Offline Reinforcement Learning in the Wild

Nadav Cohen

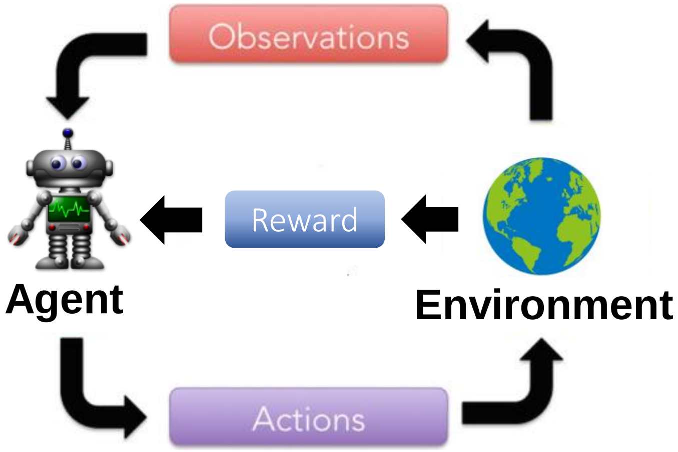
Tel Aviv University & Imubit

Israel Machine Vision Conference (IMVC) 2025

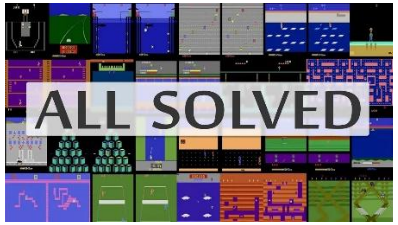
Reinforcement Learning (RL)

Goal

Design **agent** that steers an **environment** to maximize a **reward**



Applications



Computer gaming

Playing Go

Autonomous driving

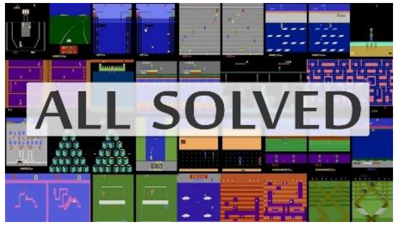
Medical treatment

Manufacturing optimization

Learning via Trial & Error

Learning an agent typically entails **trial & error** in environment

Feasible in some applications; **prohibitively costly/dangerous** in others



Computer gaming

Playing Go

Autonomous driving

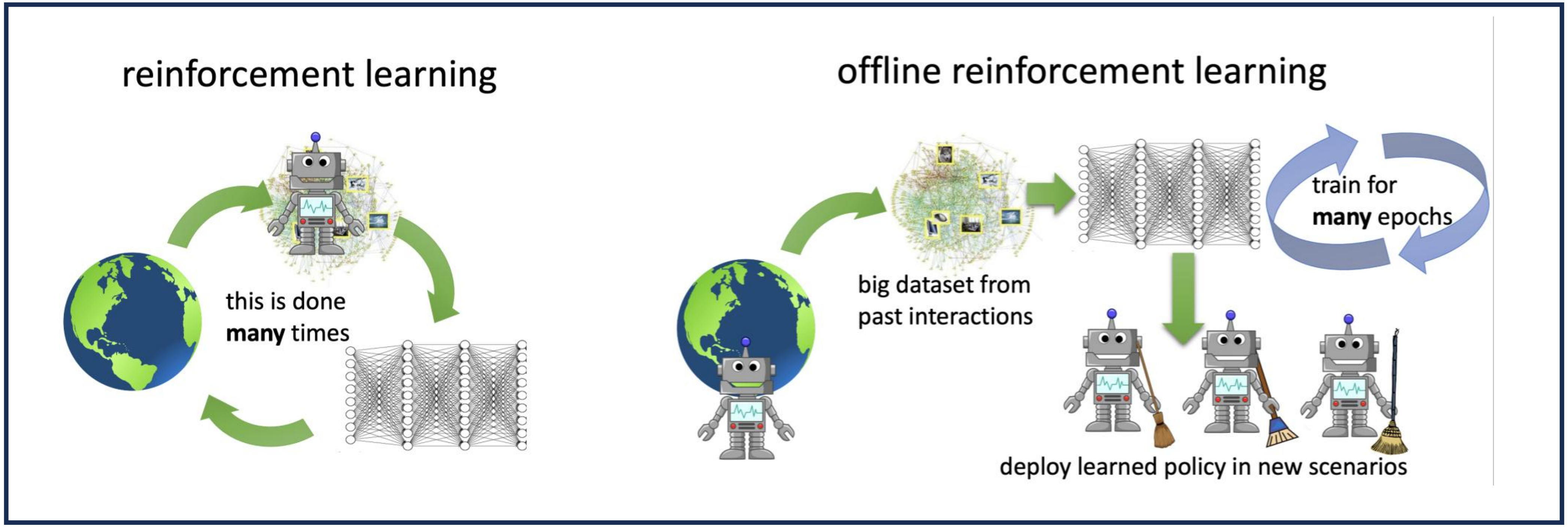
Medical treatment

Manufacturing optimization

Offline RL

Goal

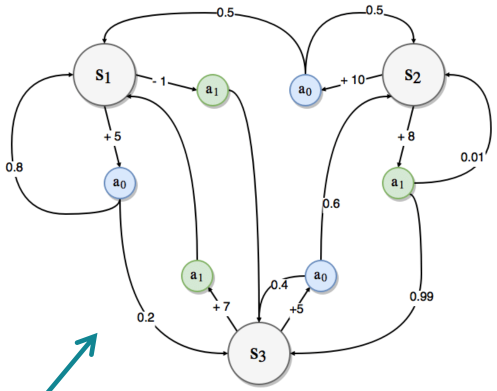
Learn an agent without trial & error in environment



Conventional Offline RL Methods

Designed for Markov Decision Process (MDP) environments

MDP environment



MDP environment is fully observable:
its observations reveal its full state

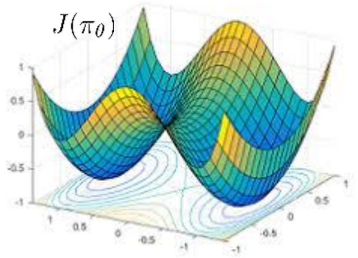
Conventional offline RL methods

Value-based methods

	A_1	A_2	...	A_M
S_1	$Q(S_1, A_1)$	$Q(S_1, A_2)$		$Q(S_1, A_M)$
S_2	$Q(S_2, A_1)$	$Q(S_2, A_2)$		$Q(S_2, A_M)$
$\vdots$			$\ddots$	$\vdots$
S_N	$Q(S_N, A_1)$	$Q(S_N, A_2)$	...	$Q(S_N, A_M)$

$$Q(s,a) = r(s,a) + \gamma \max_{a} Q(s', a)$$

Policy-based methods



$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\hat{R}(\tau, t)\right]$$

Challenge: many real-world environments are not fully observable

Autonomous driving

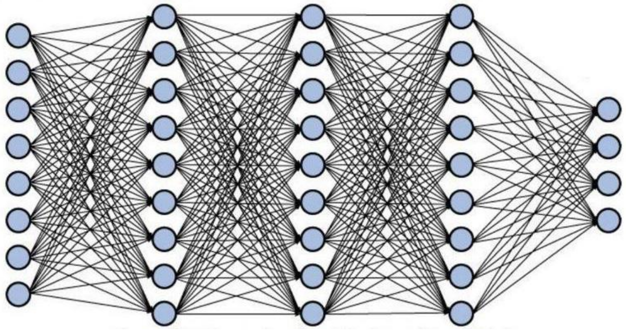
Medical treatment

Manufacturing optimization

An Appeal to Supervised Learning

In supervised learning:

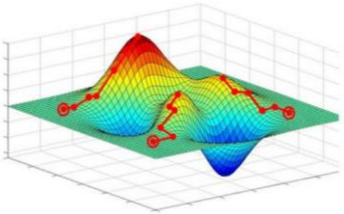
Overparameterized neural networks (**NNs**) trained by **gradient descent** (**GD**) led to a breakthrough



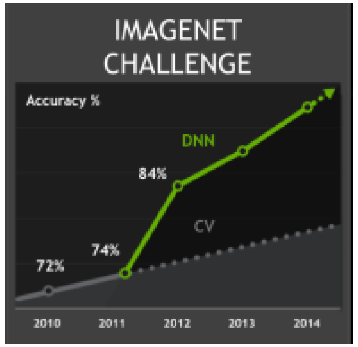
# of learned weights

$\gg$

Training set size



GD training



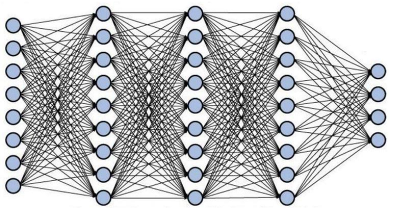
Breakthrough results

Q: Can a similar approach be taken in offline RL?

An Approach to Offline RL Inspired by Supervised Learning

Step 1 Learn Environment Model

Overparameterized NN trained by GD over pre-recorded data



Overparam NN

Time	Action	Observation
t	a_t	o_t
$t+1$	a_{t+1}	o_{t+1}
$t+2$	a_{t+1}	o_{t+1}
...	...	...

Pre-recorded data

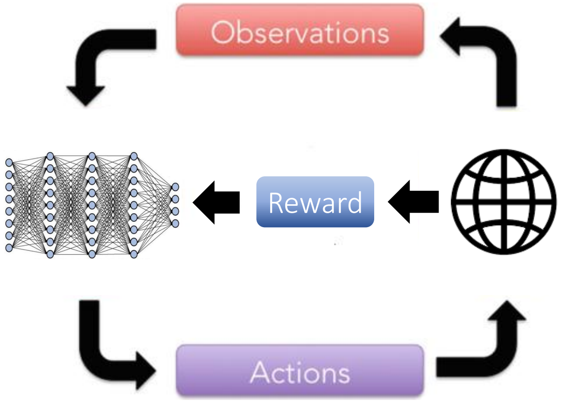
GD training

Learned environment model

Step 2 Learn Agent

Overparameterized NN trained by GD over learned environment model

Overparam NN



Learned environment model

GD training

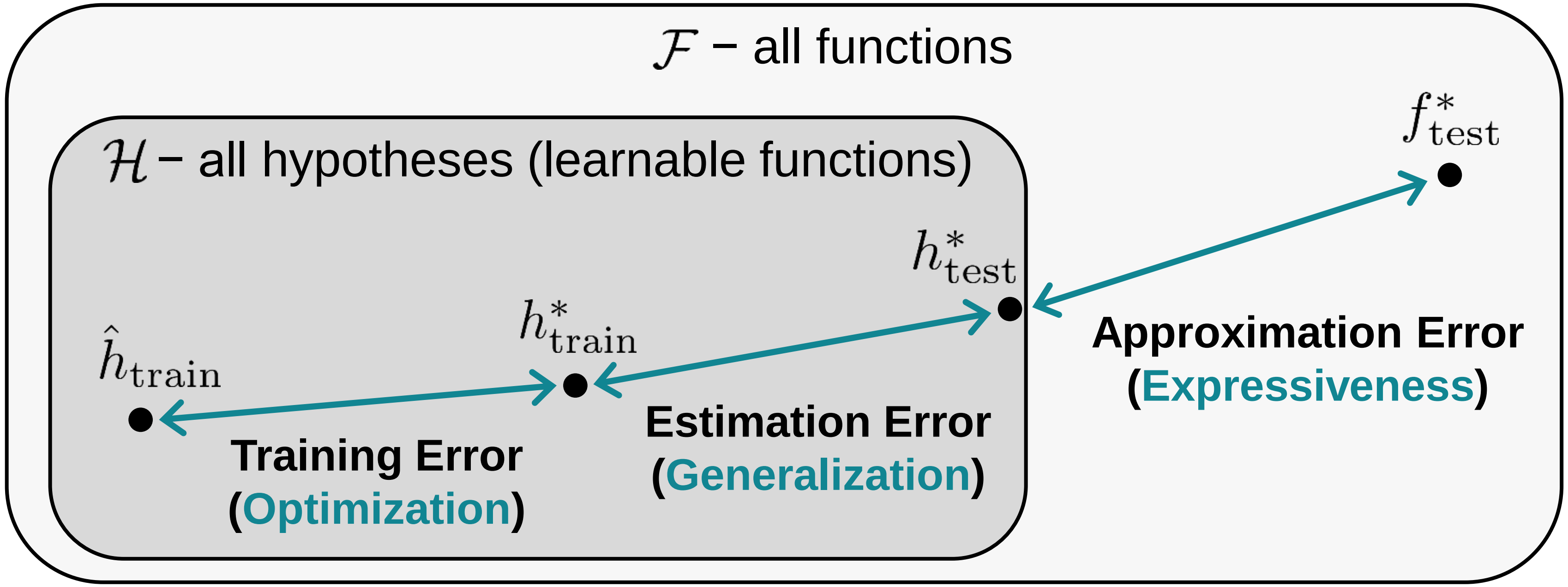
Learned agent

Q: Can this approach work well enough in critical environments?

Medical treatment ?

Manufacturing optimization ?

Three Pillars of Statistical Learning: Expressiveness, Generalization and Optimization



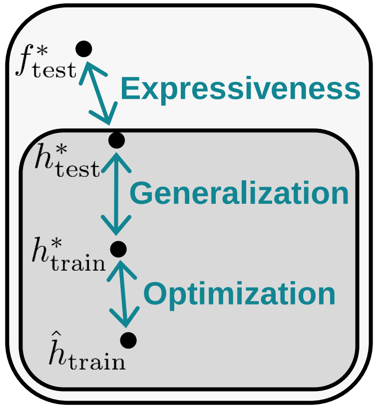
f^*_{test} – optimal function (minimizer of test error over $\mathcal{F}$)

h^*_{test} – optimal hypothesis (minimizer of test error over $\mathcal{H}$)

h^*_{train} – empirically optimal hypothesis (minimizer of train error over $\mathcal{H}$)

$\hat{h}_{\text{train}}$ – returned hypotheis

Three Pillars in Supervised Learning



Various theoretical guarantees:

Expressiveness

[Telgarsky 15']

[Eldan and Shamir 15']

[Cohen et al. 16']

[Raghu et al. '16]

[Levine el al. 17']

[Razin et al. 22']

...

Generalization

[Lampinen and Ganguli 19']

[Arora et al. 19']

[Advani and Saxe 20']

[Chizat and Bach 20']

[Razin and Cohen 20']

[Razin et al. 21]

...

Optimization

[Saxe et al. 14']

[Bartlett el al. 18']

[Arora et al. 18']

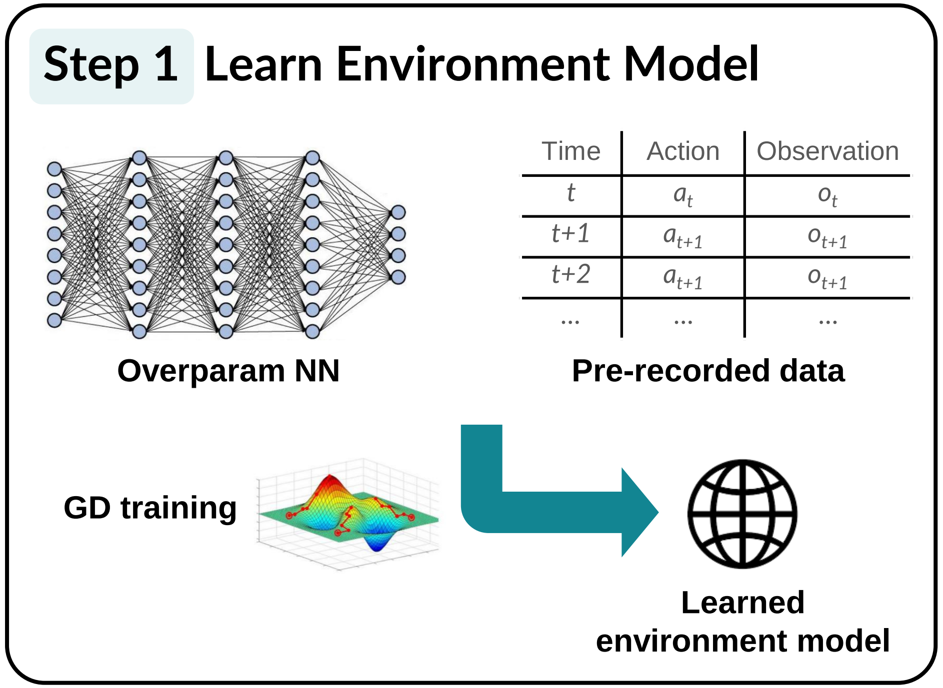
[Arora et al. 19']

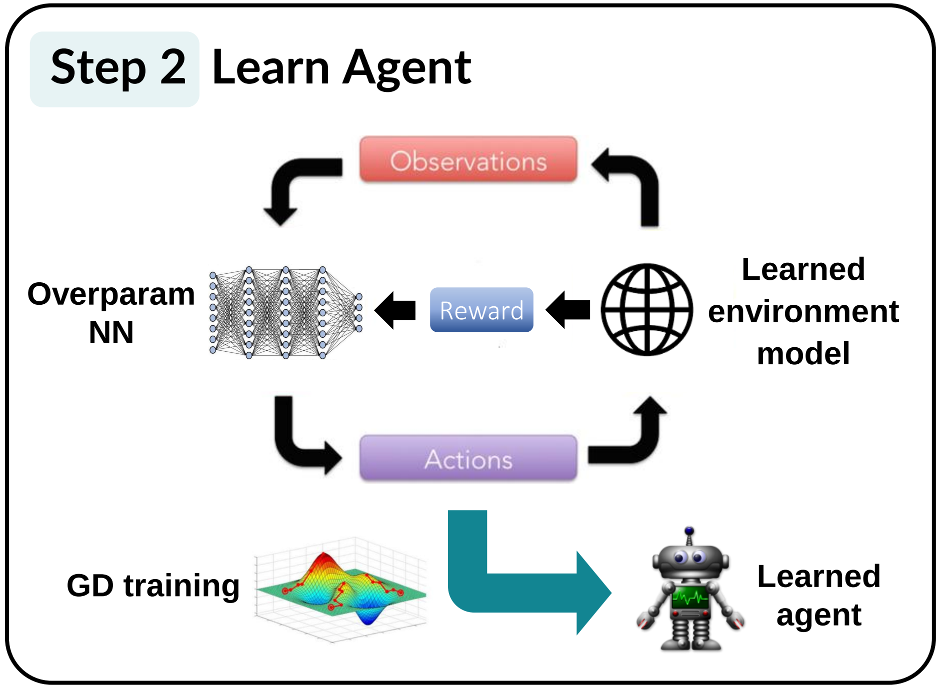
[Ji and Telgarsky 20']

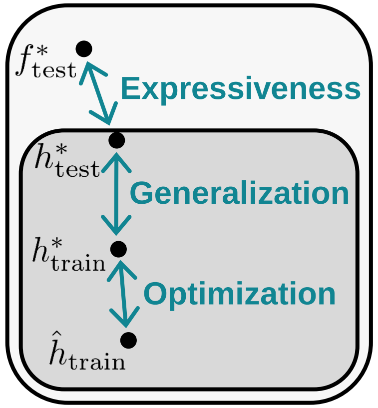
[Elkabetz and Cohen 21']

...

Three Pillars in Offline RL







Significant challenges:

- **Expressiveness**: capacity of NN arch to reach low test error is **highly obscured** by **dynamics**
- **Generalization**: test distribution can **vastly differ** from train (**distribution shift**)
- **Optimization**: train loss is **extremely complex** (GD faces **instability**, **vanishing gradients**, etc.)

Three Pillars in Offline RL (cont.)

Nascent theory gives positive indications:

On the Implicit Bias of Gradient Descent for Temporal Extrapolation

Cohen-Karlik + Ben David + C + Globerson
AISTATS 2022

Learning Low Dimensional State Spaces with Overparameterized Recurrent Neural Nets

Cohen-Karlik + Menuhin-Gruman + Giryes + C + Globerson
ICLR 2023

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

Razin* + Alexander* + Cohen-Karlik + Giryes + Globerson + C
ICML 2024

Provable Benefits of Complex Parameterizations for Structured State Space Models

Ran-Milo + Lumbroso + Cohen-Karlik + Giryes + Globerson + C
NeurIPS 2024

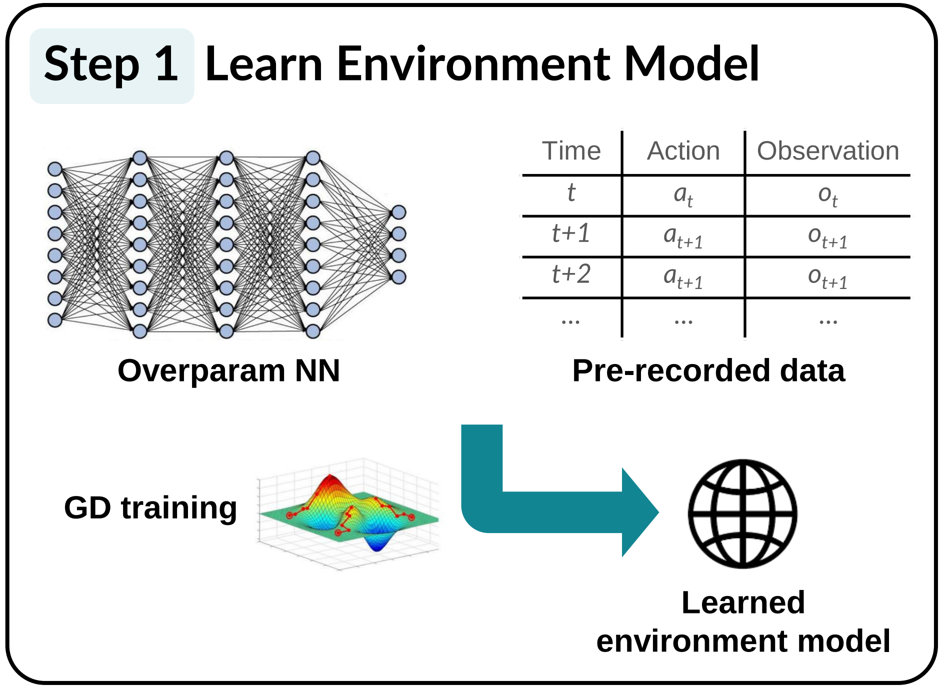
The Implicit Bias of Structured State Space Models Can Be Poisoned with Clean Labels

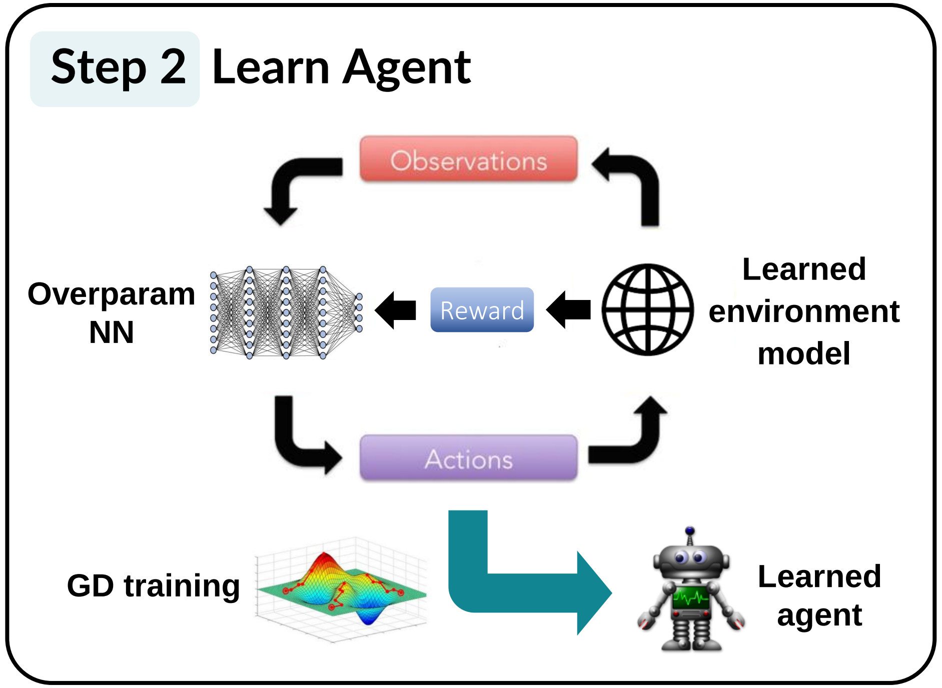
Slutzky* + Alexander* + Razin + C
Under Review 2025

Implicit Bias of Neural Networks for Control: A Tendency for Safety (tentative)

Slutzky + Alexander + Nagel + C
Work in Progress 2025

Offline RL in the Wild?





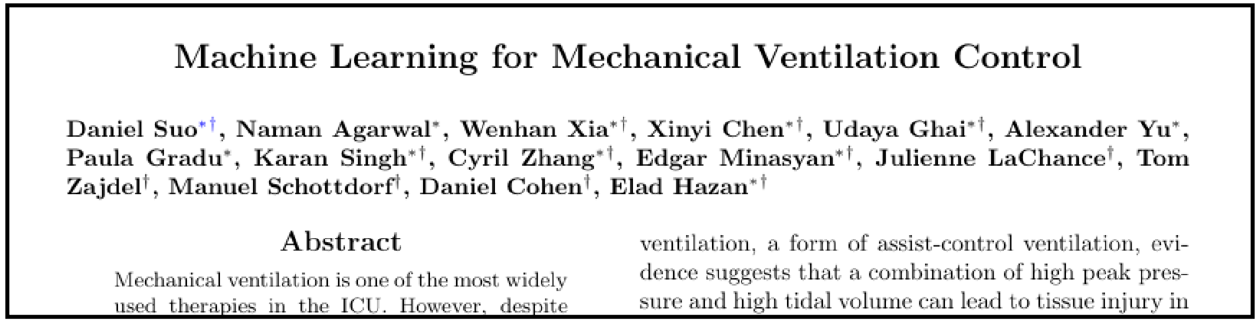
in the wild

Q: Can this approach work well enough in critical environments?

Medical treatment ?

Manufacturing optimization ?

Case Study I: Medical Treatment


Machine Learning for Mechanical Ventilation Control

Daniel Suo[*,†], Naman Agarwal[*], Wenhan Xia[*,†], Xinyi Chen[*,†], Udaya Ghai[*,†], Alexander Yu[*], Paula Gradu[*], Karan Singh[*,†], Cyril Zhang[*,†], Edgar Minasyan[*,†], Julienne LaChance[†], Tom Zajdel[†], Manuel Schottdorf[†], Daniel Cohen[†], Elad Hazan[*,†]

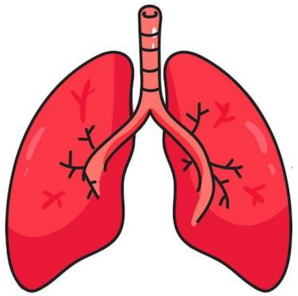
Abstract

Mechanical ventilation is one of the most widely used therapies in the ICU. However, despite

ventilation, a form of assist-control ventilation, evidence suggests that a combination of high peak pressure and high tidal volume can lead to tissue injury in

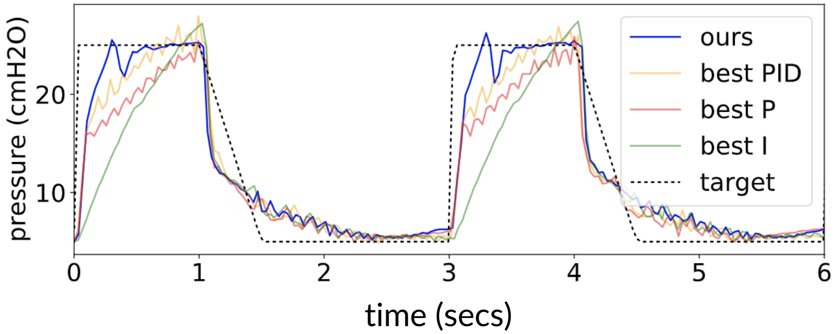
Step 1 Learn Environment Model

Use pre-recorded data for learning an NN lungs model



Step 2 Learn Agent

Use learned lungs model for learning an NN mechanical ventilator controller



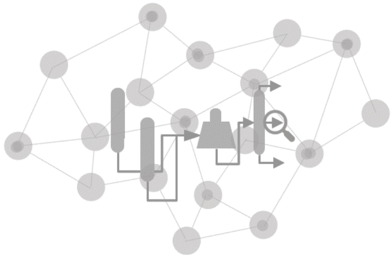
Critical environment? ✓

In the wild? ✗

Case Study II: Manufacturing Optimization

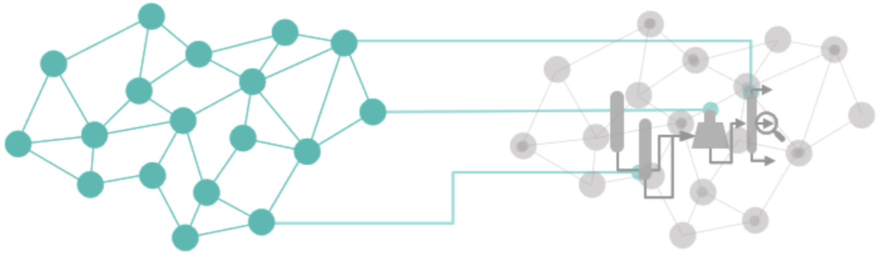
Step 1 Learn Environment Model

Use pre-recorded data for learning an NN plant model



Step 2 Learn Agent

Use learned plant model for learning an NN controller



Critical environment? ✓

In the wild? ✓

Case Study II: Manufacturing Optimization (cont.)

70+ Applications

30+ Process Plants

6+ Years of Model Engagement

15-30% Reduced Natural Gas Usage

1-3% Yield Improvement

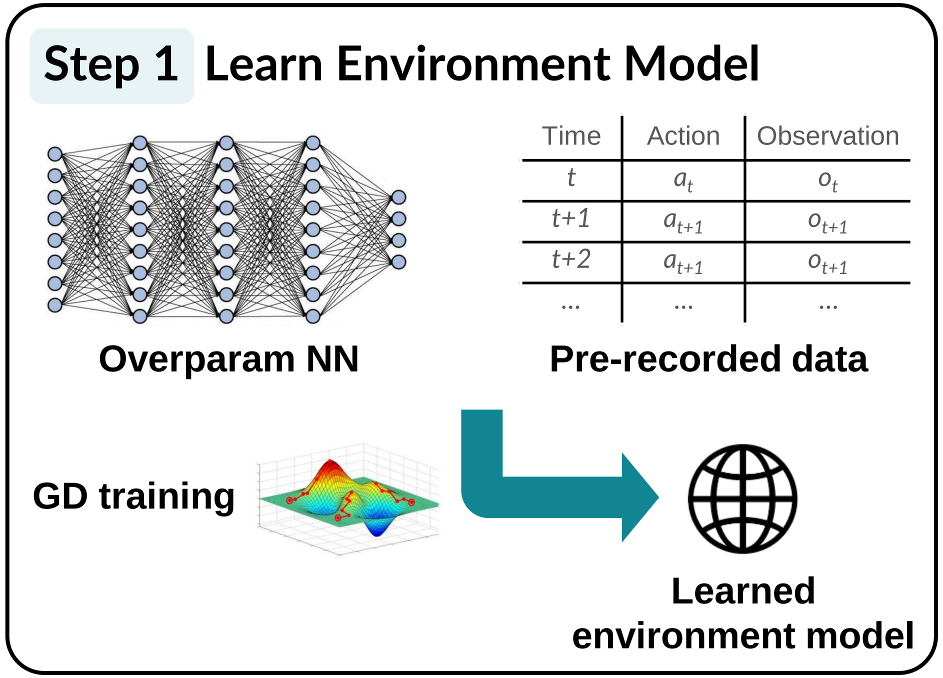
Optimized by IMUBIT

Chevron
ExxonMobil
BAZAN GROUP
Hunt Refining Company
Big West Oil LLC
Oxbow
Flint Hills resources
HF Sinclair
preem
INVISTA
Monroe Energy
ASH GROVE A CRH COMPANY
CITGO
Eagle Materials

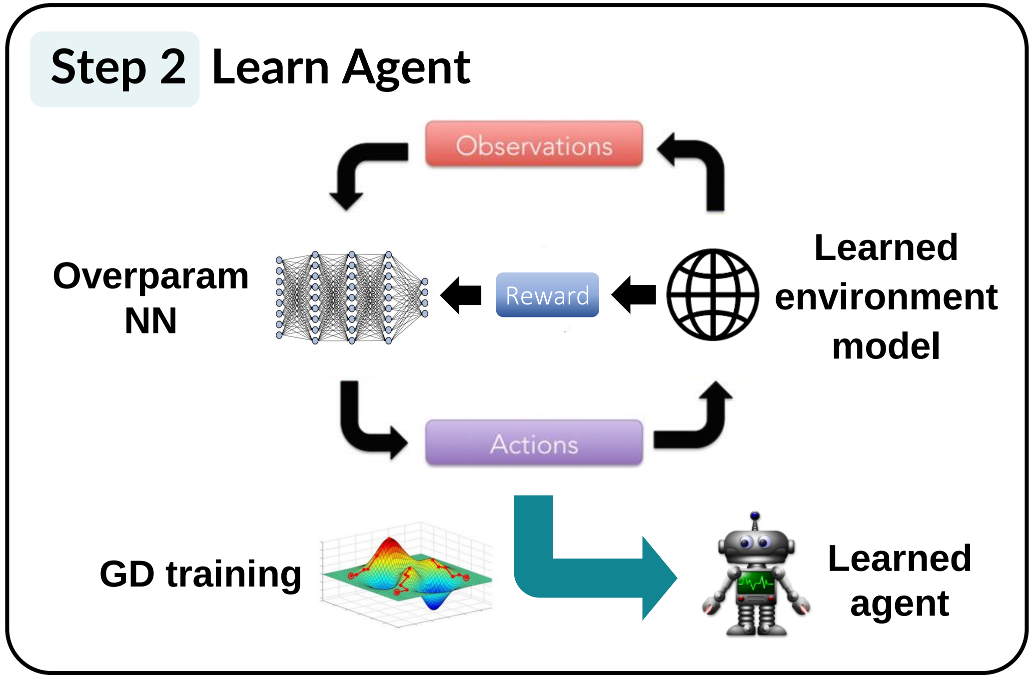
Conclusion

In critical applications, **trial & error is prohibitively costly/dangerous** ➡ **RL** must be **offline**

Supervised learning success of **overparameterized NNs** trained by **GD** inspires offline RL approach :



Time	Action	Observation
t	a_t	o_t
$t+1$	a_{t+1}	o_{t+1}
$t+2$	a_{t+1}	o_{t+1}
...	...	...



Nascent **theory supports** the approach

Approach **successfully demonstrated** in critical application **in the wild**!

Perspective

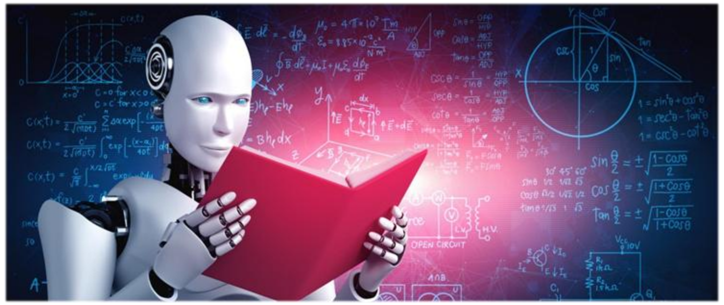
Practical progress in **AI is currently driven by trial & error**

Less suitable for critical applications

Medical treatment

Manufacturing optimization

For AI to proliferate in critical applications, **theory may be necessary**



Thank You!

Theoretical research supported by:
ERC Starting Grant (101164614); Apple; Google; Meta; Yandex; ISF Grant (1780/21); Blavatnik Foundation; Adelis Foundation; Tel Aviv University Center for AI and Data Science; and Amnon and Anat Shashua