Predict Responsibly: Increasing Fairness by Learning to Defer

David Madras, Toniann Pitassi, Richard Zemel

University of Toronto, Vector Institute

December 8, 2017

The Judge and the Black-Box

The Judge and the Black-Box

The Judge and the Black-Box

“0.6”

The Judge and the Black-Box

“0.6”

What does the prediction “0.6” mean?
What qualities should it have?

What We Want From Black Box Predictions

What We Want From Black Box Predictions

1. Accuracy

What We Want From Black Box Predictions

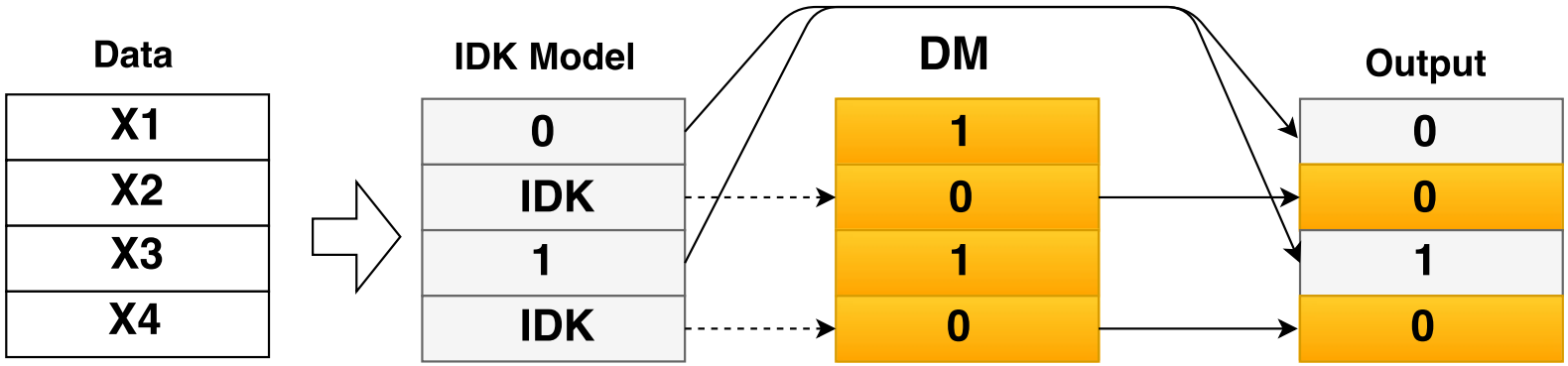
1. Accuracy
2. Fairness

What We Want From Black Box Predictions

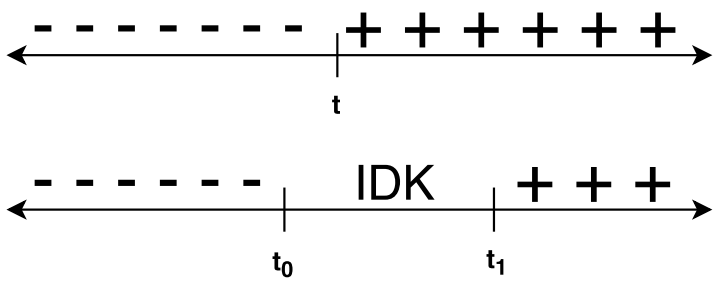
1. Accuracy
2. Fairness
3. Responsibility — Ability to say "I Don't Know"

Why Say IDK?

- Judge is **external decision maker (DM)** - may have more knowledge
- Can seek out extra information on difficult cases
- Can assess qualitative or difficult-to-codify features
- Can access privacy-sensitive information



Learning to Punt



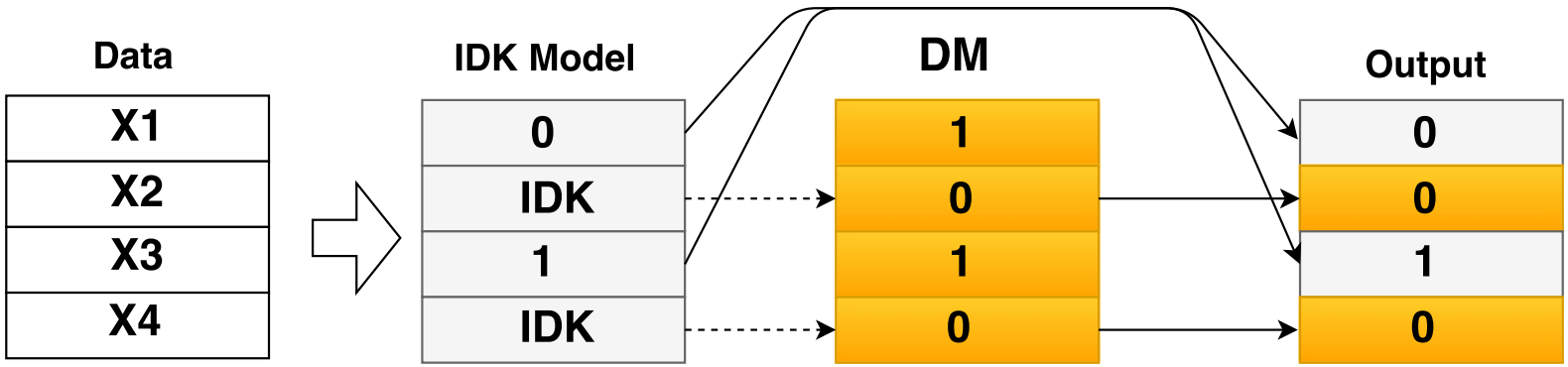
- "Positive", "Negative", and "IDK"
- Learn two thresholds: t_0, t_1
- At test time, punt to DM if $t_0 < x_i < t_1$; else, output prediction

Results - Punting

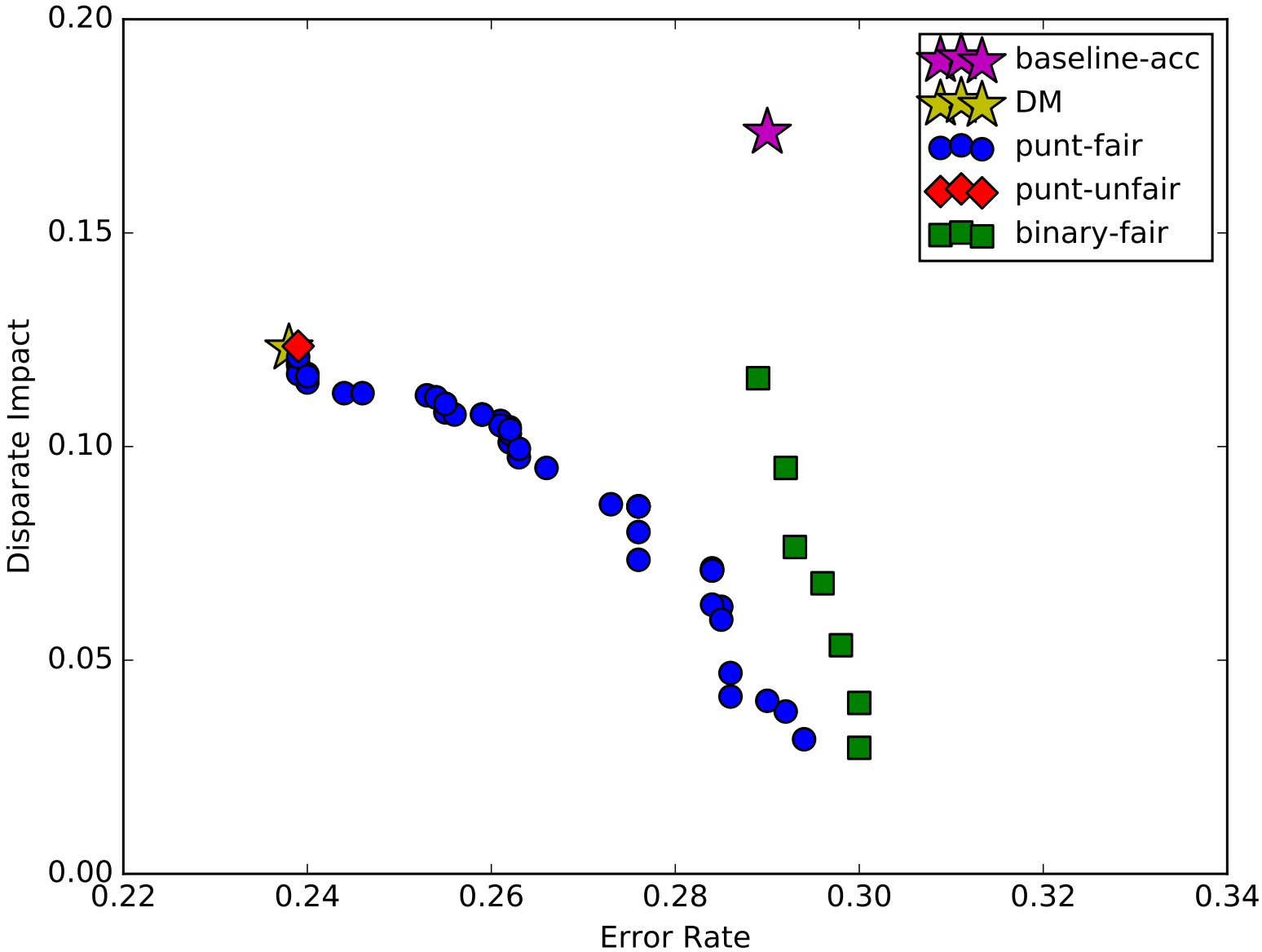
- Trained our model (2-layer NN) with fair regularization

$$\mathcal{L}_{fair} = Accuracy + \alpha \cdot Fairness$$

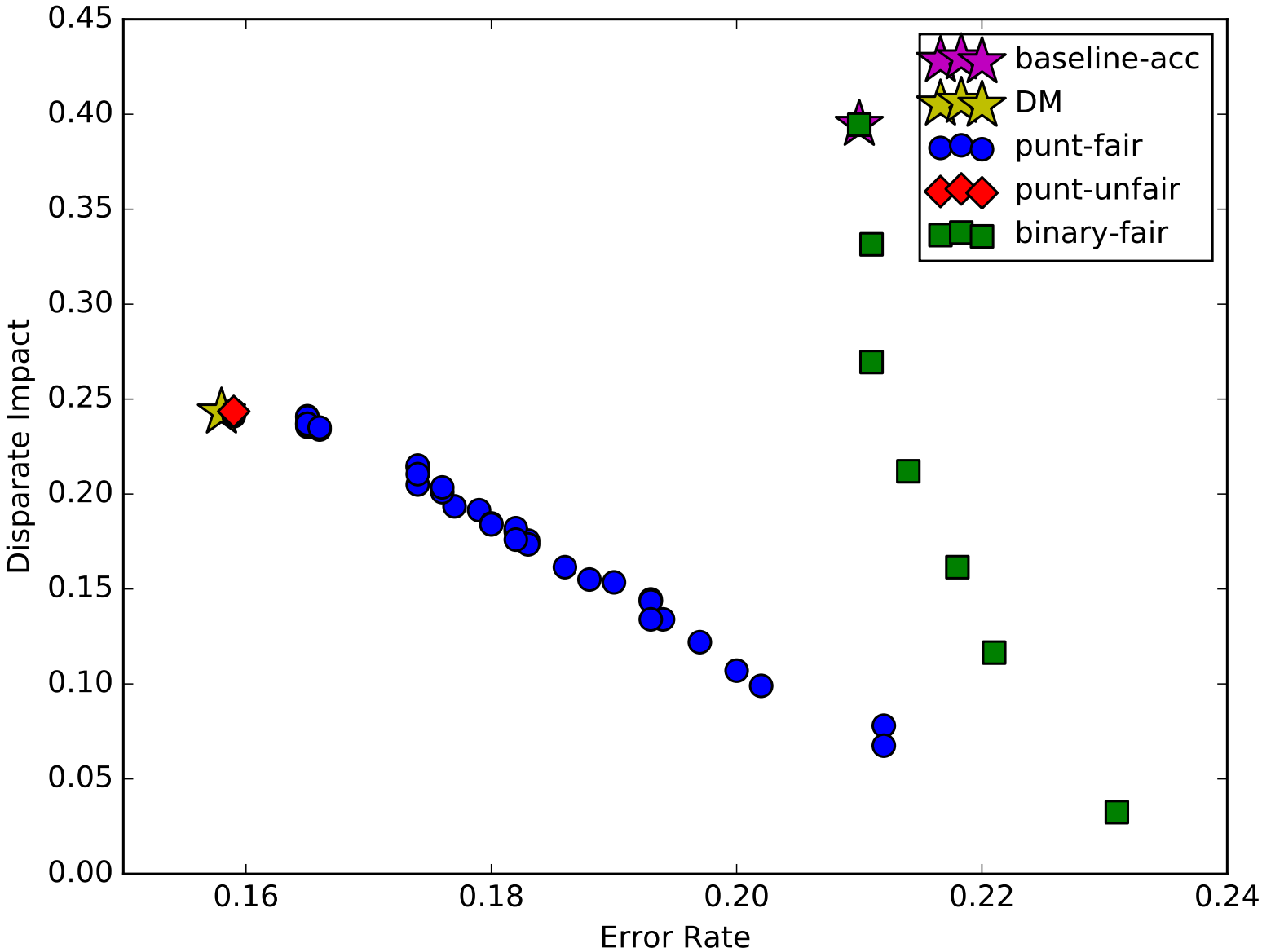
- Simulated external DM by training separate (unfair) model
- This DM received some extra attributes in training, simulating a possible real-life imbalance between DM and model



Results - COMPAS



Results - Heritage Health



DM-Aware Learning

- What if judge has access to extra info on some defendants?
  - Detailed written analysis, classified info, further inquiry
- What if judge is biased towards some types of defendants?
  - Unfairness may be concentrated on a few examples
- By using info about the DM during learning, we could punt more intelligently
- This is **learning to defer**

Learning to Defer

- Modify our model to take DM scores Y_{DM} on training set
- Use IDK output as a mixing parameter π_i
- Can describe system output Y_{sys} as function of $s \sim Bernoulli(\pi_i)$, Y_{DM}, and Y_{model}

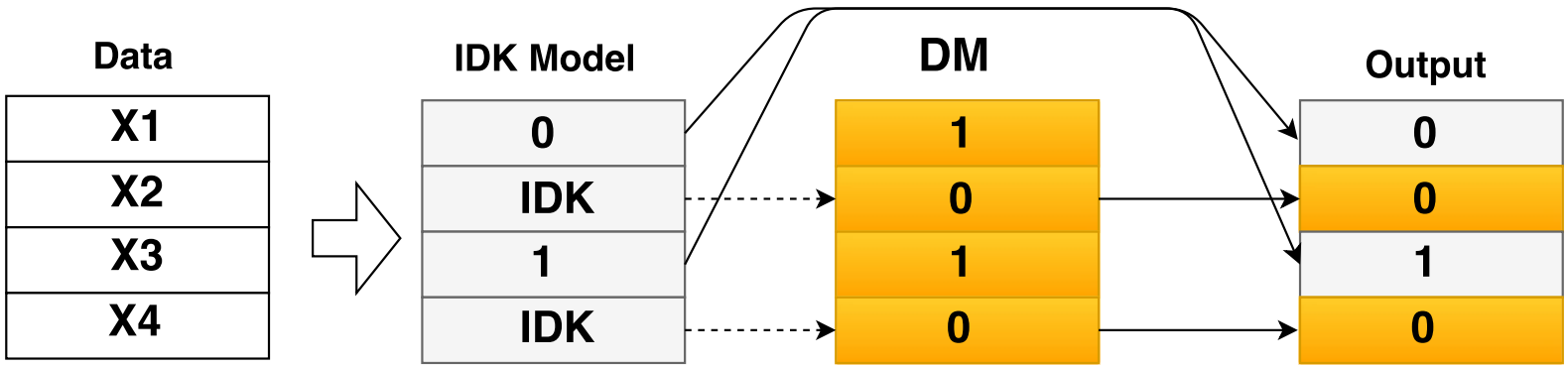
$$Y_{sys} = s \cdot Y_{DM} + (1 - s) \cdot Y_{model}$$
$$s \in \{0, 1\};\ Y_{sys}, Y_{DM}, Y_{model} \in [0, 1]$$

Learning to Defer

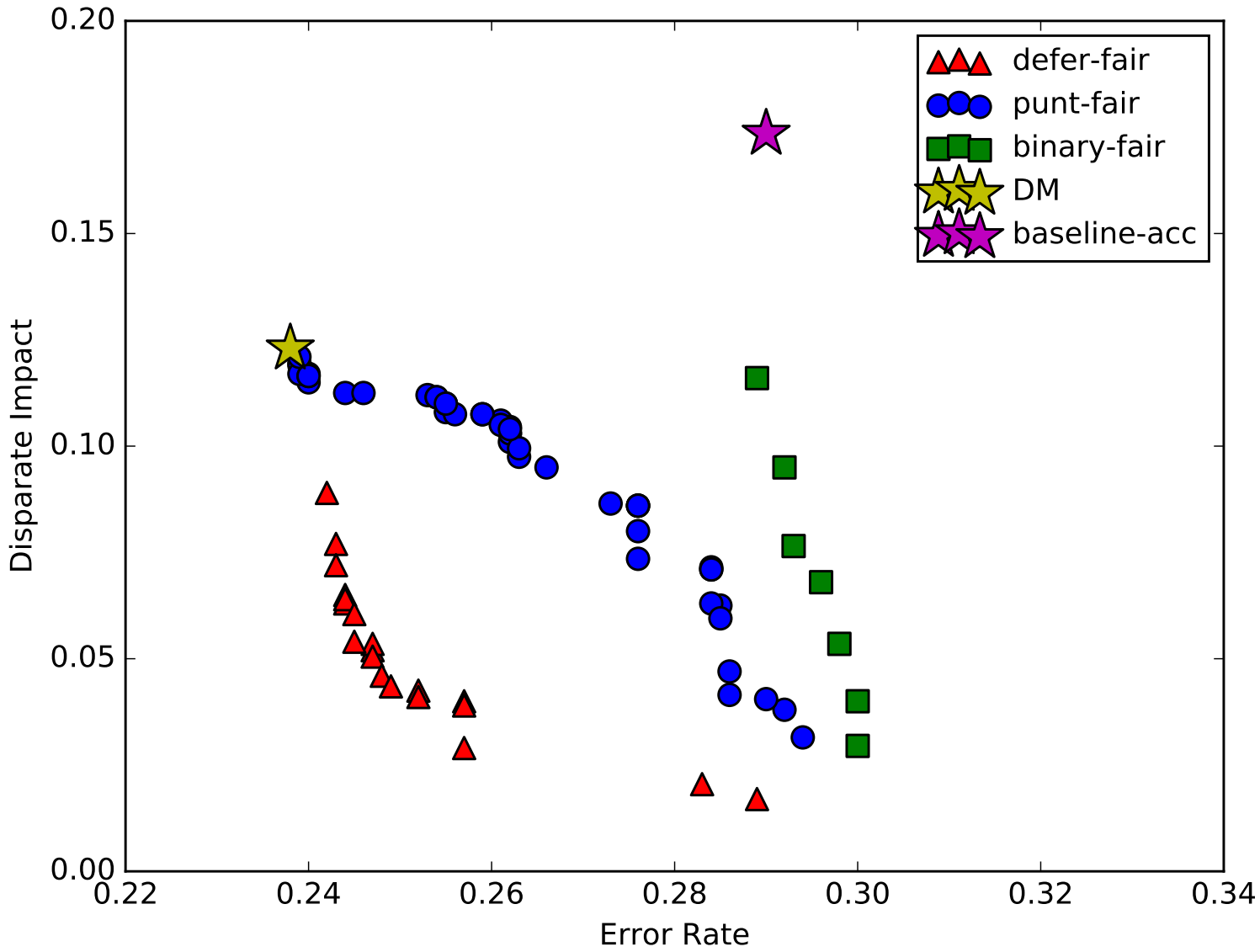
- Suppose we are optimizing some loss function $\mathcal{L}(Y, Y_{sys})$ over ground truth labels Y and system output Y_{sys}
- We can then define a new loss function $\mathcal{L}_{Defer}$

$$\begin{aligned}\mathcal{L}_{Defer}(Y, Y_{sys}) &= \mathbb{E}_s \mathcal{L}(Y, Y_{sys}) \\ &= \mathbb{E}_s \mathcal{L}(Y, s \cdot Y_{DM} + (1 - s) \cdot Y_{model})\end{aligned}$$

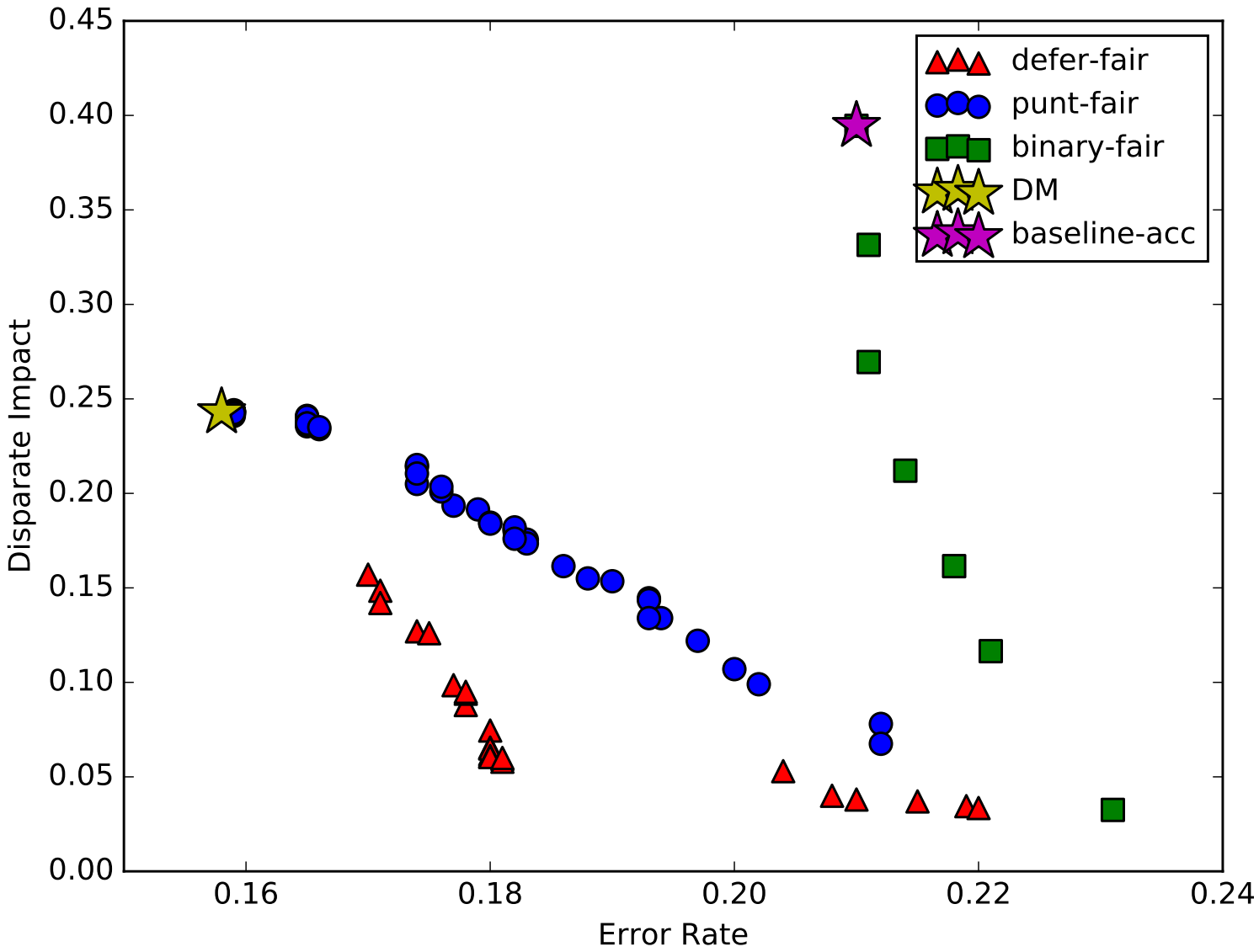
- Penalty for IDK $\approx$ DM loss on that example



Results (Learning to Defer) - COMPAS



Results (Learning to Defer) - Heritage Health



Conclusion

- We argue that it is important to consider IDK models as part of a larger pipeline
- We demonstrate that learning to defer can provide benefits above and beyond learning to punt
- Deferring intelligently can improve the entire pipeline in both accuracy and fairness

Thank you!