathcal{S}$ – initial states seen in training

$\mathcal{X}_{\text{pg}}$ – the set of **states encountered during policy gradient**

Proposition – *Exploration is Necessary for Extrapolation*

- For any $\mathbf{x} \in \mathcal{X}_{\text{pg}}^{\perp}$ the controls produced by $\mathbf{K}^{(t)}$ and $\mathbf{K}_{\text{no-ext}}$ are the same

Extrapolation Requires Exploration

$\mathbf{K}^{(t)}$ – the policy gradient controller at iteration t

$\mathcal{S}$ – initial states seen in training

$\mathcal{X}_{\text{pg}}$ – the set of **states encountered during policy gradient**

Proposition – *Exploration is Necessary for Extrapolation*

- For any $\mathbf{x} \in \mathcal{X}_{\text{pg}}^{\perp}$ the controls produced by $\mathbf{K}^{(t)}$ and $\mathbf{K}_{\text{no-ext}}$ are the same
- There exist systems s.t. $\mathcal{X}_{\text{pg}} \subseteq \operatorname{span}(\mathcal{S})$ and $\mathcal{E}(\mathbf{K}^{(t)}) = \mathcal{E}(\mathbf{K}_{\text{no-ext}})$

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient?

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient?

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient?

Simple "Shift" Setting

- Consider training initial state $\mathbf{e}_1$
  — first standard basis vector

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient?

Simple "Shift" Setting

- Consider training initial state $\mathbf{e}_1$
  — first standard basis vector
- The trajectory steered by $\mathbf{K}^{(0)}$
  is $\mathbf{e}_1, \mathbf{A}\mathbf{e}_1, \ldots, \mathbf{A}^{H-1}\mathbf{e}_1$

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient? ✓

Simple "Shift" Setting

- Consider training initial state $\mathbf{e}_1$ — first standard basis vector
- The trajectory steered by $\mathbf{K}^{(0)}$ is $\mathbf{e}_1, \mathbf{A}\mathbf{e}_1, \ldots, \mathbf{A}^{H-1}\mathbf{e}_1$

➡

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

Ensures trajectory spans whole state space

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient? ✓

Simple "Shift" Setting

- Consider training initial state $\mathbf{e}_1$ — first standard basis vector
- The trajectory steered by $\mathbf{K}^{(0)}$ is $\mathbf{e}_1, \mathbf{A}\mathbf{e}_1, \ldots, \mathbf{A}^{H-1}\mathbf{e}_1$

→

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

Ensures trajectory spans whole state space

Proposition

Policy gradient converges to $\mathbf{K}_{\text{pg}}$, which minimizes the training cost and:

$$\mathcal{E}(\mathbf{K}_{\text{pg}}) << \mathcal{E}(\mathbf{K}_{\text{no-ext}})$$

Extrapolation in Exploration-Inducing Setting

Q: Exploration is necessary for extrapolation, but can it be sufficient? ✓

Simple "Shift" Setting

- Consider training initial state $\mathbf{e}_1$ — first standard basis vector
- The trajectory steered by $\mathbf{K}^{(0)}$ is $\mathbf{e}_1, \mathbf{A}\mathbf{e}_1, \ldots, \mathbf{A}^{H-1}\mathbf{e}_1$

→

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

Ensures trajectory spans whole state space

Proposition

Policy gradient converges to $\mathbf{K}_{\text{pg}}$, which minimizes the training cost and:

$$\mathcal{E}(\mathbf{K}_{\text{pg}}) << \mathcal{E}(\mathbf{K}_{\text{no-ext}})$$

where perfect extrapolation is attained when the horizon $H \to \infty$

Extrapolation in Typical Setting

Q: We saw two ends of a spectrum, but which type of extrapolation do we typically get?

Extrapolation in Typical Setting

Q: We saw two ends of a spectrum, but which type of extrapolation do we typically get?

Typical Setting

- Arbitrary training initial state $\mathbf{x}_0$

Extrapolation in Typical Setting

Q: We saw two ends of a spectrum, but which type of extrapolation do we typically get?

Typical Setting

- Arbitrary training initial state $\mathbf{x}_0$
- Random $\mathbf{A}$ with entries sampled indepently from $\mathcal{N}(0, 1/\underset{\text{state dimension}}{D})$

Extrapolation in Typical Setting

Q: We saw two ends of a spectrum, but which type of extrapolation do we typically get?

Typical Setting

- Arbitrary training initial state $\mathbf{x}_0$
- Random $\mathbf{A}$ with entries sampled indepently from $\mathcal{N}(0, 1/\underbrace{D}_{\text{state dimension}})$

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

Extrapolation in Typical Setting

Q: We saw two ends of a spectrum, but which type of extrapolation do we typically get?

Typical Setting

- Arbitrary training initial state $\mathbf{x}_0$
- Random $\mathbf{A}$ with entries sampled indepently from $\mathcal{N}(0, 1/\underbrace{D}_{\text{state dimension}})$

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\big[\mathcal{E}(\mathbf{K}^{(1)})\big] \leq \mathbb{E}\big[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\big] - \Omega\big(\eta \cdot \tfrac{H^2}{D}\big)$$

Extrapolation in Typical Setting

Q: We saw two ends of a spectrum, but which type of extrapolation do we typically get?

Typical Setting

- Arbitrary training initial state $\mathbf{x}_0$
- Random $\mathbf{A}$ with entries sampled indepently from $\mathcal{N}(0, 1/\underset{\substack{\uparrow \\ \text{state dimension}}}{D})$

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\big[\mathcal{E}(\mathbf{K}^{(1)})\big] \leq \mathbb{E}\big[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\big] - \Omega\big(\eta \cdot \tfrac{H^2}{D}\big)$$

Additionally, extrapolation occurs with high probability if D is large

Extrapolation in Typical Setting: Proof Idea and Limitations

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\left[\mathcal{E}(\mathbf{K}^{(1)})\right] \leq \mathbb{E}\left[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\right] - \Omega\left(\eta \cdot \frac{H^2}{D}\right)$$

Additionally, extrapolation occurs with high probability if D is large

Proof Idea:

Extrapolation in Typical Setting: Proof Idea and Limitations

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\big[\mathcal{E}(\mathbf{K}^{(1)})\big] \leq \mathbb{E}\big[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\big] - \Omega\big(\eta \cdot \tfrac{H^2}{D}\big)$$

Additionally, extrapolation occurs with high probability if D is large

Proof Idea:

- Intuition: Random system generically induces exploration

Extrapolation in Typical Setting: Proof Idea and Limitations

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\big[\mathcal{E}(\mathbf{K}^{(1)})\big] \leq \mathbb{E}\big[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\big] - \Omega\big(\eta \cdot \tfrac{H^2}{D}\big)$$

Additionally, extrapolation occurs with high probability if D is large

Proof Idea:

- Intuition: Random system generically induces exploration
- Convert intuition to formal guarantee via tools from random matrix theory and topology

Extrapolation in Typical Setting: Proof Idea and Limitations

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\big[\mathcal{E}(\mathbf{K}^{(1)})\big] \leq \mathbb{E}\big[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\big] - \Omega\Big(\eta \cdot \frac{H^2}{D}\Big)$$

Additionally, extrapolation occurs with high probability if D is large

Proof Idea:

- Intuition: Random system generically induces exploration
- Convert intuition to formal guarantee via tools from random matrix theory and topology

Limitations: Condition on learning rate + only second iterate of policy gradient

Extrapolation in Typical Setting: Proof Idea and Limitations

Theorem

If the learning rate η is sufficiently small, a single iteration of policy gradient leads to non-trivial extrapolation:

$$\mathbb{E}\left[\mathcal{E}(\mathbf{K}^{(1)})\right] \leq \mathbb{E}\left[\mathcal{E}(\mathbf{K}_{\text{no-ext}})\right] - \Omega\left(\eta \cdot \frac{H^2}{D}\right)$$

Additionally, extrapolation occurs with high probability if D is large

Proof Idea:

- Intuition: Random system generically induces exploration
- Convert intuition to formal guarantee via tools from random matrix theory and topology

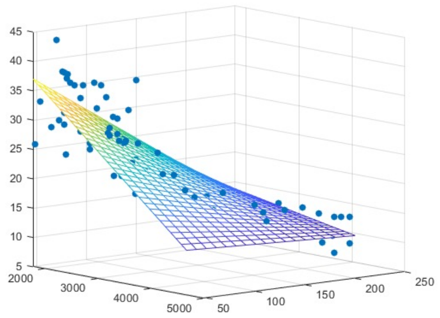
Limitations: Condition on learning rate + only second iterate of policy gradient

experiments suggest these limitations may be alleviated

Implicit Bias in Optimal Control $\neq$ Euclidean Norm Minimization

Implicit Bias in Optimal Control $\neq$ Euclidean Norm Minimization

Supervised Learning

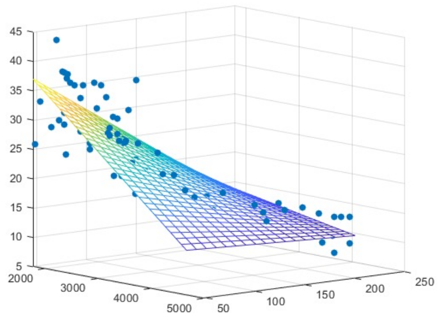


Task: Linear regression

Known (e.g. Zhang et al. 2017)**:** Implicit bias minimizes **Euclidean norm**

Implicit Bias in Optimal Control ≠ Euclidean Norm Minimization

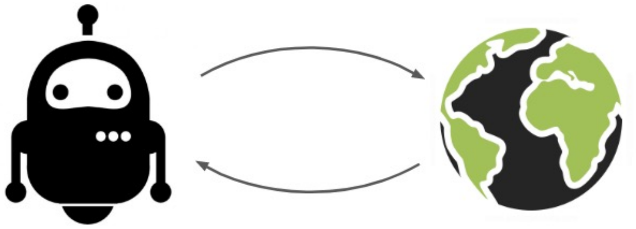
Supervised Learning



Task: Linear regression

Known (e.g. Zhang et al. 2017)**:** Implicit bias minimizes **Euclidean norm**

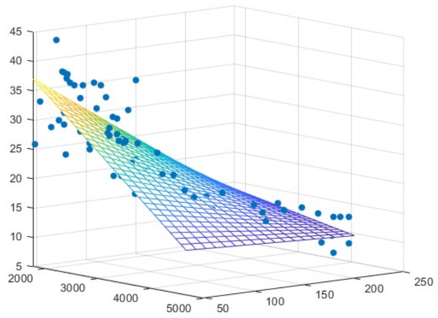
Optimal Control



Task: LQR

Implicit Bias in Optimal Control $\neq$ Euclidean Norm Minimization

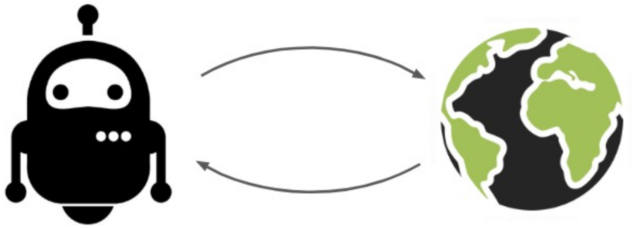
Supervised Learning



Task: Linear regression

Known (e.g. Zhang et al. 2017)**:** Implicit bias minimizes **Euclidean norm**

Optimal Control

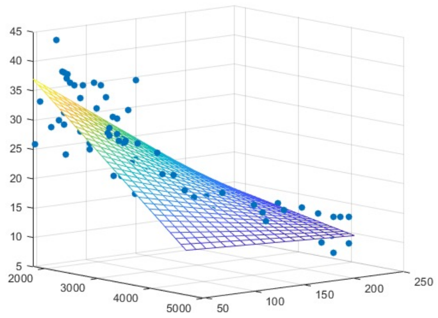


Task: LQR

Our Work: Implicit bias does **not** minimize **Euclidean norm**

Implicit Bias in Optimal Control $\neq$ Euclidean Norm Minimization

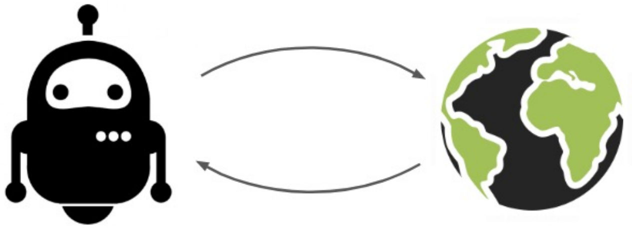
Supervised Learning



Task: Linear regression

Known (e.g. Zhang et al. 2017)**:** Implicit bias minimizes Euclidean norm

Optimal Control



Task: LQR

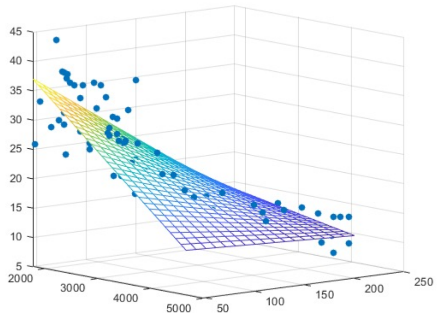
Our Work: Implicit bias does not minimize Euclidean norm

Corollary

Among controllers minimizing the training cost, $\mathbf{K}_{\text{no-ext}}$ has the minimal Euclidean norm

Implicit Bias in Optimal Control $\neq$ Euclidean Norm Minimization

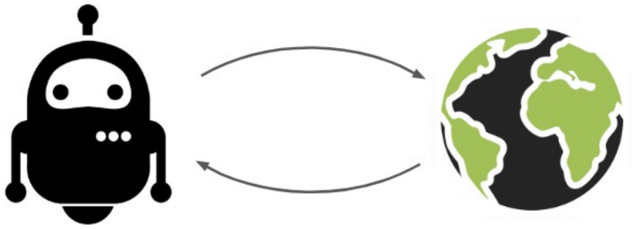
Supervised Learning



Task: Linear regression

Known (e.g. Zhang et al. 2017)**:** Implicit bias minimizes **Euclidean norm**

Optimal Control



Task: LQR

Our Work: Implicit bias does **not** minimize **Euclidean norm**

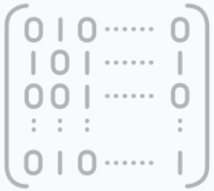
Corollary

Among controllers minimizing the training cost, $\mathbf{K}_{\text{no-ext}}$ has the minimal Euclidean norm

➡ Extrapolation implies policy gradient **does not implicitly minimize Euclidean norm**

Main Contributions: Effect of Implicit Bias on Extrapolation

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?

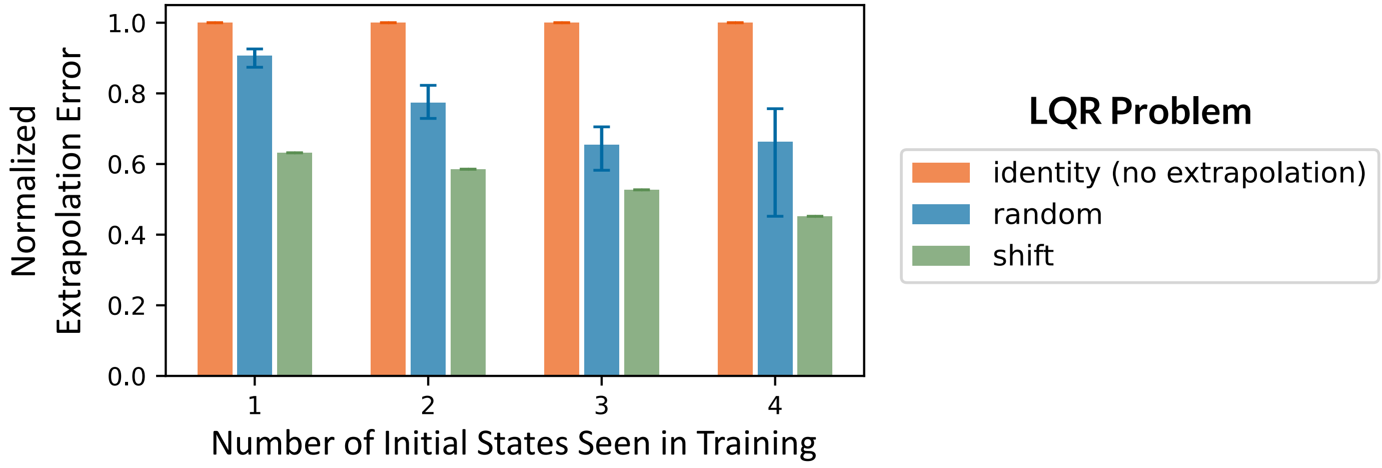


Theory for the Linear Quadratic Regulator (LQR) Problem:
Extrapolation depends on an **interplay between the system and initial states seen in training**

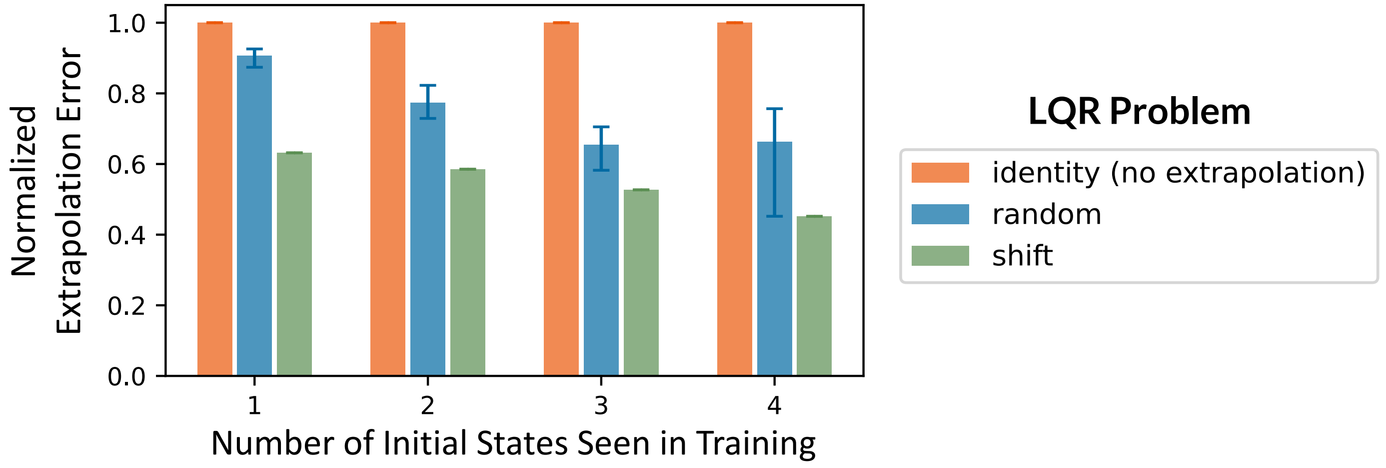
Experiments:
Support theory for LQR and demonstrate its conclusions apply to **non-linear systems and neural network controllers**

Experiments: Analyzed LQR Problems

Experiments: Analyzed LQR Problems



Experiments: Analyzed LQR Problems



In accordance with our theory:

No extrapolation occurs under the identity system, while for the shift and random systems we have non-trivial extrapolation (yet not perfect)

Experiments: Non-Linear Systems and Neural Network Controllers

Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration
from initial states seen in training

Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration from initial states seen in training ➡ Linear controller typically extrapolates

Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration from initial states seen in training ➡ Linear controller typically extrapolates

Experiments: Phenomenon extends to **non-linear systems** and **neural network controllers**

Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration from initial states seen in training → Linear controller typically extrapolates

Experiments: Phenomenon extends to **non-linear systems** and **neural network controllers**

Pendulum Control Problem

(analogous experiments for a quadcopter control problem)

Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration from initial states seen in training ➡ Linear controller typically extrapolates

Experiments: Phenomenon extends to **non-linear systems** and **neural network controllers**

Pendulum Control Problem

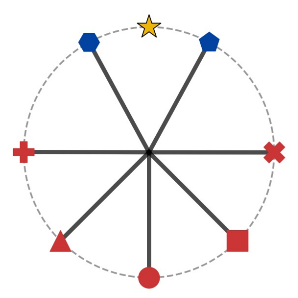
(analogous experiments for a quadcopter control problem)

★ target state

● initial state seen in training

● initial state unseen in training

Initial States (time step 0)



Experiments: Non-Linear Systems and Neural Network Controllers

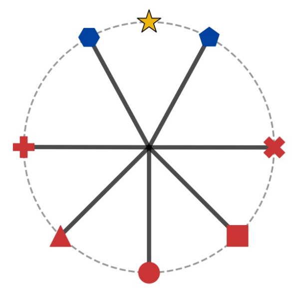
Our Theory: Linear system induces exploration from initial states seen in training ➡ Linear controller typically extrapolates

Experiments: Phenomenon extends to **non-linear systems** and **neural network controllers**

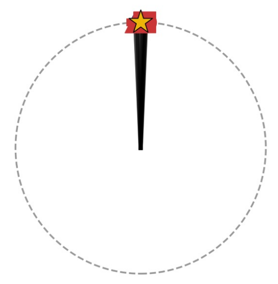
Pendulum Control Problem
(analogous experiments for a quadcopter control problem)

☆ target state
● initial state seen in training
● initial state unseen in training

Initial States (time step 0)



Policy Gradient Controller
Final States (time step 100)



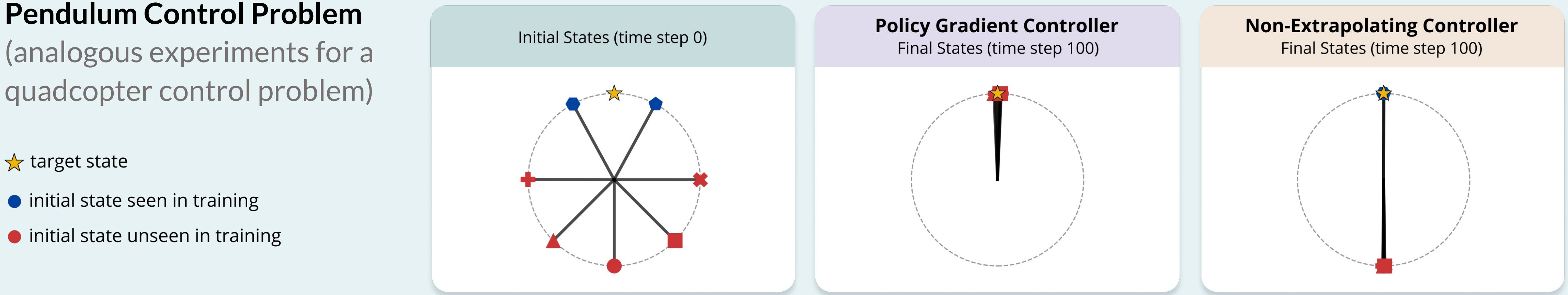
Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration from initial states seen in training ➡ Linear controller typically extrapolates

Experiments: Phenomenon extends to **non-linear systems** and **neural network controllers**

Pendulum Control Problem

(analogous experiments for a quadcopter control problem)



Experiments: Non-Linear Systems and Neural Network Controllers

Our Theory: Linear system induces exploration from initial states seen in training ➡ Linear controller typically extrapolates

Experiments: Phenomenon extends to **non-linear systems** and **neural network controllers**

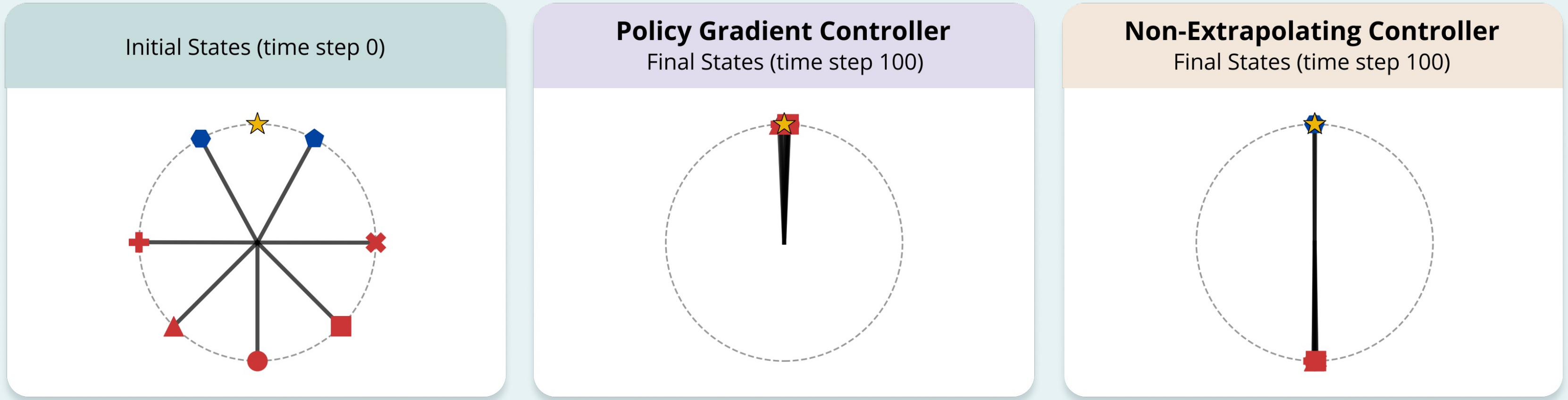
Pendulum Control Problem

(analogous experiments for a quadcopter control problem)

☆ target state

● initial state seen in training

● initial state unseen in training



ⓘ **The controller learned via policy gradient extrapolates despite existence of non-extrapolating controllers**

Conclusion: Implicit Bias of Policy Gradient in Optimal Control

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?

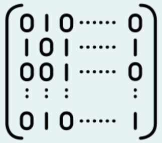
Conclusion: Implicit Bias of Policy Gradient in Optimal Control

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?

Theory for the LQR Problem:
Extrapolation depends on exploration induced by the system from initial states seen in training

Conclusion: Implicit Bias of Policy Gradient in Optimal Control

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?

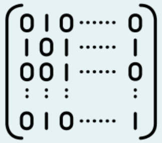


Theory for the LQR Problem: Extrapolation depends on exploration induced by the system from initial states seen in training

Experiments: Support theory for LQR and demonstrate its conclusions apply to non-linear systems and neural network controllers

Conclusion: Implicit Bias of Policy Gradient in Optimal Control

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?



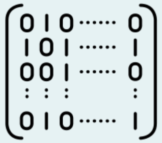
Theory for the LQR Problem: Extrapolation depends on exploration induced by the system from initial states seen in training

Experiments: Support theory for LQR and demonstrate its conclusions apply to non-linear systems and neural network controllers

Going Forward:

Conclusion: Implicit Bias of Policy Gradient in Optimal Control

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?



Theory for the LQR Problem: Extrapolation depends on exploration induced by the system from initial states seen in training

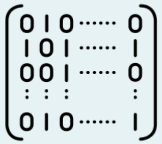
Experiments: Support theory for LQR and demonstrate its conclusions apply to non-linear systems and neural network controllers

Going Forward:

- Theory for non-linear systems and neural network controllers

Conclusion: Implicit Bias of Policy Gradient in Optimal Control

Q: To what extent does the implicit bias of policy gradient lead to extrapolation to initial states unseen in training?



Theory for the LQR Problem: Extrapolation depends on exploration induced by the system from initial states seen in training

Experiments: Support theory for LQR and demonstrate its conclusions apply to non-linear systems and neural network controllers

Going Forward:

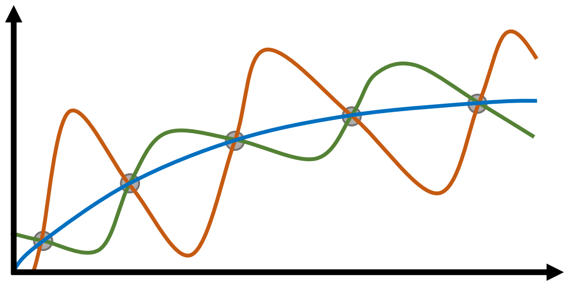
- Theory for non-linear systems and neural network controllers
- Enhancing extrapolation via methods for selecting initial states to train on

Outlook

Optimization and Implicit Bias in Optimal Control/Reinforcement Learning

Optimization and Implicit Bias in Optimal Control/Reinforcement Learning

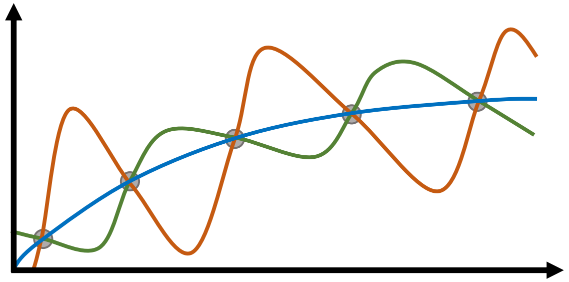
Supervised Learning



Optimization and implicit bias have been extensively studied

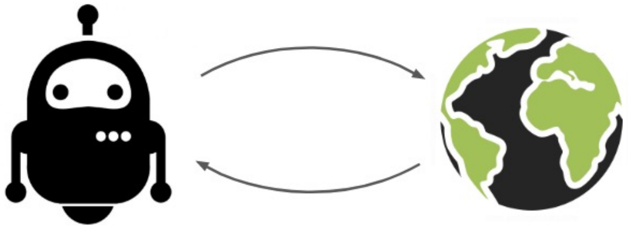
Optimization and Implicit Bias in Optimal Control/Reinforcement Learning

Supervised Learning



Optimization and implicit bias have been extensively studied

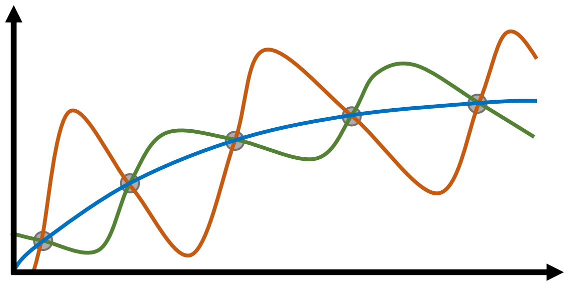
Optimal Control/Reinforcement Learning



Our Results: Optimization and implicit bias can substantially differ from those in supervised learning, hence require dedicated study

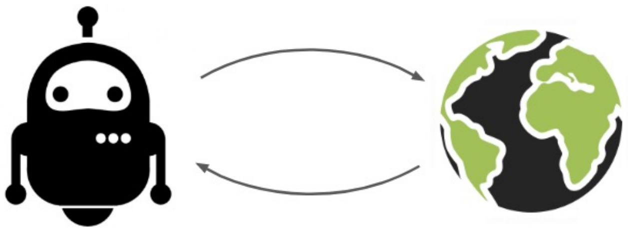
Optimization and Implicit Bias in Optimal Control/Reinforcement Learning

Supervised Learning



Optimization and implicit bias have been extensively studied

Optimal Control/Reinforcement Learning



Our Results: Optimization and implicit bias can substantially differ from those in supervised learning, hence require dedicated study

Studying optimization and implicit bias in optimal control/reinforcement learning may allow addressing their unique challenges

Thank You!

Work supported by:
Apple scholars in AI/ML PhD fellowship, Google Research Scholar Award, Google Research Gift, the Yandex Initiative in Machine Learning, the Israel Science Foundation (grant 1780/21), Len Blavatnik and the Blavatnik Family Foundation, Tel Aviv University Center for AI and Data Science, and Amnon and Anat Shashua