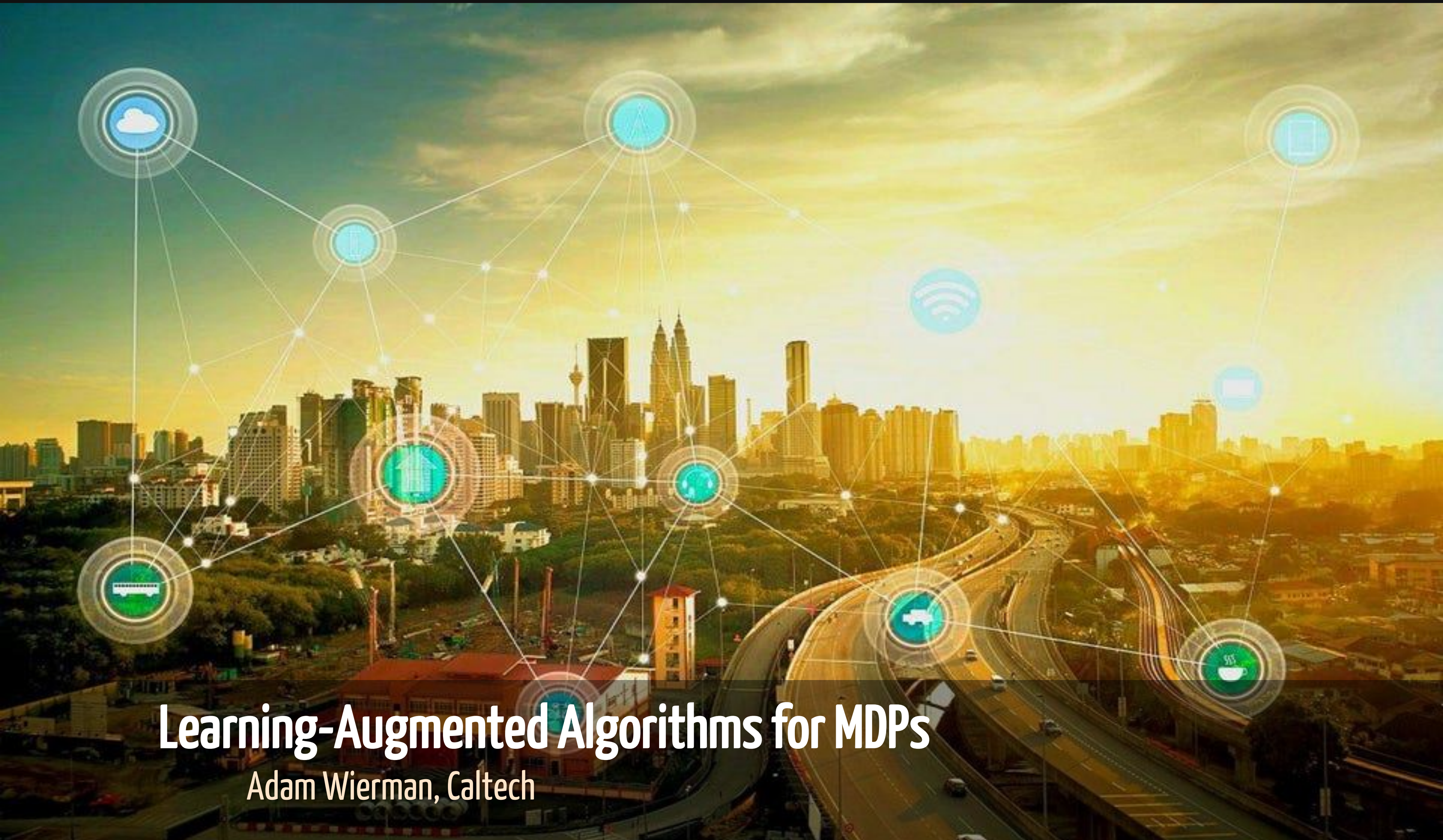


Learning-Augmented Algorithms for MDPs

Adam Wierman, Caltech

Tongxin Li

Nico Christianson

Tinashe Handina

Chris Yeh

Yiheng Lin

amazon

vmware

NSF

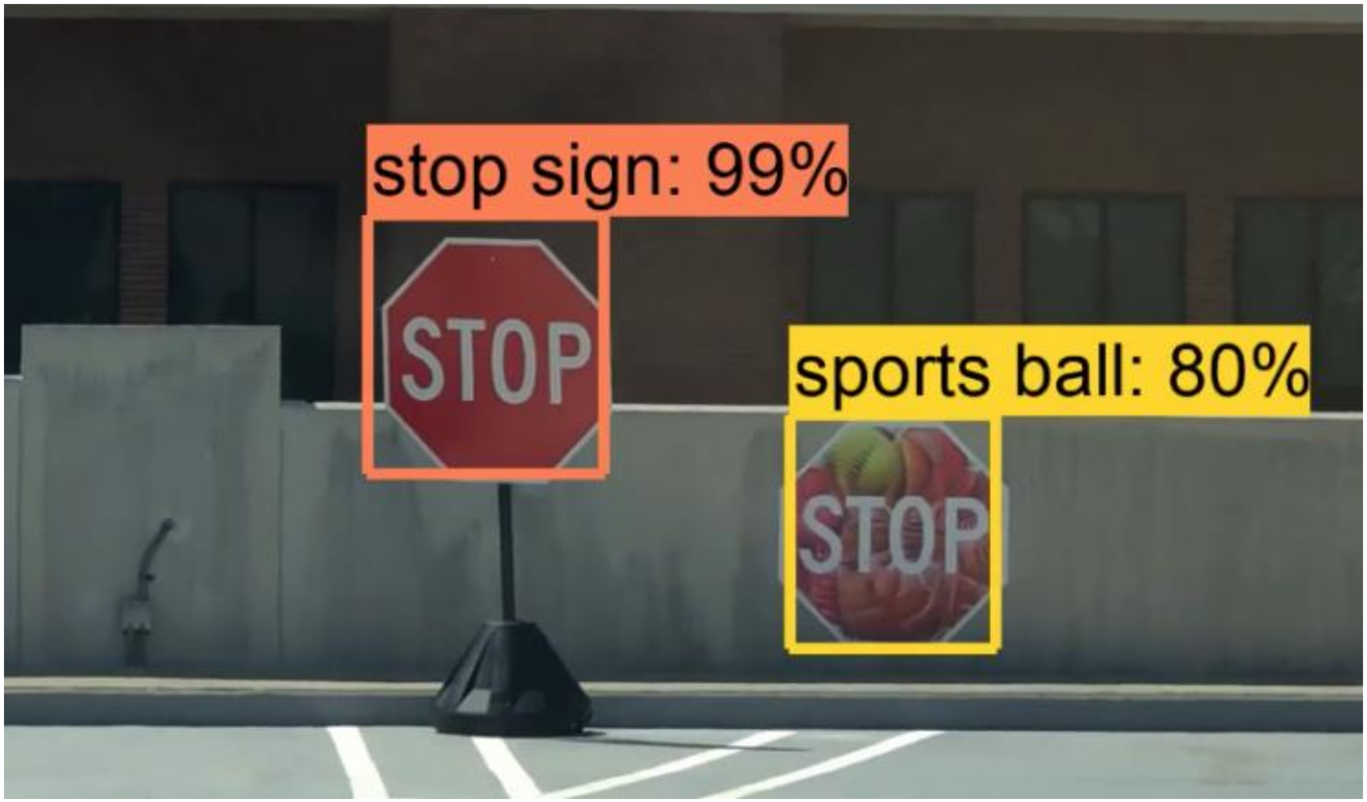
RESNICKINSTITUTE
science + energy + sustainability
CALIFORNIA INSTITUTE OF TECHNOLOGY

BEYOND LIMITS

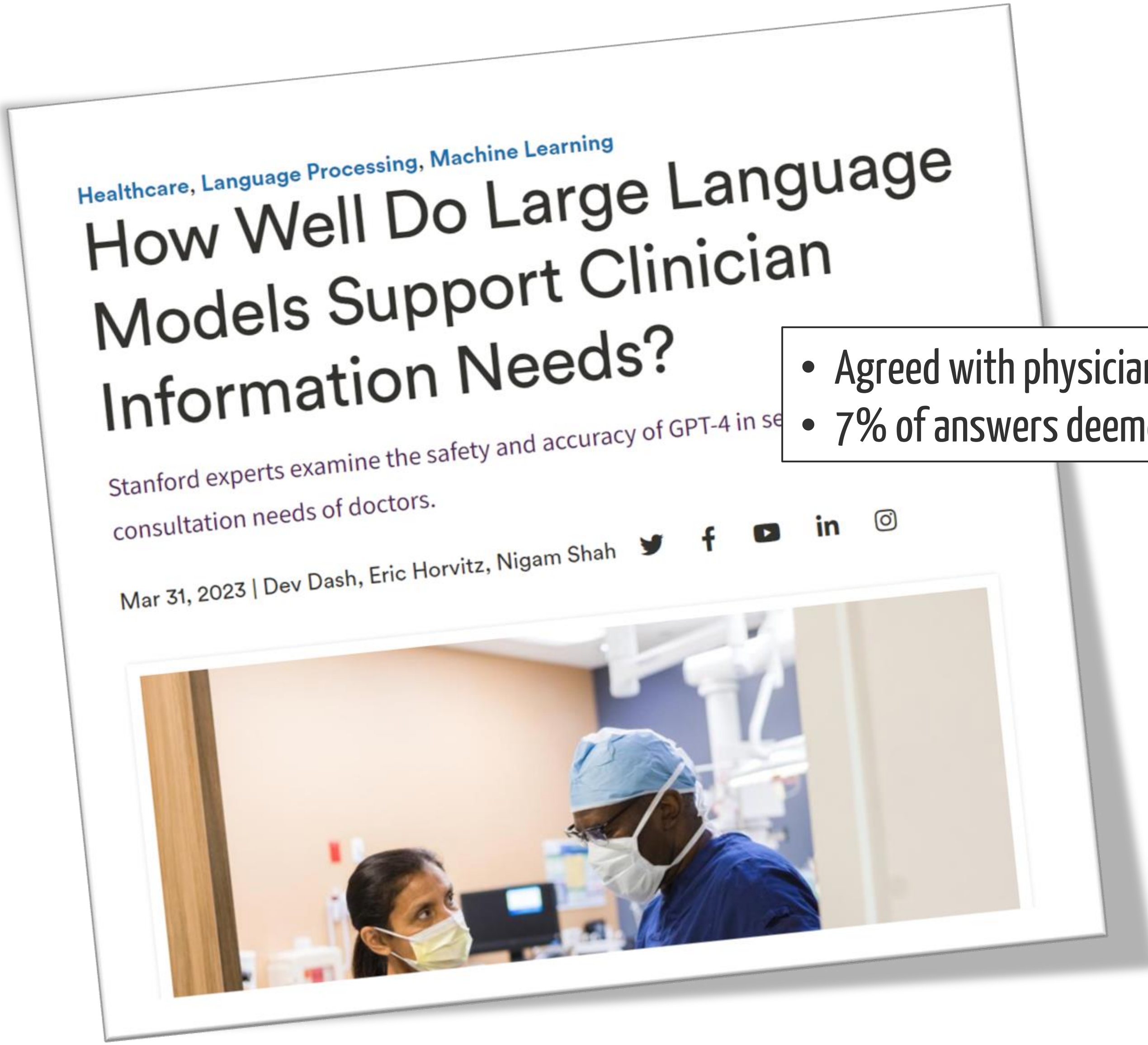
MIT Lincoln Laboratory


AI can potentially give us more resilient, sustainable, and autonomous energy systems...
Are AI tools ready?

[Pei et al 2017]



[Pei et al 2017]



Healthcare, Language Processing, Machine Learning

How Well Do Large Language Models Support Clinician Information Needs?

Stanford experts examine the safety and accuracy of GPT-4 in se

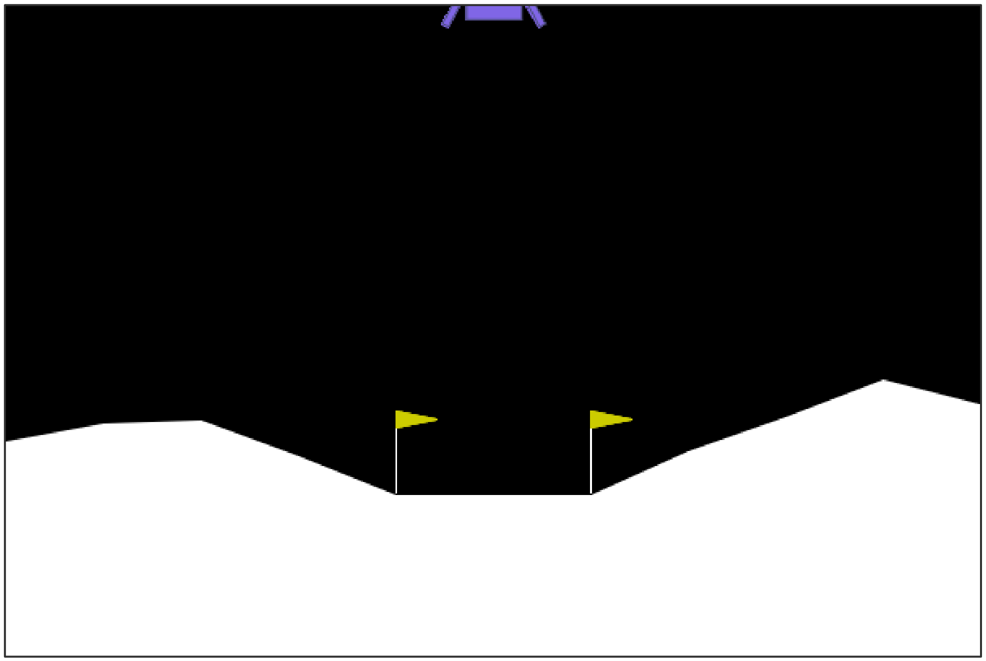
consultation needs of doctors.

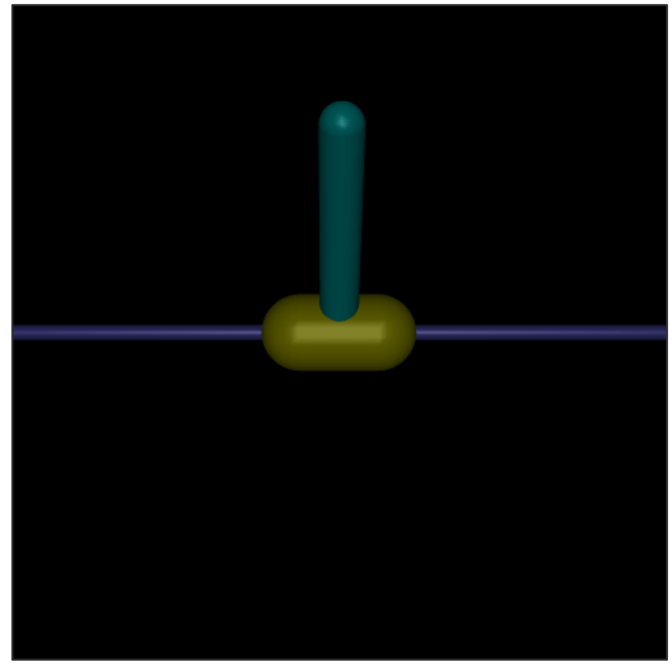
Mar 31, 2023 | Dev Dash, Eric Horvitz, Nigam Shah

[Lee et al 2023]

- Agreed with physicians 41% of time
- 7% of answers deemed harmful by physicians

Most algorithms are benchmarked on toy environments





Most algorithms are benchmarked on toy environments

Energy systems must deal with

- physical constraints
- distribution shifts
- distributed, multi-agent control

Introducing Caltech/UCSD SustainGym

Five environments (so far):

1. Adaptive EV charging (local and multi-location)
2. Grid-scale battery storage management for price arbitrage
3. Data center dynamic capacity management (VCCs, local and global)
4. Cogeneration management of a plant producing steam and electricity
5. Smart building management to meet temperature requirements

Caltech/UCSD Collaboration led by Christopher Yeh with co-authors:
Victor Li, Rajeev Datta, Julio Arroyo, Nicolas Christianson, Chi Zhang, Yize Chen, Mohammad Hosseini, Azarang Golmohammadi, Yuanyuan Shi, Yisong Yue

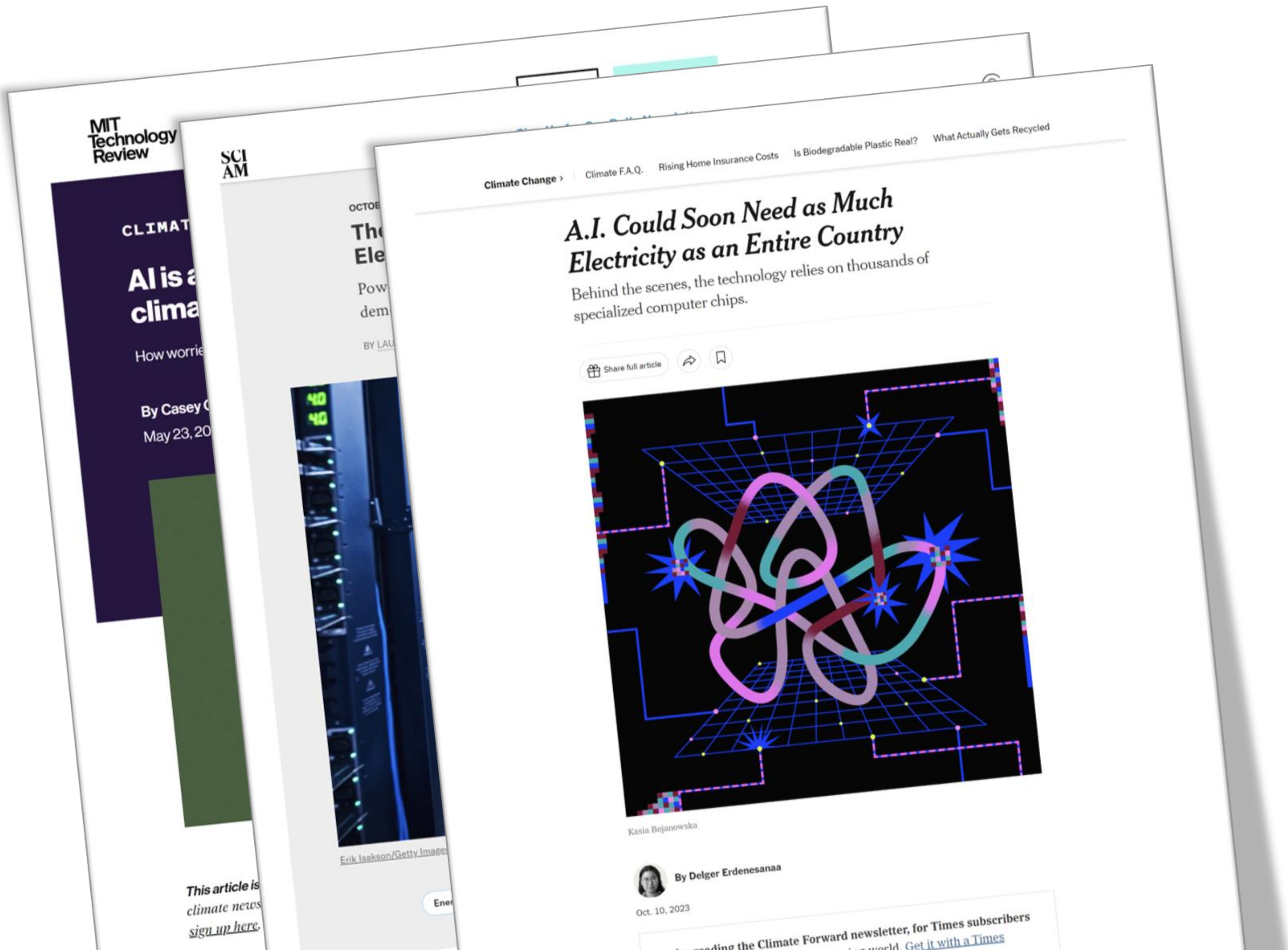
Introducing Caltech/UCSD SustainGym

Environments feature

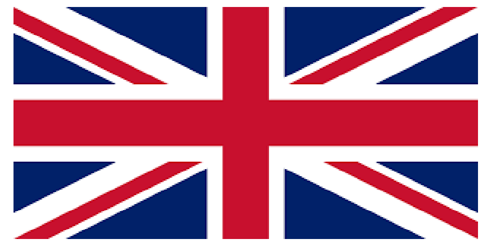
- Focus on marginal carbon emissions
- Real-world data and models from industry partners
- Distribution shifts in demand & environmental parameters
- Physical constraints
- Mix of discrete and continuous actions
- Multi-agent settings

An example: Carbon-first Data Centers

AI’s environmental footprint is enormous

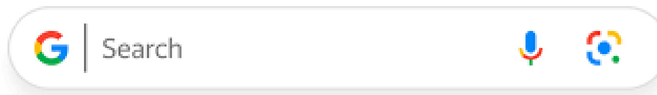


>



Data centers use 50% more electricity than the UK

Data centers make up >20% electricity use in Ireland

GPT-3.5 ~ 2x 

/ query

GPT-3.5 ~ 33x classic AI /query

OpenAI ChatGPT **4.0** ~100 x GPT-3.5 to train

[Patel & Ahmed]
[Luccioni, Jernite, Strubell]

...and utilities are just giving up!

HOME > NEWS > NORTH AMERICA

Dominion Energy admits it can't meet data center power demands in Virginia

The high-voltage lines simply can't handle more power, says the utility

July 29, 2022 By: Peter Judge Have your say

North American utility Dominion Energy says it may not be able to meet demands for power in Ashburn, Northern Virginia, delaying building projects in the world's fastest-growing data center hub by many years.

Dominion has told customers that it has power supplies, but can no longer guarantee to deliver the quantity of electricity customers want via overhead powerlines. If these warnings prove true, this could stall projects with billions invested, and Loudoun County's tax revenue would take a severe hit if the hub of data centers in Ashburn stalls. For now, local authorities and industry bodies are struggling to understand the sudden warning from Dominion.

Dominion supplies electricity in Virginia, North Carolina, and South Carolina, as well as natural gas to parts of the US. In the data center-rich counties of Loudoun, Prince William, and Fauquier, most of the electricity is carried by overhead powerlines marching along roads - a delivery method that has led to protests.

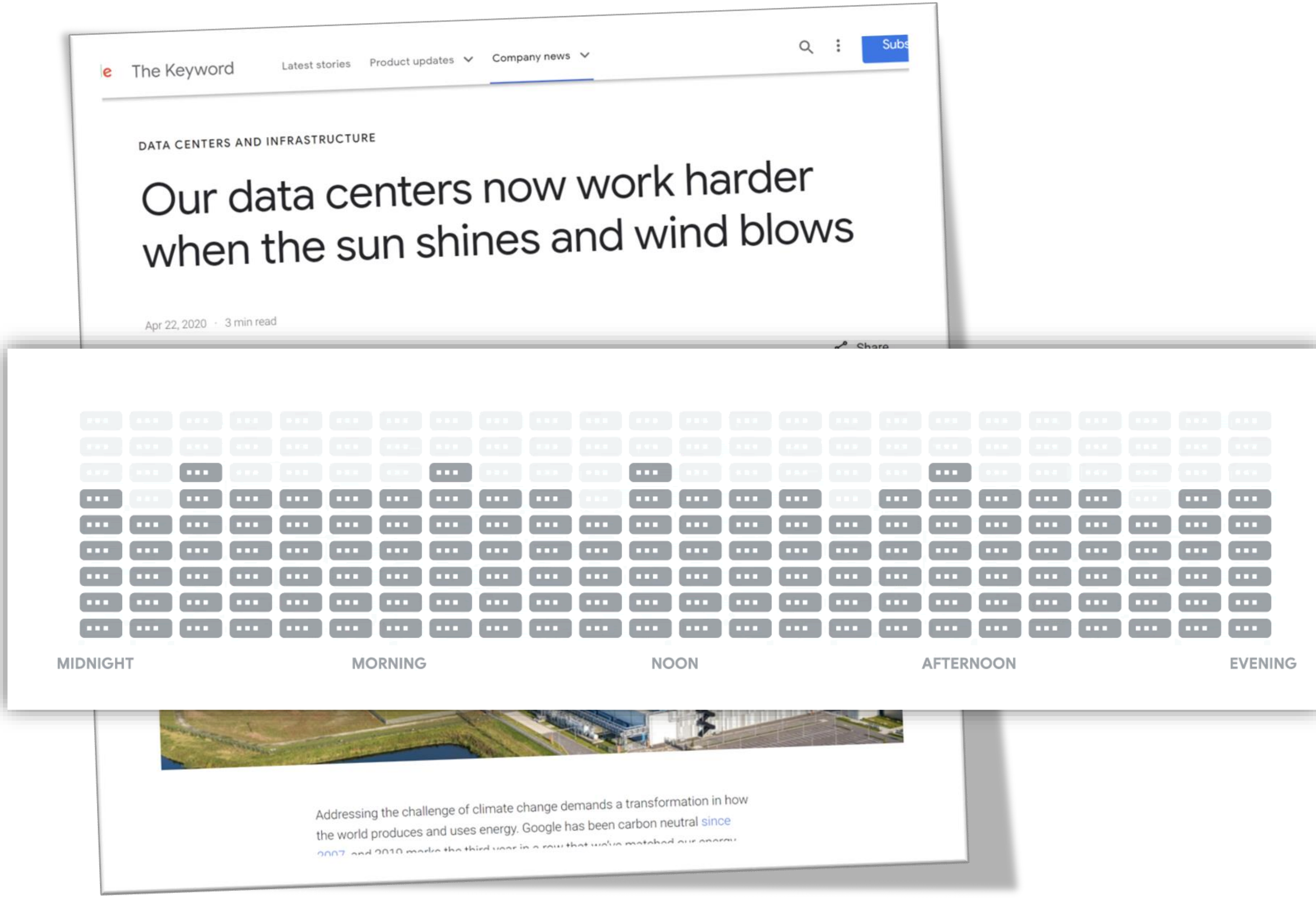
Loudoun County has 26 million square feet of data center space, with 5 million more in development and many more projects planned. Data center equipment taxes provide one-third of the County's tax income, but has

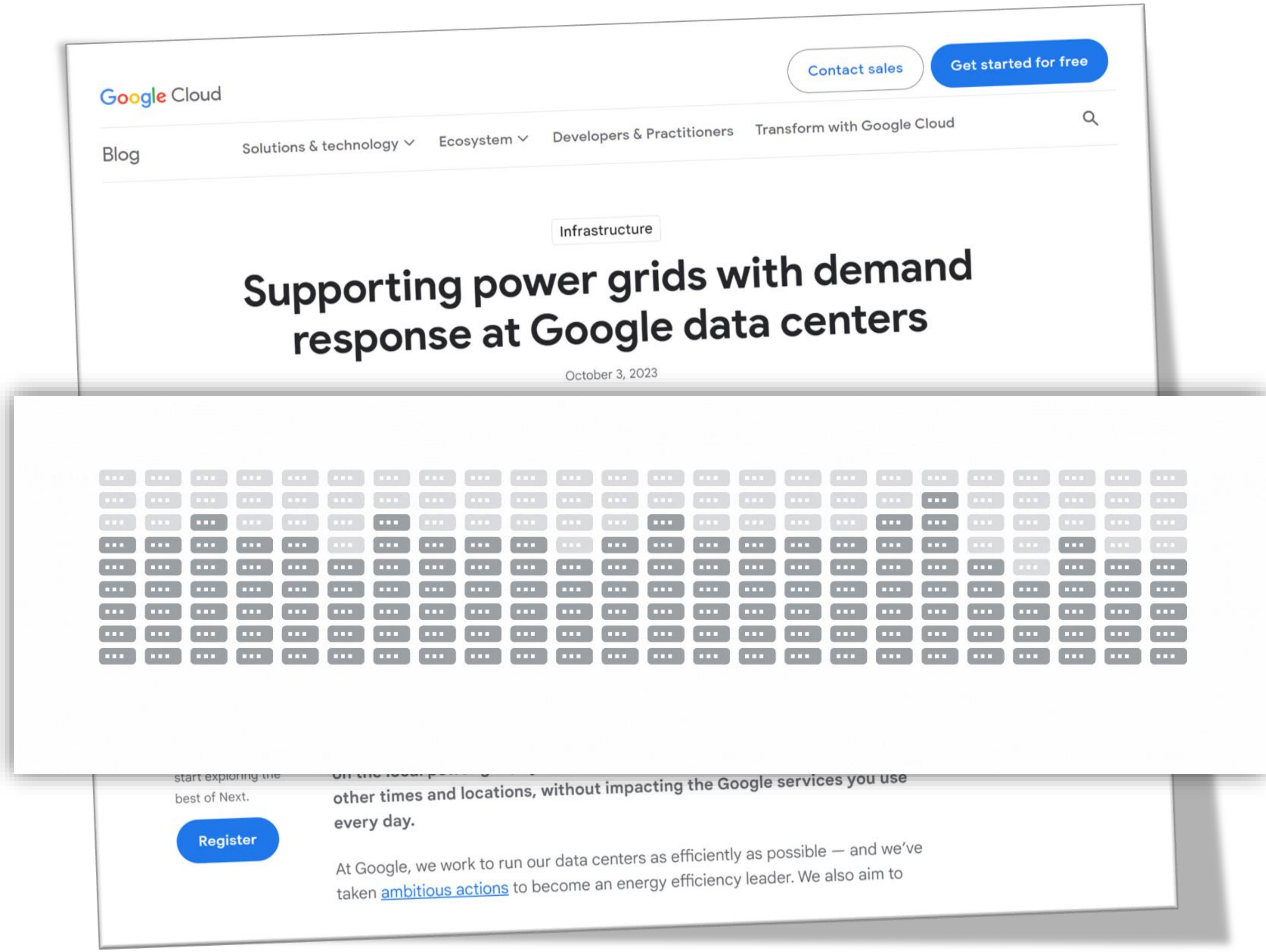
Infrastructure > Data Centres

(Image credit: Getty Images)

Data centres around the world are facing pressure to become more sustainable

Data centers <u>must</u> be adaptive & grid-integrated


The Keyword
Latest stories
Product updates
Company news
Subs
DATA CENTERS AND INFRASTRUCTURE
Our data centers now work harder when the sun shines and wind blows
Apr 22, 2020 · 3 min read
Share
MIDNIGHT
MORNING
NOON
AFTERNOON
EVENING
Addressing the challenge of climate change demands a transformation in how the world produces and uses energy. Google has been carbon neutral since



Google Cloud

Contact sales

Get started for free

Blog

Solutions & technology

Ecosystem

Developers & Practitioners

Transform with Google Cloud

Infrastructure

Supporting power grids with demand response at Google data centers

October 3, 2023

start exploring the best of Next.

Register

other times and locations, without impacting the Google services you use every day.

At Google, we work to run our data centers as efficiently as possible — and we've taken ambitious actions to become an energy efficiency leader. We also aim to


DeepMind

Blog

BLOG POST
RESEARCH

DeepMind AI Reduces Google Data Centre Cooling Bill by 40%

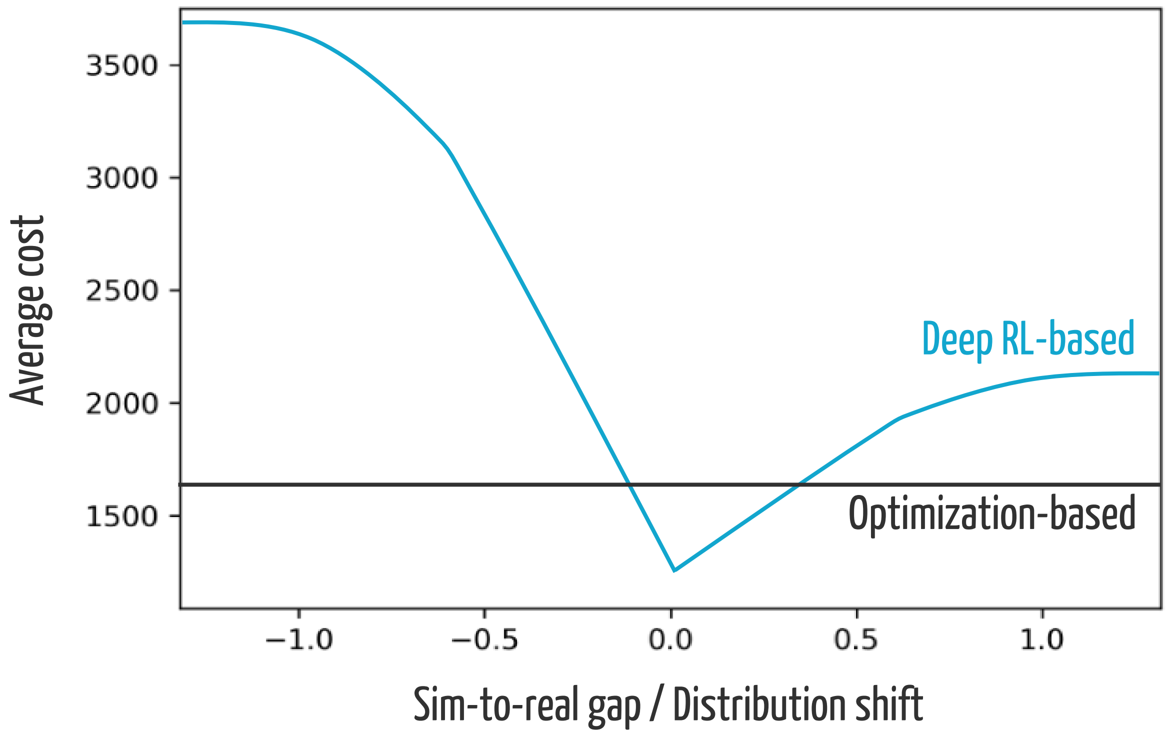
rom smartphone assistants to image recognition and translation, machine learning already helps us in our veryday lives. But it can also help us to tackle some of the world's most challenging physical problems – uch as energy consumption. Large-scale commercial and industrial systems like data centres consume a ɔt of energy, and while much has been done to stem the growth of energy use, there remains a lot more to lo given the world's increasing need for computing power.

ɛducing energy usage has been a major focus for us over the past 10 years: we have built our own super-fficient servers at Google, invented more efficient ways to cool our data centres and invested heavily in reen energy sources, with the goal of being powered 100 percent by renewable energy. Compared to five ears ago, we now get around 3.5 times the computing power out of the same amount of energy, and we ontinue to make many improvements each year.

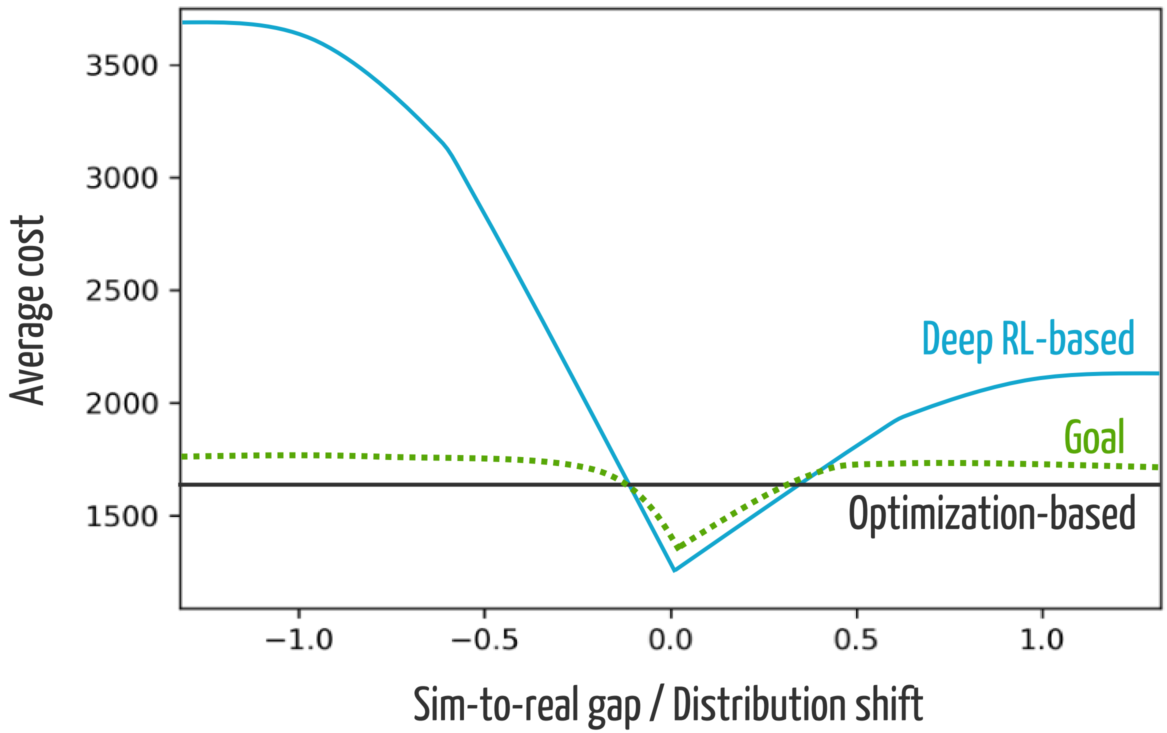
But ML/AI tools are not in use in practice...

<u>Can't afford to "fail at scale"</u>

Example: Capacity provisioning with on-site solar & storage

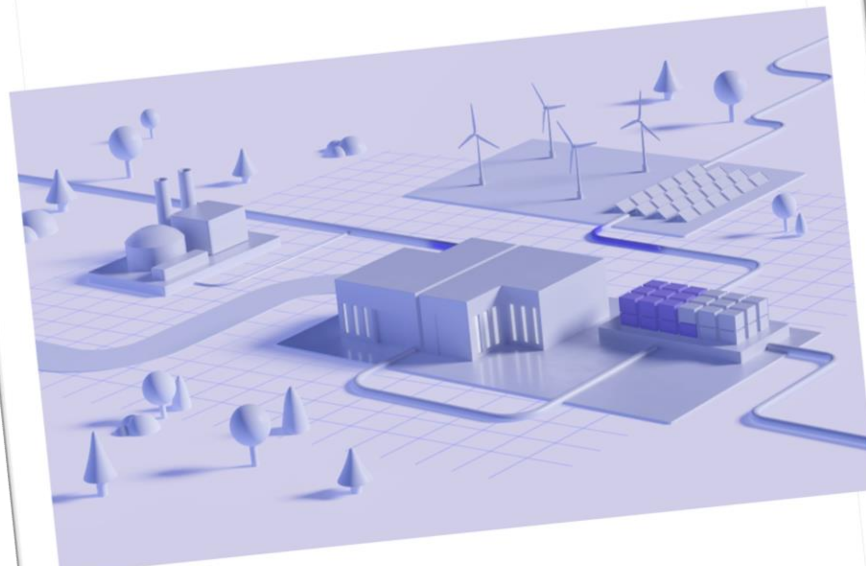


Example: Capacity provisioning with on-site solar & storage



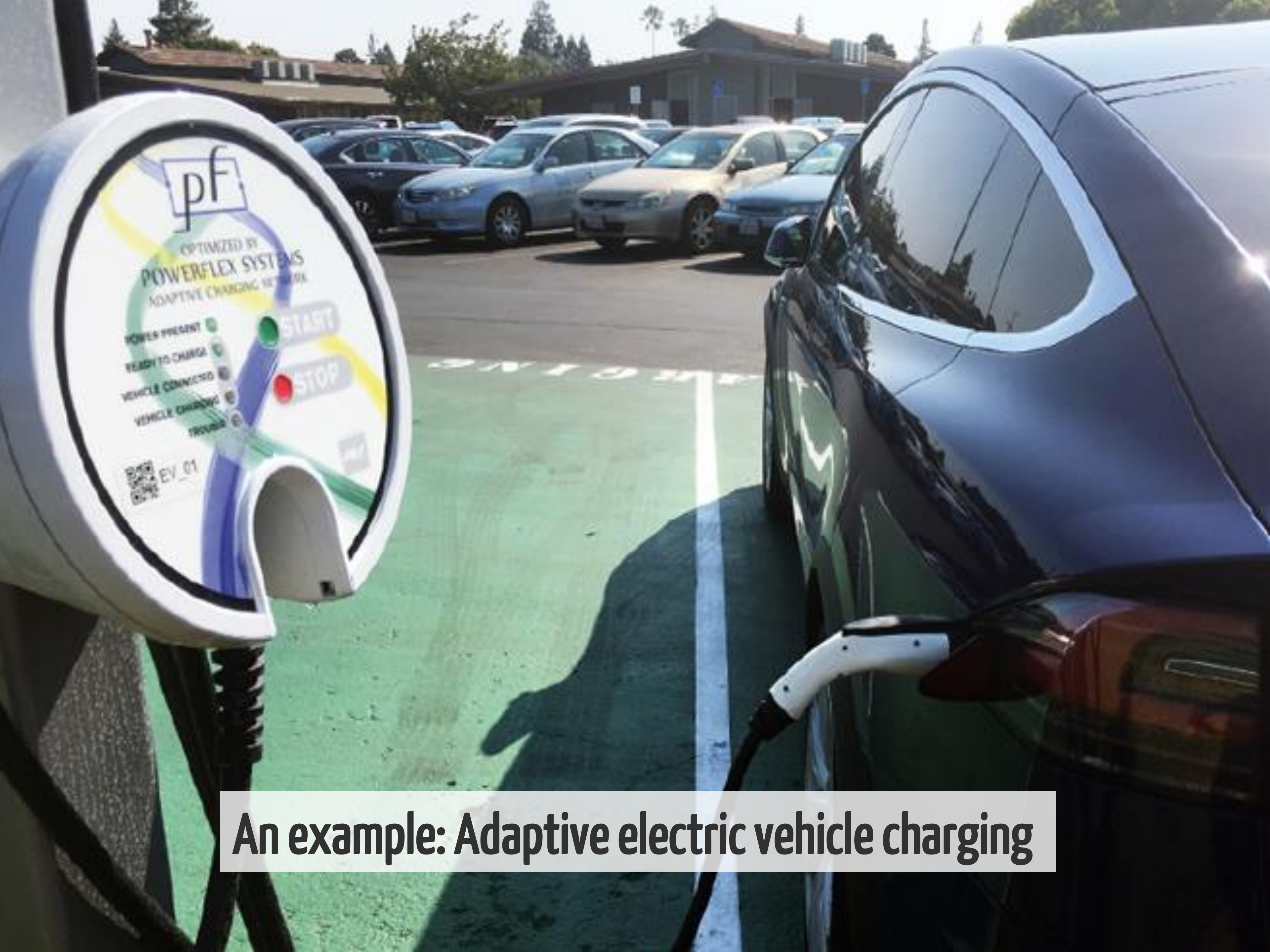
verrus

The world's most flexible and sustainable data centers.

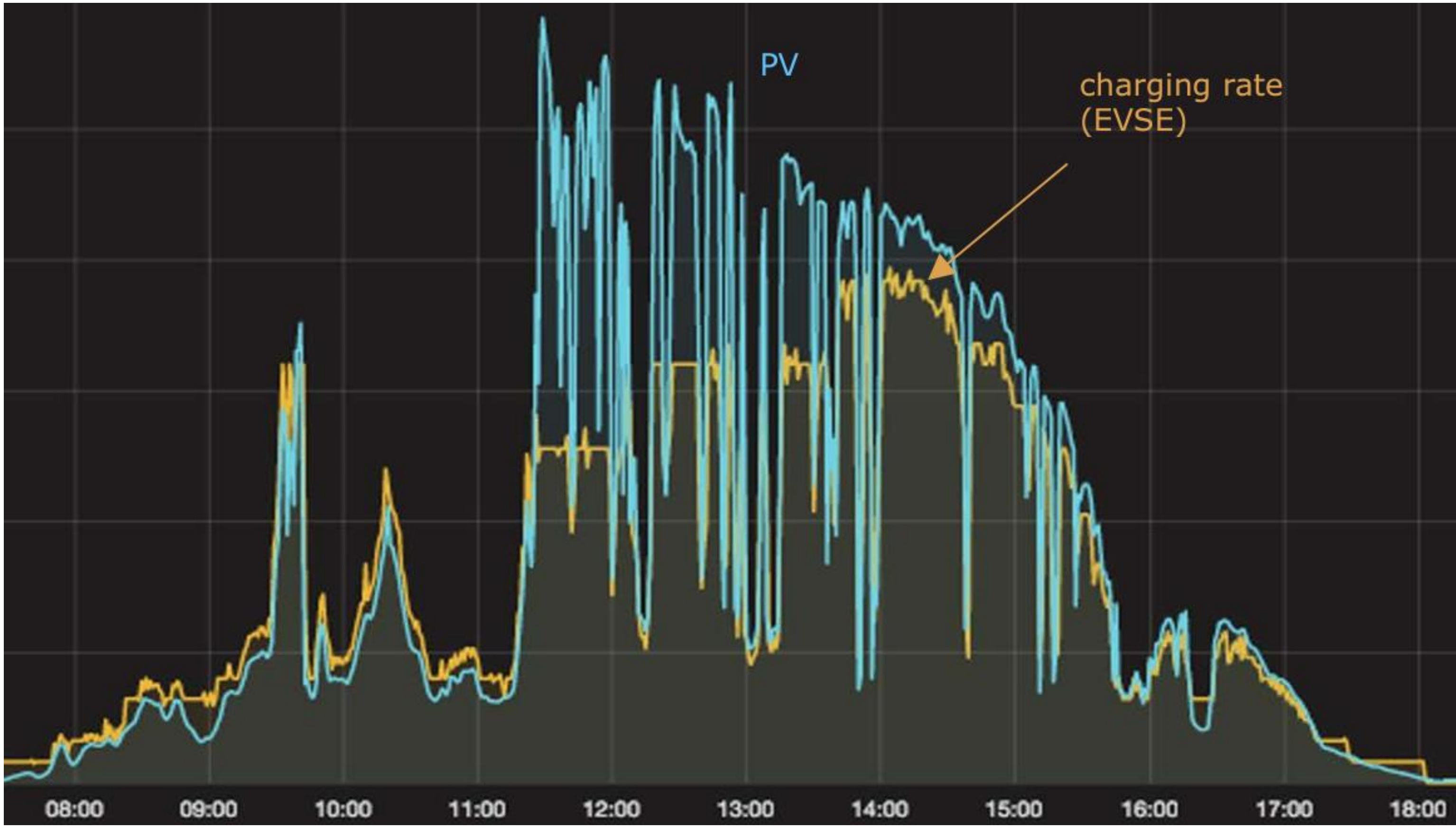


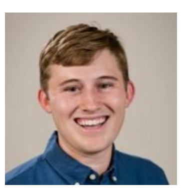
Most data centers will be obsolete within years, unable to

Pause motion



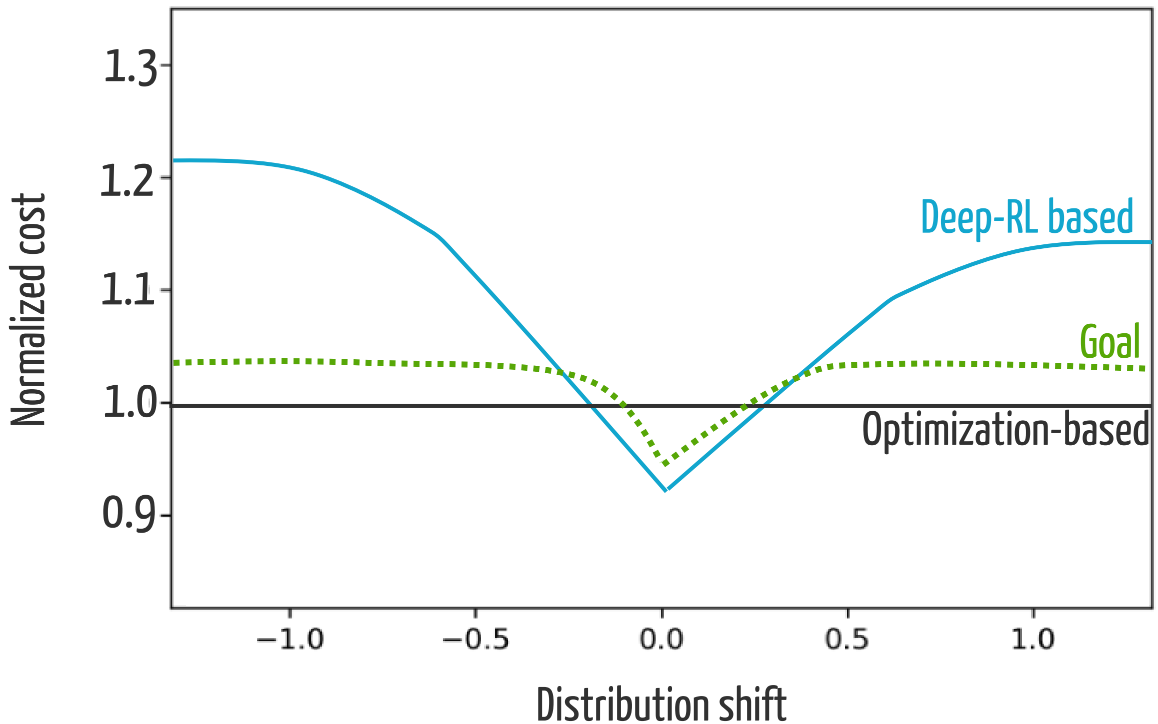
An example: Adaptive electric vehicle charging

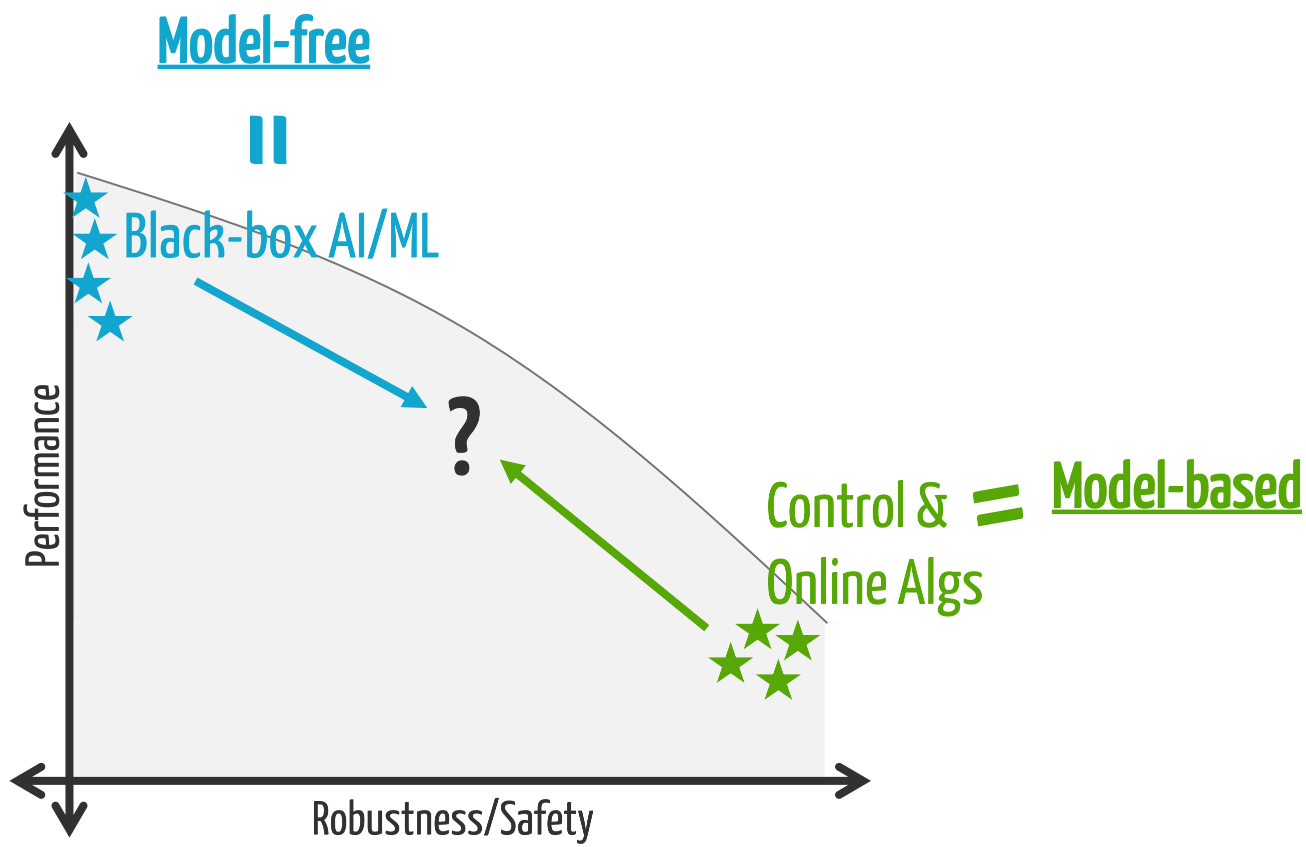


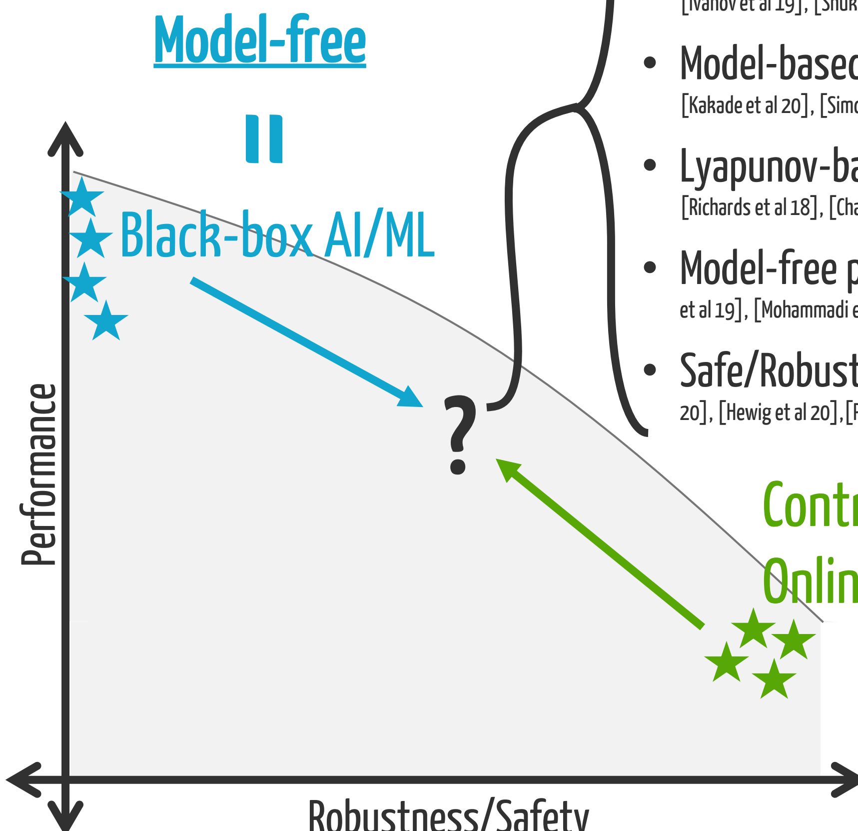


powerflex
EDF Renewables

Example: EV charging at JPL with co-located solar generation

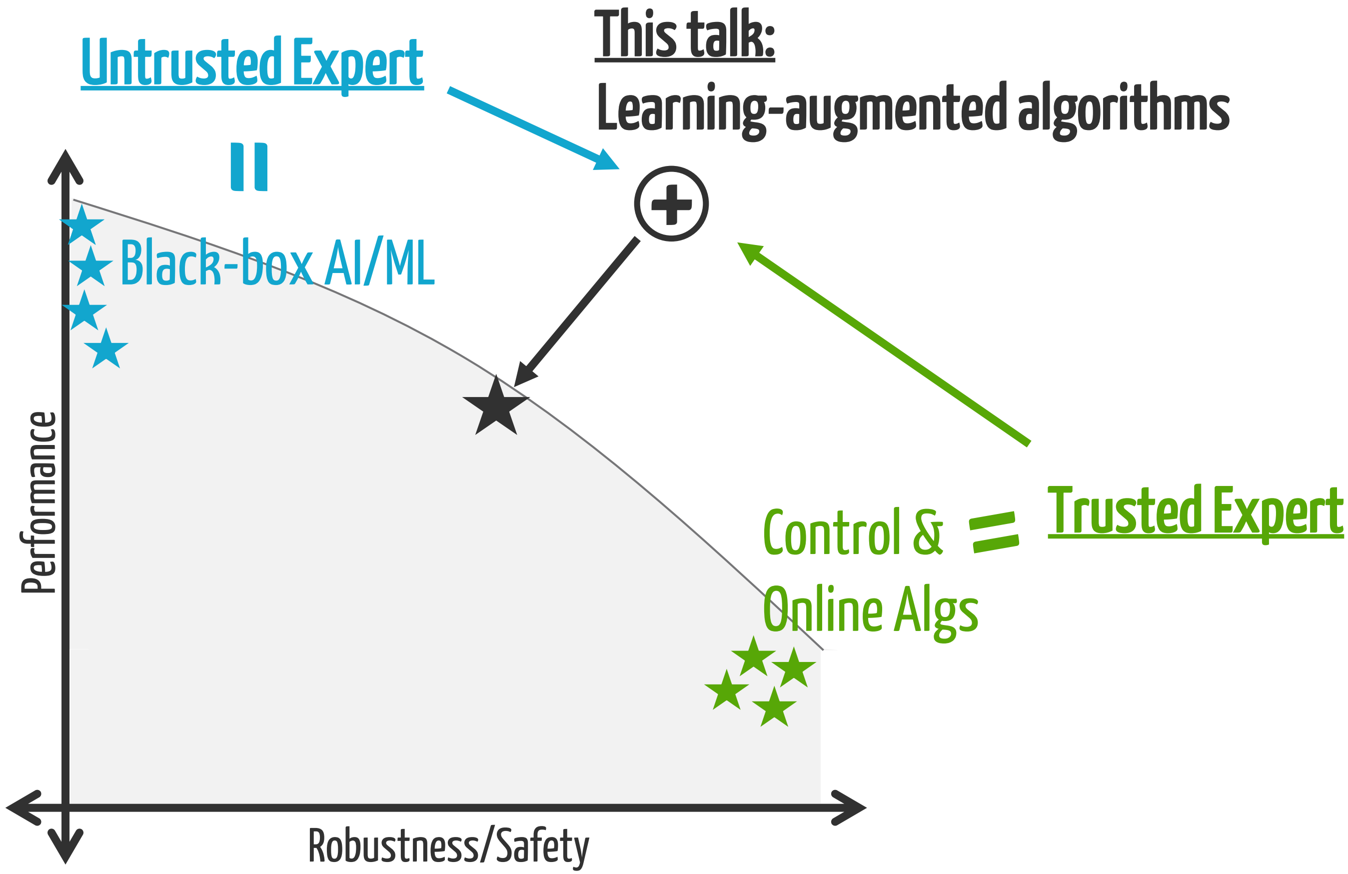



Model-free
=
Black-box AI/ML
?
Control &
Online Algs
=
Model-based
Performance
Robustness/Safety



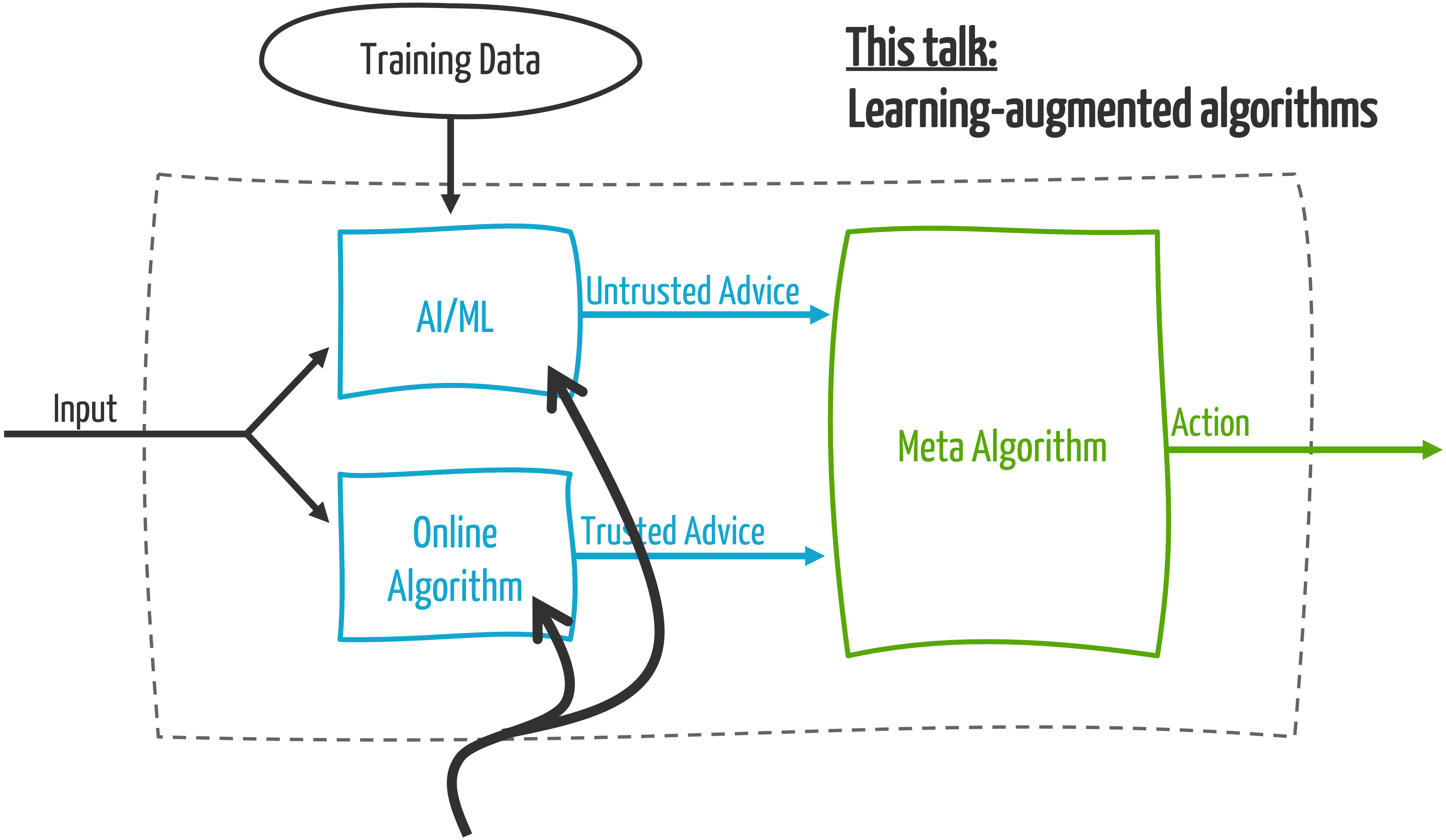
- **Robust/adversarial training** [Ehlers 17], [Katz et al 17], [Maganti 17], [Dutta et al 18], [Tjeng et al 18], [Gehr 18,], [Salman 19], [Bak 20], [Fazlyab 19, 22], [Robey et al 21,22], [Eastwood et al 23]. ...
- **Post-training verification** [Huang et al 17], [Kuper et al 18], [Ivanov et al 19], [Shukla et al 19], [Matni et al 20], [Fazlyab et al 22], ...
- **Model-based RL in dynamical systems** [Recht 19], [Kakade et al 20], [Simchowitz & Foster 20], [Lale et al 21], ...
- **Lyapunov-based policy learning** [Chow et al 18], [Richards et al 18], [Chang et al 19], [Jin et al 20], [Shi et al 21], ...
- **Model-free policy search** [Fazel et al 18], [Malik et al 18], [Bu et al 19], [Mohammadi et al 19], [Li et al 19], [Qu et al 20], ...
- **Safe/Robust RL** [Garcia & Fernandez 15], [Fisac et al 19], [Taylor et al 20], [Hewig et al 20],[Panaganti at al 21, 22], [Shi et al 21, 22], ...

Control & Online Algs = Model-based



This talk:
Learning-augmented algorithms
Untrusted Expert
=
Black-box AI/ML
Control &
Online Algs
=
Trusted Expert
Performance
Robustness/Safety

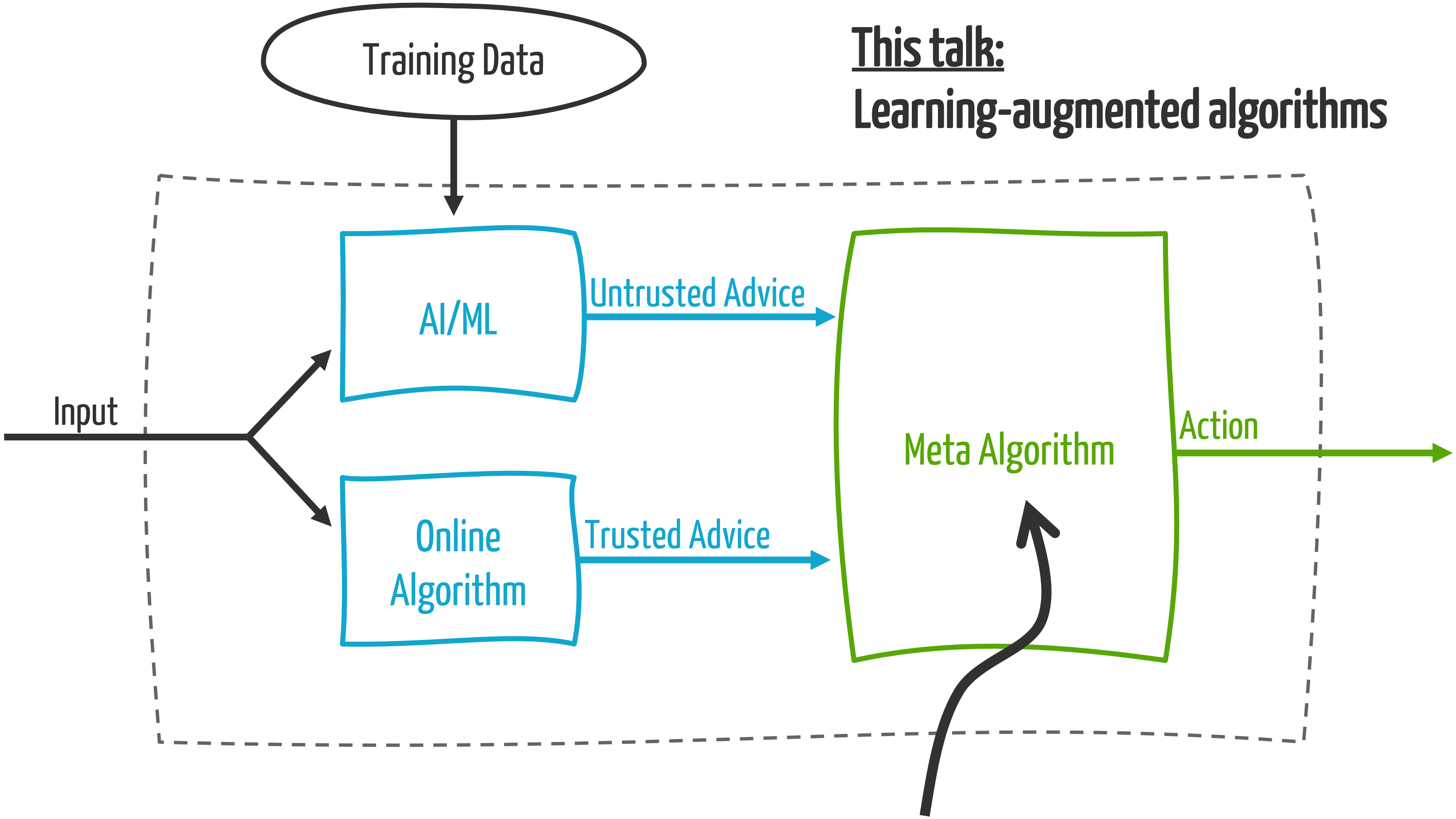
<u>This talk:</u>
Learning-augmented algorithms



Treated as black boxes

Allows adoption of new AI tools by combining with current trusted approach

This talk:
Learning-augmented algorithms



How should advice be used?
Switch between them? Combine them? Hedge?

bicompetitive guarantee

Goal 1: Consistency

(Nearly) Match the performance of the untrusted expert (AI tool), when it does well.

$$Cost(Alg) \leq (1+\delta)Cost(Untrusted)$$

Goal 2: Robustness

Always ensure a worst-case performance guarantee.

$Cost(Alg) \leq \gamma_{Alg}\ Cost(Opt)$, where γ_{Alg} is "close to" $\gamma_{trusted}$

Goal 3: Smoothness

Trade off between robustness and consistency smoothly in prediction error.

Goal 4: Frugality / Succinctness

Use only as much advice as necessary to be robust and consistent.

Skip for today

The study of learning augmented algorithms with <u>untrusted advice</u> is exploding

Introduced by [Lykouris & Vassilvitskii, 2018] in the context of online caching

Since then, studied in a wide variety of settings:

- ski rental [Purohit et al 18] [Angelopoulos et al 19] [Bamas et al 20] [Wei & Zhang 20], ...
- bloom filters [Mitzenmacher 18]
- online set cover [Bamas et al 20]
- online matching [Antoniadis et al 20]
- metrical task systems [Antoniadis et al 20]
- Scheduling [Scully et al 22]
- data center capacity [Rutten & Mukherjee 21]
- demand response [Lee et al 21]
- online optimization [Christianson et al 21]
- online conversion problems [Sun et al 21]
- convex body chasing [Christianson et al 21]
- linear quadratic control [Li et al 21]
- Online knapsack [Sun et al 22]

Bibliography of 200+ papers at https://algorithms-with-predictions.github.io/

The study of learning augmented algorithms with untrusted advice is exploding

Introduced by [Lykouris & Vassilvitskii, 2018] in the context of online caching

Since then, studied in a wide variety of settings:

- ski rental [Purohit et al 18] [Angelopoulos et al 19] [Bamas et al 20] [Wei & Zhang 20], ...
- bloom filters [Mitzenmacher 18]
- online set cover [Bamas et al 20]
- online matching [Antoniadis et al 20]
- metrical task systems [Antoniadis et al 20]
- Scheduling [Scully et al 22]
- data center capacity [Rutten & Mukherjee 21]
- demand response [Lee et al 21]
- online optimization [Christianson et al 21]
- online conversion problems [Sun et al 21]
- convex body chasing [Christianson et al 21]
- linear quadratic control [Li et al 21]
- Online knapsack [Sun et al 22]

Real applications in industry have emerged:

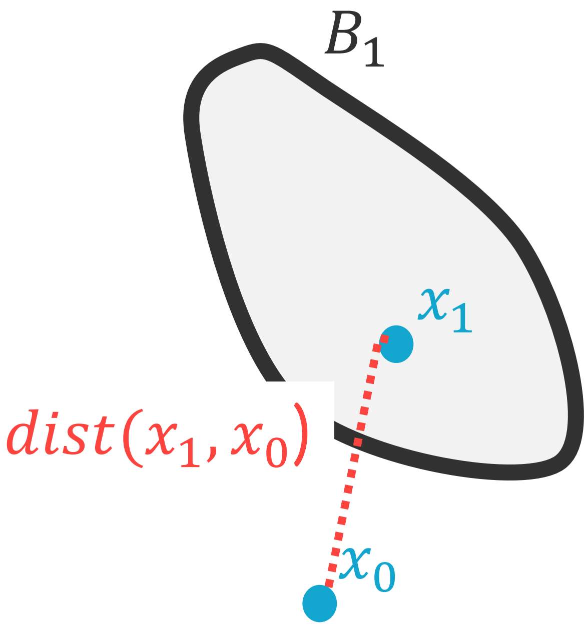
video streaming, co-generation management, data center capacity management, robotic manipulation, drone trajectory planning, ...

This talk: **Algorithm design** & **fundamental limits** on the use of learning-augmented algorithms.

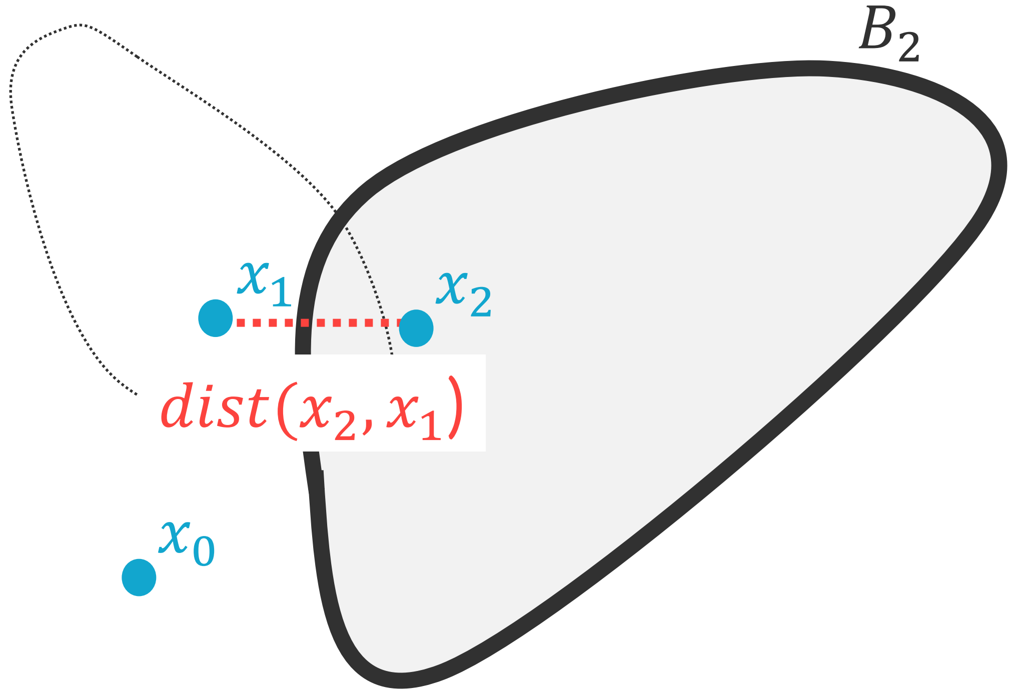
Examples:

1. **Appetizer:**
   Convex Body Chasing → Carbon-aware data centers

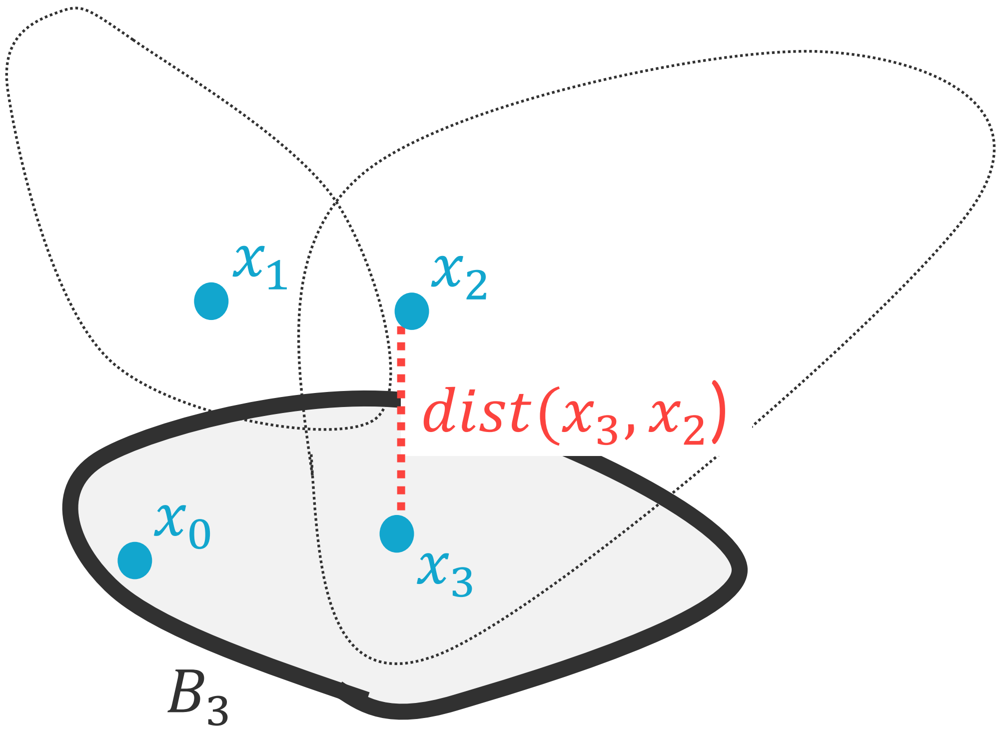
2. **Main Course:**
   MDPs → Adaptive EV charging

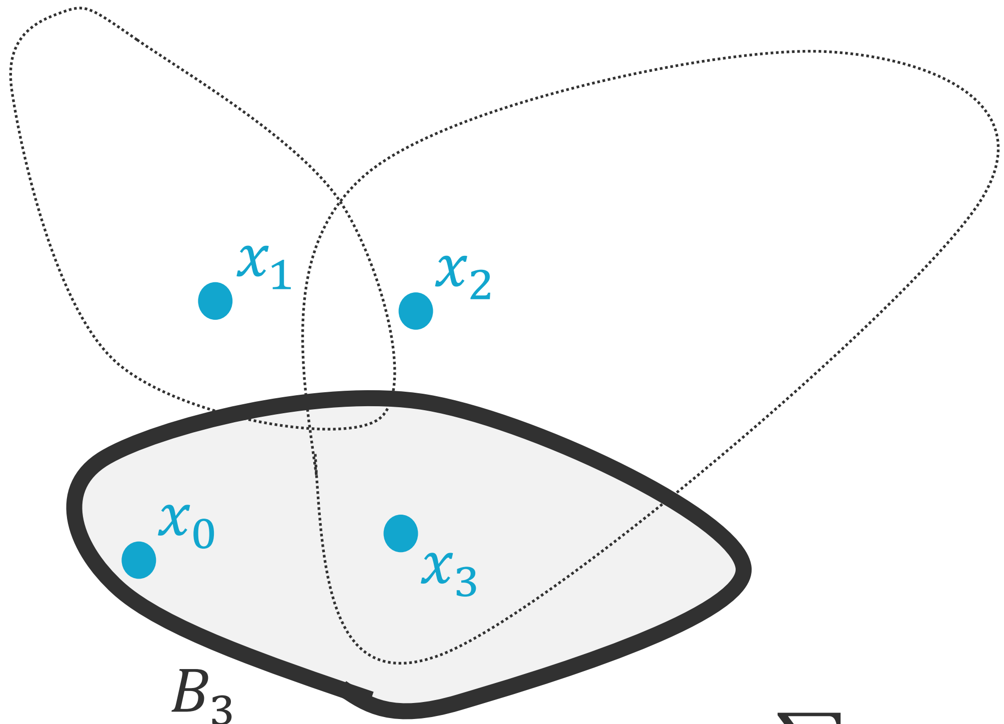


B_1
x_1
$dist(x_1, x_0)$
x_0

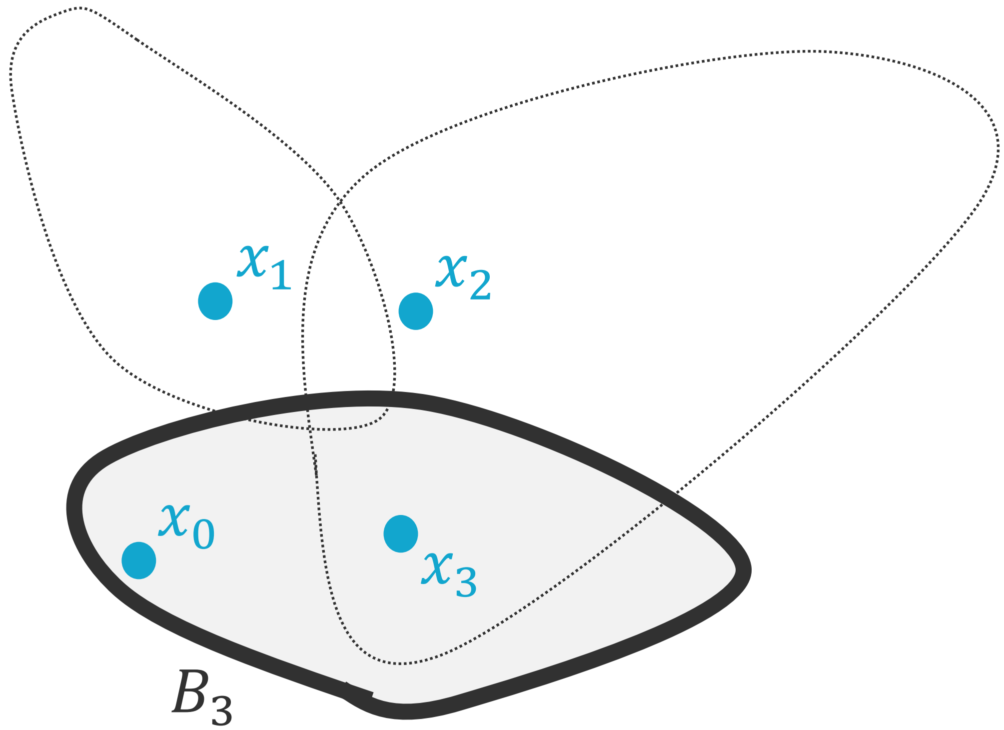


B_2
x_1
x_2
$dist(x_2, x_1)$
x_0


x_1
x_2
$dist(x_3, x_2)$
x_0
x_3
B_3



$$\min_{x_t \in B_t} \sum_t dist(x_t, x_{t-1})$$



How do you decide where to move
without knowing the future?

Convex body chasing has a long history & many applications

Applications to data centers, video streaming, robotics, drone trajectory tracking, "learning to control" and "safe control", among others.

Exciting algorithmic progress in recent years [Antoniadis et al 16], [Bansal et al 20], [Bubeck et al 19], [Sellke 20], [Argue 20], [Bubeck et al 20], [Argue 21], ...

Theorem [Bubeck et al 20]. Moving to the Steiner point of the body each round obtains an $O\left(\min\left(d, \sqrt{d\log(T)}\right)\right)$-competitive ratio. Any online algorithms is $\Omega(\sqrt{d})$.

dimension of action space

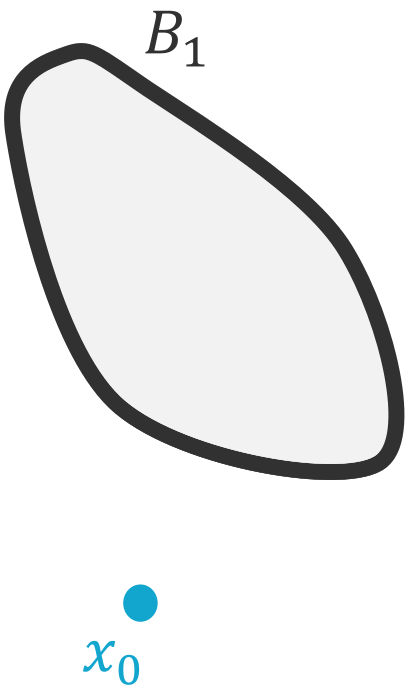
Convex body chasing has a long history & many applications

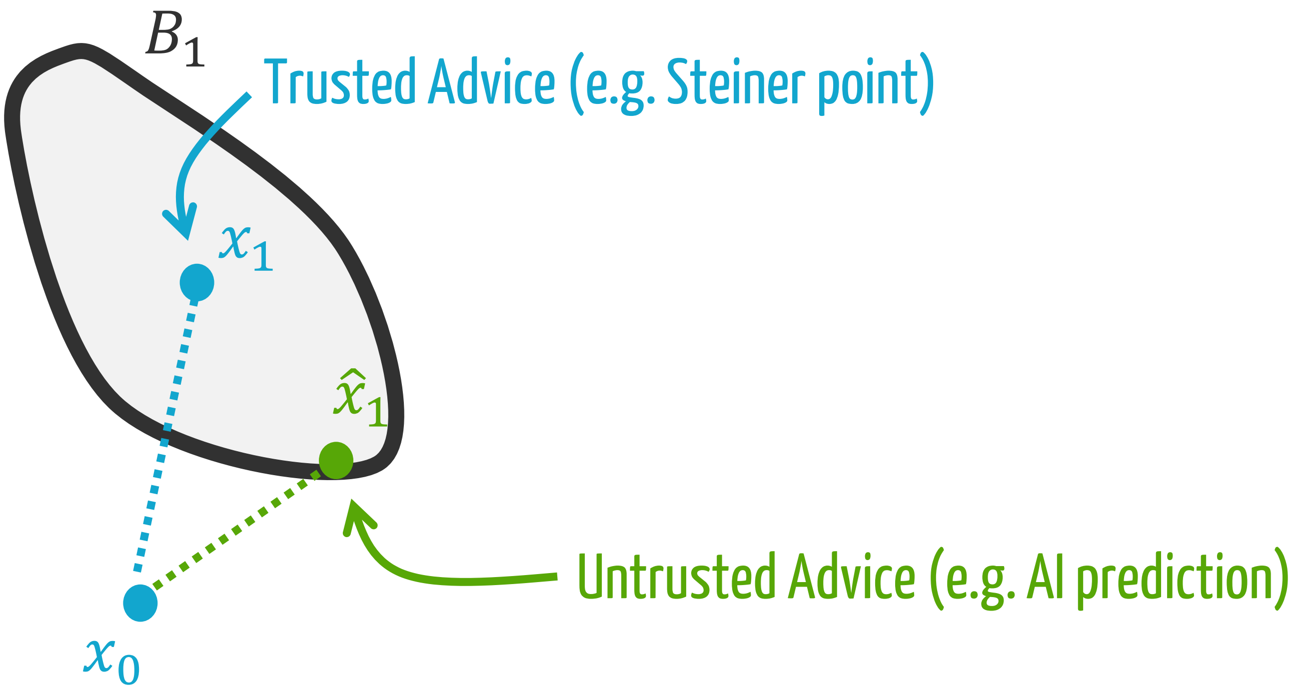
Applications to data centers, video streaming, robotics, drone trajectory tracking, “learning to control” and “safe control”, among others.

Exciting algorithmic progress in recent years [Antoniadis et al 16], [Bansal et al 20], [Bubeck et al 19], [Sellke 20], [Argue 20], [Bubeck et al 20], [Argue 21], ...

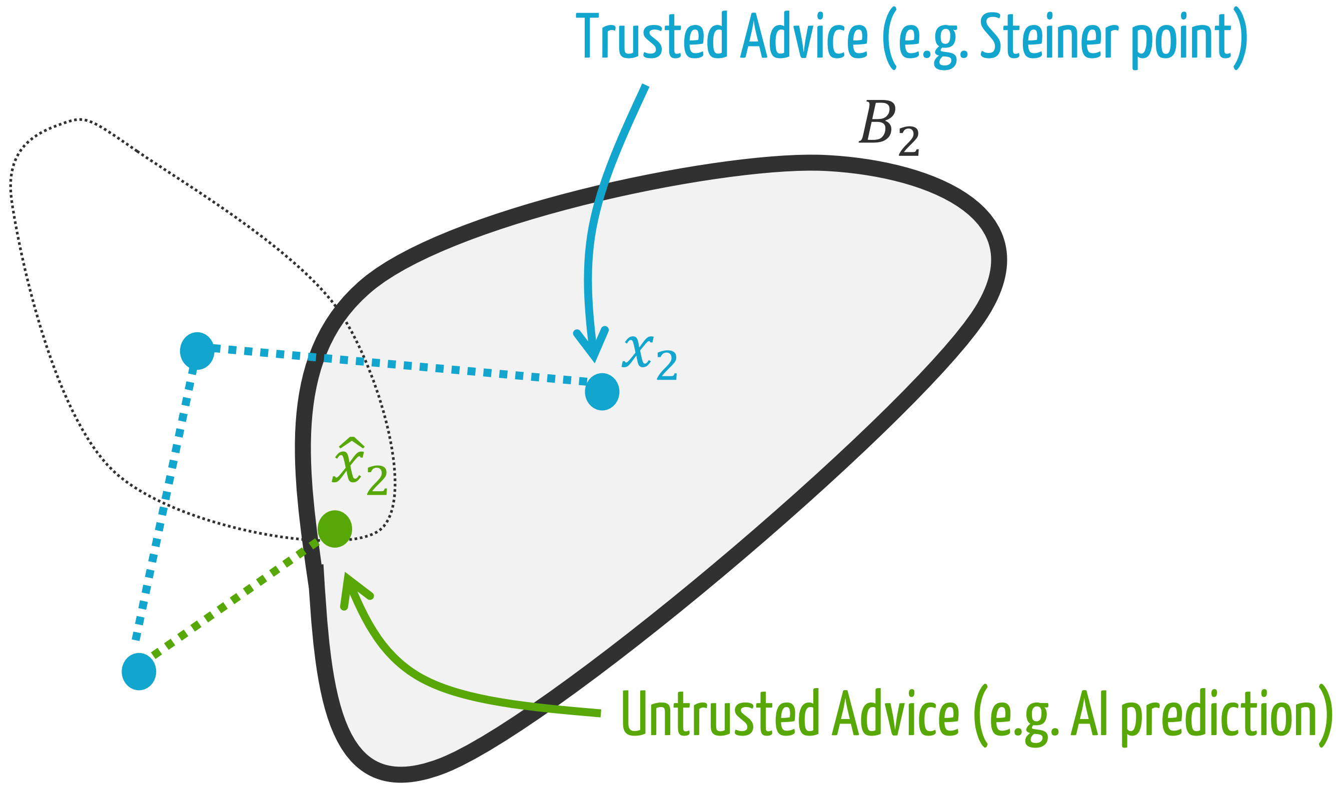
Theorem [Bubeck et al 20]. Moving to the Steiner point of the body each round obtains an $O\left(\min\left(d, \sqrt{d \log(T)}\right)\right)$-competitive ratio. Any online algorithms is $\Omega(\sqrt{d})$.

Choices of algorithm are quite conservative. Advice can help.

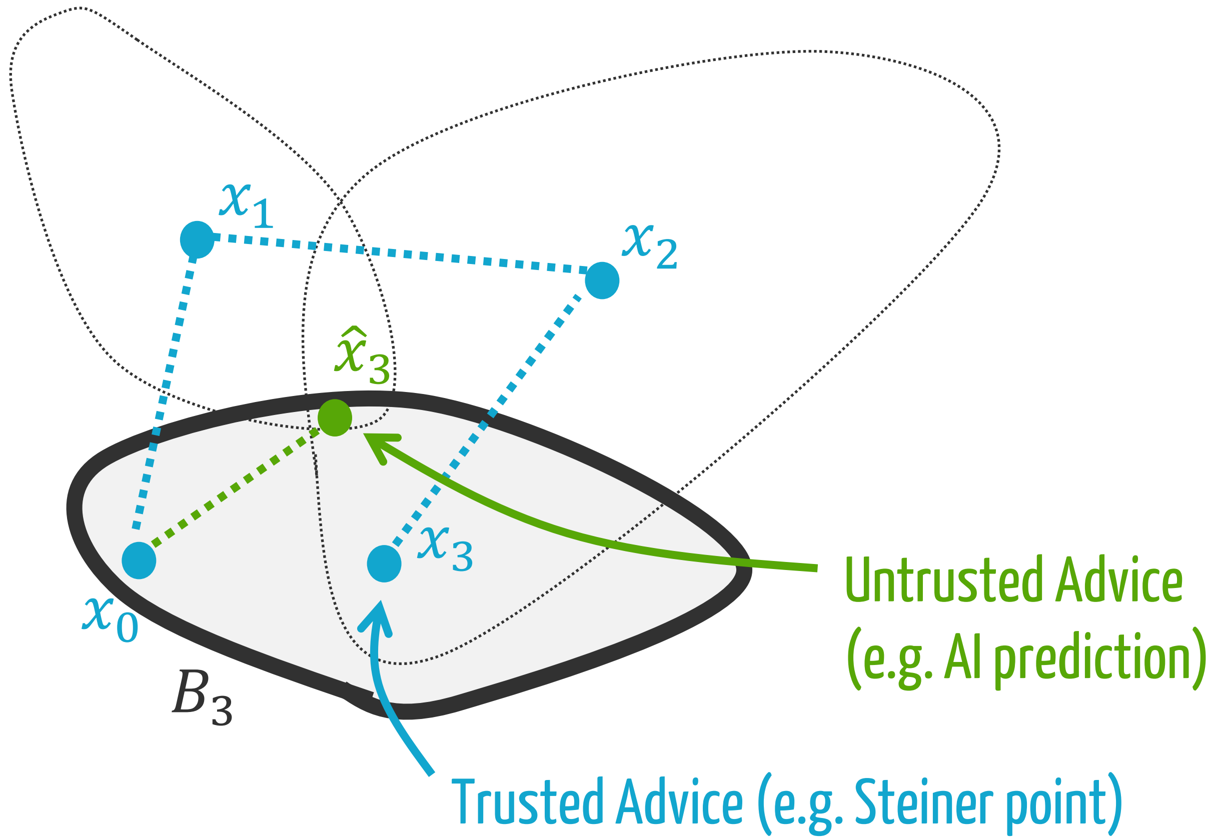

B_1
x_0



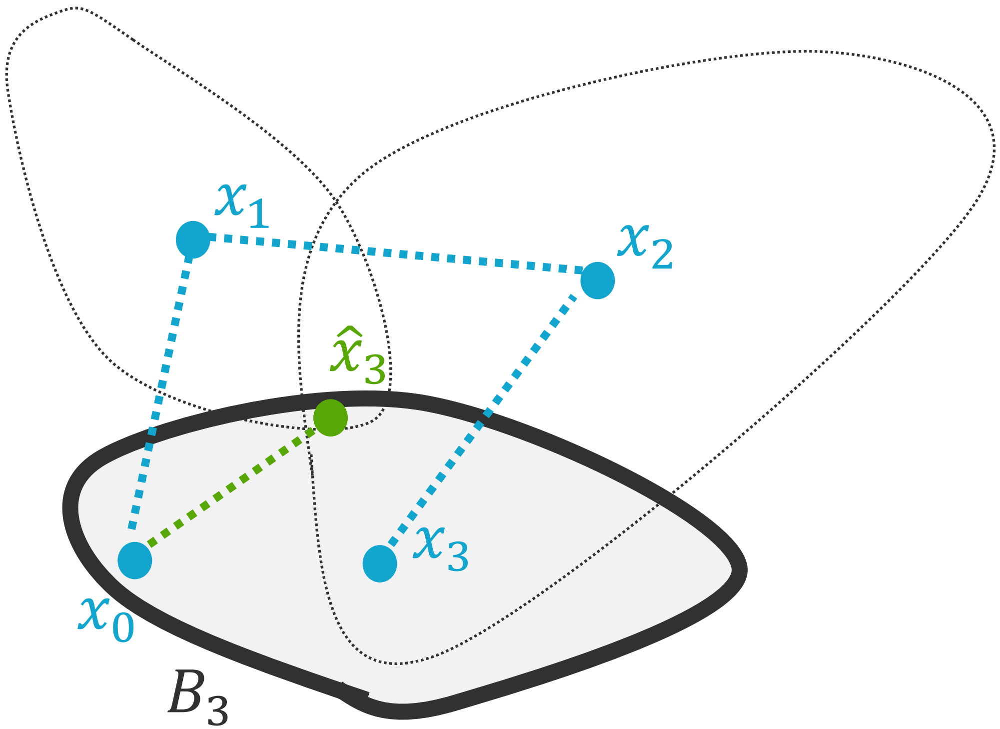
B_1
Trusted Advice (e.g. Steiner point)
x_1
$\hat{x}_1$
Untrusted Advice (e.g. AI prediction)
x_0



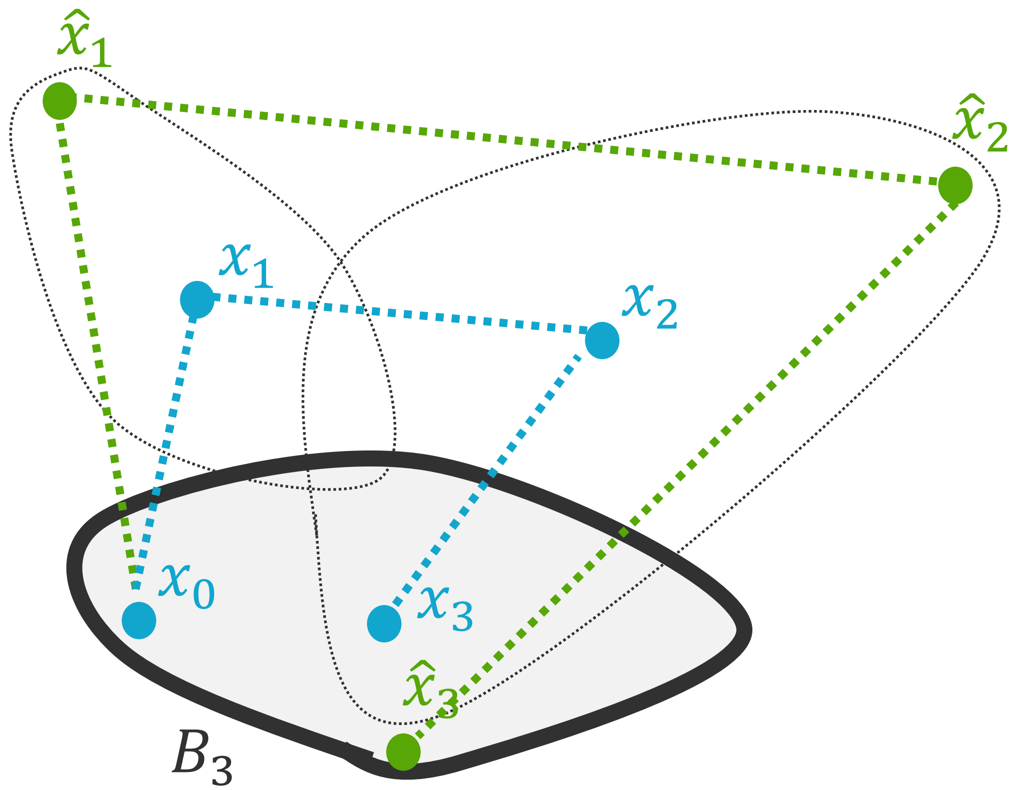
Trusted Advice (e.g. Steiner point)
B_2
x_2
$\hat{x}_2$
Untrusted Advice (e.g. AI prediction)



x_1
x_2
$\hat{x}_3$
x_3
x_0
B_3
Untrusted Advice
(e.g. AI prediction)
Trusted Advice (e.g. Steiner point)



But the advice could have been bad...



But the advice could have been bad...

A primer on learning-augmented algorithm design

Attempt 1: A Switching Algorithm

1. Follow the untrusted advice until total distance traveled is r.
2. Follow the trusted advice until total distance traveled is r.
3. Set $r \leftarrow 2r$

Treats advice as
black boxes.

Attempt 1: A Switching Algorithm

1. Follow the untrusted advice until total distance traveled is r.
2. Follow the trusted advice until total distance traveled is r.
3. Set $r \leftarrow 2r$ and repeat.

Optimize to bias toward consistency

Theorem. For nested convex body chasing, the switching algorithm is $(1+\delta)$-consistent & $O(dD/\delta)$-robust.

diameter of action space

Attempt 1: A Switching Algorithm

1. Follow the untrusted advice until total distance traveled is r.
2. Follow the trusted advice until total distance traveled is r.
3. Set $r \leftarrow 2r$ and repeat.

Optimize to bias toward consistency

Theorem. For nested convex body chasing, the switching algorithm is $(1+\delta)$-consistent & $O(dD/\delta)$-robust.

"Best of both worlds": Black-box AI/ML imbued with robustness guarantee. Constant factor loss in robustness yields near-optimal consistency.

A Fundamental Limit

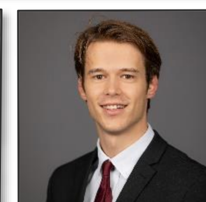
Theorem. For general convex body chasing, any switching algorithm that is robust must be at least 3-consistent.

Theorem. For nested convex body chasing, the switching algorithm is $(1+\delta)$-consistent & $O(dD/\delta)$-robust.

A Fundamental Limit

<u>Theorem.</u> For general convex body chasing, any switching algorithm that is robust must be at least 3-consistent.

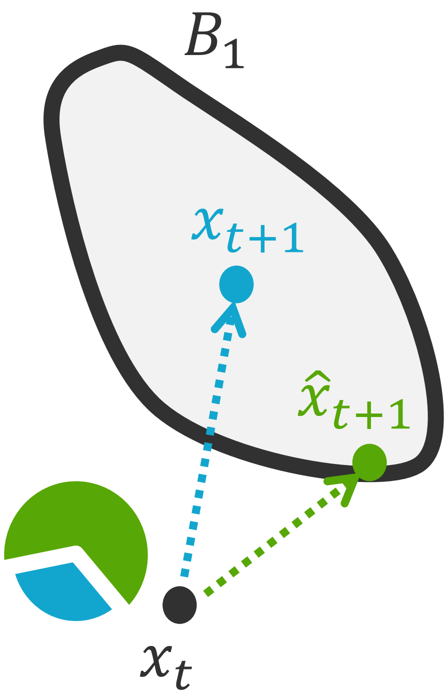
<u>Theorem.</u> For general convex body chasing, any memoryless algorithm that is robust cannot have non-trivial consistency.

Consistency better than if advice had been ignored

Attempt 2: A Bandit Algorithm

Apply multiplicative weights a la [Blum & Burch 2000]



<u>Multiplicative Weights [Blum & Burch 2000]</u>

Update weights for each expert

$$w_{ALG_i}^{t+1} = w_{ALG_i}^{t} \cdot (1-\beta)^{Cost_{t,t}(ALG_i)/D}$$

Update probability of following each expert

$$p_i^{t+1} = {}^{w_{ALG_i}}\!/_{\sum w_{ALG_i}}$$

Switch to other expert with probability proportional to mass transferred from $p_{ALG_i}^{t}$ to $p_{ALG_j}^{t+1}$

Attempt 2: A Bandit Algorithm

Apply multiplicative weights a la [Blum & Burch 2000]

Theorem [Antoniadis et al 2020]. For general convex body chasing, multiplicative weights has cost

$$(1+\delta) \cdot 4\eta Cost(Untrusted) + O(D/\delta) \text{ [Consistency]}$$

and

$$(1+\delta) \cdot O(d) Cost(Opt) + O(D/\delta) \text{ [Robustness]}$$

Aggregate prediction quality of untrusted advice

Attempt 2: A Bandit Algorithm

Apply multiplicative weights a la [Blum & Burch 2000]

<u>Theorem [Antoniadis et al 2020].</u> For general convex body chasing, multiplicative weights has cost

$$(1+\delta)\cdot 4\eta Cost(Untrusted) + O(D/\delta) \text{ [Consistency]}$$

and

$$(1+\delta)\cdot O(d)Cost(Opt) + O(D/\delta) \text{ [Robustness]}$$

Multiplicative Weights has been used to incorporate untrusted advice broadly.
(This result extends to metrical task systems, MTS.)

Attempt 2: A Bandit Algorithm

Apply multiplicative weights a la [Blum & Burch 2000]

Theorem [Antoniadis et al 2020]. For general convex body chasing, multiplicative weights has cost

$$(1+\delta)\cdot 4\eta Cost(Untrusted) + O(D/\delta)$$ [Consistency]

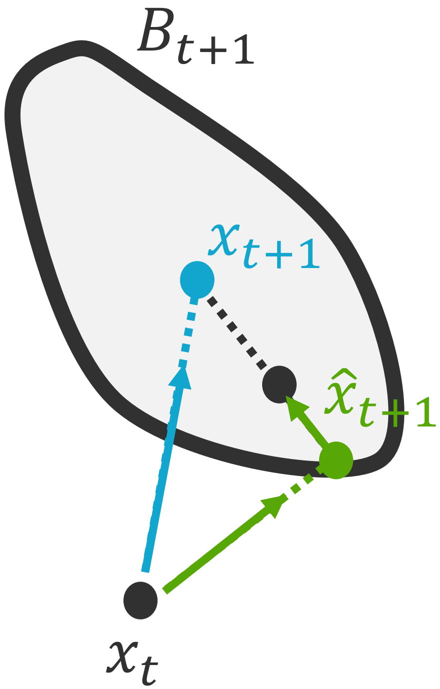
and

$$(1+\delta)\cdot O(d)Cost(Opt) + O(D/\delta)$$ [Robustness]

Diameter dependence

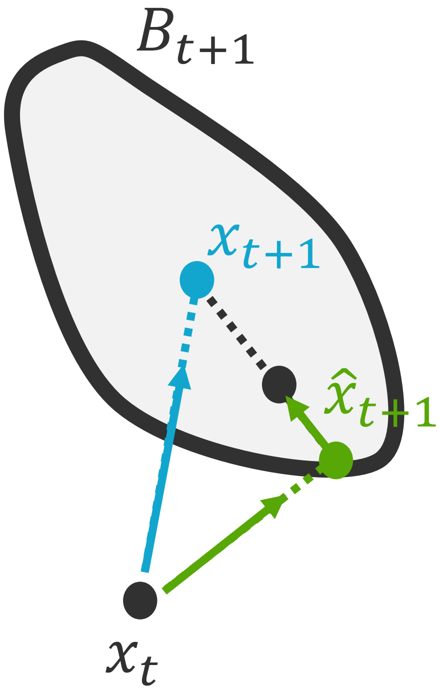
Attempt 3: Exploiting Convexity

Adaptively choose a convex combination of the two advice points.



Attempt 3: Exploiting Convexity

Adaptively choose a convex combination of the two advice points.



<u>Bicompetitive Line Chasing</u>

If $Cost_{0,t}(x) > \delta \cdot Cost_{0,t}(\hat{x})$

then follow $\hat{x}_{t+1}$

Else, take a greedy step from $\hat{x}_{t+1}$ toward x_{t+1}

with a series of radial projections depending on

$Cost_{t,t}(\hat{x})$ and $dist(\hat{x}_t, x_t)$.

Attempt 3: Exploiting Convexity

Adaptively choose a convex combination of the two advice points.

<u>Theorem.</u> For general convex body chasing, the interpolation algorithm is $(\sqrt{2}+\delta)$-consistent & $O(d/\delta^2)$-robust.

Dependence on the diameter D is gone!

Attempt 3: Exploiting Convexity

Adaptively choose a convex combination of the two advice points.

<u>Theorem.</u> For general convex body chasing, the interpolation algorithm is $(\sqrt{2}+\delta)$-consistent & $O(d/\delta^2)$-robust.

Adding robustness means sacrificing performance of black-box AI.

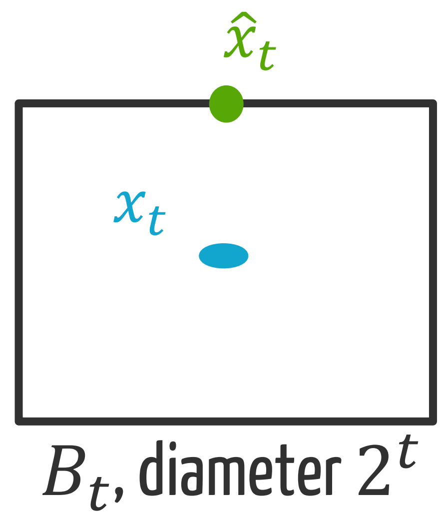
<u>Is this a fundamental limit</u>?

A Fundamental Limit

<u>Theorem.</u> For general convex body chasing, given a C-competitive algorithm, any $(1+\delta)$-consistent algorithm is $2^{\Omega(1/\delta)}C$-robust.

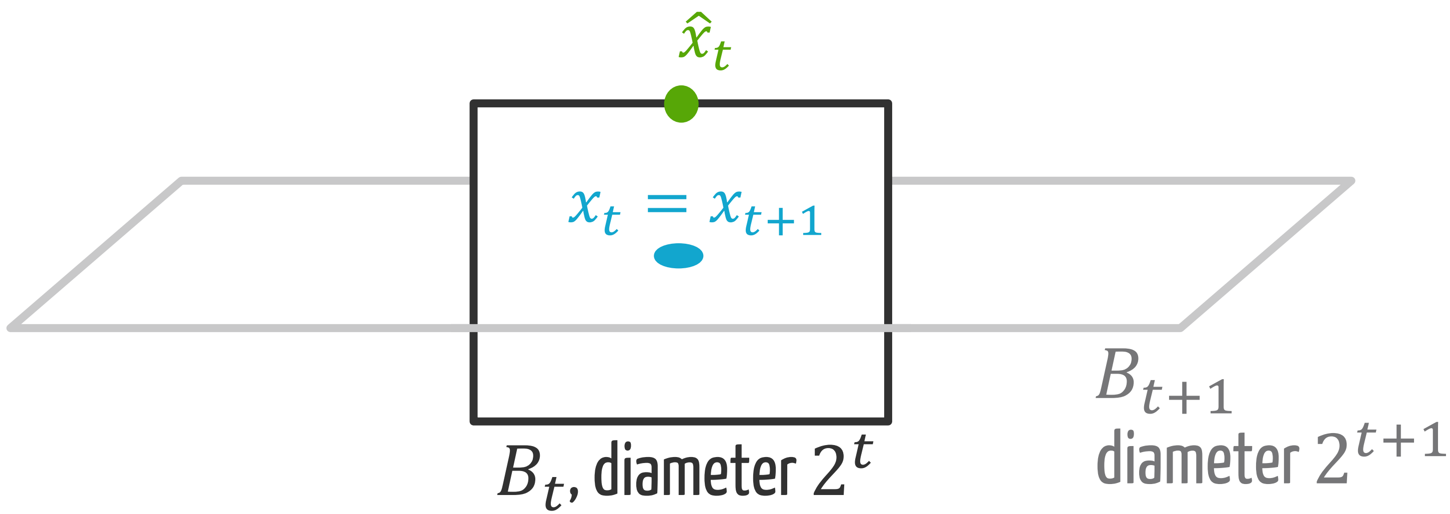
A Fundamental Limit

<u>Theorem.</u> For general convex body chasing, given a C-competitive algorithm, any $(1+\delta)$-consistent algorithm is $2^{\Omega(1/\delta)}C$-robust.



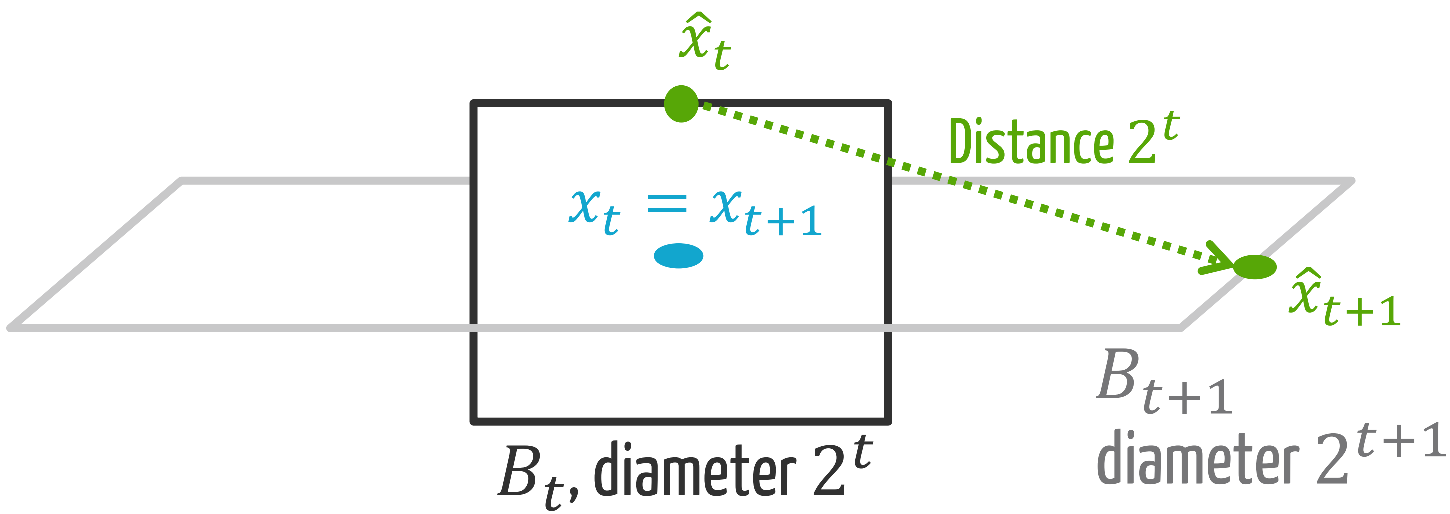
A Fundamental Limit

Theorem. For general convex body chasing, given a C-competitive algorithm, any $(1+\delta)$-consistent algorithm is $2^{\Omega(1/\delta)}C$-robust.



A Fundamental Limit

<u>Theorem.</u> For general convex body chasing, given a C-competitive algorithm, any $(1+\delta)$-consistent algorithm is $2^{\Omega(1/\delta)}C$-robust.

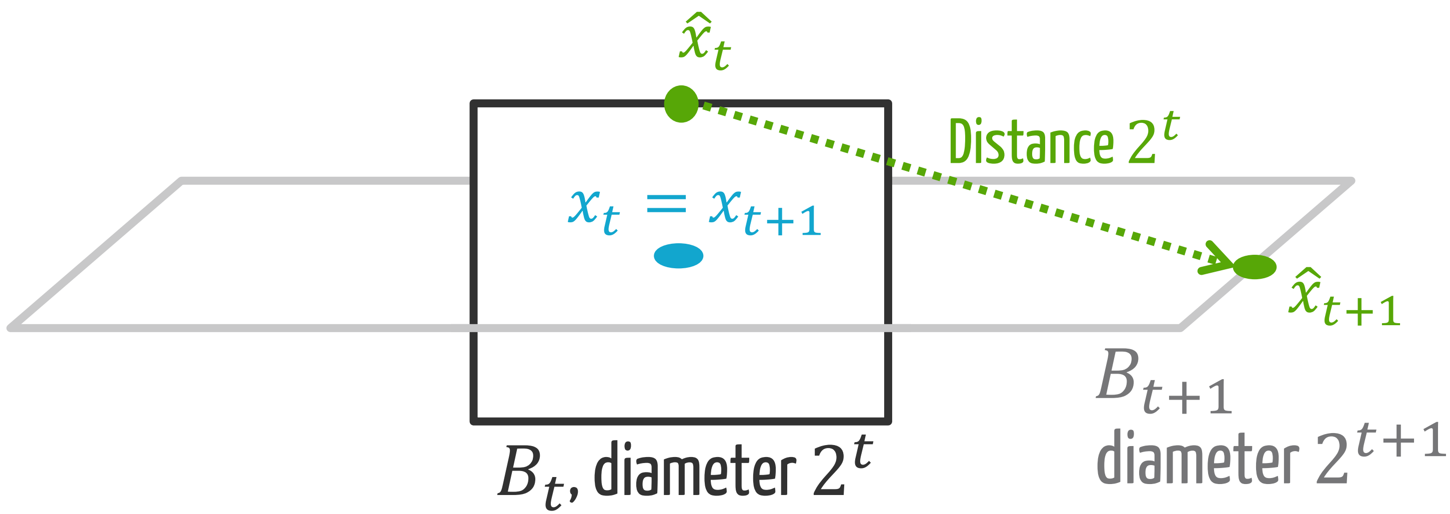


Key Property: $Cost_{0,t+1}(\hat{x}) = dist(\hat{x}_{t+1}, x_{t+1})$

(Note: $L1$ distance, not Euclidean distance.)

A Fundamental Limit

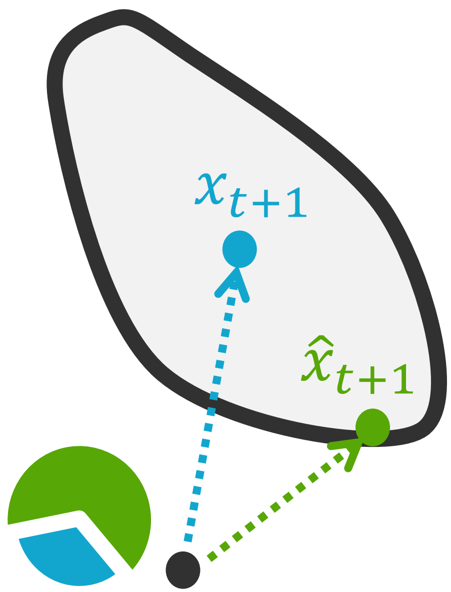
<u>Theorem.</u> For general convex body chasing, given a C-competitive algorithm, any $(1+\delta)$-consistent algorithm is $2^{\Omega(1/\delta)}C$-robust.



1. Any consistent algorithm must start following $\hat{x}_t$.
2. No algorithm can move more than $\delta/2$ probability to x_t in any round.

So, at $T = 1/\delta$, only ½ probability can be on x_T, which means the total cost is at least $2^T = 2^{1/\delta}$.

An Optimal Algorithm: Distance-Adaptive Robust weight Transport (DART)



<u>DART</u>

If $Cost_{0,t}(x) > \delta/4 \cdot Cost_{0,t}(\hat{x})$

then follow $\hat{x}_{t+1}$

Else, update probability of following the advice

$$p_{ADV}^{t+1} = \max\left(p_{ADV}^{t} - \frac{\delta Cost_{t,t}(\hat{x}_t)}{4dist(\hat{x}_t, x_t)}, 0\right)$$

Sample action through optimal transport plan (Wasserstein-1) for $p_{ALG_i}^{t} \rightarrow p_{ALG_j}^{t+1}$

An Optimal Algorithm: Distance-Adaptive Robust weight Transport (DART)

Theorem. For general convex body chasing, DART is $(1+\delta)$-consistent and $2^{O(1/\delta)}O(d)$-robust.

An Optimal Algorithm: Distance-Adaptive Robust weight Transport (DART)

<u>Theorem.</u> For general convex body chasing, DART is $(1+\delta)$-consistent and $2^{O(1/\delta)}O(d)$-robust.

<u>Theorem.</u> For convex body chasing with bounded diameter DART is $(1+\delta)$-consistent and $O(1/\delta)$-robust with an additive $O(D/\delta)$.

<u>Theorem.</u> For metrical task systems DART is $(1+\delta)$-consistent and$2^{O(1/\delta)}O(\log^2 n)$-robust.

<u>Theorem.</u> For k-server, DART is $(1+\delta)$-consistent and $O(k/\delta)$-robust.

<u>Theorem.</u> For k-function chasing in $\mathbb{R}$, DART is $(1+\delta)$-consistent and $O(k/\delta)$-robust.

An Optimal Algorithm: Distance-Adaptive Robust weight Transport (DART)

<u>Theorem.</u> For general convex body chasing, DART is $(1+\delta)$-consistent and $2^{O(1/\delta)}O(d)$-robust.

Matches state of the art

<u>Theorem.</u> For convex body chasing with bounded diameter DART is $(1+\delta)$-consistent and $O(1/\delta)$-robust with an additive $O(D/\delta)$.

1st w/o D dependence

<u>Theorem.</u> For metrical task systems DART is $(1+\delta)$-consistent and$2^{O(1/\delta)}O(\log^2 n)$-robust.

Prior: $O(1/\delta^{k-1})$

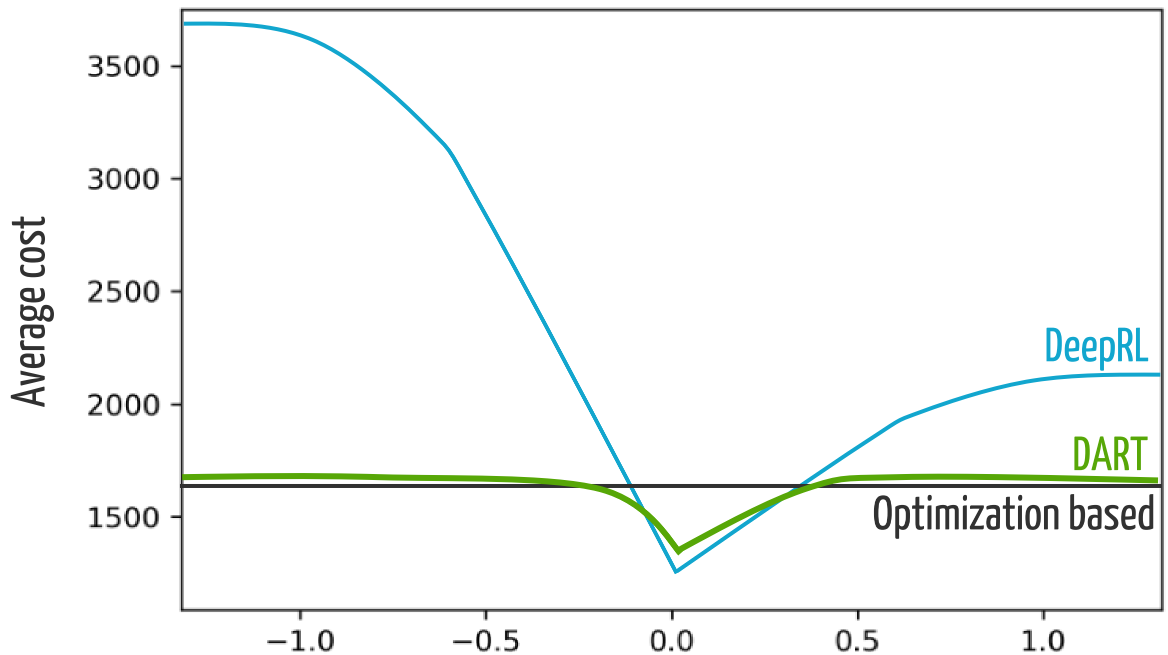
<u>Theorem.</u> For k-server, DART is $(1+\delta)$-consistent and $O(k/\delta)$-robust.

1st w/o D dependence

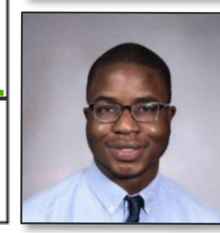
<u>Theorem.</u> For k-function chasing in $\mathbb{R}$, DART is $(1+\delta)$-consistent and $O(k/\delta)$-robust.

An example: Carbon-First Data Centers

Example: Capacity provisioning with on-site solar & storage



Sim-to-real gap / Distribution shift



Exploiting convexity yields optimal robustness-consistency tradeoffs in online convex body chasing.

Many open problems remain

- What if there are long-term constraints on actions?
- What if decisions need to be decentralized?
- What if there are multiple predictions?
- What about the stochastic model?

...

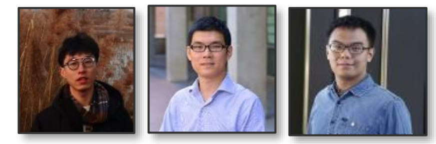
This talk: **Algorithm design** **&** **fundamental limits** **on the use of learning-augmented algorithms.**

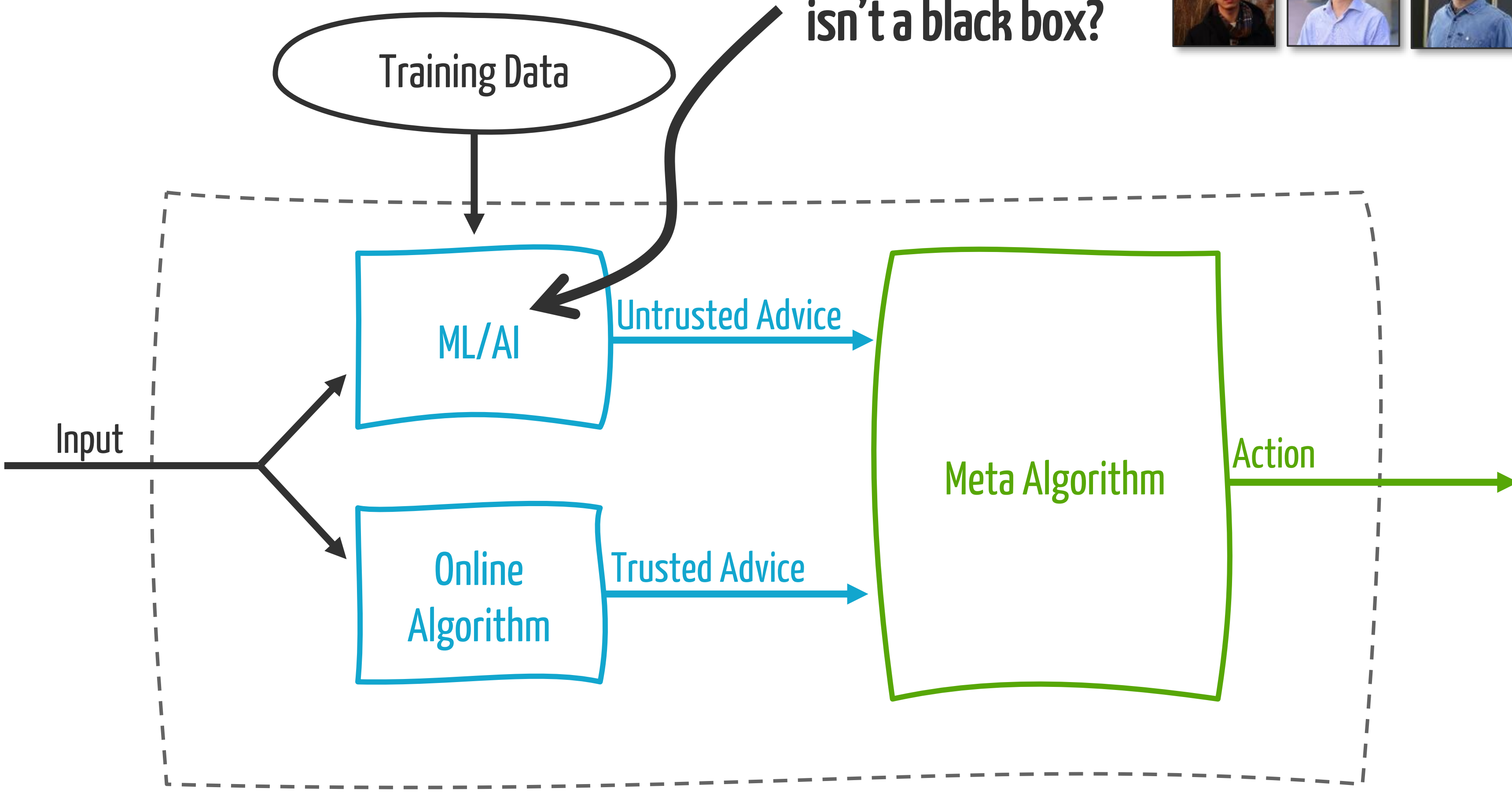
Examples:

1. **Appetizer:**
   Convex Body Chasing → Carbon-aware data centers

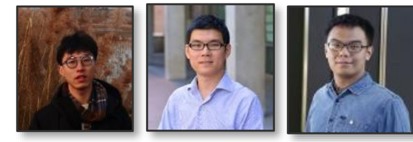
2. **Main Course:**
   MDPs → Adaptive EV charging

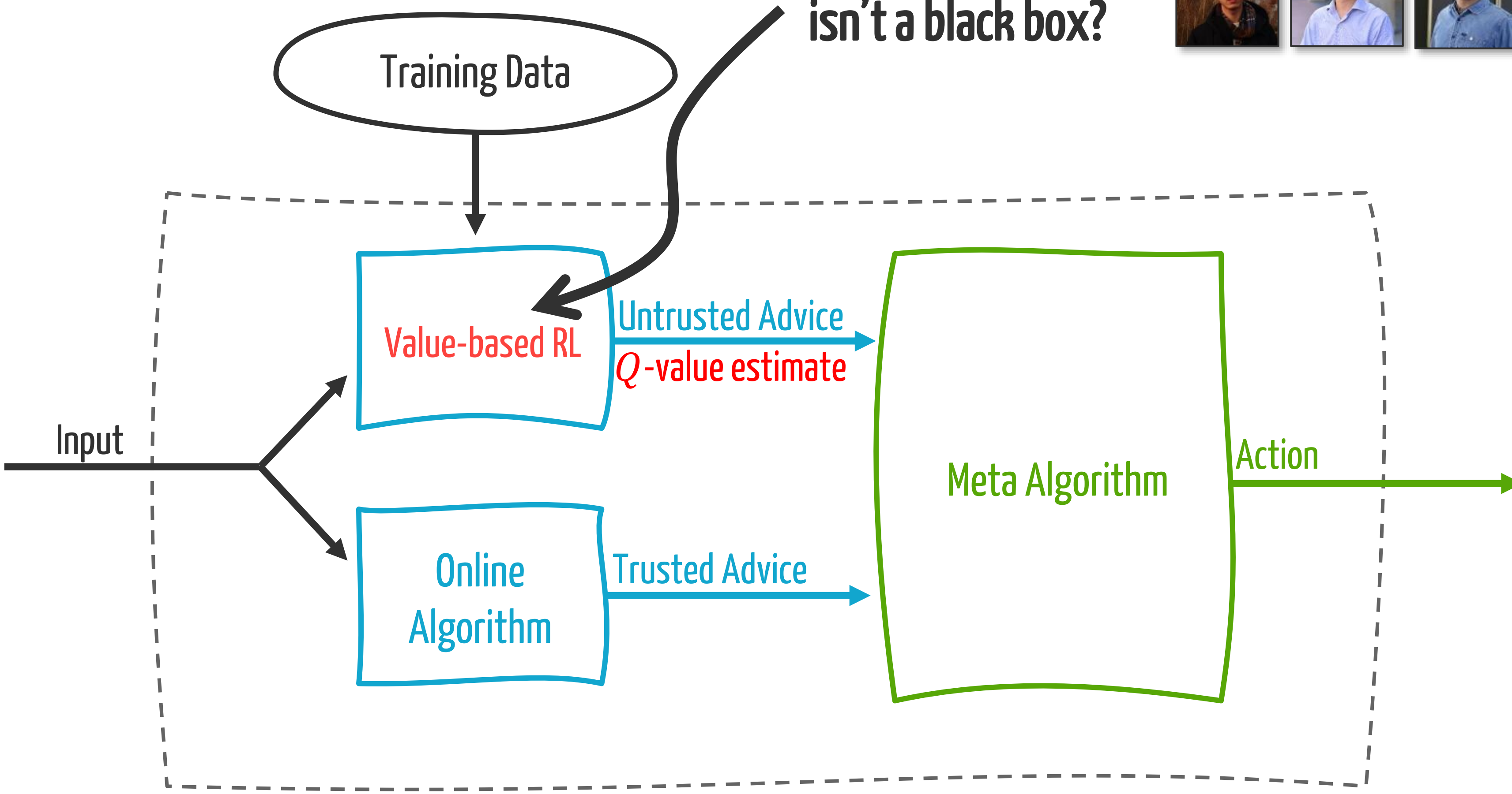
What if the learning isn’t a black box?





What if the learning isn't a black box?





Can untrusted Q-value advice improve upon black-box advice in terms of robustness-consistency tradeoffs in MDPs?

First asked by [Golowich, Moitra. Can Q-Learning be Improved with Advice? COLT 2022]

Markov Decision Processes

Finite-horizon Markov Decision Process (MDP) represented by $(\mathcal{X}, \mathcal{U}, T, P, c)$

- $\mathcal{X}$ is the state space, with norm $\|\cdot\|_{\mathcal{X}}$ (continuous or finite)
- $\mathcal{U}$ is the action space, with norm $\|\cdot\|_{\mathcal{U}}$ (continuous or finite)
- T is the horizon
- P_t is the transition kernel at step t
- c_t is the cost function at step t
- Single trajectory (not episodic)

Environment

$x_t \sim P_t(\cdot \mid x_{t-1}, u_{t-1})$

solar generation, arrivals/departures, etc.

x_t

policy π

$u_t = \pi_t(x_t)$

cost: $c_t(x_t, u_t)$

Markov Decision Processes

Finite-horizon Markov Decision Process (MDP) represented by $(\mathcal{X}, \mathcal{U}, T, P, c)$

Goal: minimize expected cost $J(\pi) = \mathbb{E}_{P,\pi}[\sum_t c_t(x_t, \pi_t(x_t))]$

Optimal Cost: $J^* = \inf_{\pi} J(\pi)$

Q-value: $Q_t^*(x, u) = \inf_{\pi} \mathbb{E}_{P,\pi}[\sum_{\tau=t}^{T-1} c_\tau(x_\tau, u_\tau) | x_t = x, u_t = u]$

Environment

$x_t \sim P_t(\cdot \,| x_{t-1}, u_{t-1})$

solar generation, arrivals/departures, etc.

x_t

policy π

$u_t = \pi_t(x_t)$

cost: $c_t(x_t, u_t)$

Markov Decision Processes with Untrusted Advice

Black-box advice vs. **Grey-box advice**

Untrusted policy $\tilde{\pi}$
suggests action $\tilde{u}_t$

Standard model
of untrusted advice
(as in appetizer)

Untrusted policy $\tilde{\pi}$
suggests action $\tilde{u}_t$ &
estimate $\tilde{Q}_t$ of long-term
cost impact of action

From a value-based policy, i.e.,
$\tilde{u}_t = \mathrm{argmin}_v \tilde{Q}_t(x_t, v)$

Gives information about the
expected gain from the action!

Markov Decision Processes with Untrusted Advice

Black-box advice

Untrusted policy $\tilde{\pi}$
suggests action $\tilde{u}_t$

vs.

Grey-box advice

Untrusted policy $\tilde{\pi}$
suggests action $\tilde{u}_t$ &
estimate $\tilde{Q}_t$ of long-term
cost impact of action

Sim-to-real translation

Multi-task learning

Markov Decision Processes with Untrusted Advice

Black-box advice

Untrusted policy $\tilde{\pi}$ suggests action $\tilde{u}_t$

vs.

Grey-box advice

Untrusted policy $\tilde{\pi}$ suggests action $\tilde{u}_t$ & estimate $\tilde{Q}_t$ of long-term cost impact of action

Consistency: π is k-consistent if $J(\pi) \leq k \cdot J^*$ for any MDP & perfect predictions.

Robustness: π is l-robust if $J(\pi) \leq l \cdot J^*$ for any MDP and any predictions.

How do learning-augmented algorithms work?

Given trusted advice ($\bar{u}_t$) and untrusted advice ($\tilde{u}_t$), how do we determine the action u_t?

Four typical designs

We saw these in the appetizer

1. Switching algorithms
2. Bandit Algorithms (randomized switching)
3. (Fixed) Convex Combination
4. Adaptive Convex Combination
   a.k.a. Projection-based algorithms

How do learning-augmented algorithms work?

Given trusted advice ($\bar{u}_t$) and untrusted advice ($\tilde{u}_t$), how do we determine the action u_t?

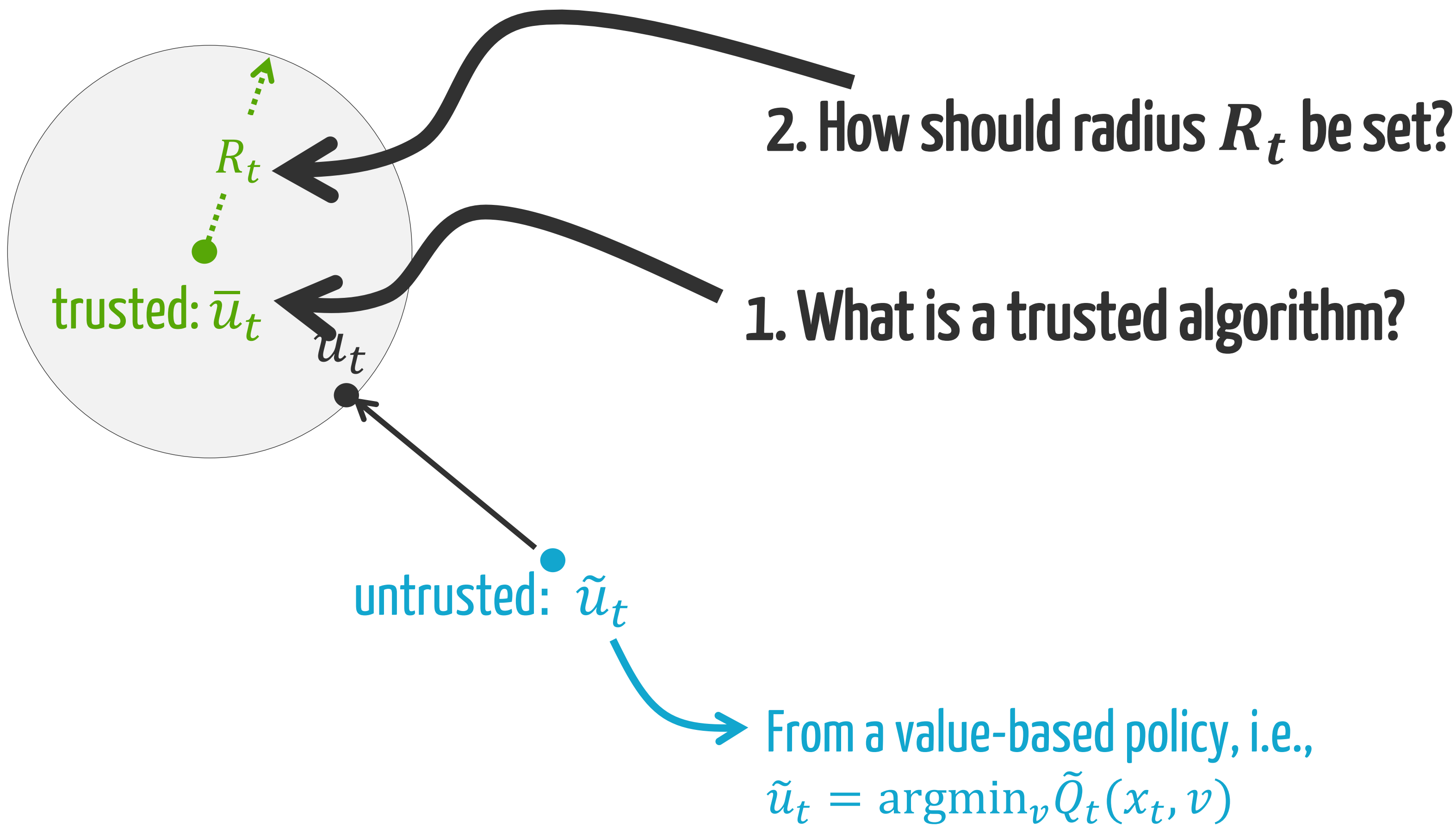
e.g. won't guarantee statility in LTV systems

Four typical designs

1. ~~Switching algorithms~~
2. ~~Bandit Algorithms (randomized switching)~~
3. ~~(Fixed) Convex Combination~~
4. Adaptive Convex Combination
   a.k.a. Projection-based algorithms

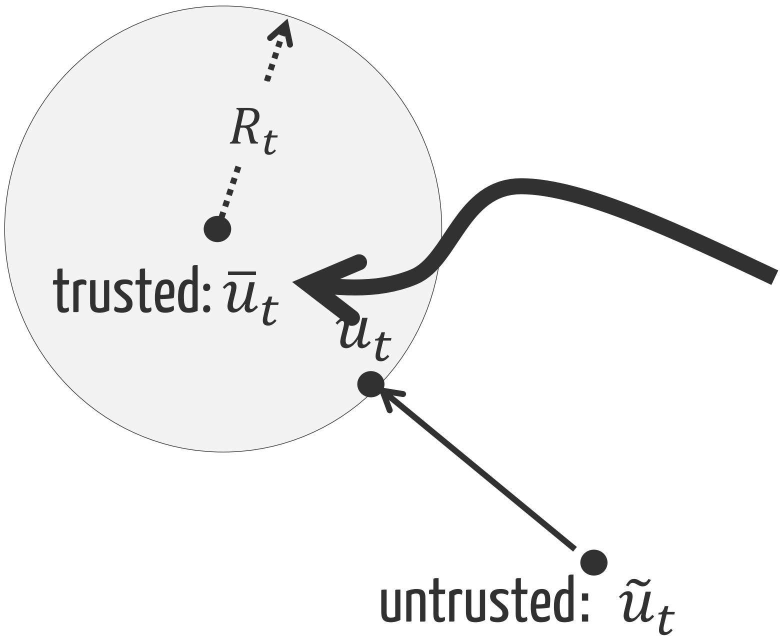
Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.



Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.

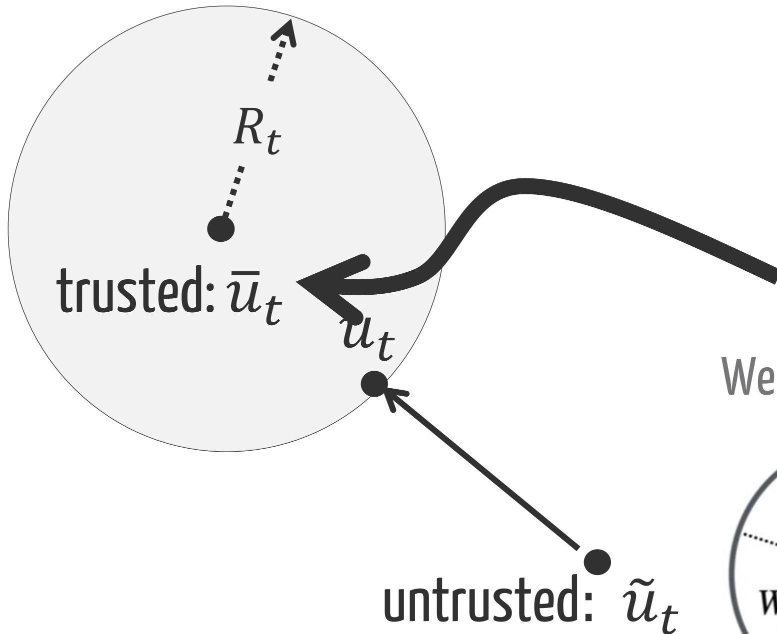


1. What is a trusted algorithm?

Goal: Define "trust" for black box algorithms

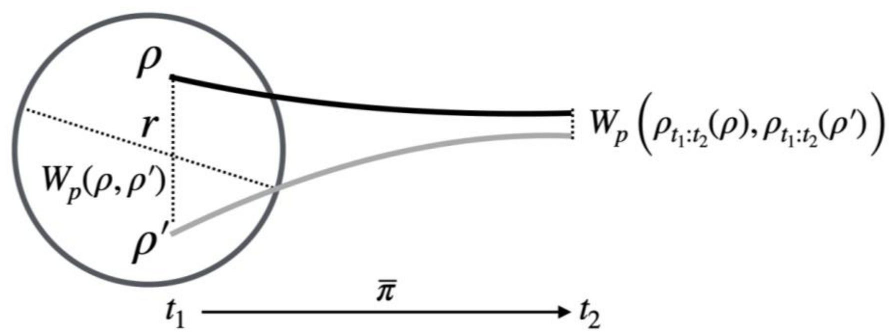
Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.



1. What is a trusted algorithm?

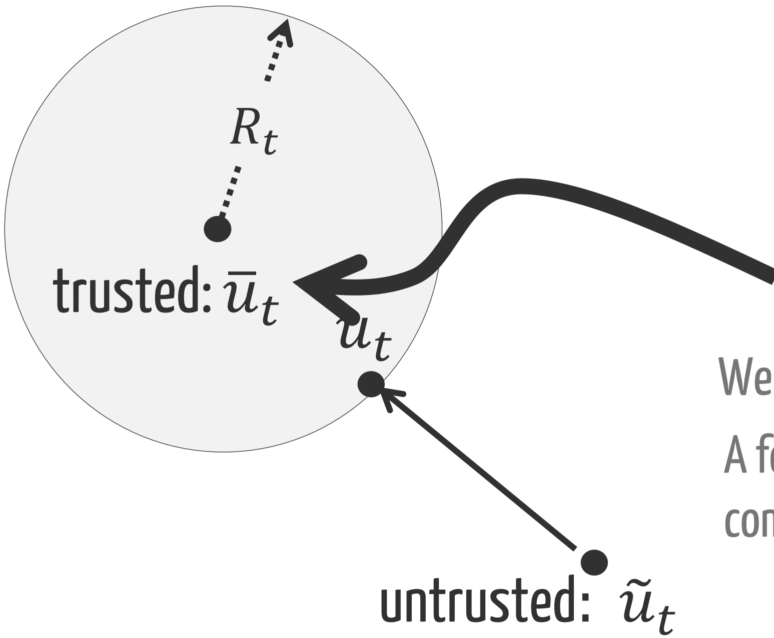
We call an algorithm trusted if it is Wasserstein-robust



Definition. r-locally p-Wasserstein-robust if for any $t_1 < t_2$ and pair of actions ρ, ρ' within Wasserstein distance r, $W_p\left(\rho_{t_1:t_2}(\rho), \rho_{t_1:t_2}(\rho')\right) \leq s(t_2 - t_1) W_p(\rho, \rho')$ for some function s satisfying $\sum_t s(t) \leq C$, for a constant C.

Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.



1. What is a trusted algorithm?

We call an algorithm trusted if it is Wasserstein-robust

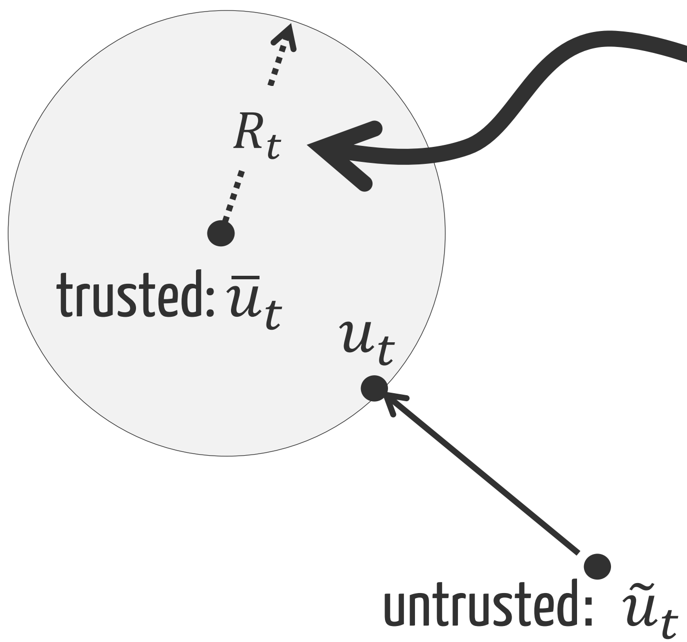
A form of perturbation stability satisfied by many common policies (see paper), e.g.,

- Robust MPC in LTV
- Discrete MDP: Any policy with regular Markov chain.

See [Lin et al 21,22,23] for a broader context.
Other applications in multi-agent RL, regret-optimal control, adaptive control, policy selection, ...

Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.



2. How should radius R_t be set?

For black-box advice, information is limited.

Simple idea: $R_t = \lambda ||\tilde{u}_t - \bar{u}_t||$,
where $\lambda \in [0,1]$ is a “trust parameter”.

<u>Theorem:</u> PROP with black box predictions is

- $1 + O\big((1-\lambda)D\big)$ consistent and
- $ROB + O(\lambda D)$ robust

diameter of action space

approximation ratio of the robust policy

Theorem: PROP with black box predictions is

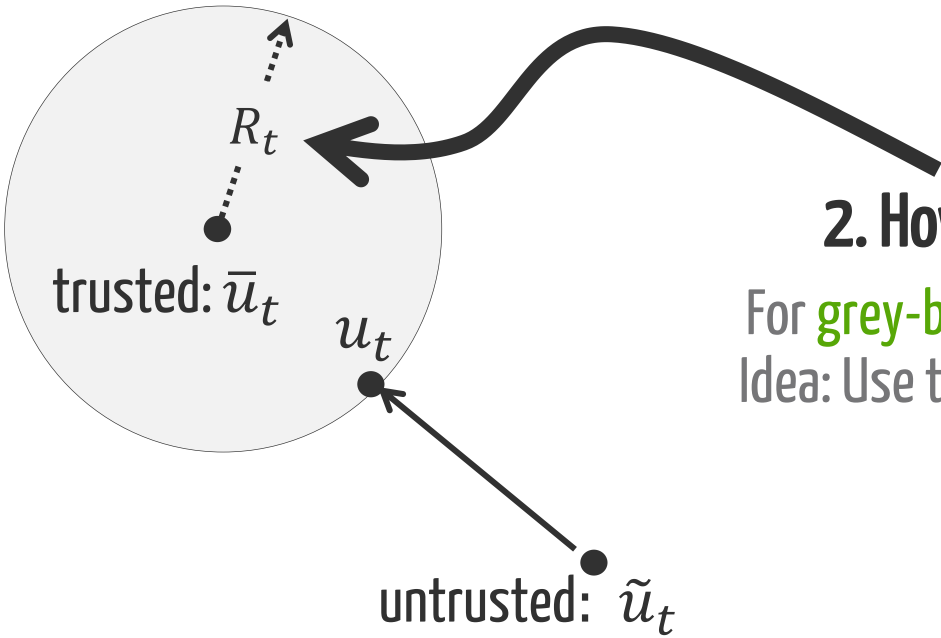
- $1 + O\big((1-\lambda)D\big)$ consistent and
- $ROB + O(\lambda D)$ robust

Theorem: No projection-based algorithm with black box predictions can be

- $1 + o\big((1-\lambda)D\big)$ consistent and
- $ROB + o(\lambda D)$ robust.

Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.



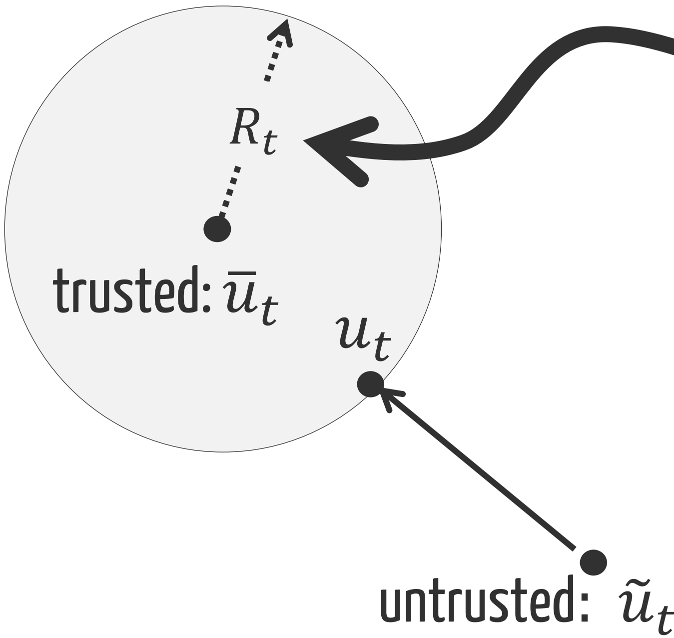
2. How should radius R_t be set?

For grey-box advice, extra information valuable.
Idea: Use the Temporal Difference (TD) error (δ_s)

$$\delta_s \uparrow \Rightarrow R_t \downarrow$$

Projection Pursuit (PROP)

Project untrusted advice onto ball around trusted advice.
Set radius R_t to ensure robustness-consistency tradeoff.



2. How should radius R_t be set?

For grey-box advice, extra information valuable.
Idea: Use the Temporal Difference (TD) error (δ_s)

$$R_t = \left[\|\tilde{u}_t - \bar{u}_t\| - \frac{\beta}{L_Q} \sum_{s=1}^{t} \delta_s(x_s, x_{s-1}, u_{s-1}) \right]^+$$

where L_Q is the Lipschitz coefficient of Q &
β is a trust parameter

Approximate TD error

$$c_{s-1}(x_{s-1}, u_{s-1}) + \inf_{v} \tilde{Q}_s(x_s, v) - \tilde{Q}_{s-1}(x_{s-1}, u_{s-1})$$

Theorem: PROP with black box predictions is

- $1 + O\big((1-\lambda)D\big)$ consistent and
- $ROB + O(\lambda D)$ robust

Theorem: No projection-based algorithm with black box predictions can be

- $1 + o\big((1-\lambda)D\big)$ consistent and
- $ROB + o(\lambda D)$ robust.

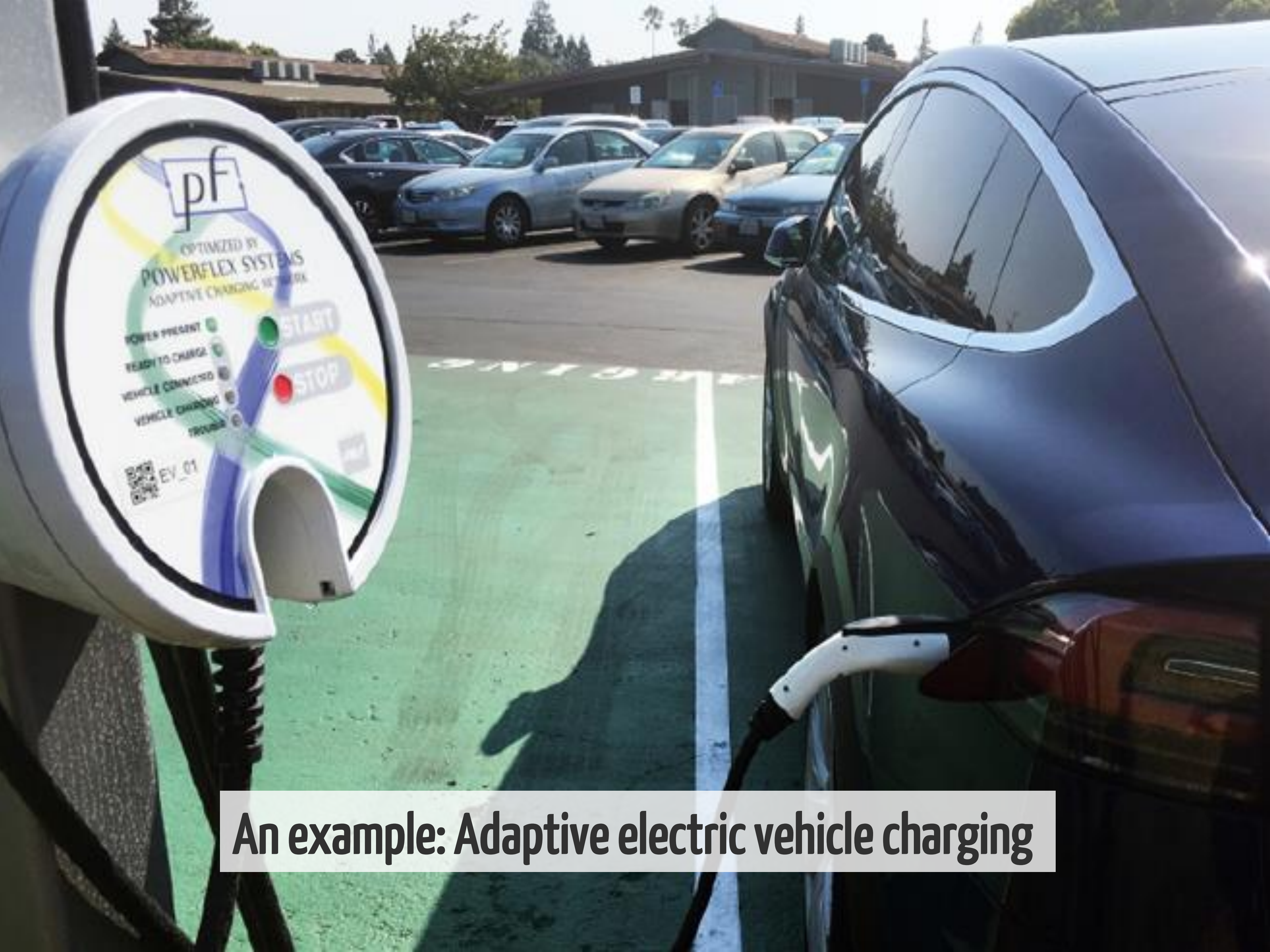
Significant improvement from Q-value predictions!

Theorem: PROP with grey-box predictions and $\beta = 1$ is

- 1-consistent and
- $ROB + o(1)$ robust.



An example: Adaptive electric vehicle charging

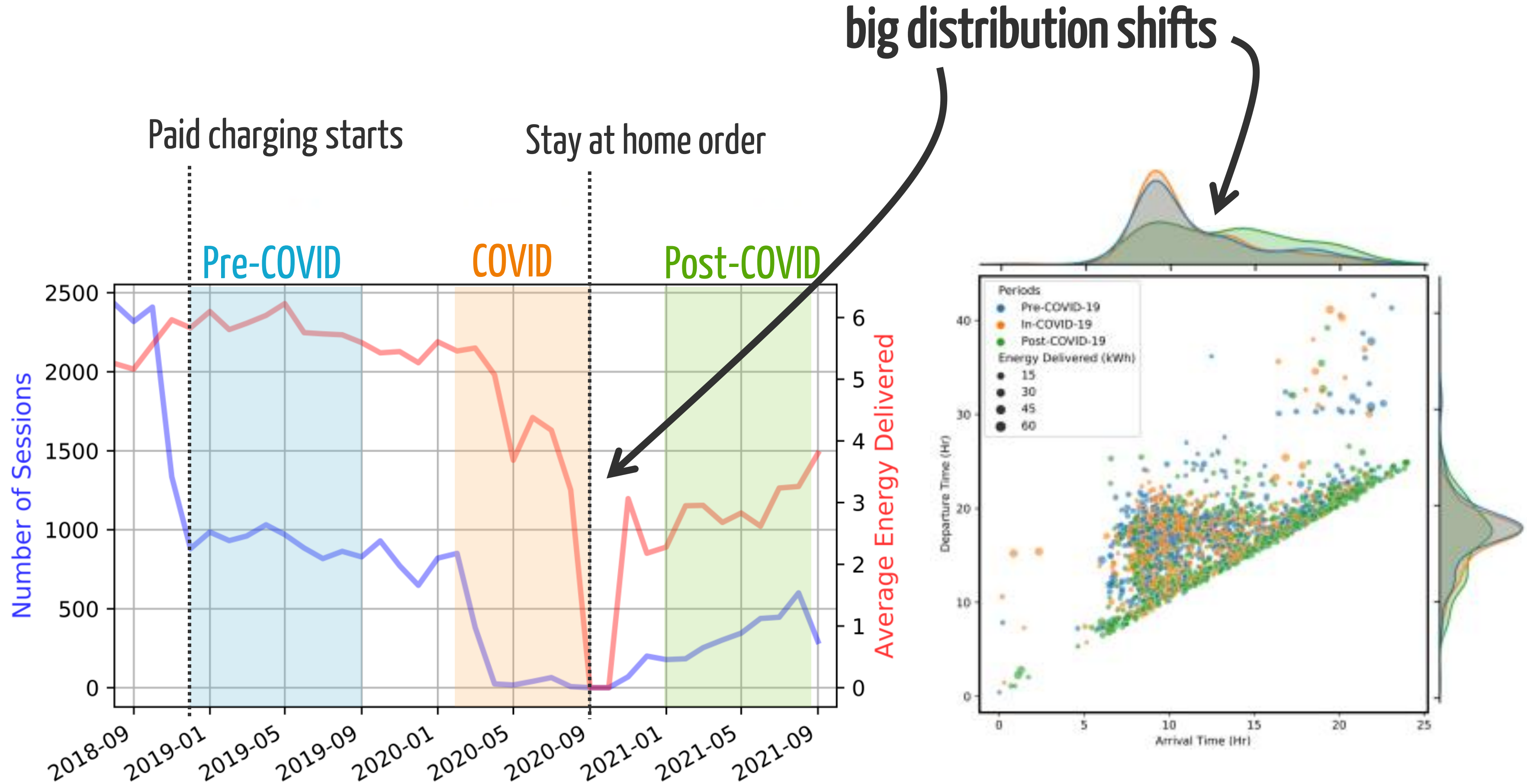


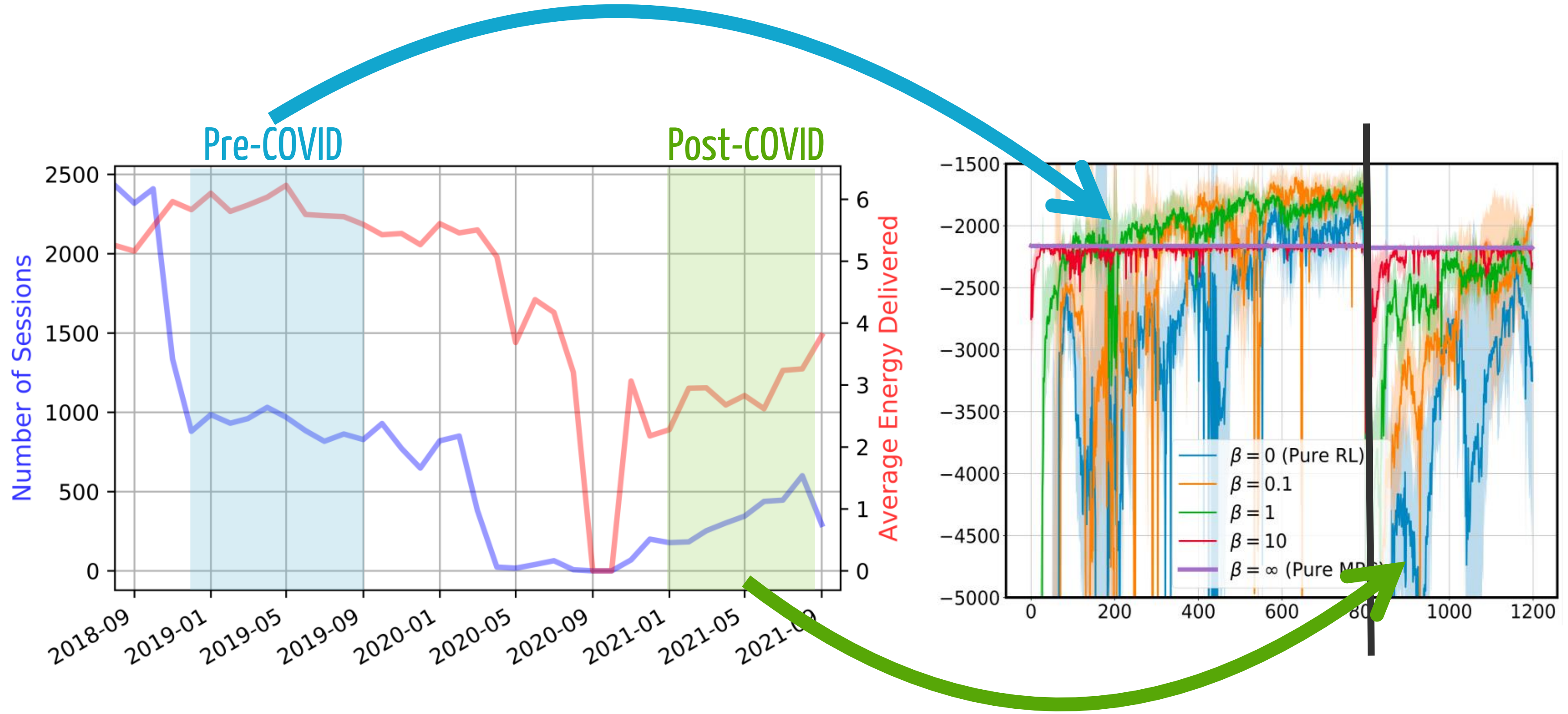
EV Charging at Caltech

$$x_{t+1} = \underbrace{A_t x_t + B_t u_t}_{\text{battery dynamics}} + \underbrace{f_t(x_t, u_t)}_{\text{uncertain residuals}}$$

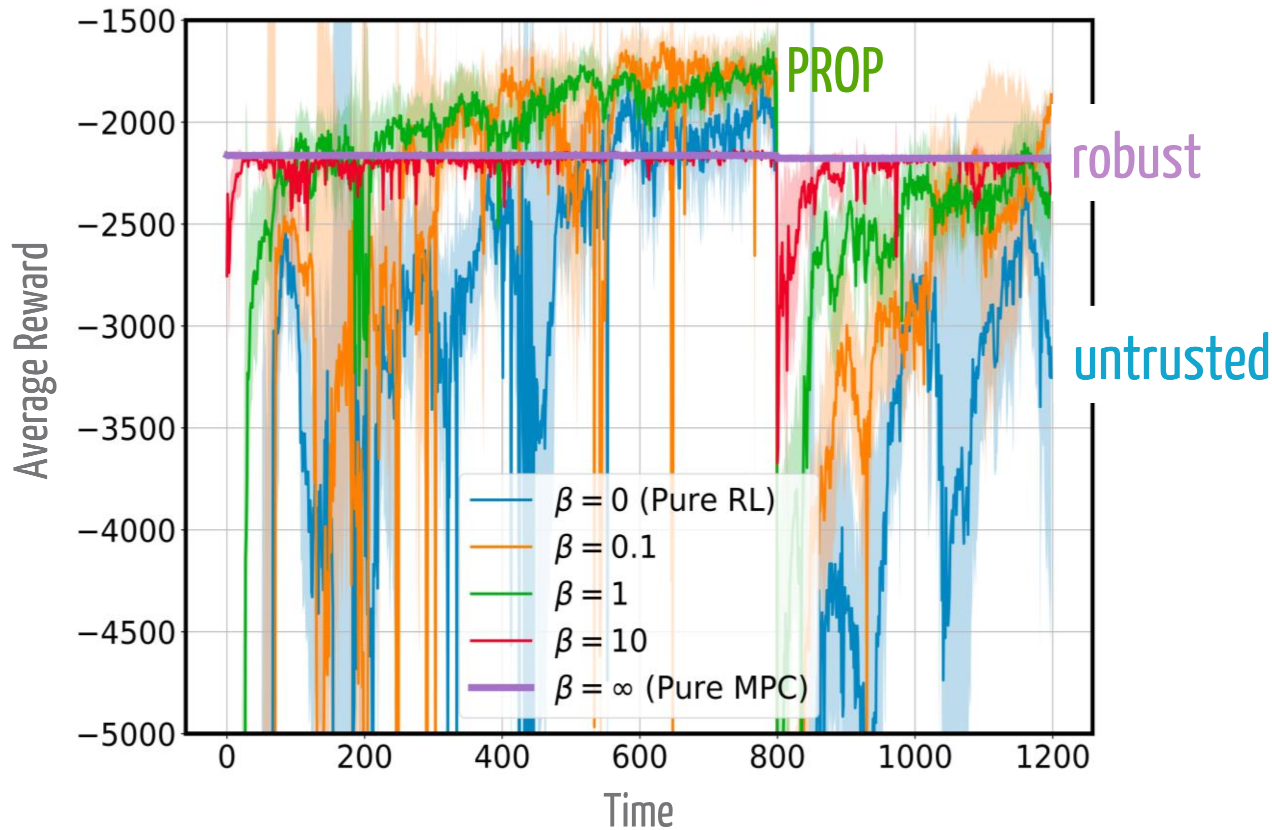
Trusted algorithm ($\overline{\pi}$): Robust MPC depends only on LTV battery dynamics

Untrusted algorithm ($\tilde{\pi}$): RL can learn residuals better (if no distribution shift)


big distribution shifts
Paid charging starts
Stay at home order
Pre-COVID
COVID
Post-COVID
Number of Sessions
Average Energy Delivered
2018-09
2019-01
2019-05
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
2021-09
Periods
Pre-COVID-19
In-COVID-19
Post-COVID-19
Energy Delivered (kWh)
Departure Time (Hr)
Arrival Time (Hr)



Pre-COVID
Post-COVID
Number of Sessions
Average Energy Delivered
2018-09
2019-01
2019-05
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
$\beta = 0$ (Pure RL)
$\beta = 0.1$
$\beta = 1$
$\beta = 10$



PROP exploits predictions while maintaining robustness to distribution shift

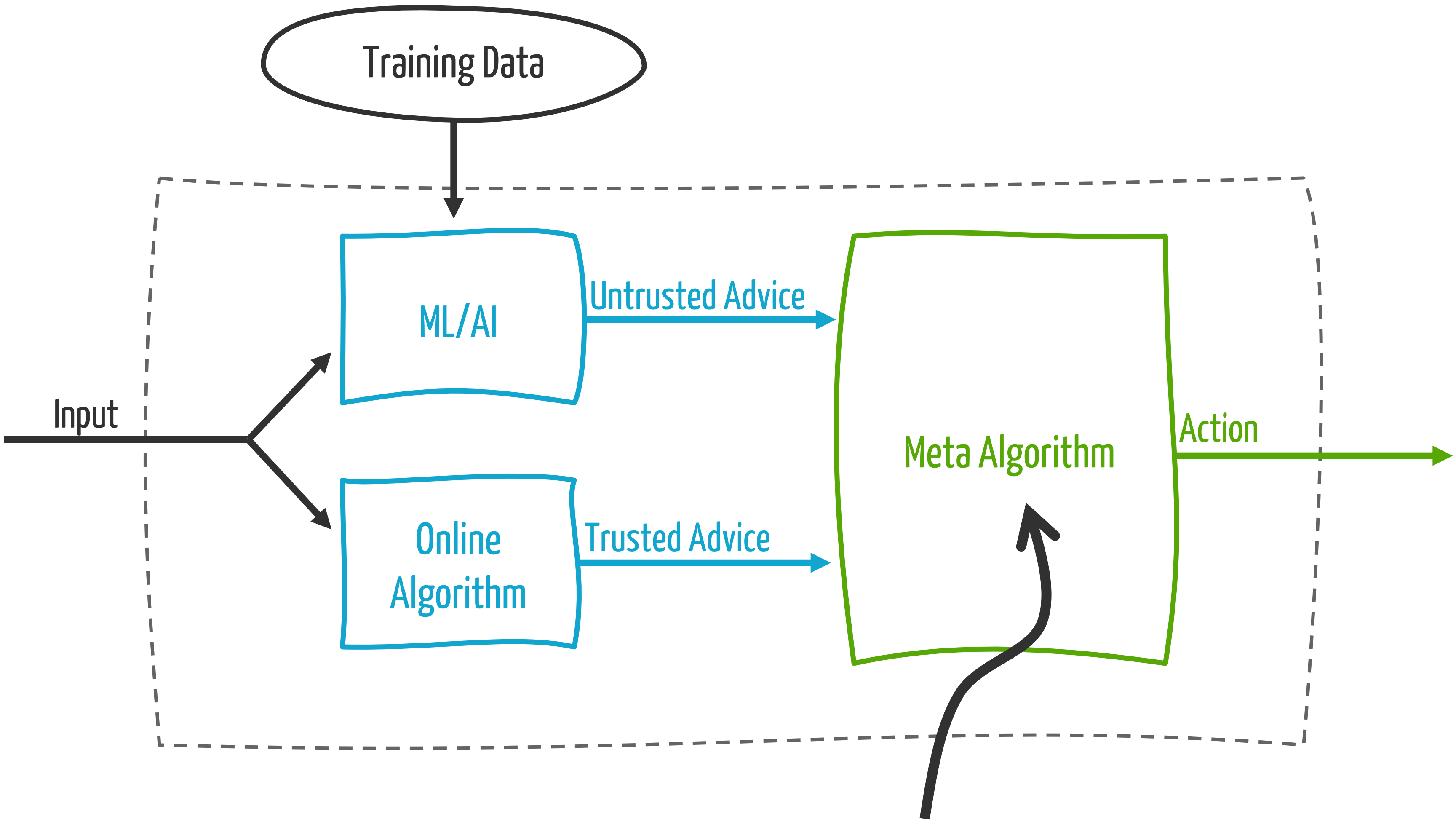
Q-value advice can improve upon black-box advice in terms of robustness-consistency tradeoffs in MDPs.

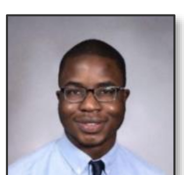
Many open problems remain

- Improved lower bounds on grey-box or black-box advice?
- Improved algorithms?
- End to end analysis including sample complexity trade-offs?
- Other forms of "grey box" information?
- Benefits from other forms of advice, e.g., predictions of P_t?

...

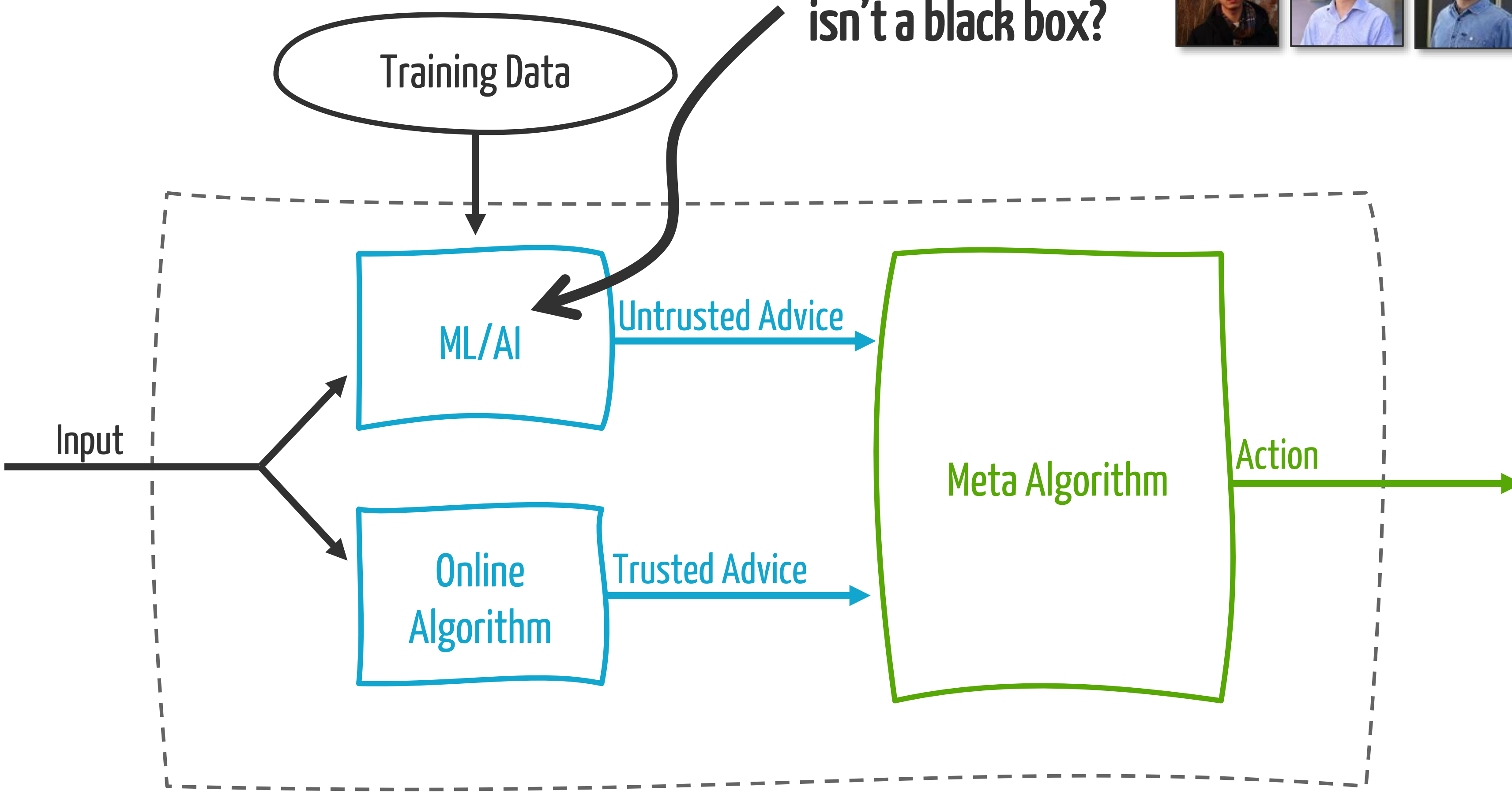
This is just the tip of the iceberg for understanding learning augmented algorithms...

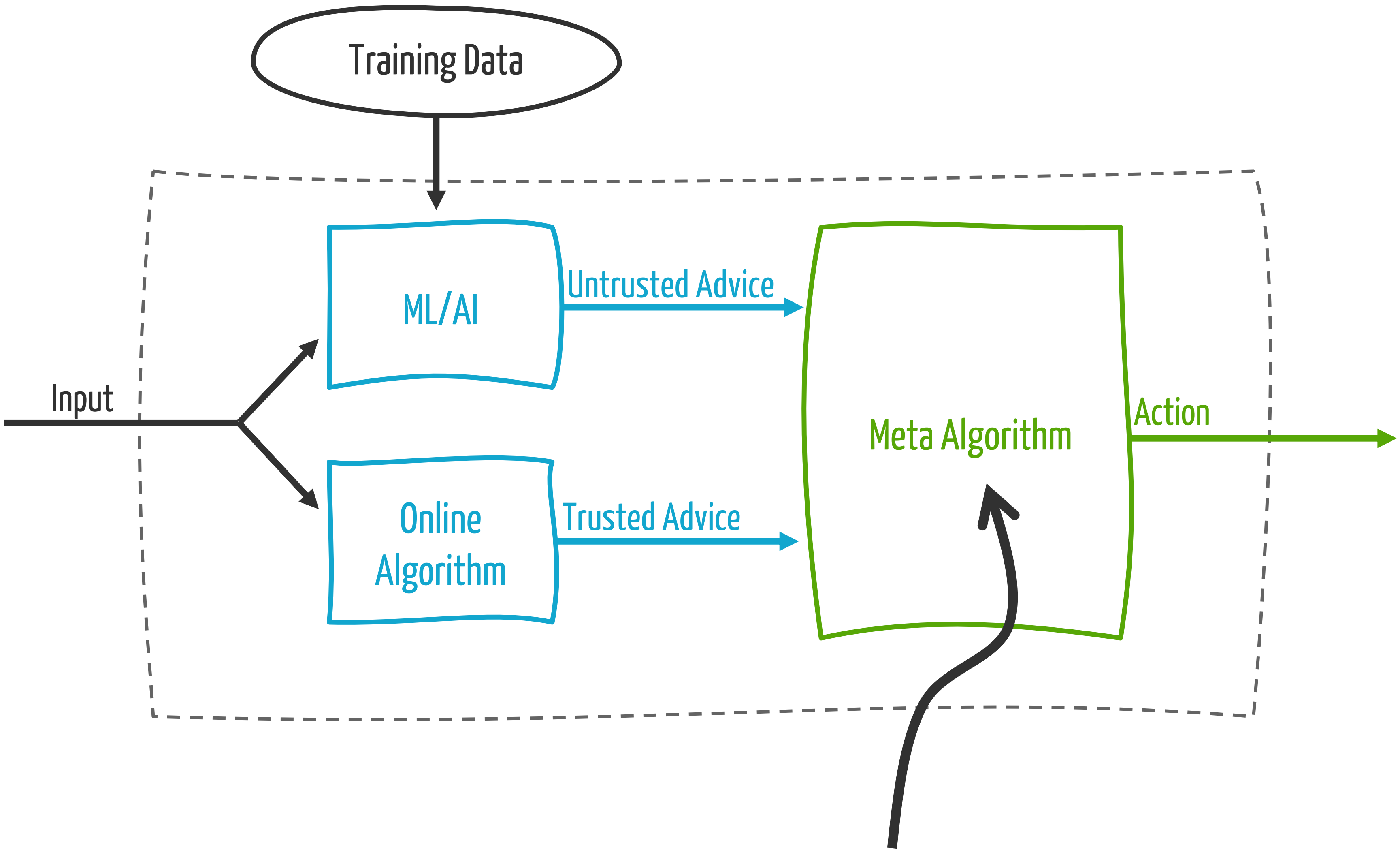




How should advice be used?
Switch between them? Combine them?

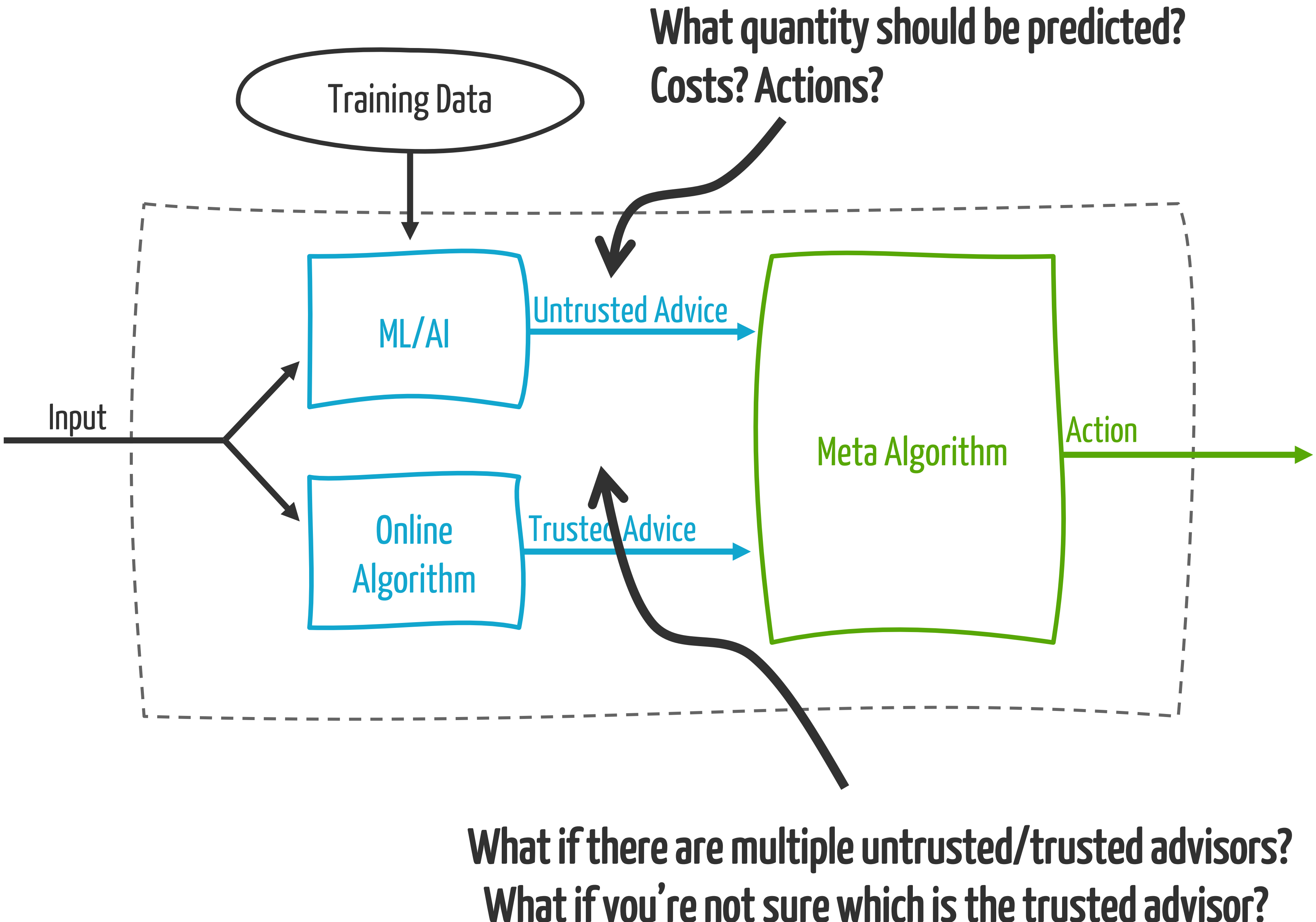
What if the learning isn't a black box?



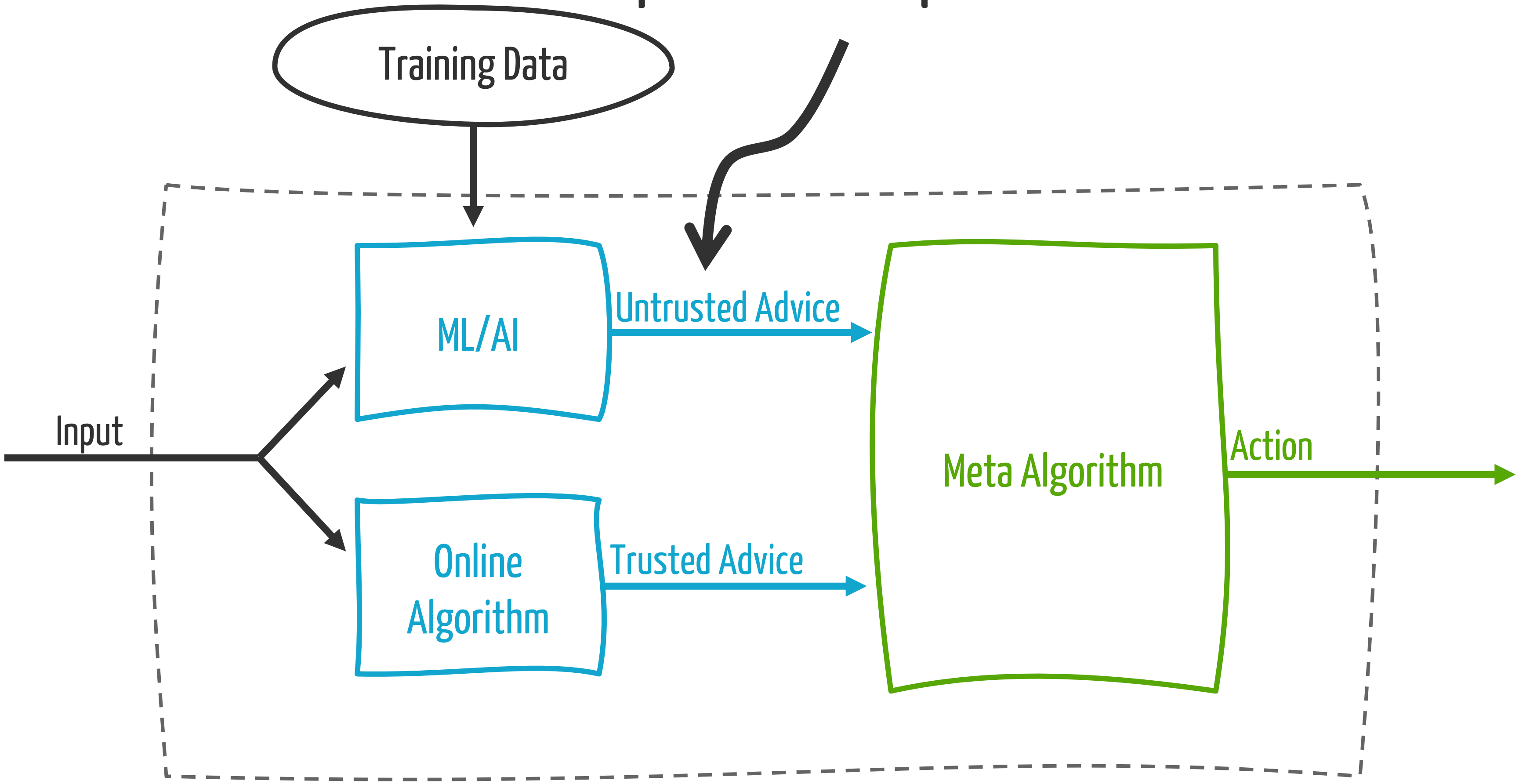


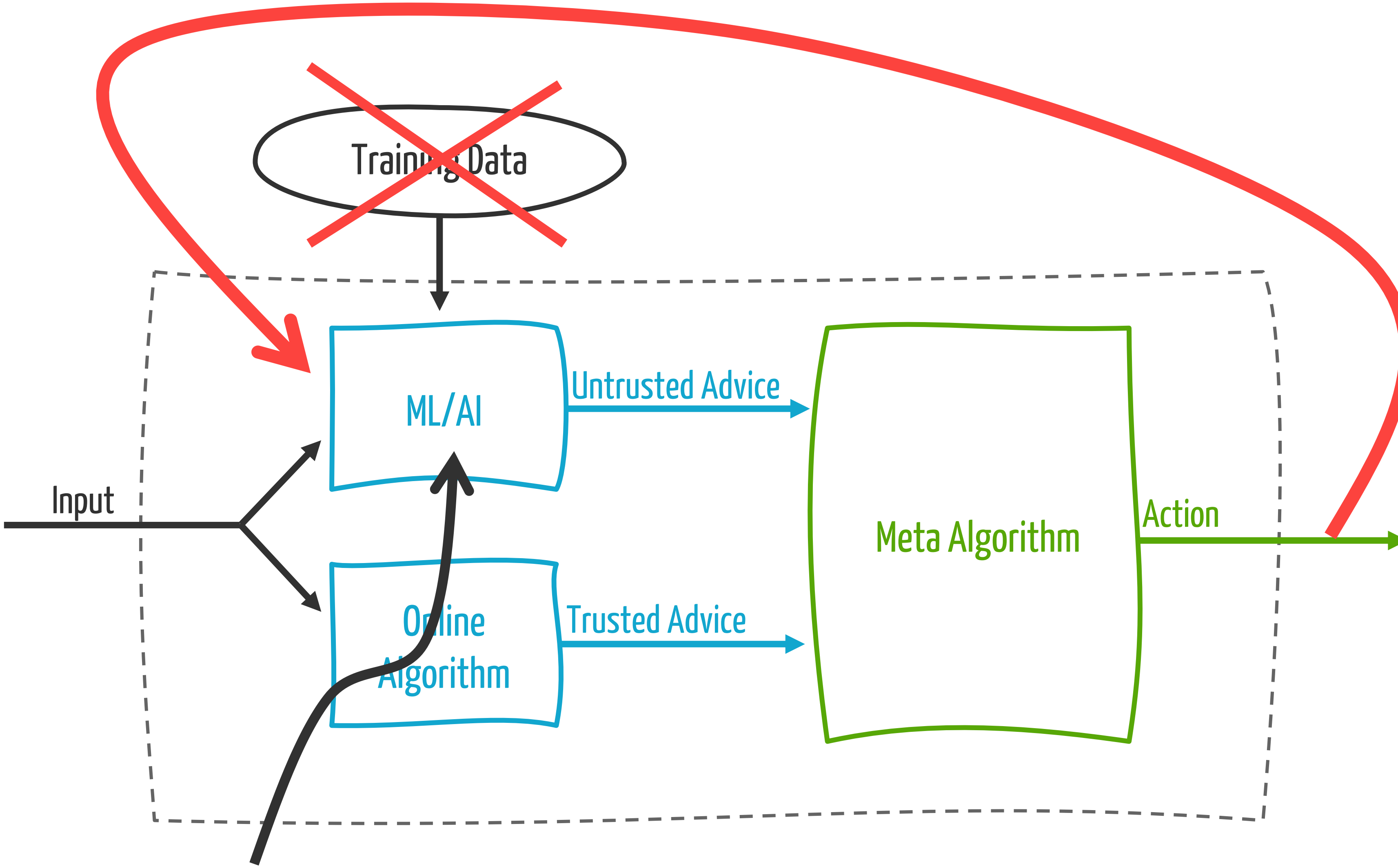
Can we move beyond robustness & consistency?

Average-case? Smoothness? Frugality? Memory-dependence?

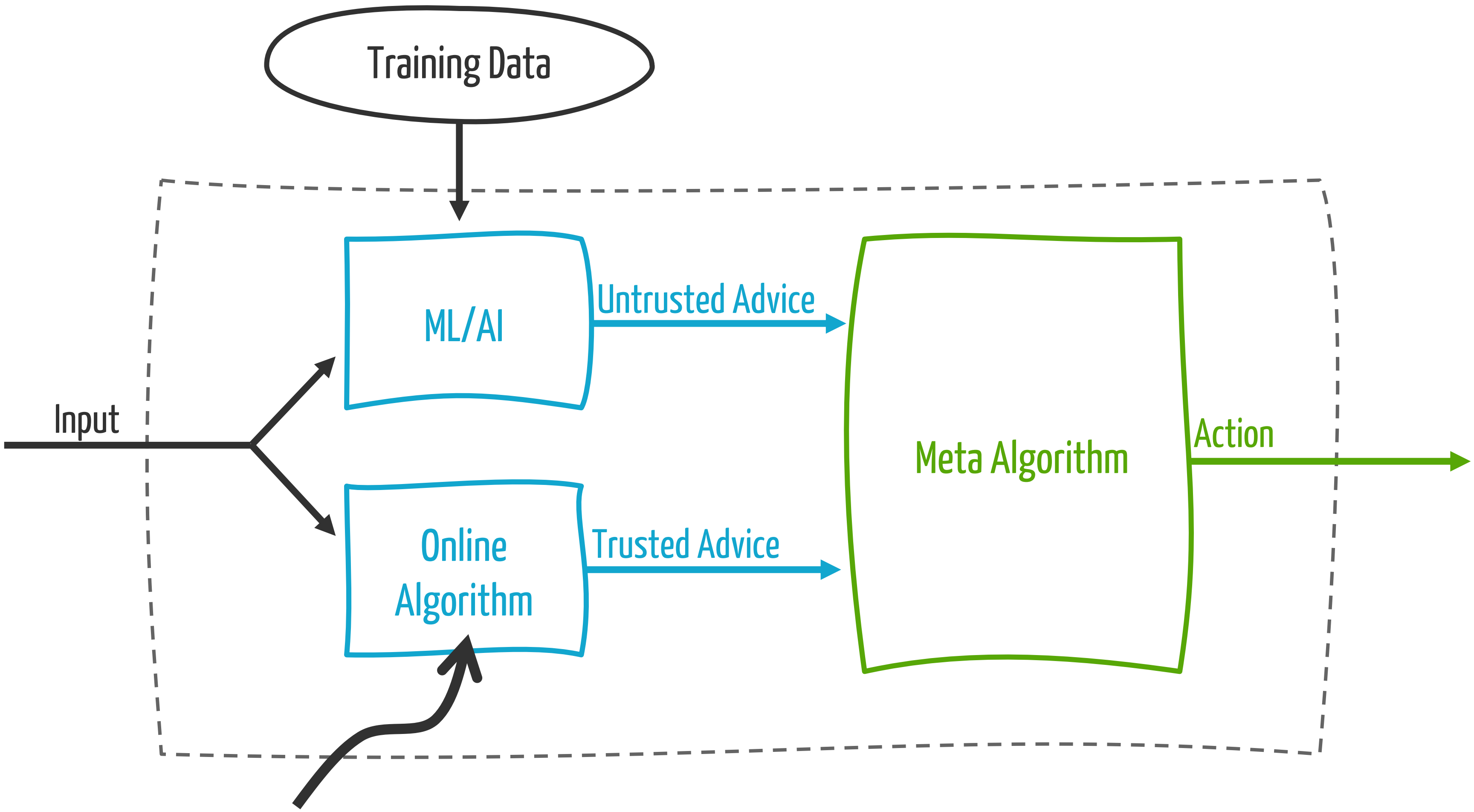

What quantity should be predicted?
Costs? Actions?
Training Data
ML/AI
Untrusted Advice
Input
Online
Algorithm
Trusted Advice
Meta Algorithm
Action
What if there are multiple untrusted/trusted advisors?
What if you're not sure which is the trusted advisor?

What is the value of uncertainty quantification of predictions?





What if the ML model is trained online?



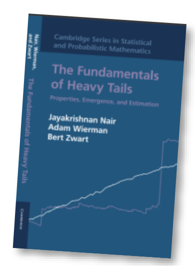
What if the model needs to be learned?

Learning-Augmented Algorithms for MDPs

Adam Wierman, Caltech

- T Li, R Yang, G Qu, G Shi, C Yu, A Wierman, S Low. Robustness and Consistency in Linear Quadratic Control with Untrusted Predictions. Sigmetrics 2022
- C Yeh, J Yu, Y Shi, A Wierman. Robust Online Voltage Control with an Unknown Grid Topology. E-Energy 2022.
- N Christianson, T Handina, A Wierman. Chasing Convex Bodies and Functions with Black-Box Advice. COLT 2022.
- Y Hu, G Qu, A Wierman. On the Sample Complexity of Stabilizing LTI Systems on a Single Trajectory. NeurIPS 2022.
- N Christianson, J Chen, A Wierman. Optimal Robustness-Consistency Tradeoffs for Learning-Augmented Metrical Task Systems. AIStats 2023.
- D Rutten, N Christianson, D Mukherjee, A Wierman. Online Non-convex Optimization with Untrusted Advice. Sigmetrics 2023.
- J Yu, D Ho, A Wierman. Online Stabilization of Unknown Networked Systems with Communication Constraints. Sigmetrics 2023.
- T Li, R Yang, G Qu, Y Lin, A Wierman, S Low. Certifying Black-Box Policies with Stability for Nonlinear Control. IEEE J of Control Sys. 2023.
- Y Lin, J Preiss, E Anand, Y Li, Y Yue, A Wierman. Online Adaptive Controller Selection in Time Varying Systems. NeurIPS 2023.
- T Li, Y Lin, S Ren, A Wierman, S Ren. Beyond Black-Box Advice: Learning-Augmented Algorithms for MDPs with Q-Value Predictions. NeurIPS 2023.
- B. Sun, J. Huang, N. Christianson, M. Hajiesmaili, A Wierman. Online Algorithms with Uncertainty-Quantified Predictions. ICML 2024.
- N Bhuyan, D Mukherjee, A Wierman. Best of both worlds: Stochastic and adversarial convex function chasing. ICML 2024

Case studies done using SustainGym



New(ish) book on heavy tails!