# Decision-focused learning: theory, applications, challenges

Andrew Perrault

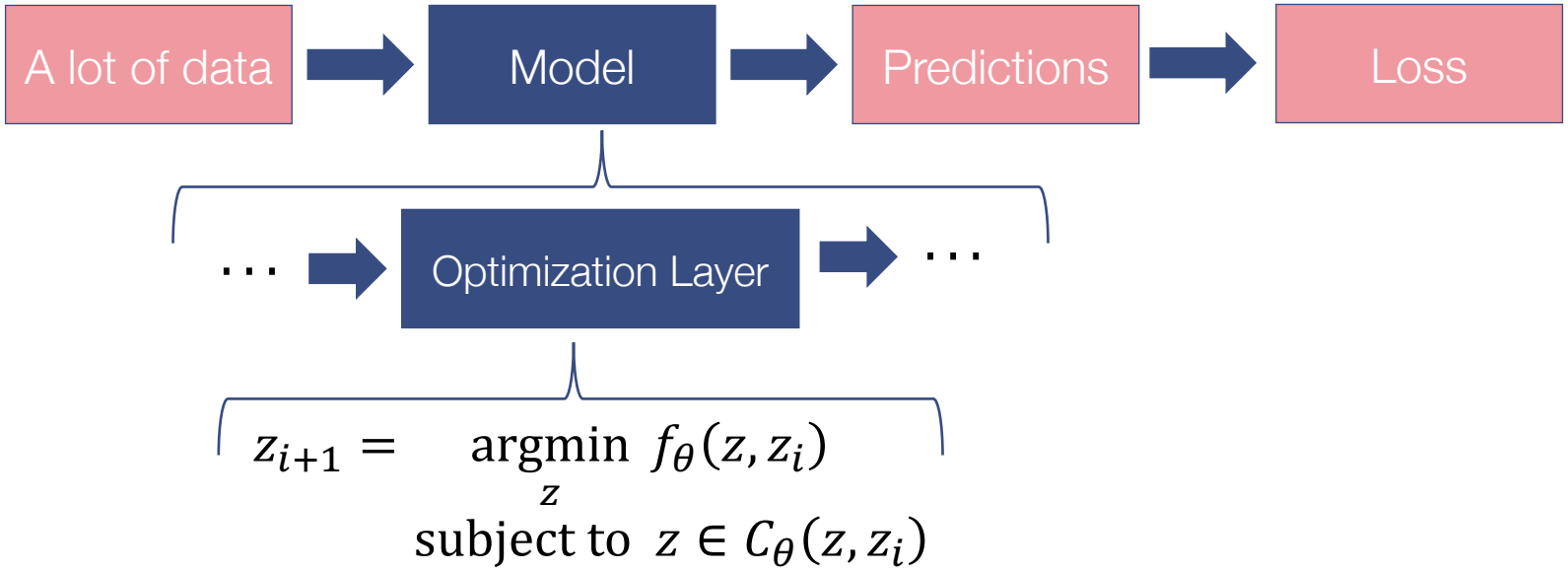
Department of Computer Science and Engineering

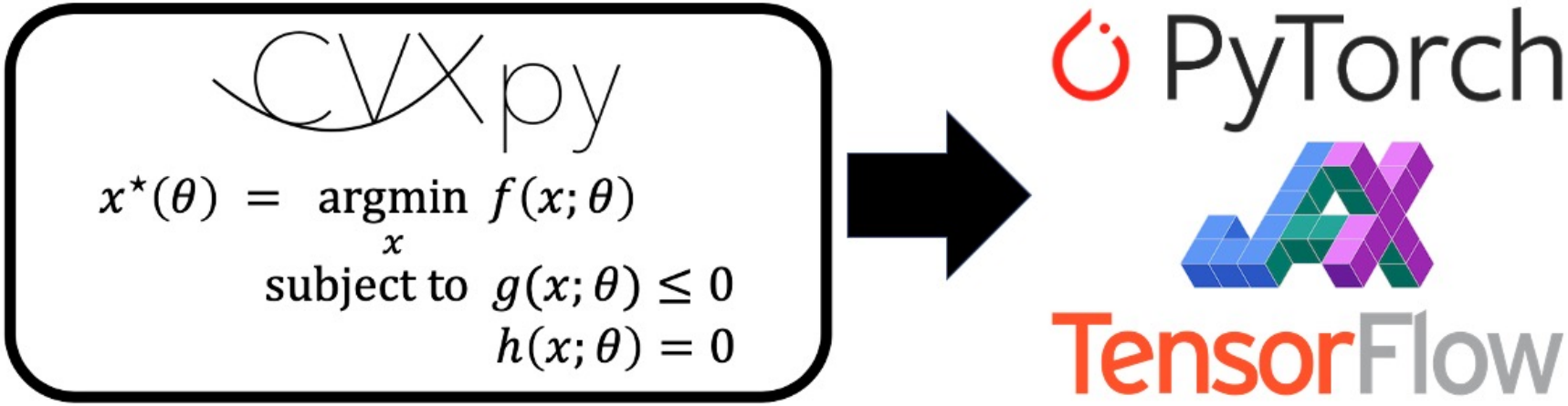
The Ohio State University

IJCAI 2022


THE OHIO STATE UNIVERSITY

# Brandon's talk recap



# Agenda

What is decision-focused learning? Why should you care?

Decision-focused learning for sequential decision problems (NeurIPS 2021)

Learning convex & smooth loss functions for decision-focused learning (arXiv)

# Notation: predict-then-optimize

Observable feature/context vector $x \in X$

Unobserved ground truth environment parameter $\theta^* \in \Theta$

$x$ and $\theta^*$ drawn jointly from fixed distribution $\mathcal{D}$

Decision/control variable $z \in Z$

Objective function $f(z, \theta^*)$

Goal: $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

Wilder et al. (AAAI 2019)

# Notation: predict-then-optimize

Observable feature/context vector $x \in X$

Unobserved ground truth environment parameter $\theta^* \in \Theta$

$x$ and $\theta^*$ drawn jointly from fixed distribution $\mathcal{D}$

Observed at training time, but not test time

Decision/control variable $z \in Z$

Objective function $f(z, \theta^*)$

Goal: $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

# Notation: predict-then-optimize

Observable feature/context vector $x \in X$

Unobserved ground truth environment parameter $\theta^* \in \Theta$

$x$ and $\theta^*$ drawn jointly from fixed distribution $\mathcal{D}$

Decision/control variable $z \in Z$

Objective function $f(z, \theta^*)$

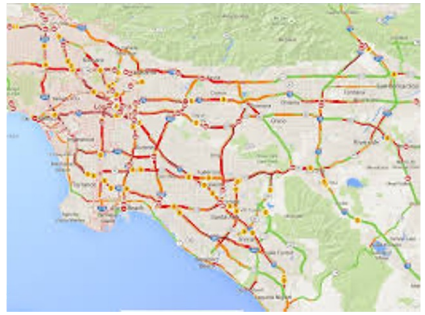
Goal: $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

**Predict-then-optimize routing**

**Context $x$**

**Decision: route $z$**

**Objective $f(x, z)$: minimize travel time under traffic**



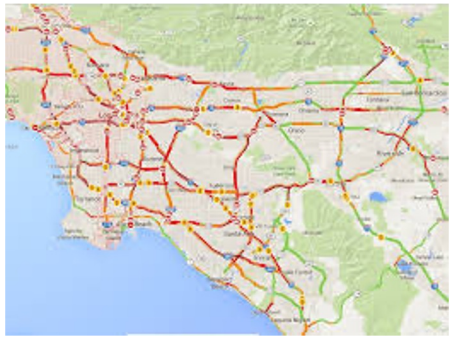
Wilder et al. (AAAI 2019)

# Motivation of predict-then-optimize

**Data =**

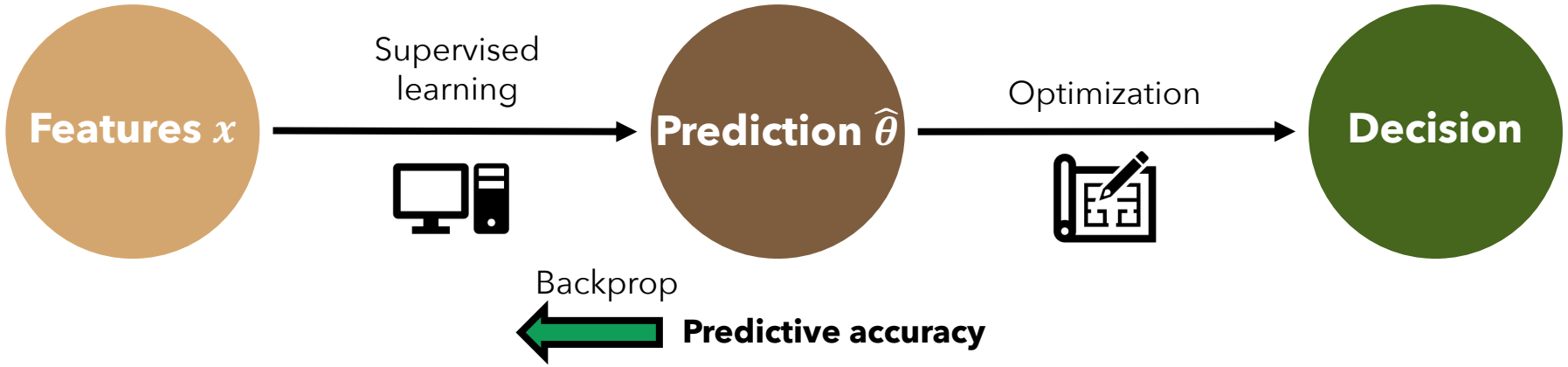
$$\{( \quad , \quad )\}$$

$i$



$i$

# Two-stage pipeline for predict-then-optimize



# What do we lose by simplifying?

$\min_{z \in Z} f(z, \mathbb{E}(\theta|x))$ instead of $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

Two-stage

What we want

# What do we lose by simplifying?

$\min_{z \in Z} f(z, \mathbb{E}(\theta|x))$ instead of $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

Consider $f(z, \theta) = \sum_i \theta_i z_i$. Then,

$$\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta) = \min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_i z_i$$

$$= \min_{z \in Z} \sum_i \mathbb{E}[\theta_i|x] z_i = \min_{z \in Z} f(z, \mathbb{E}(\theta|x))$$

# Product of predictions

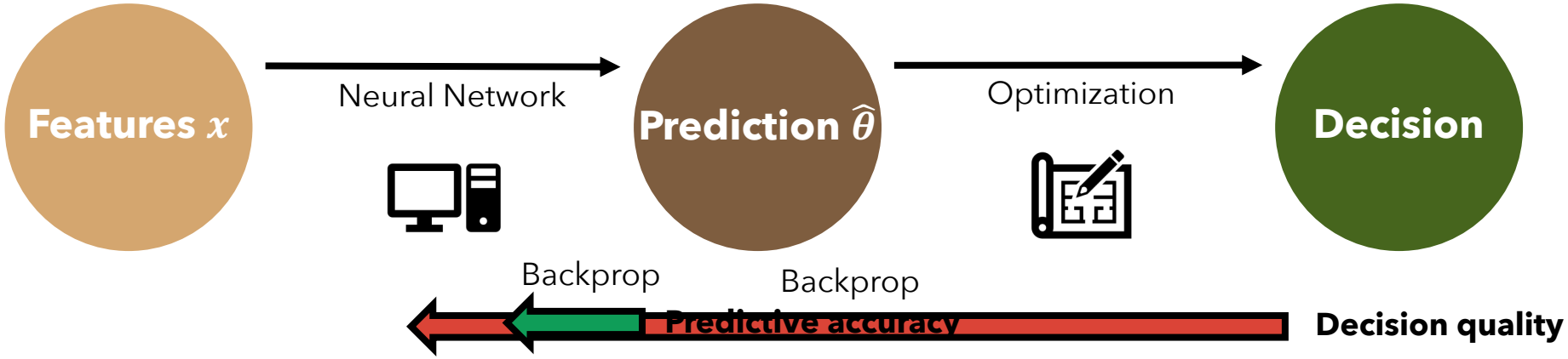
$\min_{z \in Z} f(z, \mathbb{E}(\theta|x))$ instead of $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

Consider $f(z, \theta) = \sum_i \theta_{2i}\theta_{2i+1}z_{2i}$. Then,

$$\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta) = \min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_{2i}\theta_{2i+1}z_{2i}$$

$$\neq \min_{z \in Z} \sum_i \mathbb{E}[\theta_{2i}|x]\mathbb{E}[\theta_{2i+1}|x]z_{2i}$$

# Does differentiable optimization help?




Wilder et al. (AAAI 2019)

Theorem (Cameron et al., 2022): There exists a class of problems (two-stage minimum cost flow from Agrawal et al., 2012) such that:

- $\mathbf{DFL} = \mathbf{OPT}$
- $\frac{\mathbf{2S}}{\mathbf{DFL}}$ is at least $\Omega(2^d)$, where $d = \dim(\mathrm{Z}) = \dim(\Theta)$

# Optimizing policies in unknown MDPs

Environment (MDP)

RL solver

RL policy

# Planning for Community Health Workers

Key tasks:
Maternal health checkups
Behavioral health interventions
Medication adherence support

Photo credit: Pippa Ranger

# Adherence monitoring for TB

Tuberculosis (TB): 500,000 deaths/year, 3 million infected in India

Curing TB requires 6 months of daily medication

Symptoms disappear quickly

Mate, Killian, Xu, P, Tambe (NeurIPS 2020)

# TB medication adherence

today | February | 2016

enrollment/end | received | shared phone numbers | manual | not on 99DOTS | missed (hover for reason)

**ART Programs**

Showing: NKL

| ID # | Attention Required | Patient |
|---|---|---|
| 6204 | MEDIUM | |
| 6214 | MEDIUM | |
| 6218 | MEDIUM | |
| 6231 | MEDIUM | |
| 6232 | MEDIUM | |
| 6235 | MEDIUM | |
| 6260 | MEDIUM | |
| 6264 | MEDIUM | |
| 6272 | MEDIUM | |
| 6287 | MEDIUM | |
| 6312 | MEDIUM | |
| 6334 | HIGH | |

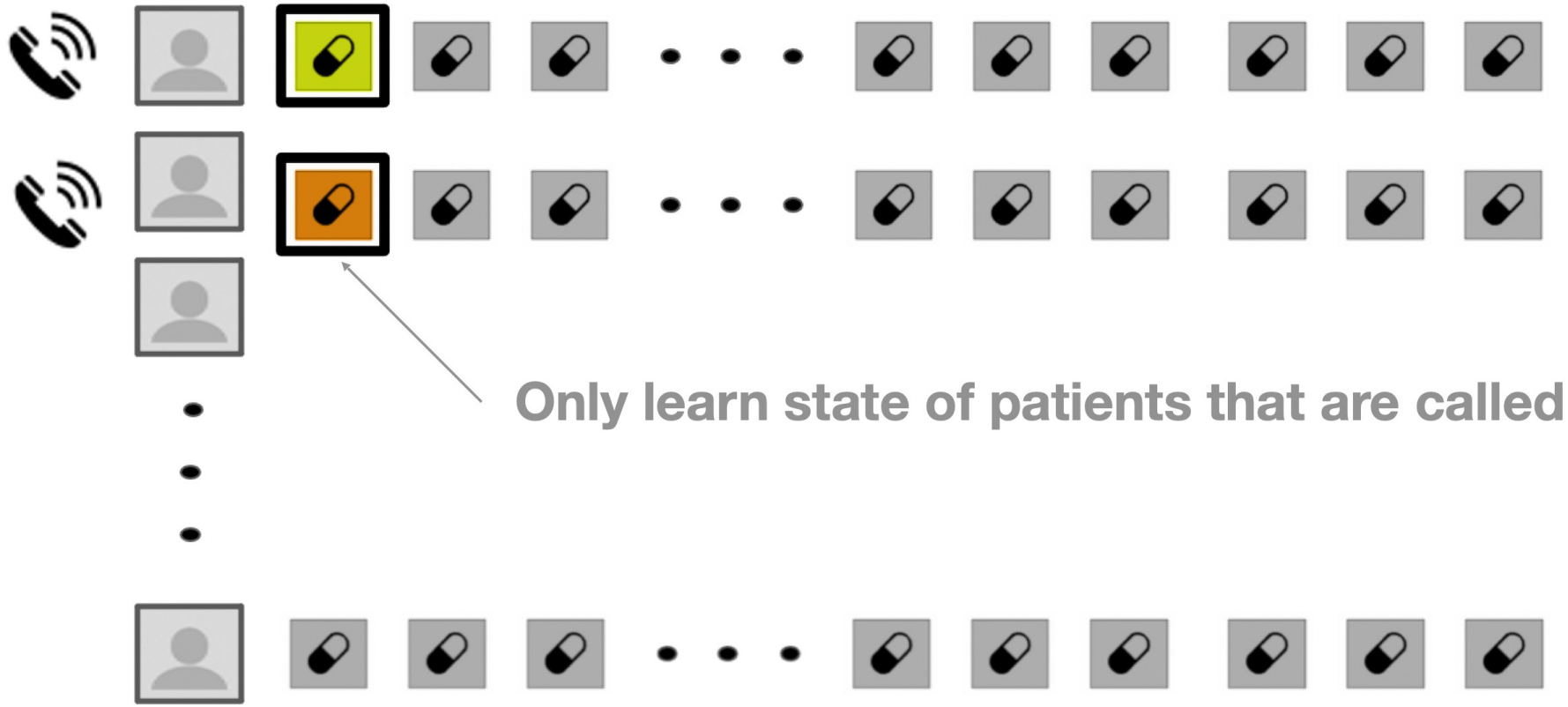
Everwell

Mate, Killian, Xu, P, Tambe (NeurIPS 2020)

# Adherence example

6 months

# Adherence example

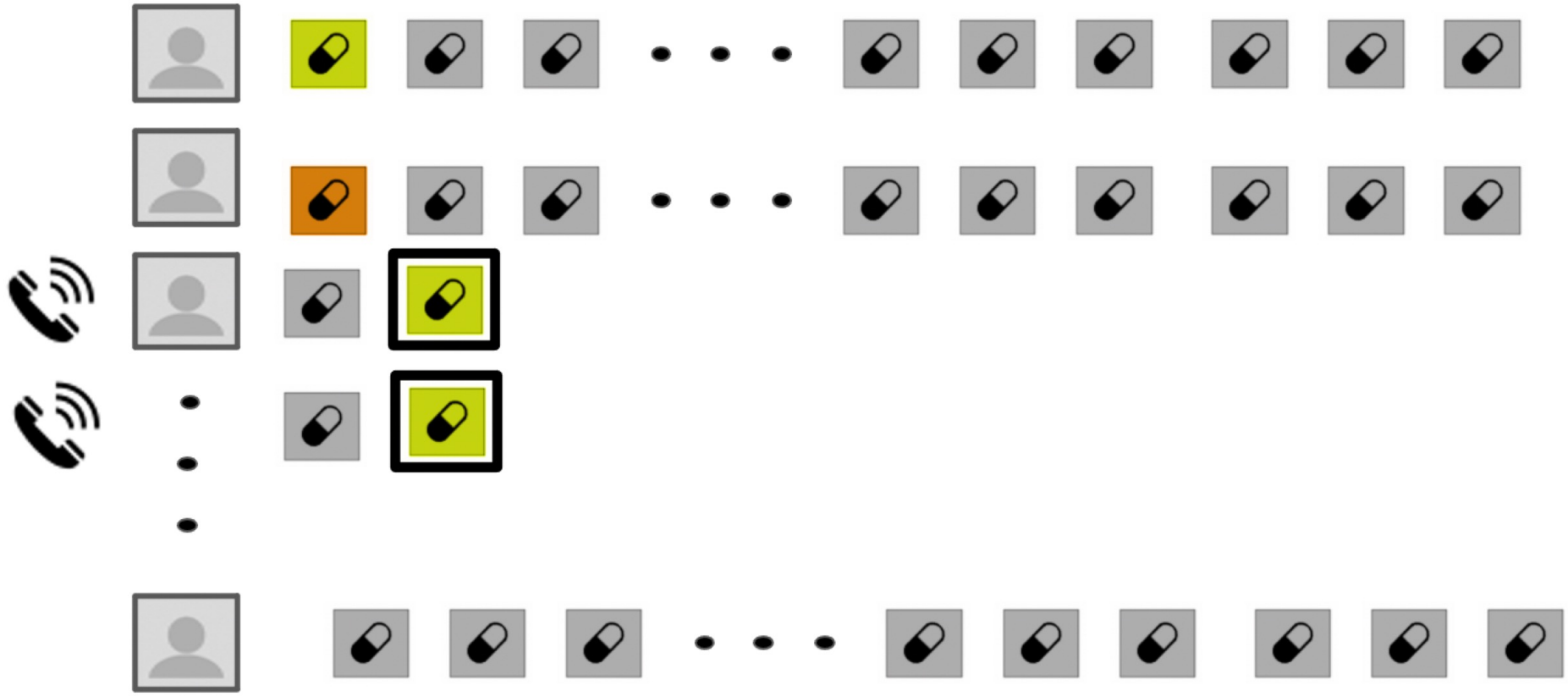



Adhering

Not adhering

Unknown

# Adherence example



Adhering

Not adhering

Unknown

# POMDP formulation

Each patient is represented by a partially observable Markov decision process (POMDP)

Shared action budget across all processes

States 

Actions 

Observations 

Transitions $P_n^{\text{call}}$ $P_n^{\text{no call}}$

Rewards  = 1  = 0

Goal: find optimal policy $\pi^*$—a mapping from the action-observation history to actions that maximizes reward

# POMDP formulation

Each patient is represented by a partially observable Markov decision process (POMDP)

Shared action budget across all processes

States 

Actions 

Observations 

Transitions $P_n$ $P_n$

Rewards  = 1  = 0

Unknown and different per patient!

Goal: find optimal policy $\pi^*$—a mapping from the action-observation history to actions that maximizes reward

# Patient data

Patient context $x$

Trajectory $\{(s_t, a_t, r_t, s_{t+1})\}$

# Predict-then-optimize?

Observable feature/context vector $x \in X$

Unobserved ground truth environment parameter $\theta^* \in \Theta$

$x$ and $\theta^*$ drawn jointly from fixed distribution $\mathcal{D}$

Decision/control variable $z \in Z$

Objective function $f(z, \theta^*)$

Goal: $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

# Predict-then-optimize?

Observable feature/context vector
$x \in X$

Unobserved ground truth environment parameter $\theta^* \in \Theta$

$x$ and $\theta^*$ drawn jointly from fixed distribution $\mathcal{D}$

Not available at training time–only have trajectories

Decision/control variable $z \in Z$

Objective function $f(z, \theta^*)$

Goal: $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

# Contributions

How to differentiate through the optimal policy in sequential decision problems

(and how to speed it up)

Do DFL with only trajectories (i.e., without $\theta^*$ labels)

# Problem setting

Let $(\mathcal{S}, s_0, A, T, R)$ be an *incomplete* MDP specification with:

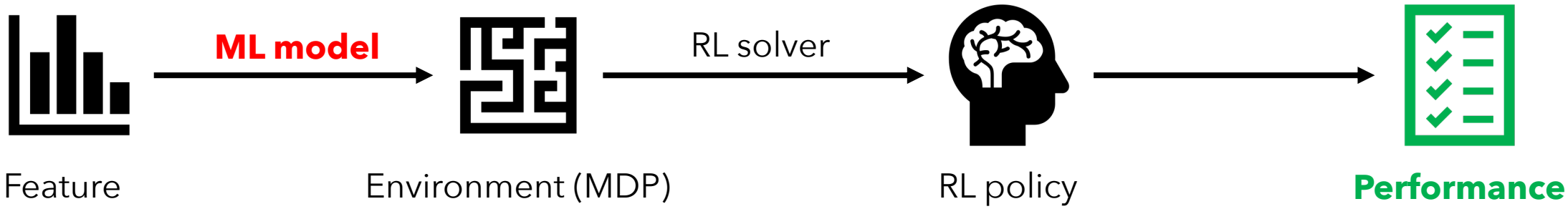
- Initial state $s_0$
- Set of states $\mathcal{S}$
- Set of actions $A$
- Possibly incomplete transition function: $T: \mathcal{S} \times A \to \Delta\mathcal{S}$
- Possible incomplete reward function: $\mathrm{R}: \mathcal{S} \times A \to \mathbb{R}$

Each MDP has a context vector $\boldsymbol{x}$

Each training MDP has a set of trajectories $\mathcal{T}$

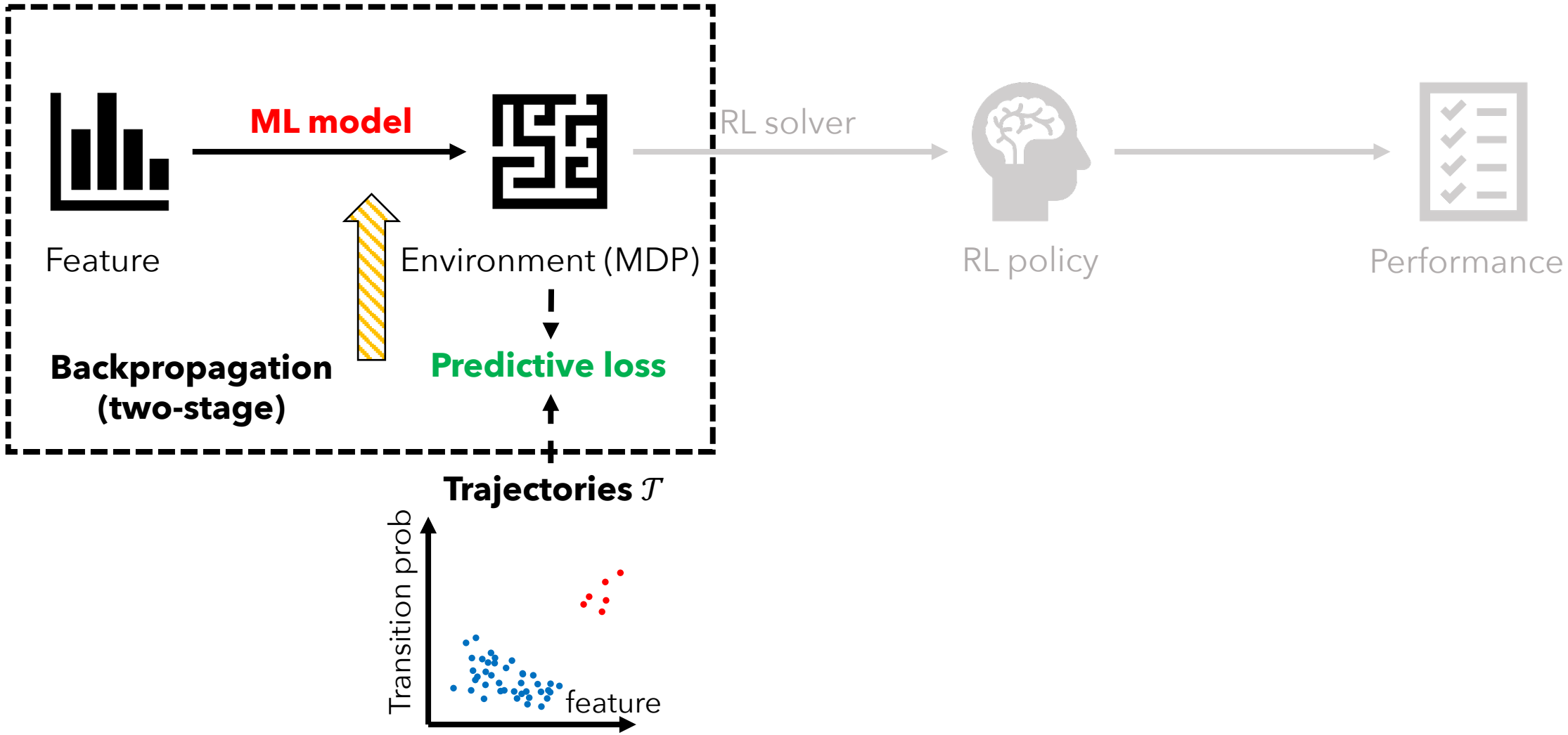
Goal: learn a predictive model $m_w$ that "completes" the MDP from $\boldsymbol{x}$ and maximizes reward on $\mathcal{T}$

# Pipeline visualized

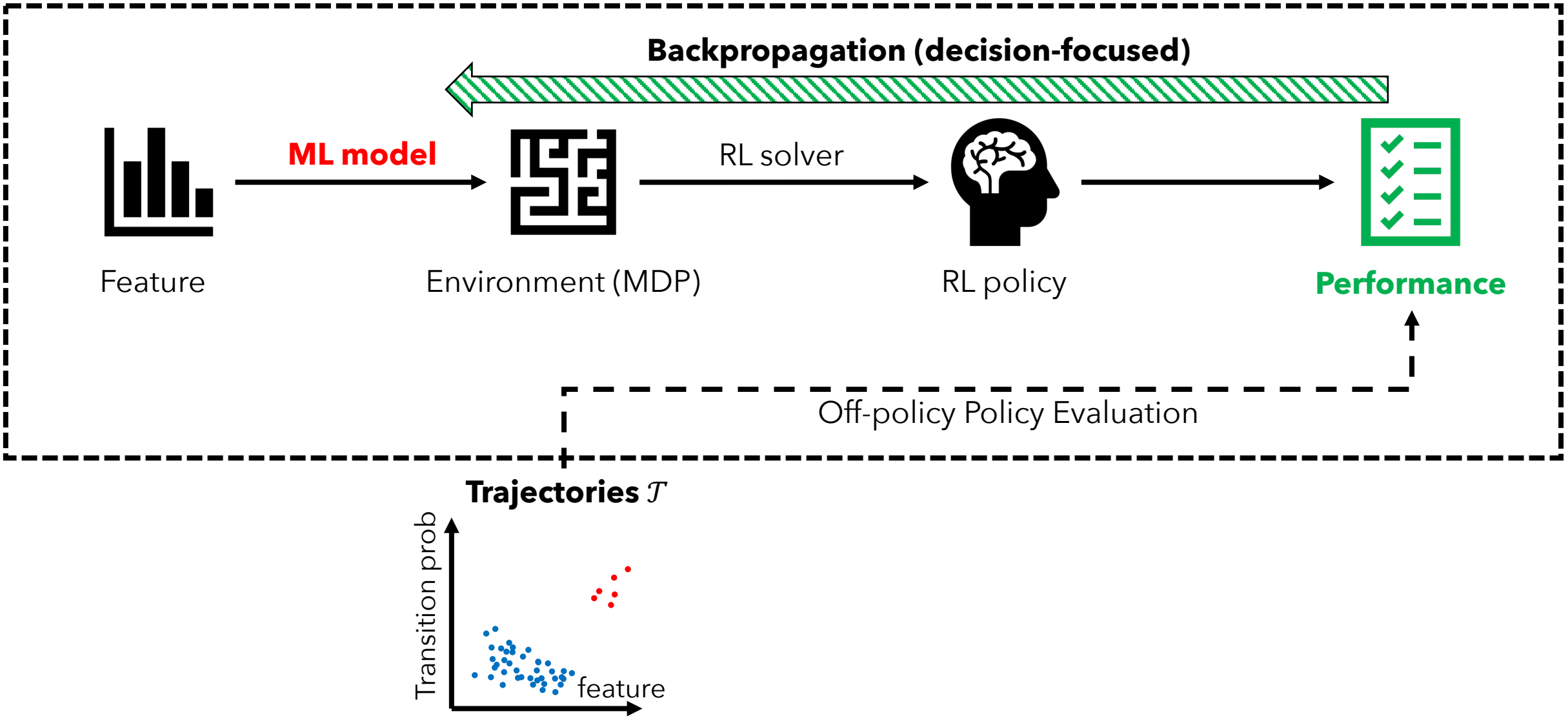



Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

# Two-stage approach

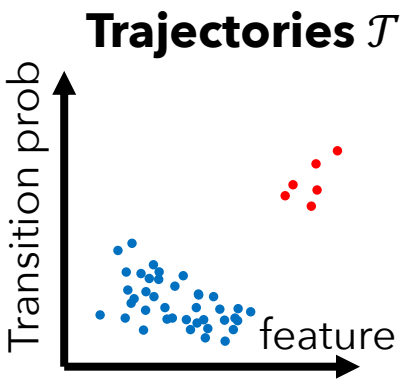


# Decision-focused approach




Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

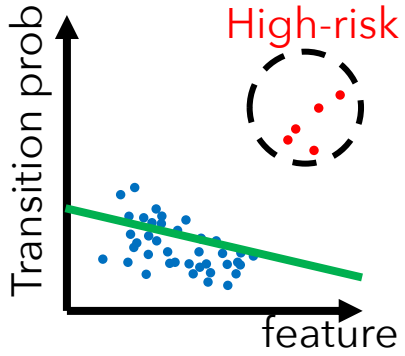
# Two-stage vs. DFL



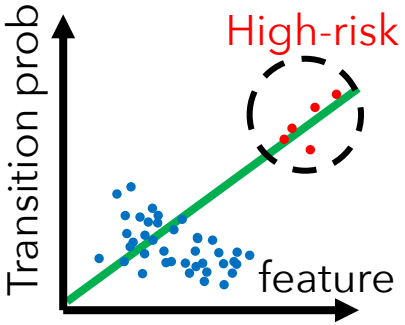

**Trajectories** $\mathcal{T}$

**Two-stage**
Focus on predictive accuracy

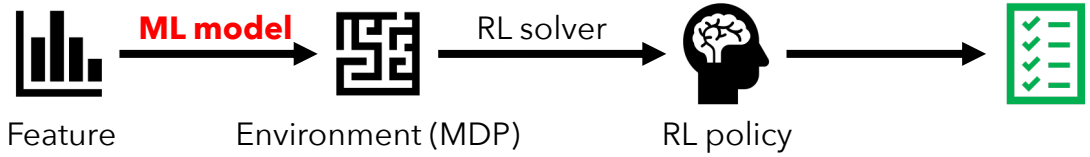
**Decision-focused**
Focus on decision quality




High predictive accuracy
Low decision quality




Low predictive accuracy
High decision quality




Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

# How DFL works

To run gradient descent, we need to compute

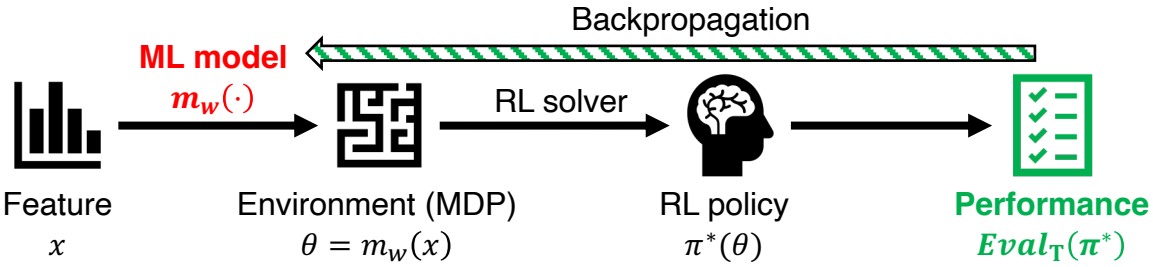
$$\frac{dEval_{\mathrm{T}}(\pi^*)}{dw} = \frac{d\theta}{dw}\frac{\boldsymbol{d\pi^*}}{\boldsymbol{d\theta}}\frac{dEval_{\mathrm{T}}(\pi^*)}{d\pi^*}$$

**Weights $w$ of predictive model**

**We use a metric from Futoma et al.**

$$\mathrm{Eval}_{\mathcal{T}}(\pi) := V^{\mathrm{CWPDIS}}(\pi) - \lambda_{\mathrm{ESS}}/\sqrt{\mathrm{ESS}(\pi)} \quad (1)$$

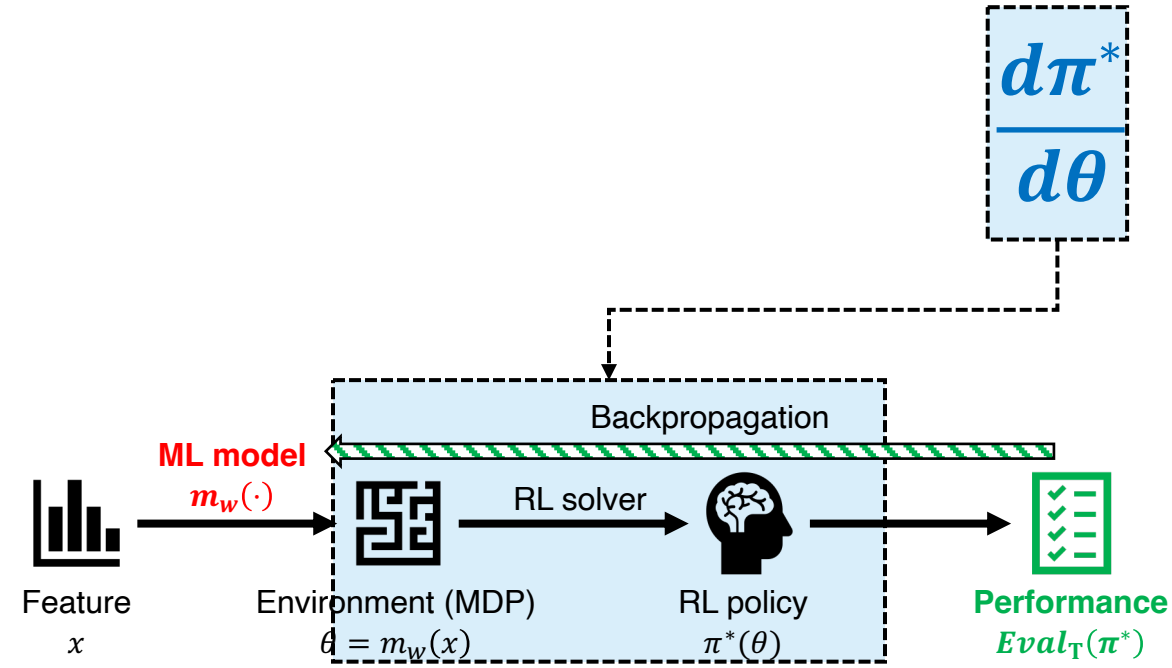
where $V^{\mathrm{CWPDIS}}(\pi) := \sum_{t=1}^{h} \gamma^t \frac{\sum_i r_{it}\rho_{it}(\pi)}{\sum_i \rho_{it}(\pi)}$ and $\mathrm{ESS}(\pi) := \sum_{t=1}^{h} \frac{(\sum_i \rho_{it})^2}{\sum_i \rho_{it}^2}$, and $\rho_{it}(\pi)$ is the ratio of the proposed policy and the behavior policy likelihoods up to time t: $\rho_{it}(\pi) := \prod_{t'=1}^{t} \frac{\pi(a_{it'}|s_{it'})}{\pi_{\mathrm{beh}}(a_{it'}|s_{it'})}$.




Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

# Leveraging optimality conditions

Differentiate through one-shot optimization problem by optimality condition (Amos et al., 2017)



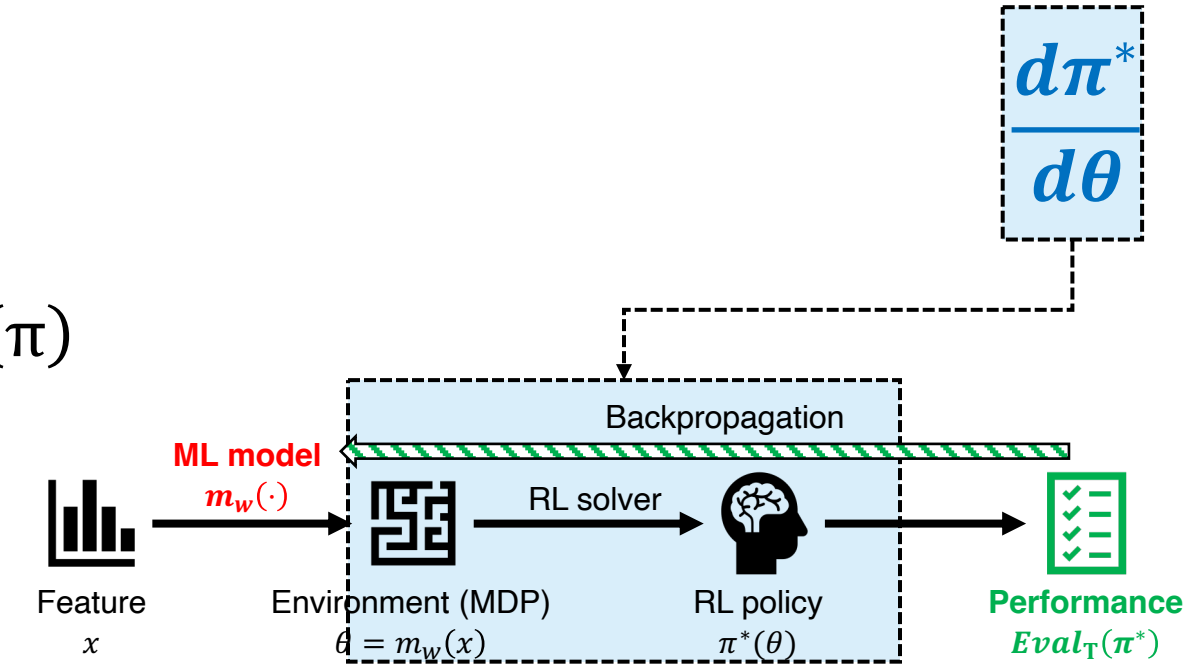
# Leveraging optimality conditions

Expected cumulative return

Policy optimality: $\pi^* = \text{argmax}_\pi \, J_\theta(\pi)$

Bellman optimality: $\pi^* = \text{argmin}_\pi \, B_\theta(\pi)$

Differentiate through one-shot optimization problem by optimality condition (Amos et al., 2017)



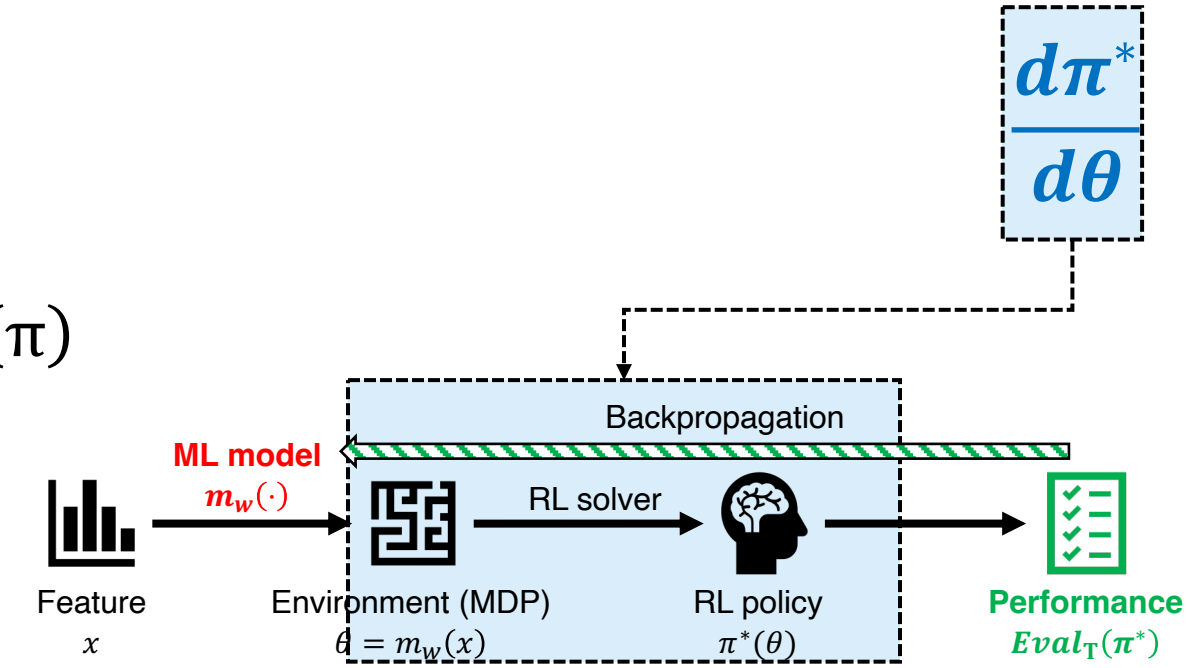
# Leveraging optimality conditions

Expected cumulative return

Policy optimality: $\pi^* = \text{argmax}_\pi \, J_\theta(\pi)$

KKT conditions: $\nabla_\pi J_\theta(\pi^*) = 0$

Bellman optimality: $\pi^* = \text{argmin}_\pi \, B_\theta(\pi)$




Differentiate through one-shot optimization problem by optimality condition (Amos et al., 2017)

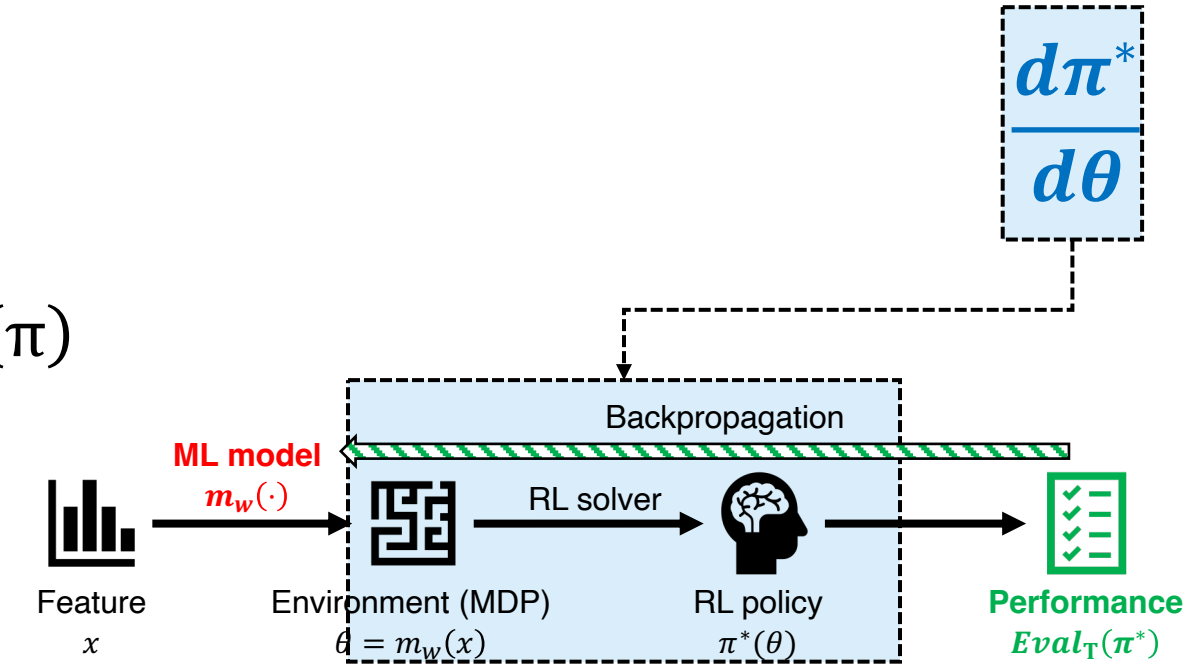
# Leveraging optimality conditions

Expected cumulative return

Policy optimality: $\pi^* = \text{argmax}_\pi \, J_\theta(\pi)$

KKT conditions: $\frac{d}{d\theta} \nabla_\pi J_\theta(\pi^*) = 0$

Bellman optimality: $\pi^* = \text{argmin}_\pi \, B_\theta(\pi)$




Differentiate through one-shot optimization problem by optimality condition (Amos et al., 2017)

# Leveraging optimality conditions

Expected cumulative return

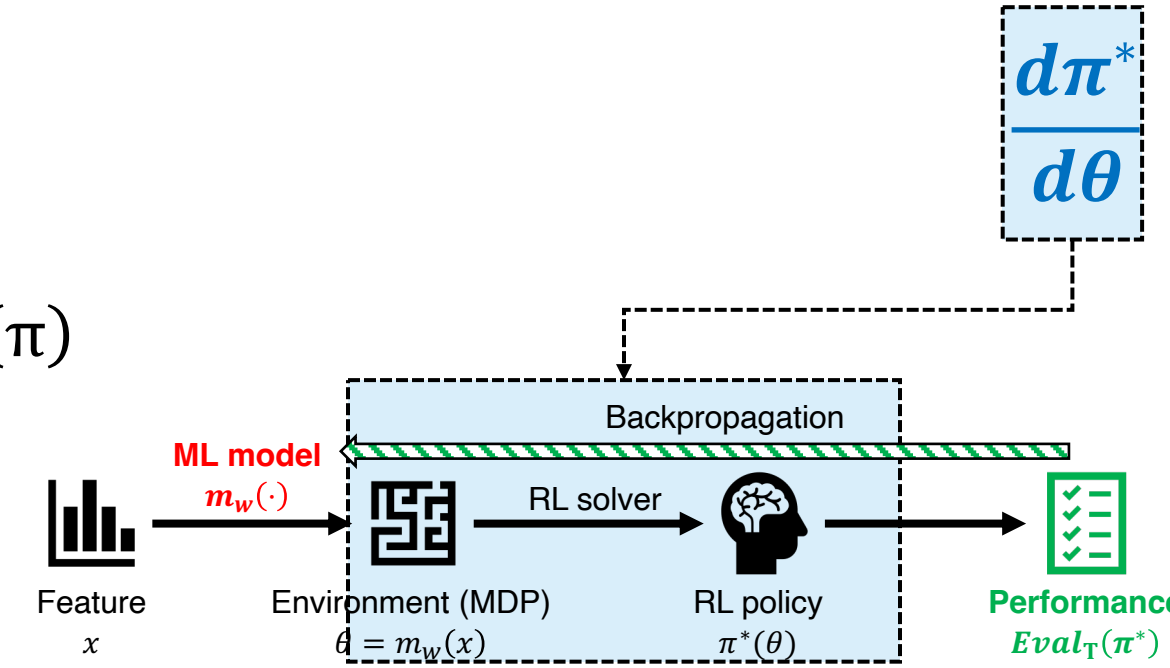
Policy optimality: $\pi^* = \text{argmax}_\pi \, J_\theta(\pi)$

KKT conditions: $\frac{d}{d\theta} \nabla_\pi J_\theta(\pi^*) = 0$

$$\nabla^2_\pi J_\theta(\pi^*) \frac{\mathbf{d}\boldsymbol{\pi}^*}{\mathbf{d}\boldsymbol{\theta}} + \nabla^2_{\theta\pi} J_\theta(\pi^*) = 0$$

Bellman optimality: $\pi^* = \text{argmin}_\pi \, B_\theta(\pi)$

Differentiate through <u>one-shot optimization problem</u> by optimality condition (Amos et al., 2017)



# Leveraging optimality conditions

Expected cumulative return (implicitly defined)

Policy optimality: $\pi^* = \text{argmax}_\pi \, J_\theta(\pi)$

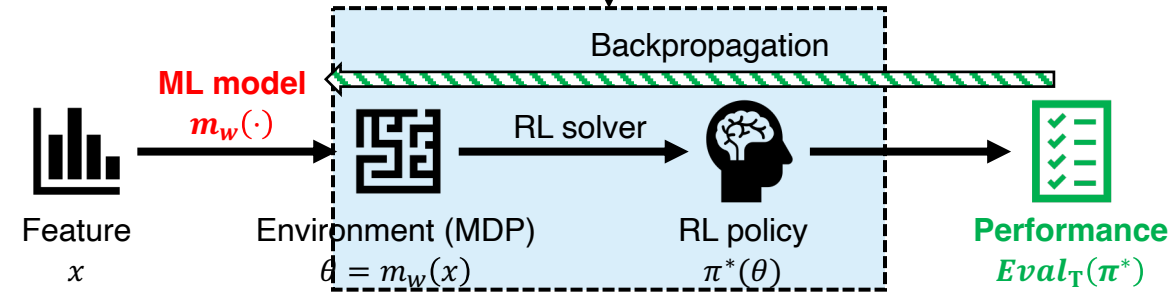
KKT conditions: $\frac{d}{d\theta} \nabla_\pi J_\theta(\pi^*) = 0$

$$\nabla^2_\pi J_\theta(\pi^*) \frac{\mathbf{d}\boldsymbol{\pi}^*}{\mathbf{d}\boldsymbol{\theta}} + \nabla^2_{\theta\pi} J_\theta(\pi^*) = 0$$

$$\frac{\boldsymbol{d\pi}^*}{\boldsymbol{d\theta}} = \left(\nabla^2_\pi J_\theta(\pi^*)\right)^{-1} \nabla^2_{\theta\pi} J_\theta(\pi^*)$$

Bellman optimality: $\pi^* = \text{argmin}_\pi \, B_\theta(\pi)$

Differentiate through one-shot optimization problem by optimality condition (Amos et al., 2017)



# Computational challenges

Policy optimality: $\pi^* = \text{argmax}_\pi J_\theta(\pi)$

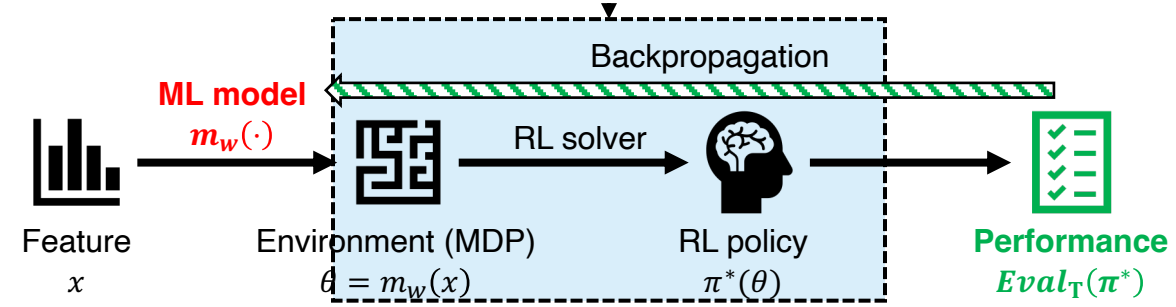
Bellman optimality: $\pi^* = \text{argmin}_\pi B_\theta(\pi)$

Different approximations of $\nabla^2_\pi J_\theta(\pi^*)$

| Method | Computation | Accuracy |
|---|---|---|
| Full Hessian | $n^2k + n^\omega$ ✗ | High |

$n$: size of $\pi$
$k$: number of trajectories to compute the policy gradient

$$\frac{d\pi^*}{d\theta} = \left(\nabla^2_\pi J_\theta(\pi^*)\right)^{-1} \nabla^2_{\theta\pi} J_\theta(\pi^*)$$




Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

# Computational challenges

Policy optimality: $\pi^* = \text{argmax}_\pi J_\theta(\pi)$

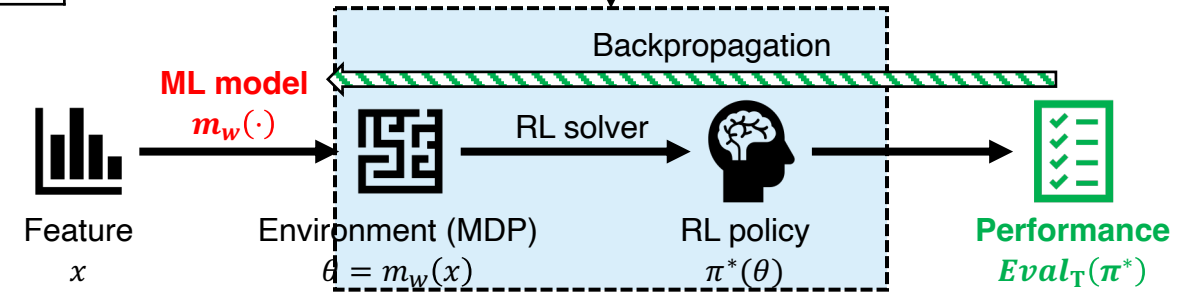
Bellman optimality: $\pi^* = \text{argmin}_\pi B_\theta(\pi)$

Different approximations of $\nabla^2_\pi J_\theta(\pi^*)$

| Method | Computation | Accuracy |
|---|---|---|
| Full Hessian | $n^2k + n^\omega$ ✗ | High |
| Identity matrix | $nk$ ✓ | Low |

$n$: size of $\pi$
$k$: number of trajectories to compute the policy gradient

$$\frac{d\pi^*}{d\theta} = \left(\nabla^2_\pi J_\theta(\pi^*)\right)^{-1} \nabla^2_{\theta\pi} J_\theta(\pi^*)$$




Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

# Computational challenges

Policy optimality: $\pi^* = \text{argmax}_\pi J_\theta(\pi)$

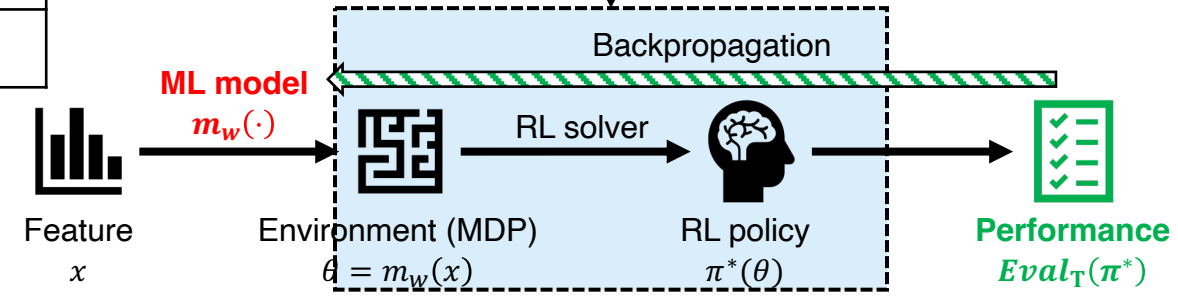
Bellman optimality: $\pi^* = \text{argmin}_\pi B_\theta(\pi)$

Different approximations of $\nabla^2_\pi J_\theta(\pi^*)$

| Method | Computation | Accuracy |
|---|---|---|
| Full Hessian | $n^2k + n^\omega$ ✗ | High |
| Identity matrix | $nk$ ✓ | Low |
| Woodbury (low-rank $k$) | $nk + k^\omega$ ✓ | Medium |

$n$: size of $\pi$
$k$: number of trajectories to compute the policy gradient

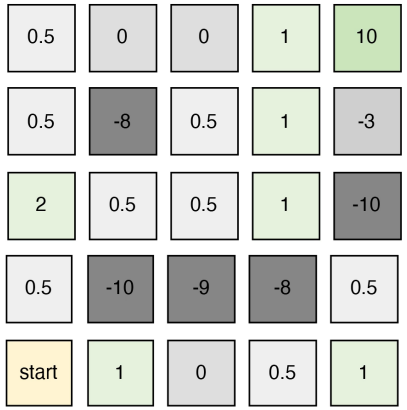
$$\frac{d\pi^*}{d\theta} = \left(\nabla^2_\pi J_\theta(\pi^*)\right)^{-1} \nabla^2_{\theta\pi} J_\theta(\pi^*)$$



# Experiments

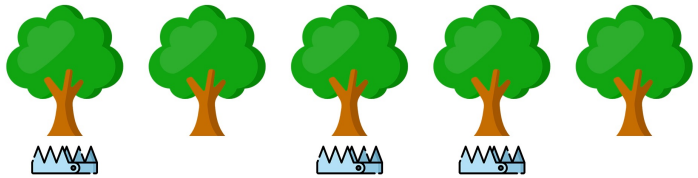
## Predict-then-optimize problems with missing MDP parameters

Reward




**Gridworld problem**
**Feature**: grid cell features
**Prediction**: grid cell reward
**Action**: movement

Transition probability



**Snare-finding problem**
**Feature**: location features
**Prediction**: snaring risk
**Action**: visit a location to remove snares

Transition probability

**Tuberculosis problem**
**Feature**: patient features
**Prediction**: patient transition probabilities
**Action**: visit a patient to intervene

# Methods

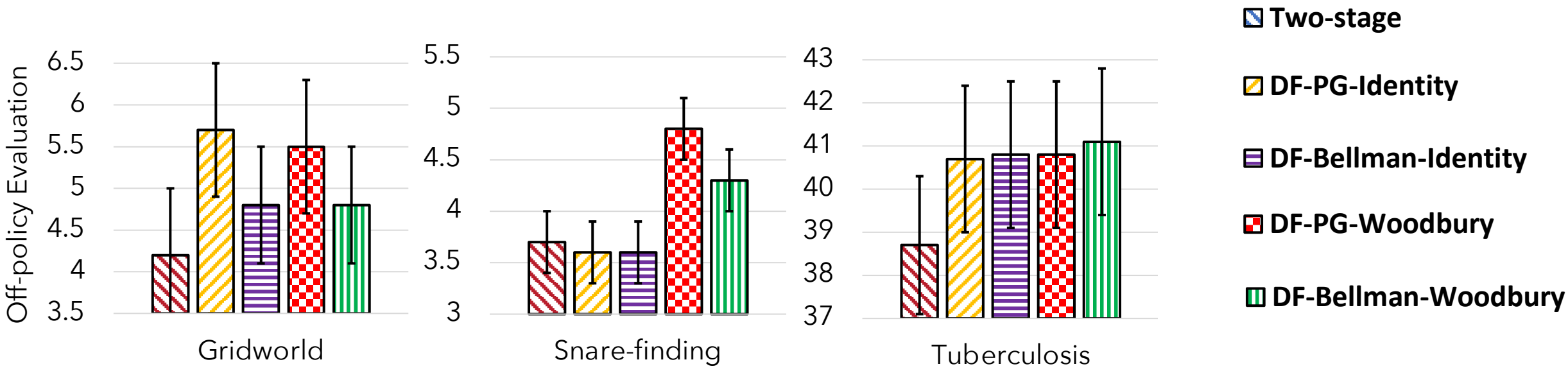
**Standard two-stage approach**

**Two-stage**

**Optimality condition**

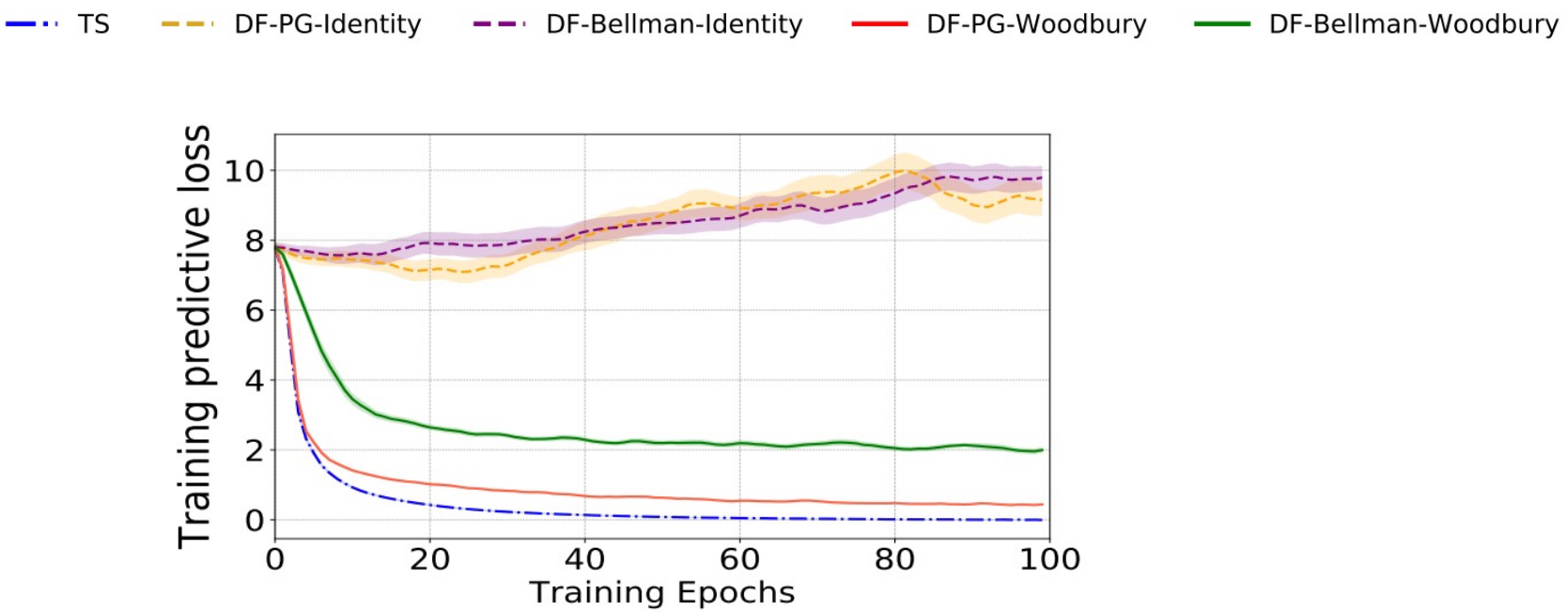
| **Approximation** | Policy gradient | Bellman error |
| --- | --- | --- |
| Identity | **DF-PG-Identity** | **DF-Bellman-Identity**<br>Futoma et al. (AISTATS 2020) |
| Woodbury (low-rank) | **DF-PG-Woodbury** | **DF-Bellman-Woodbury** |

# Results



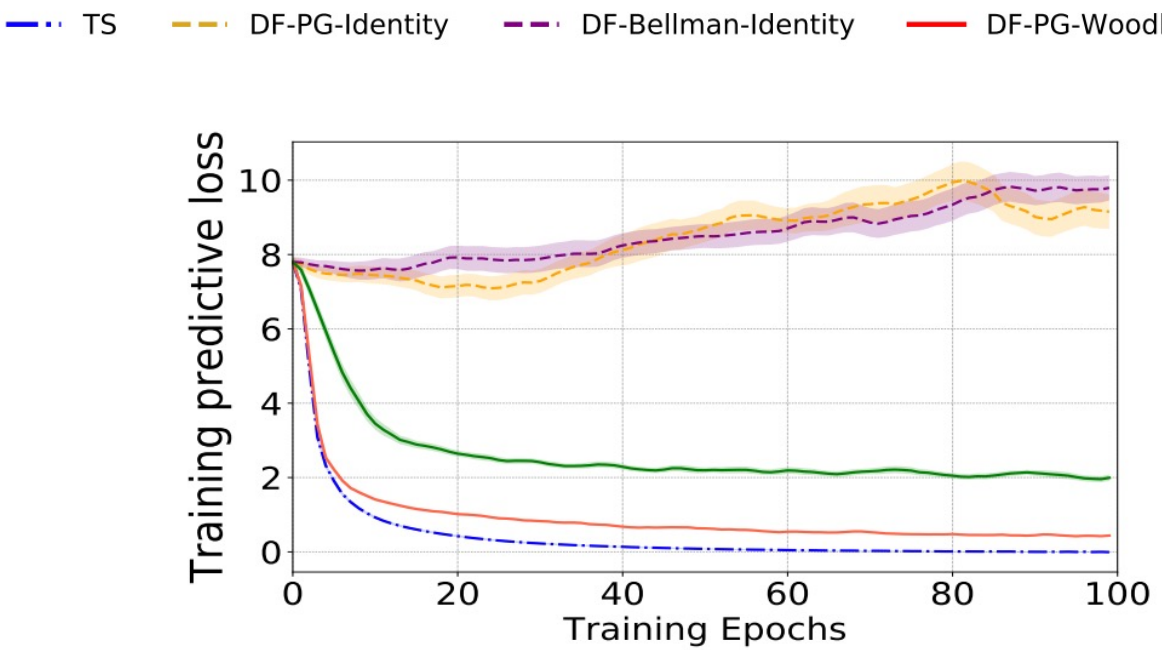

DFL outperforms two-stage
Woodbury-based are more consistent than identity-based

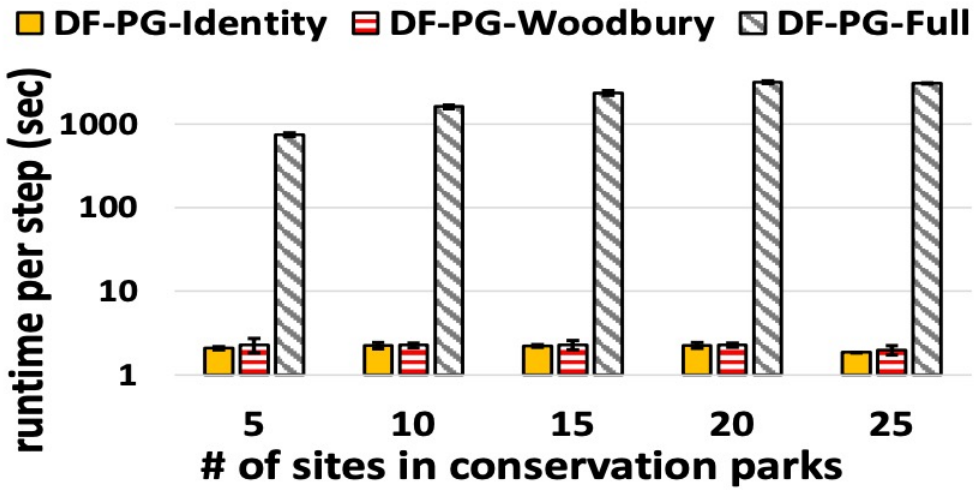
# Analysis (snare-finding)




DFL models learn less accurate models than two-stage
Hessian approximation saves ~3 orders of magnitude

# Analysis (snare-finding)




DFL models learn less accurate models than two-stage
Hessian approximation saves ~3 orders of magnitude

Wang, Shah, Chen, P, Doshi-Velez, Tambe (NeurIPS 2021)

# Why is DFL helpful?

Consider $f(z, \theta) = \sum_i \theta_i z_i$. Then,

$$\max_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_i z_i = \max_{z \in Z} \sum_i \mathbb{E}[\theta_i | x] z_i$$

# Why is DFL helpful?

Consider $f(z, \theta) = \sum_i \theta_i z_i$.

$\max_{z \in Z} \sum_i m_{w^*}(x) z_i$ instead of $\max_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_i z_i$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

# Why is DFL helpful?

Consider $f(z, \theta) = \sum_i \theta_i z_i$.

$\max_z \sum_i m_{\omega^*}(x) z_i$ instead of $\max_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_i z_i$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

$z_i \geq 0, \sum_i z_i \leq 1$

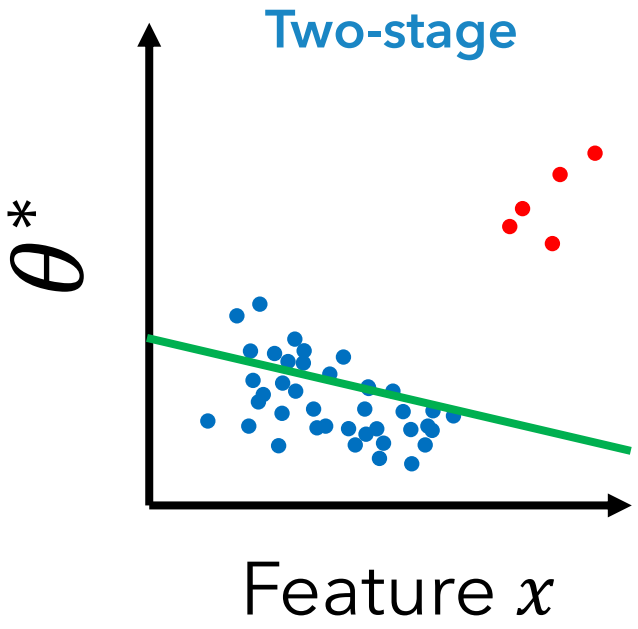
# Why is DFL helpful?

Consider $f(z, \theta) = \sum_i \theta_i z_i$.

$\max_{z} \sum_i m_{\omega^*}(x) z_i$ instead of $\max_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_i z_i$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

$z_i \geq 0, \sum_i z_i \leq 1$



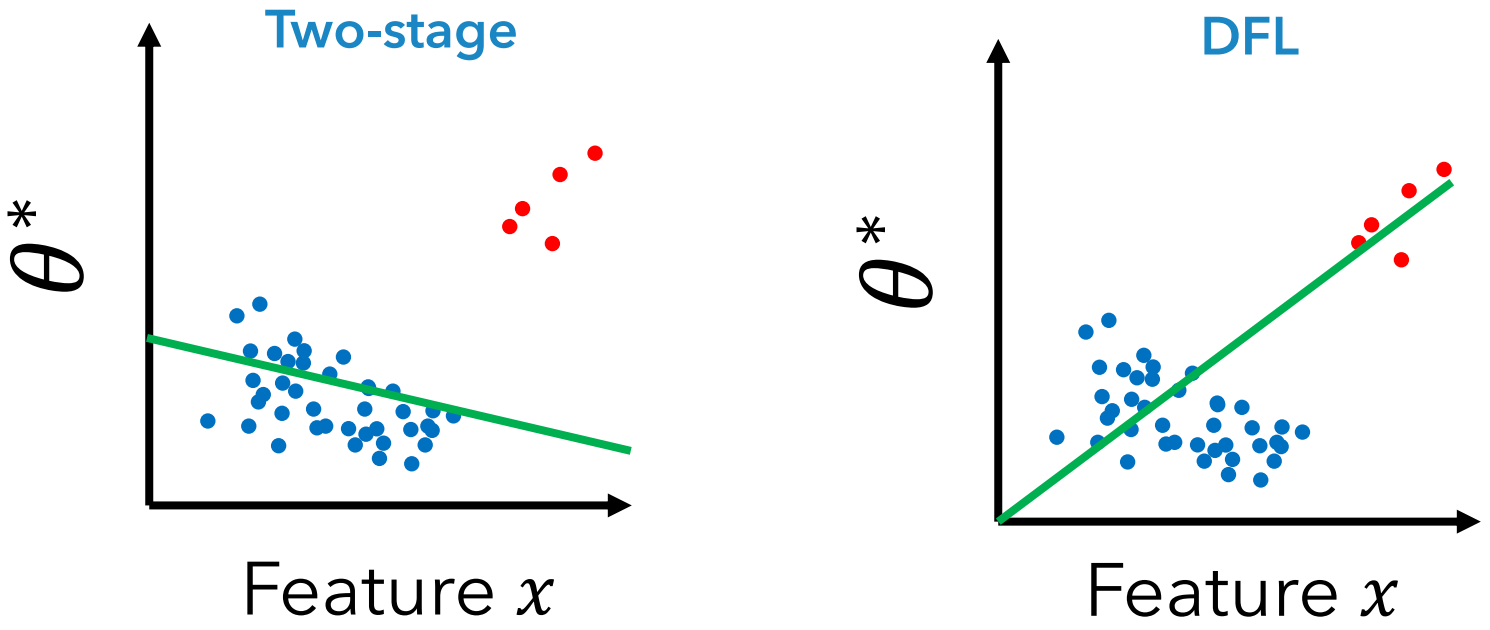
# Why is DFL helpful?

Consider $f(z, \theta) = \sum_i \theta_i z_i$.

$\max_{z} \sum_i m_{\omega^*}(x) z_i$ instead of $\max_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} \sum_i \theta_i z_i$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

$z_i \geq 0, \sum_i z_i \leq 1$

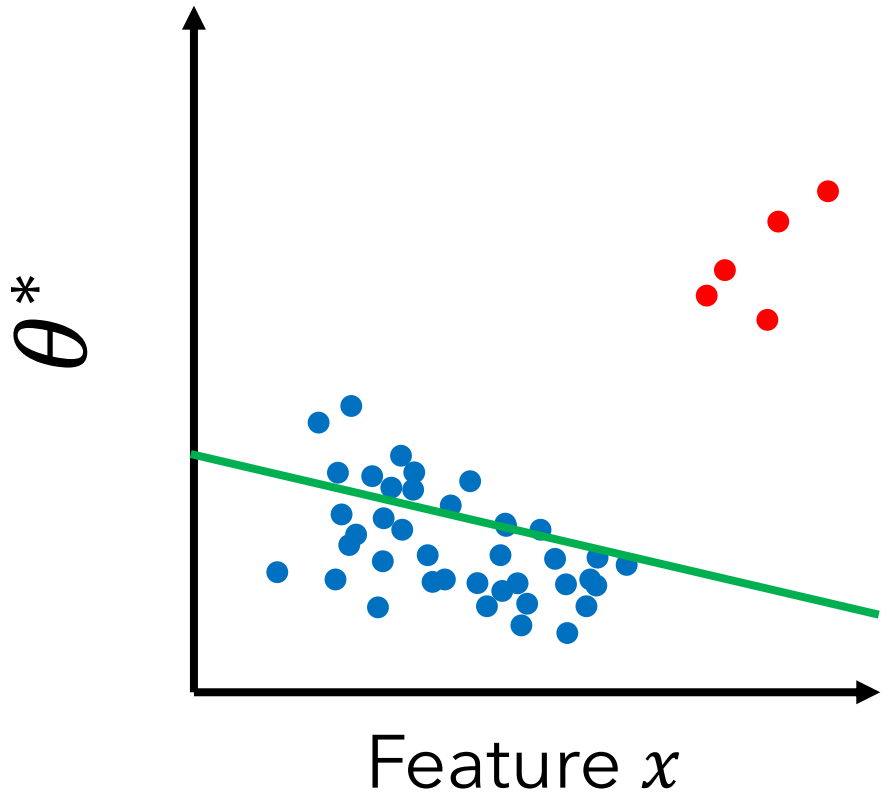


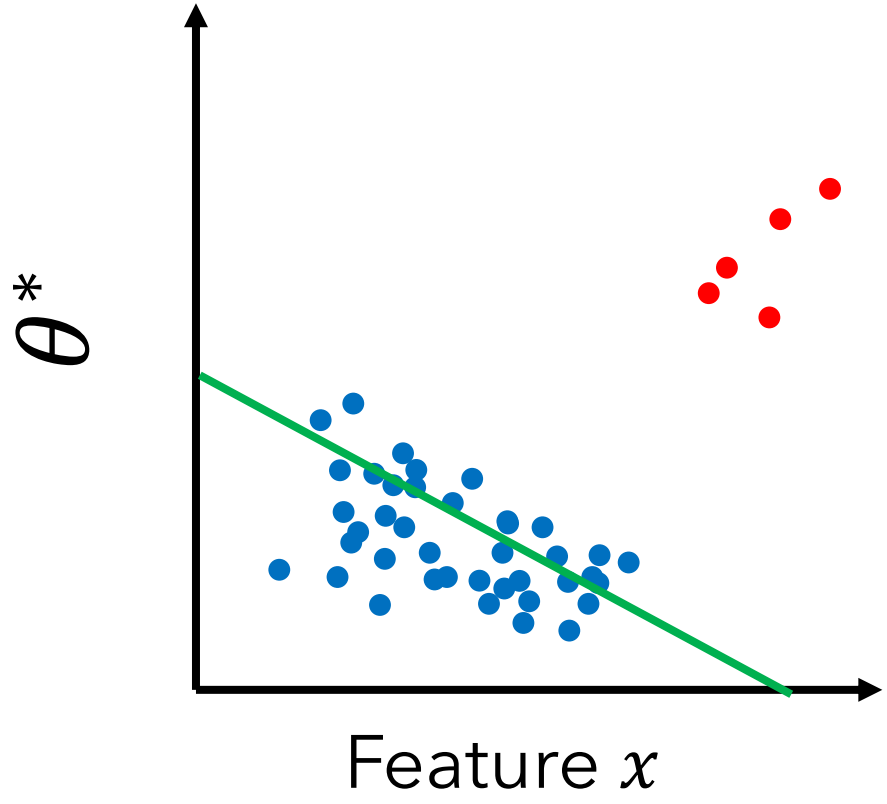
# The problem of zero gradient

$$\max_{z} \sum_{i} m_{\omega^*}(x) z_i$$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

$z_i \geq 0, \sum_i z_i \leq 1$



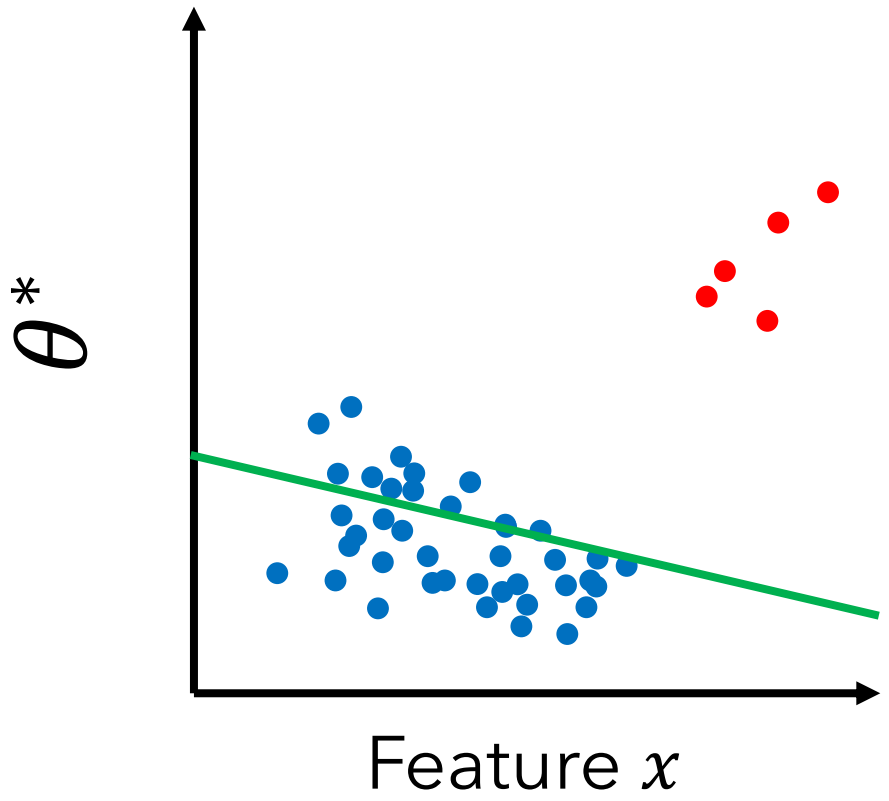



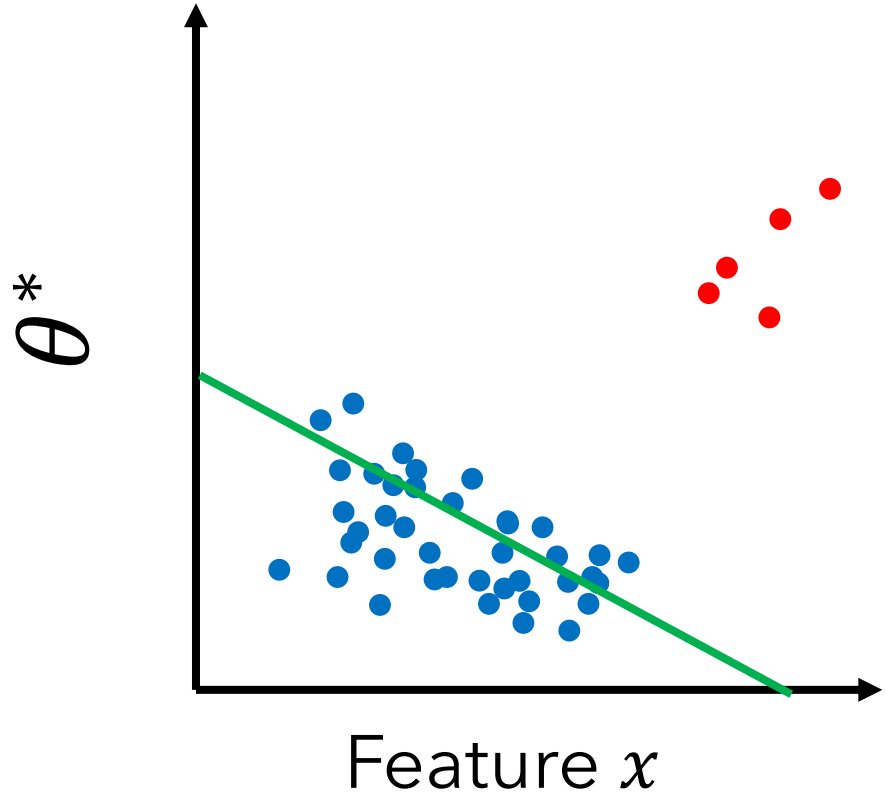
# The problem of zero gradient

$$\max_{z} \sum_{i} m_{\omega^*}(x) z_i + \lambda z_i^2$$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

$z_i \geq 0, \sum_i z_i \leq 1$



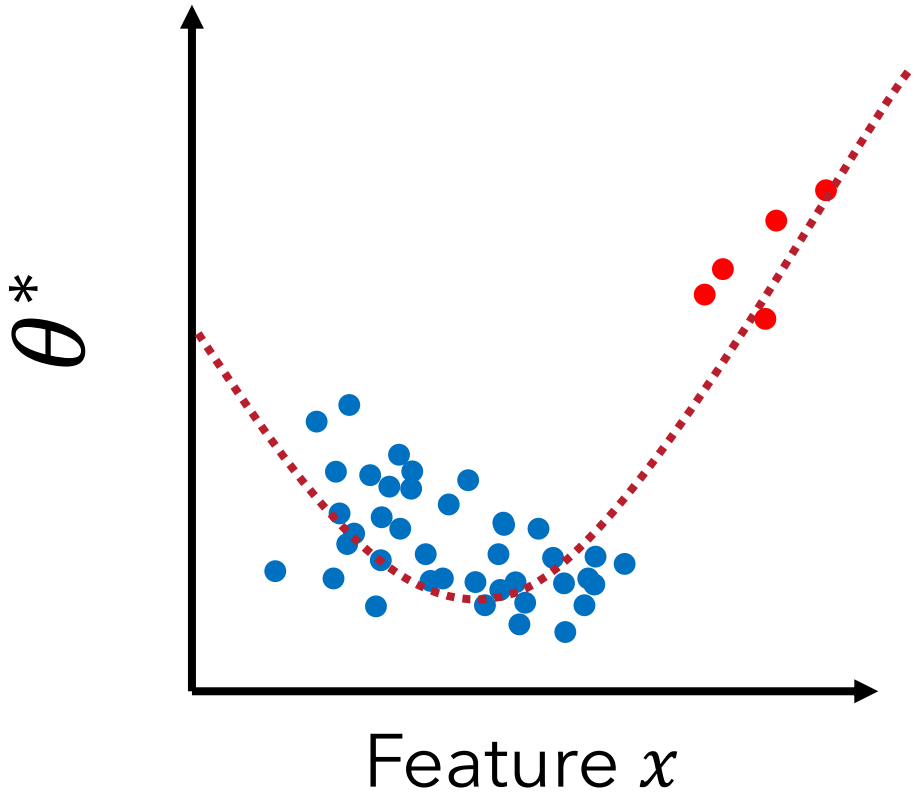



# The problem of non-convexity

$$\max_{z} \sum_{i} m_{\omega^*}(x) z_i + \lambda z_i^2$$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

$z_i \geq 0, \sum_i z_i \leq 1$

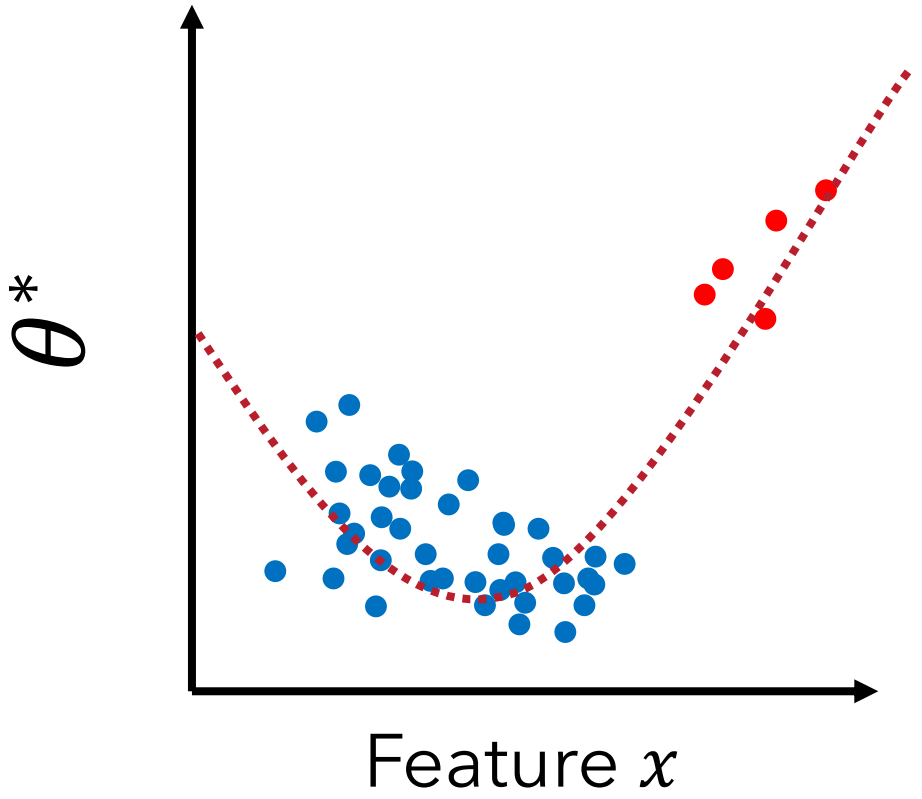


# The problem of non-convexity

$$\max_{z} \sum_{i} m_{\omega^*}(x) z_i + \lambda z_i^2$$

s.t. $w^* = \text{argmin}_w \mathcal{L}(w, \text{training data})$

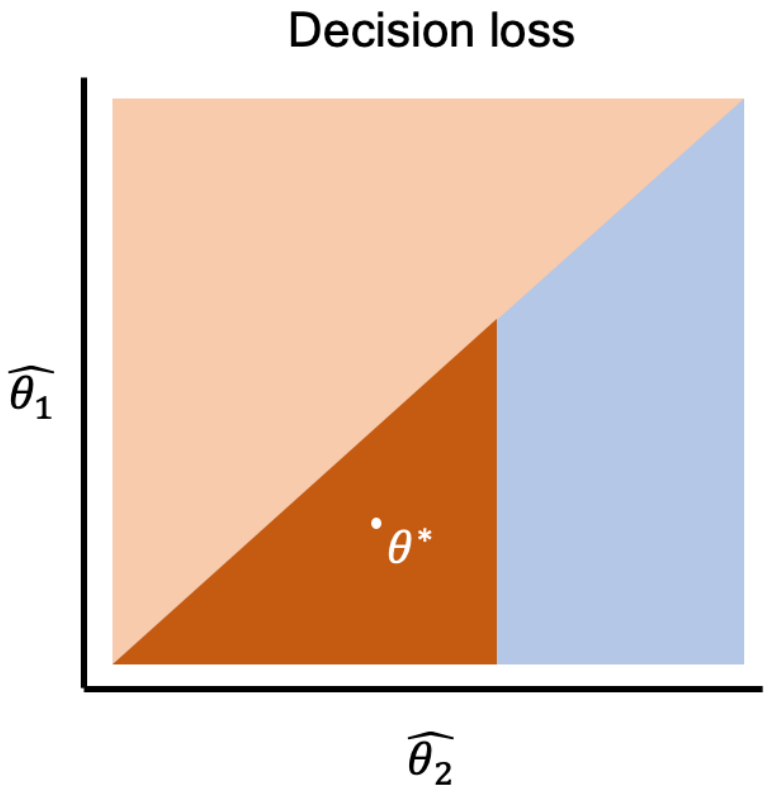
$z_i \geq 0, \sum_i z_i \leq 1$



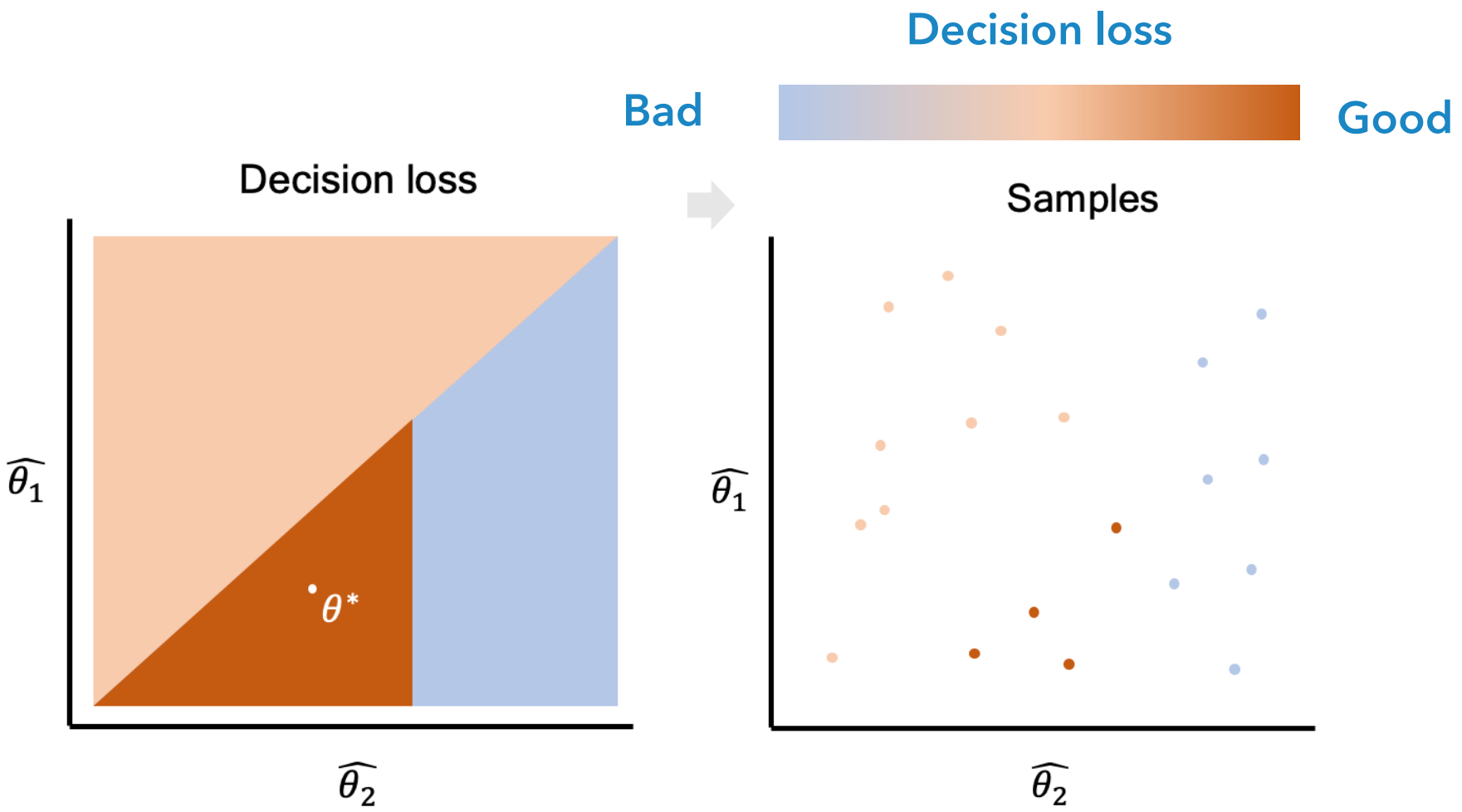

**Cool theory for linear objectives: Elmachtoub and Grigas (2022)**

Decision loss

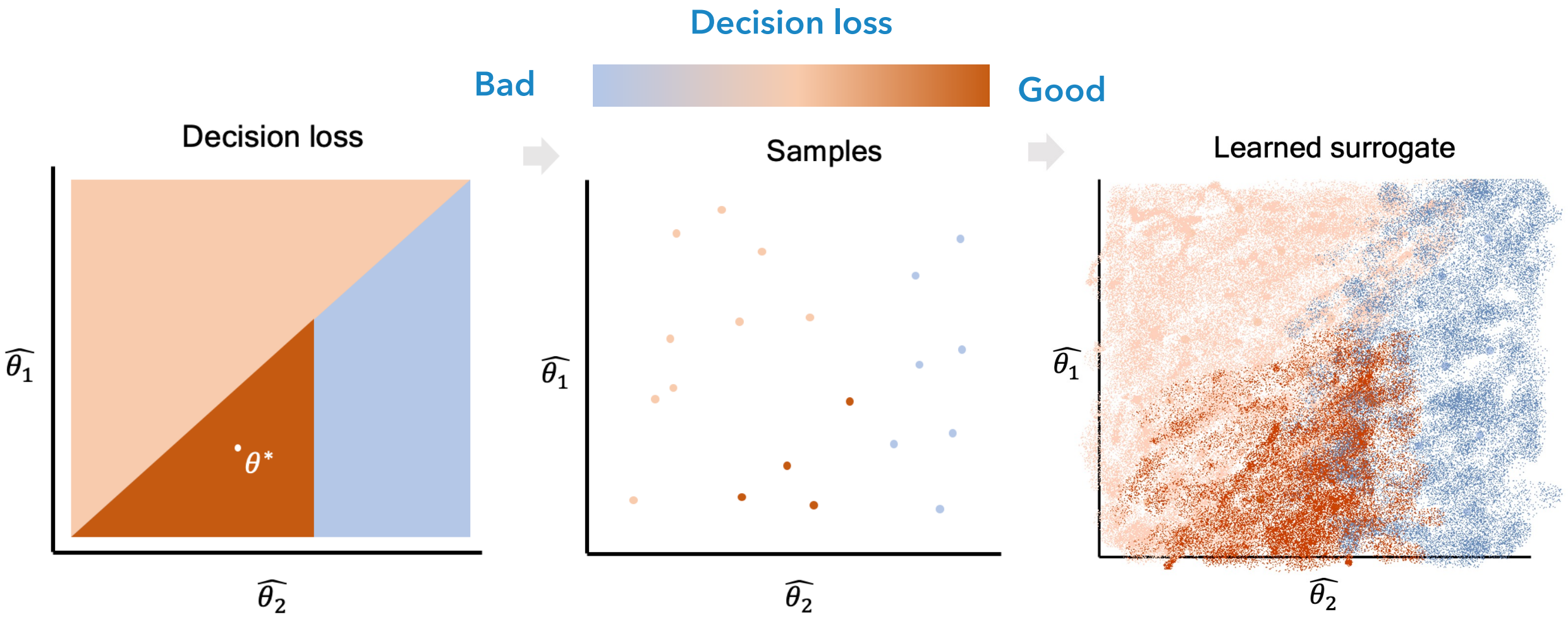
Bad Good




Shah, Wang, Wilder, P, Tambe (arXiv, 2022)




Shah, Wang, Wilder, P, Tambe (arXiv, 2022)




Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Questions

1. Where to sample from the decision loss function?

2. Where is the local minimum?

3. What function to fit?

# Notation

Predicted environment parameter $\hat{\theta}$

Decision induced by $\hat{\theta}$:
$z^*(\hat{\theta}) = \mathbf{argmin}_{z \in Z} f(z, \hat{\theta})$

Decision loss of $\hat{\theta}$:
$\ell(\hat{\theta}) = f(z^*(\hat{\theta}), \theta^*)$

**Standard DFL notation**

Observable feature/context vector $x \in X$

Unobserved ground truth environment parameter $\theta^* \in \Theta$

$x$ and $\theta^*$ drawn jointly from fixed distribution $\mathcal{D}$

Decision/control variable $z \in Z$

Objective function $f(z, \theta^*)$

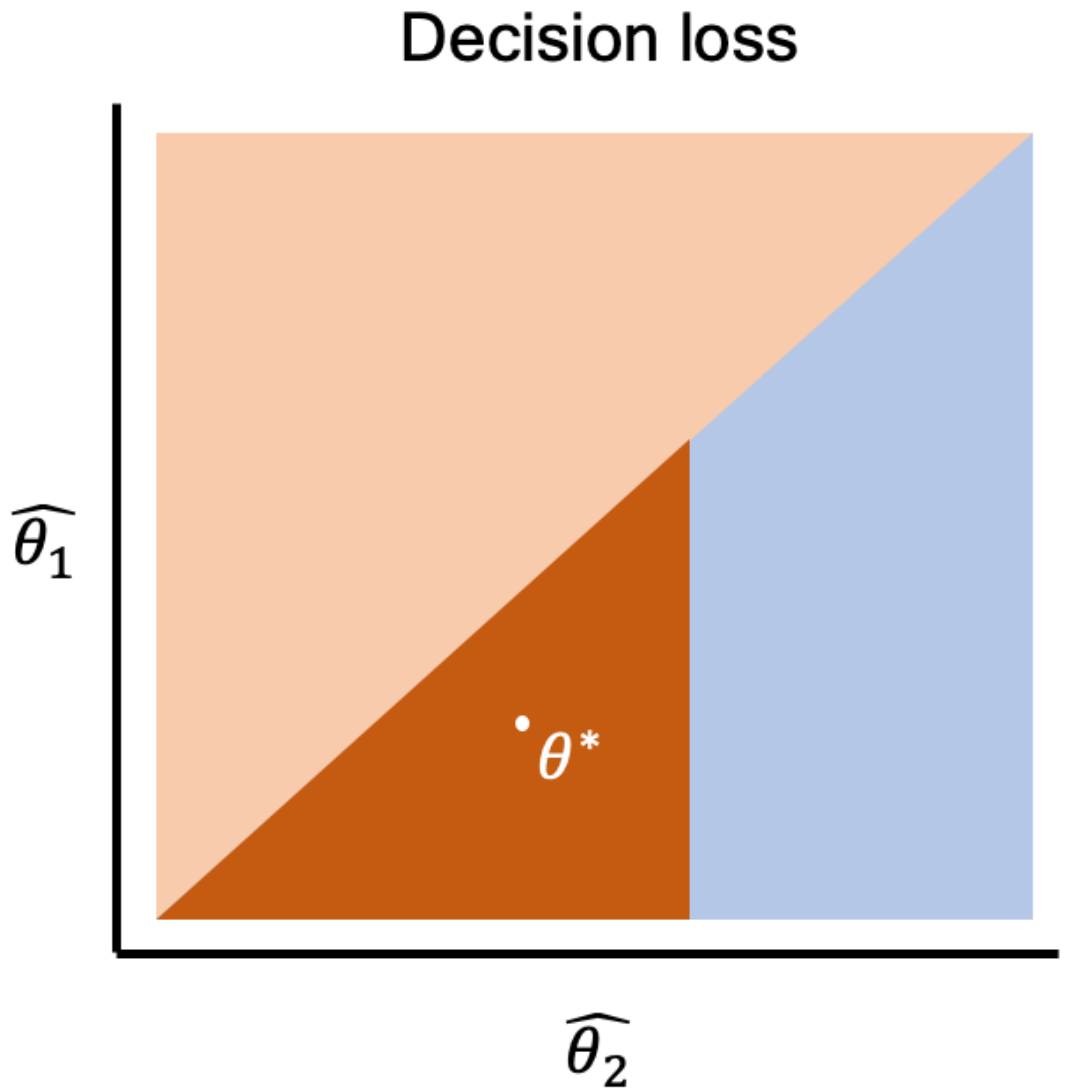
Goal: $\min_{z \in Z} \mathbb{E}_{\theta \sim P(\Theta|x)} f(z, \theta)$

1. Sample a dataset of pairs $\mathcal{D} = \{\left(\theta_j, \ell\left(\theta_j\right)\right)\}$

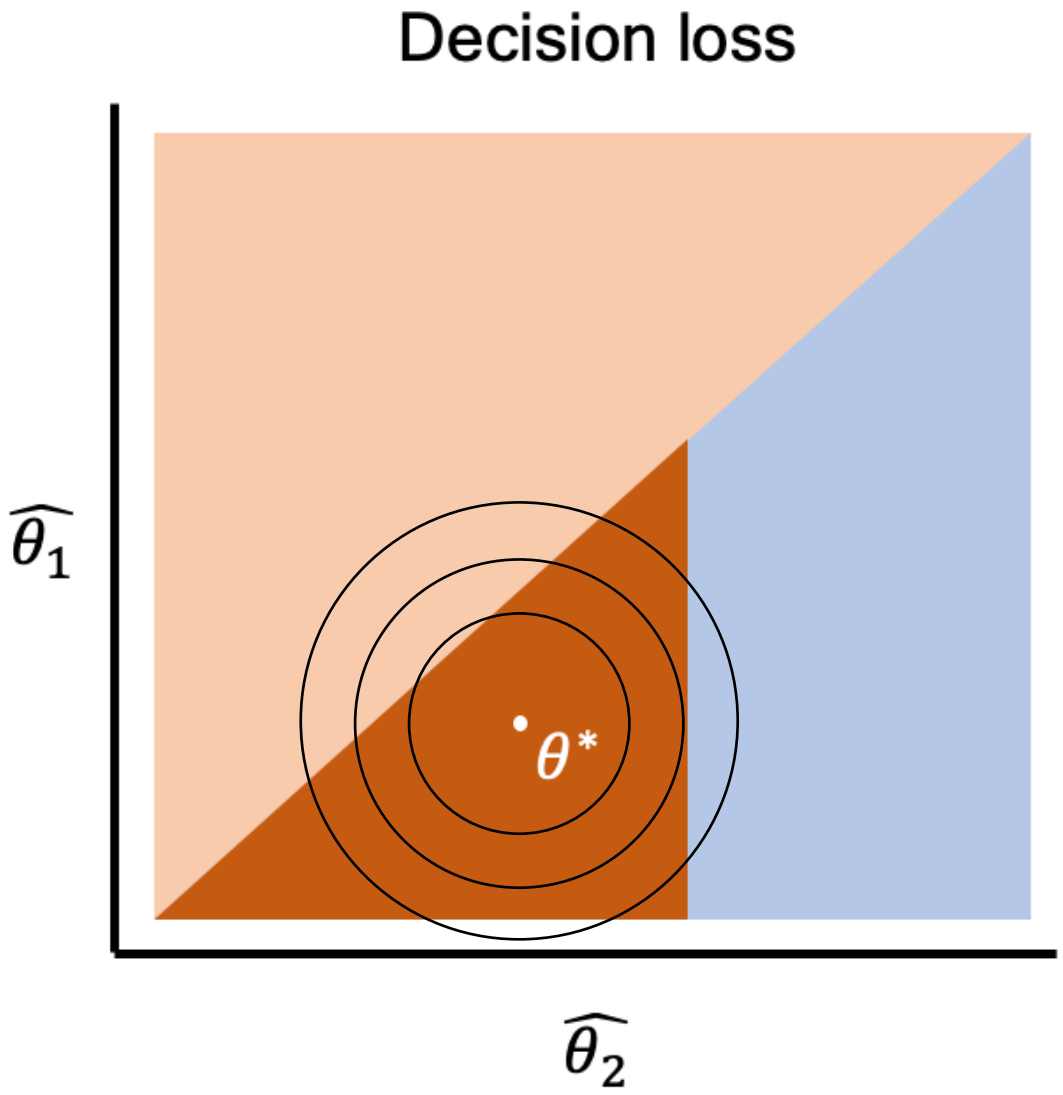
2. Fit a parametric loss $\hat{\ell}_\omega(\theta)$ to minimize MSE on training data

3. Train predictive model $m: X \to \Theta$ using $\hat{\ell}_\omega(\theta)$

Shah, Wang, Wilder, P, Tambe (arXiv, 2022)



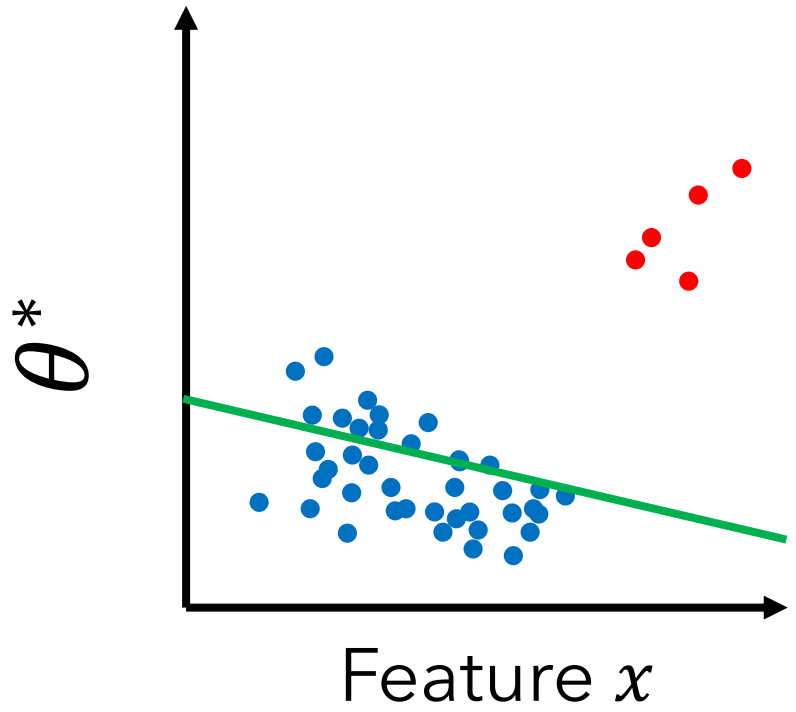

Shah, Wang, Wilder, P, Tambe (arXiv, 2022)




Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Choosing a parametric family

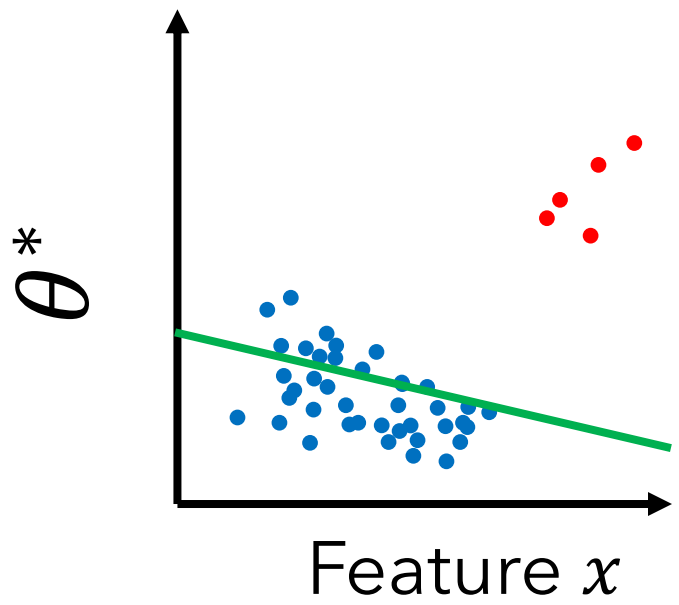
1. Capture differing importance of dimensions of $\theta$

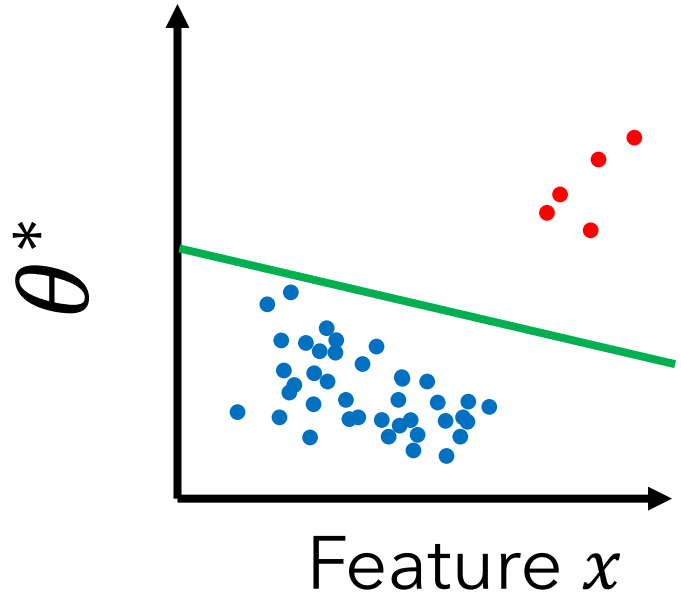



Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Choosing a parametric family

1. Capture differing importance of dimensions of $\theta$
2. Capture pairwise interactions between dimensions of $\theta$

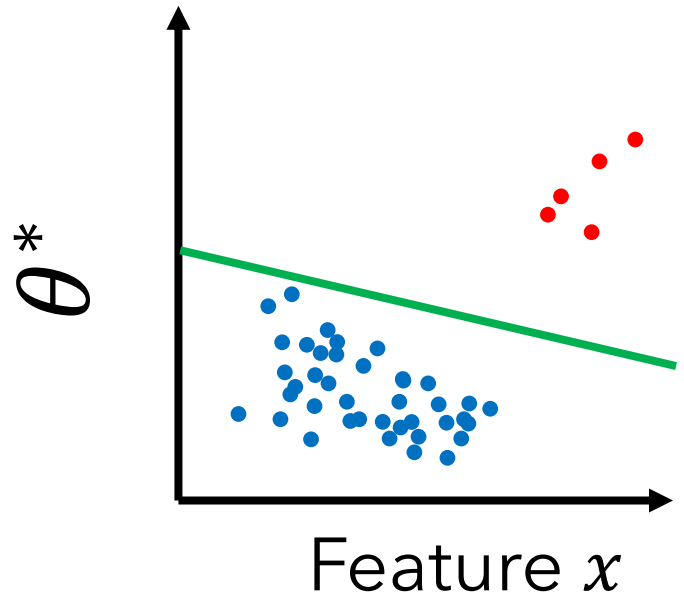



Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Choosing a parametric family

1. Capture differing importance of dimensions of $\theta$
2. Capture pairwise interactions between dimensions of $\theta$
3. Capture asymmetry in over- vs. under-predicting




Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Choosing a parametric family

Starting point: quadratic loss

$$\left(\hat{\theta} - \theta^*\right)^T H \left(\hat{\theta} - \theta^*\right)$$

where $H$ is the learned parameter.

Low-rank, PSD decomposition: $H = LL^T$

Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Choosing a parametric family

Starting point: quadratic loss

$$(\hat{\theta} - \theta^*)^T H (\hat{\theta} - \theta^*)$$

where $H$ is the learned parameter.

Low-rank, PSD decomposition: $H = LL^T$

✔ Capture differing importance of dimensions of $\theta$

✔ Capture *pairwise interactions* between entries of $\theta$

✘ Capture *asymmetry* in over vs under predicting

Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Choosing a parametric family

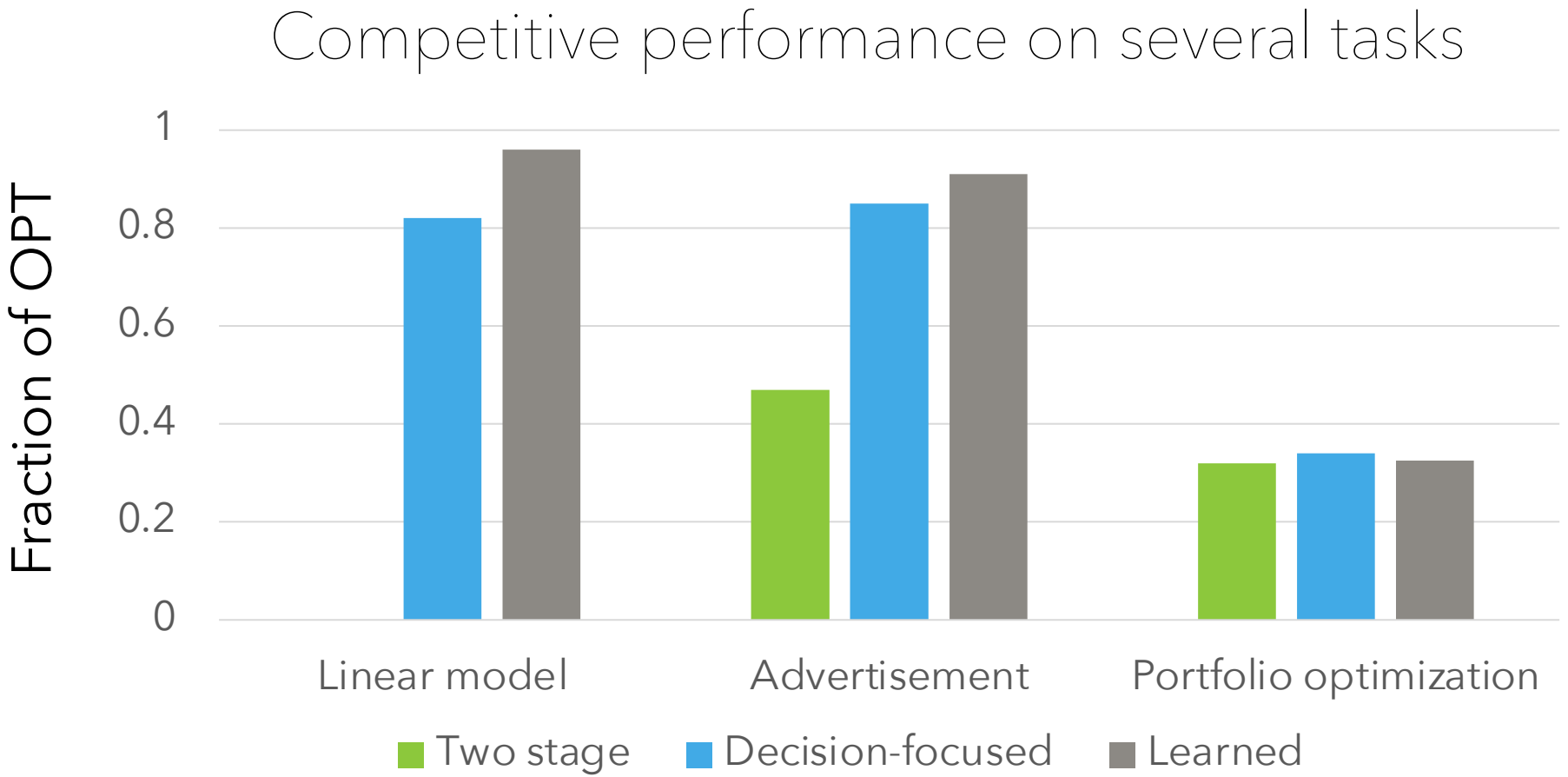
Refinement: *directed* quadratic loss.

Each $\boldsymbol{i, j}$ pair of coordinates gets a loss which depends on direction of errors for the prediction of each

Instantiate via four copies of the parameter $L$ ($H = LL^T$)

$$
\begin{aligned}
&L_{ij}^{++}, && \text{if } \hat{y}_i - y_i \geq 0 \text{ and } \hat{y}_j - y_j \geq 0 \\
&L_{ij}^{+-}, && \text{if } \hat{y}_i - y_i \geq 0 \text{ and } \hat{y}_j - y_j < 0 \\
&L_{ij}^{-+}, && \text{if } \hat{y}_i - y_i < 0 \text{ and } \hat{y}_j - y_j \geq 0 \\
&L_{ij}^{--}, && \text{otherwise}
\end{aligned}
$$

# Experiments




Shah, Wang, Wilder, P, Tambe (arXiv, 2022)

# Conclusion/future directions

1. Cost of training
2. Exploiting correlation structure
3. Accessibility of code/methods
4. Benchmarks

# References


Wilder et al. Melding the data-decisions pipeline: decision-focused learning for combinatorial optimization. AAAI 2019.

Cameron et al. The Perils of Learning Before Optimizing. AAAI 2022.

Agrawal et al. Price of correlations in stochastic optimization. Operations Research, 2012.

Wang et al. Learning MDPs from Features: Predict-Then-Optimize for Sequential Decision Problems by Reinforcement Learning. NeurIPS 2021.

Mate et al. Collapsing Bandits and Their Application to Public Health Interventions. NeurIPS 2020.

Futoma et al. POPCORN: Partially Observed Prediction COnstrained ReiNforcement learning. AISTATS 2020.

Elmachtoub and Grigas. Smart "Predict, then Optimize". Management Science, 2022.

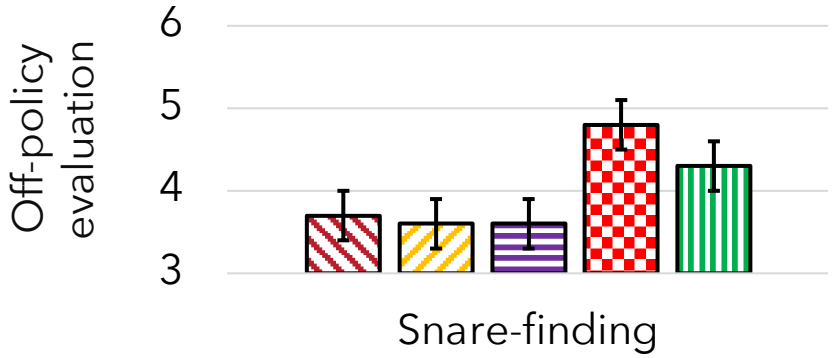
Shah et al. Decision-Focused Learning without Decision-Making: Learning Locally Optimized Decision Losses. arXiv, 2022.

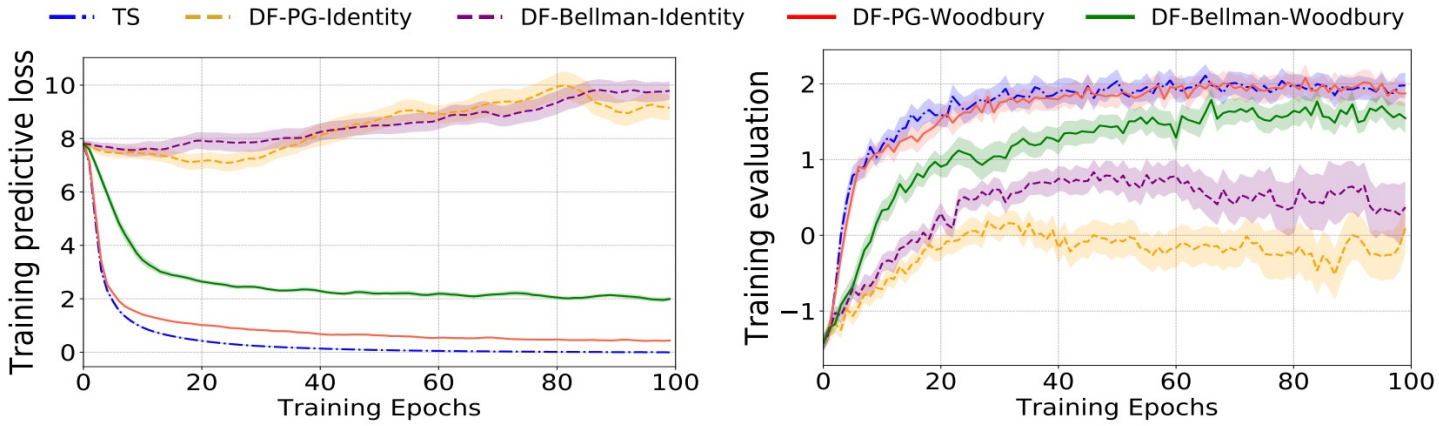
# References (tutorials)

Deep implicit layers (NeurIPS 2020): http://implicit-layers-tutorial.org/

Deep declarative networks (ECCV 2020): https://anucvml.github.io/ddn-eccvt2020/

# Analysis (snare-finding)





(a) Predictive loss in training MDPs (b) Performance in training MDPs

DFL models learn less accurate models than two-stage
Two-stage scores better on train but worse on test

# Price of correlation

Definition (price of correlation): for an objective function $f$, the price of correlation is ratio $\frac{f(x',y)}{f(x^*,y)}$, where

- $x^* = \min_{x \in X} \mathbb{E}_{y \sim P(y|\theta)} f(x,y)$
- $x' = \min_{x \in X} \mathbb{E}_{y \sim P(y|\theta)} f(x,y)$ assuming $y_i$ mutually independent