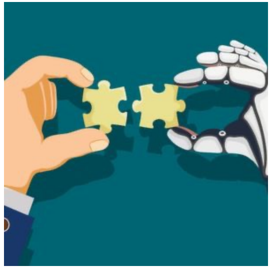


Improving Decision Making with Machine Learning, Provably

Manuel Gomez Rodriguez

Includes joint work with Eleni Straitouri, Luke Wang & Nastaran Okati

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Machine learning to improve decision making

Machine learning promises a **new generation of automated decision support systems** in many **high-stakes domains**:

Health

Hiring

Content moderation

Education

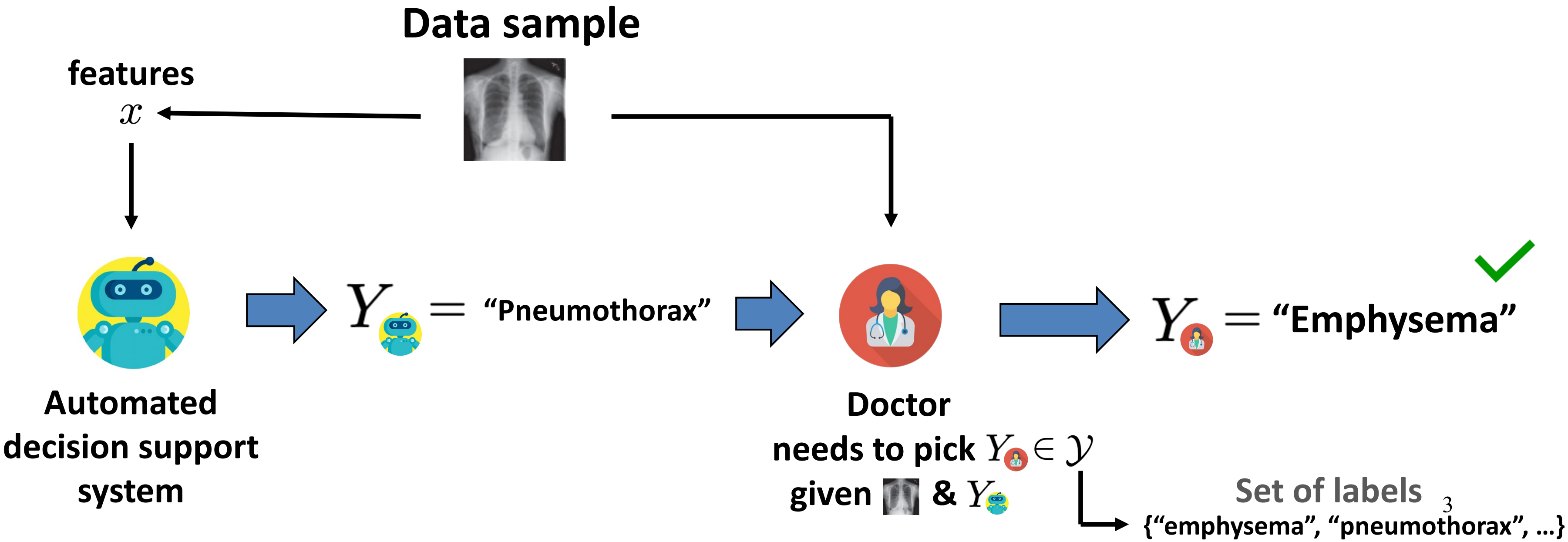
Justice

Security

Finance

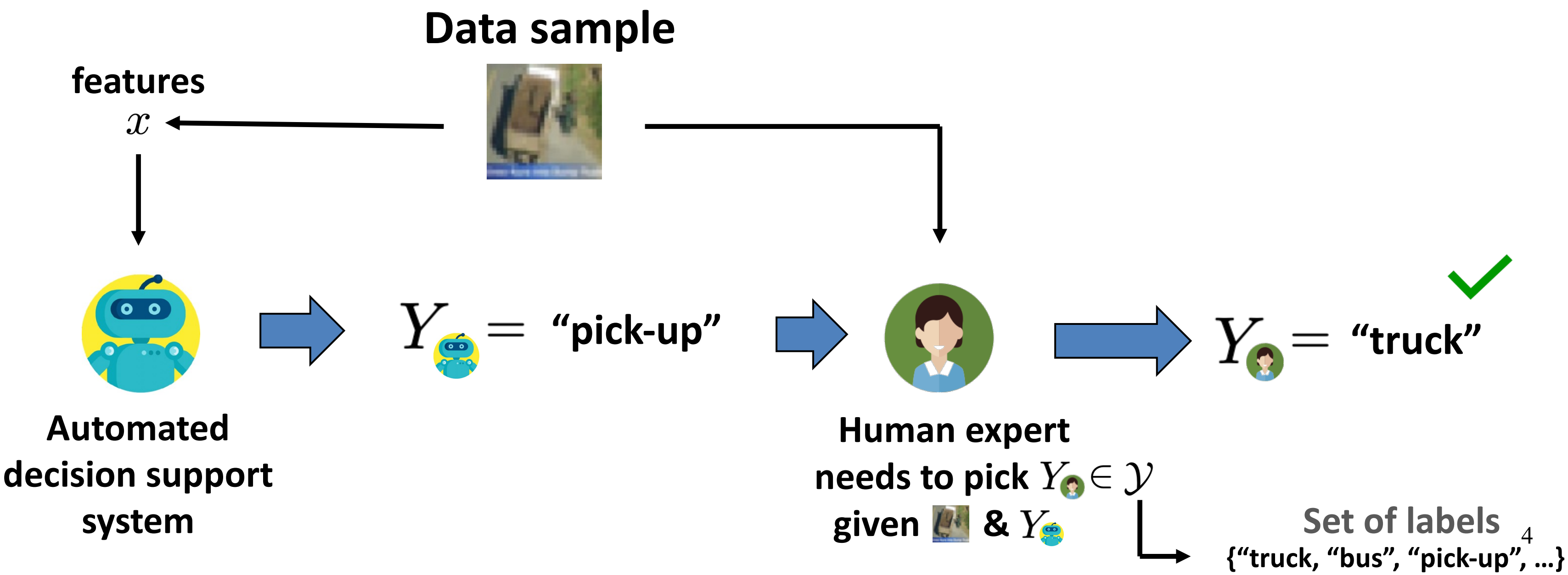
Decision support systems for classification tasks

Machine learning has mainly focused on **decision support systems** for **classification tasks**

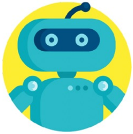


Decision support systems for classification tasks

Machine learning has mainly focused on **decision support systems** for **classification tasks**



Human experts need to understand when to trust the classifier



Automated decision support system

Y = "pick-up"

Human expert

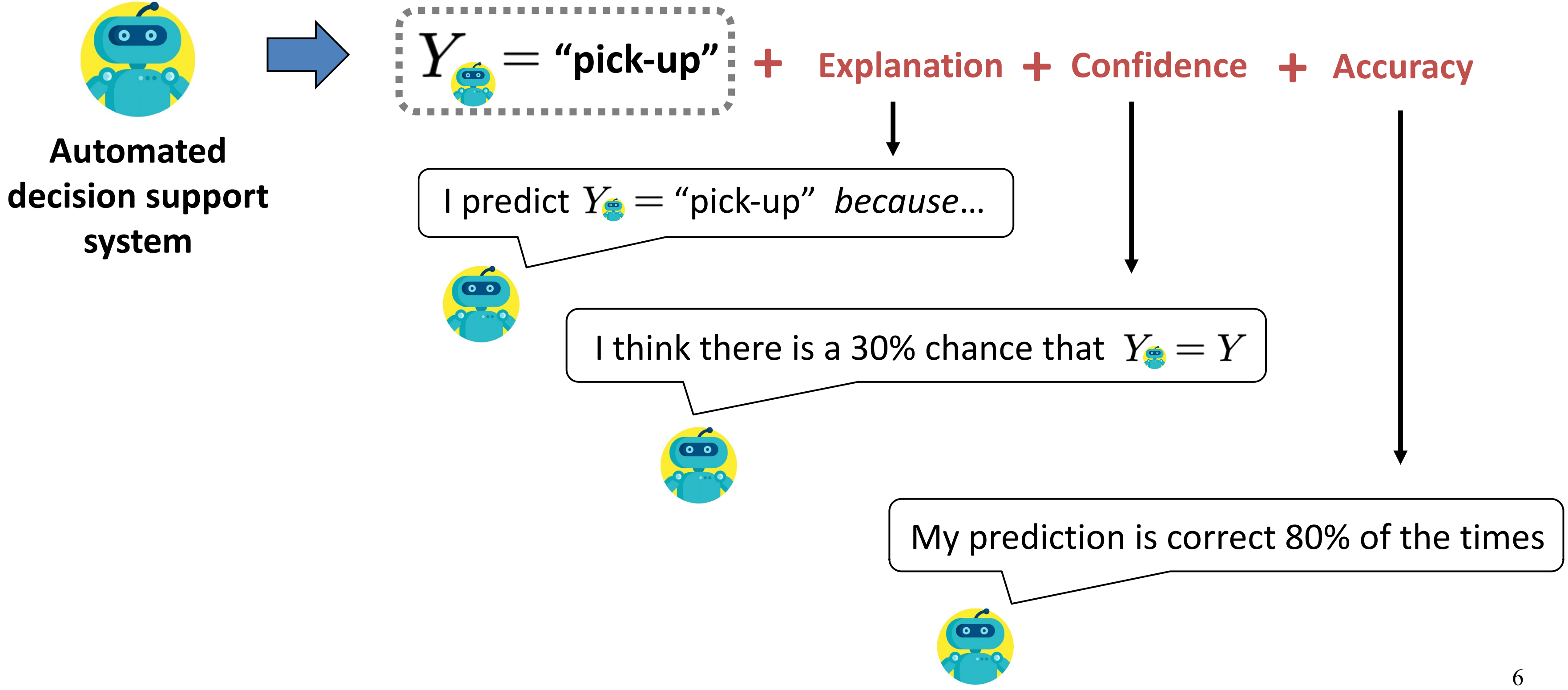
Y = "truck"

needs to pick Y
given  & Y

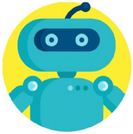
Human needs to **understand when to trust** a prediction Y made by the decision support system

- This follows from the fact that, in general, the accuracy of the system differs across data samples
- **Otherwise, they may be better off on their own**

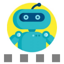
How do decision support systems *modulate* trust?



How do decision support systems *modulate* trust?



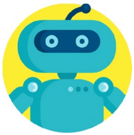
Automated decision support system

Y $=$ **"pick-up"** + Explanation + Confidence + Accuracy

Does this additional information help humans understand when to trust a prediction?

Not always. The empirical findings are mixed and seem to depend on the application domain.

How do decision support systems *modulate* trust?



Automated decision supp system

Y = **"pick-up"** + **Explanation + Confidence + Accuracy**

Not yet clear how to make sure experts do not develop a misplaced trust

Does th d

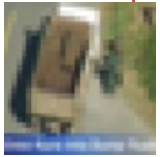
when t

Not always. The empirical findings are mixed and seem to depend on the application domain.

A decision support system that simplifies, rather than solves

Easy sample

Hard sample



Very hard sample

Existing decision support systems

$Y =$ “truck” “pick-up” “goat”

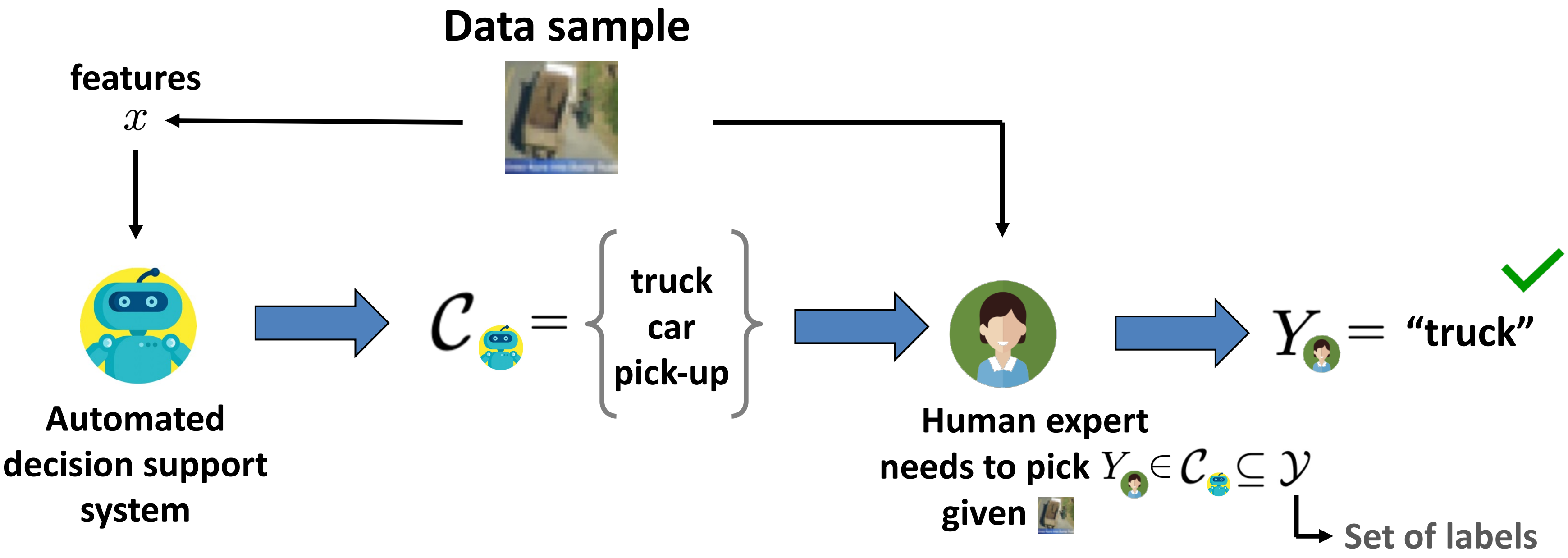
They always try to solve the classification task

Our decision support system

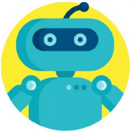
$\mathcal{C} = \{\text{truck}\}$ $\{\text{truck, car, pick-up}\}$ $\{\text{cat, dog, sheep, goat}\}$

It tries to simplify the task *as much as possible, provably*

A new type of decision support systems for classification



Humans do not need to understand when to trust the system



Automated decision support system

$\mathcal{C} = \{$ **truck**, **car**, **pick-up** $\}$

Human expert

needs to pick $Y \in \mathcal{C}$ **given** 

$Y =$ **"truck"** ✓

The human **does not need** to **understand** when to **trust** the system

→ However, we need to ensure that the subset $\mathcal{C}$ contains the **true label** Y with **high probability**

"truck"

Trustworthy subsets using conformal prediction

To ensure that the subsets $\mathcal{C}$ contain the **true label** Y with **high probability**, we rely on **conformal prediction.**

Conformal prediction (CP) is a statistical technique to construct ***trustworthy subsets*** $\mathcal{C}$

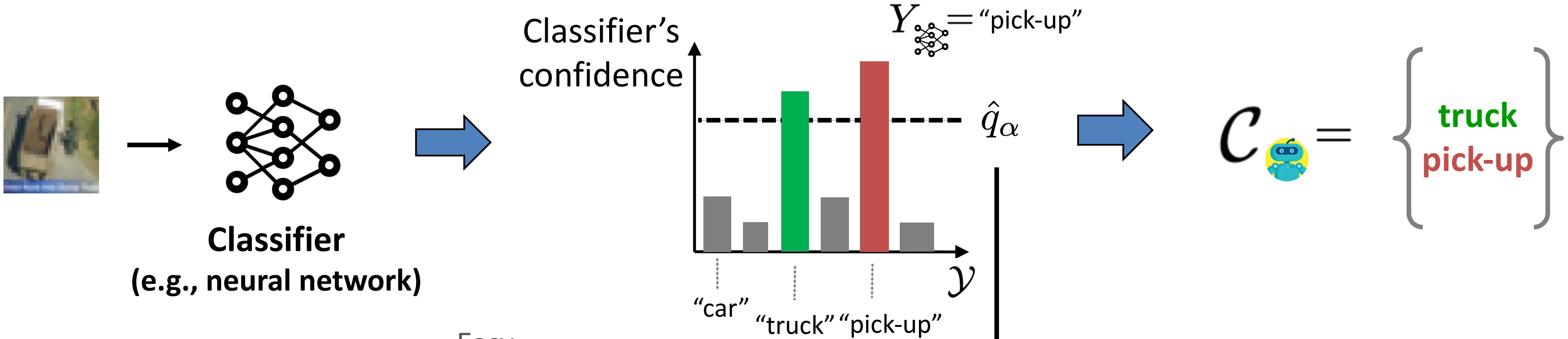
$1 - \alpha$
Desired coverage probability

CP guarantees that $\Pr(Y \in \mathcal{C}) = 1 - \alpha$

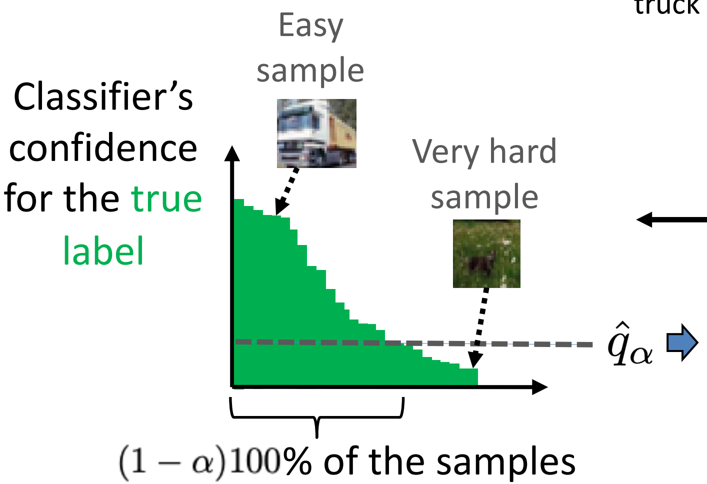
"car" "truck" "cat" "ship"

$1 - \alpha = \frac{3}{4}$ $\{$car$\}$ $\{$truck, bus$\}$ $\{$goat, cow$\}$ $\{$ship$\}$

Conformal predictors in a nutshell



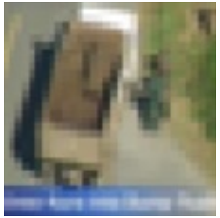
CP sets the threshold $\hat{q}_\alpha$ using a **calibration dataset** $\mathcal{D}$ of **samples** with **known** true labels



Classifier's confidence score for the true label is larger than $\hat{q}_\alpha$ for $(1-\alpha)100$ % of the samples

Conformal prediction is not magic

Depending on the desired coverage probability $1-\alpha$, the **size of the subsets** $\mathcal{C}$ constructed by a ***conformal predictor*** varies



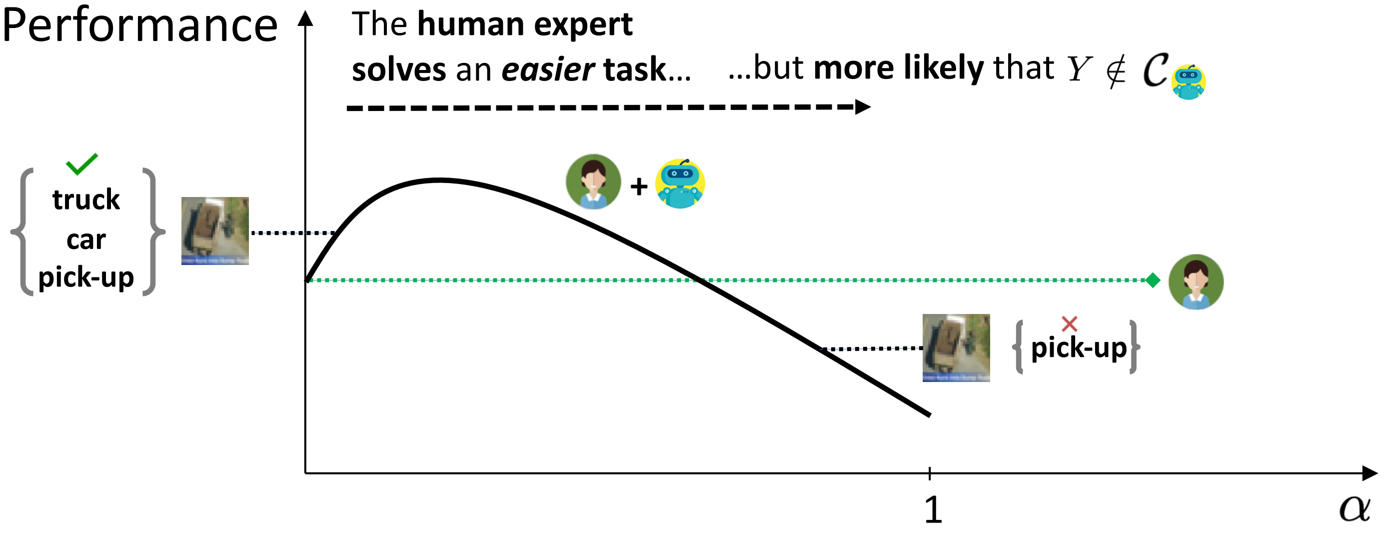
$$\mathcal{C} = \{\textbf{pick-up}\} \qquad \{\textbf{truck}, \textbf{pick-up}\} \qquad \{\textbf{truck}, \textbf{car}, \textbf{pick-up}\}$$

$$1-\alpha=0.5 \qquad 1-\alpha=0.9 \qquad 1-\alpha=0.99$$

The *larger* the desired coverage probability $1-\alpha$, the larger the subsets $\mathcal{C}$

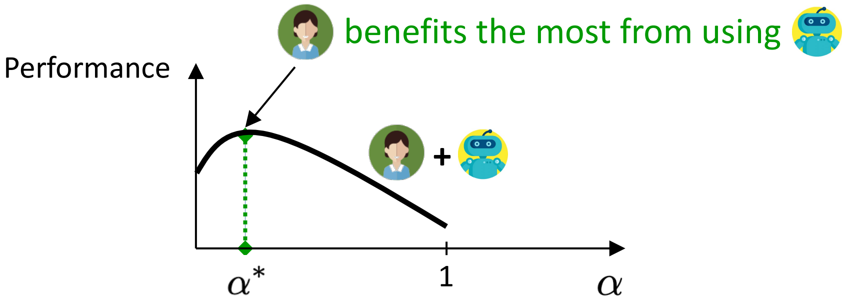
Optimizing across conformal predictors

The parameter α trade-offs **how frequently** the **system** will **mislead** the **human expert** & the **difficulty** of the **task** the **human needs to solve**



Bandit algorithms to find the optimal conformal predictor

To find the optimal parameter α^*,
we resort to **bandit algorithms**



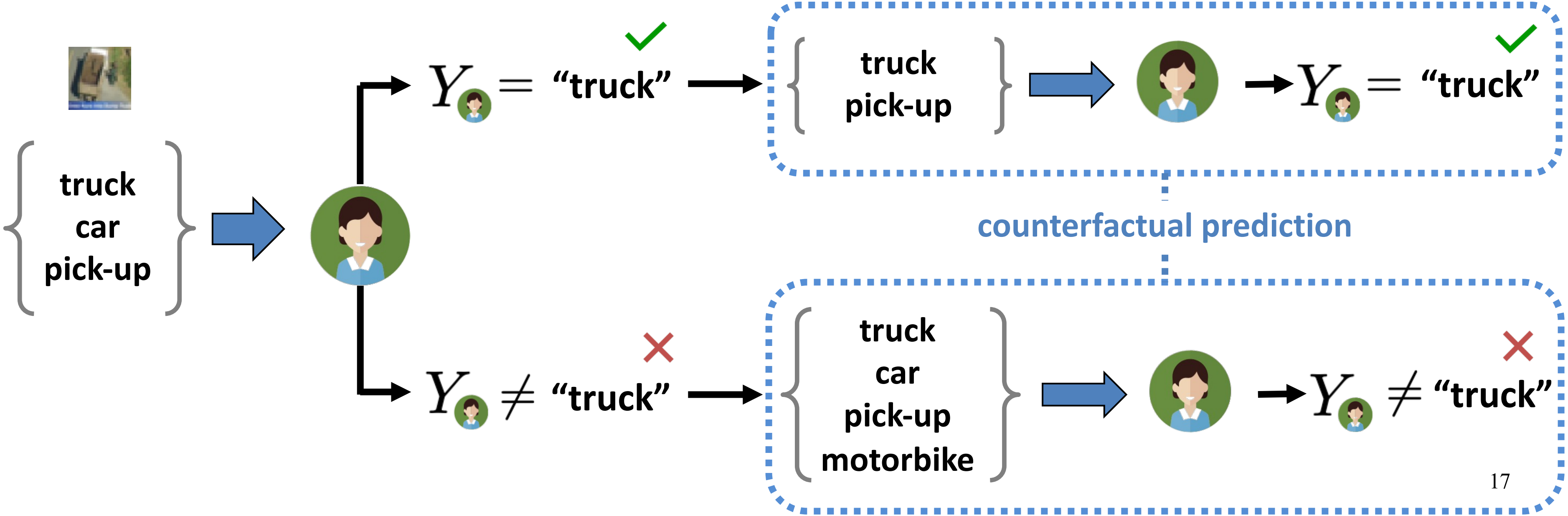
Bandit algorithms sequentially gather predictions by human experts using our system under different α values…

…$\underbrace{\textbf{prioritizing}}_{\text{many bandit algorithms, e.g., successive elimination, UCB1}}$ α values that seem **more promising over time**.

many bandit algorithms, e.g.,
successive elimination, UCB1

Efficient bandit algorithms using counterfactual monotonicity

We **speed-up** how quickly **bandit algorithms** gather predictions for different α values using a **counterfactual monotonicity assumption**



Exponential improvement in regret in successive elimination

For **successive elimination (SE)**, a well-known bandit algorithm, we show that **counterfactual monotonicity** allows for an **exponential improvement** in **regret**

number of time steps

$$\underbrace{R(T)}_{\text{regret}} = \underbrace{T \cdot \mathbb{E}_{\alpha^*}\left[\mathbb{I}\{Y = Y\}\right]}_{\text{performance under optimal } \alpha^*} - \underbrace{\sum_{t=1}^{T} \mathbb{E}_{\alpha_t}\left[\mathbb{I}\{Y = Y\}\right]}_{\text{performance under } \alpha_t \text{ values chosen by SE}}$$

Vanilla SE

$$\mathbb{E}[R(t)] \leq O(\sqrt{\underbrace{m}_{\text{size of the calibration dataset } \mathcal{D} \text{ used by CP}} t \log T})$$

SE with counterfactual monotonicity

$$\mathbb{E}[R(t)] \leq O(\sqrt{t \log m \log T})$$

Large-scale human subject study

We gather **194,407 predictions** from **2,751 human subjects** over **19,200 different pairs of natural images and subsets.**

Which one of the following categories fits better the image below?

- ○ Car
- ○ Airplane
- ○ Truck

Strict implementation

Human experts are allowed to pick $Y \in \mathcal{C} \subseteq \mathcal{Y}$

- ○ Car
- ○ Airplane
- ○ Truck
- ○ Other

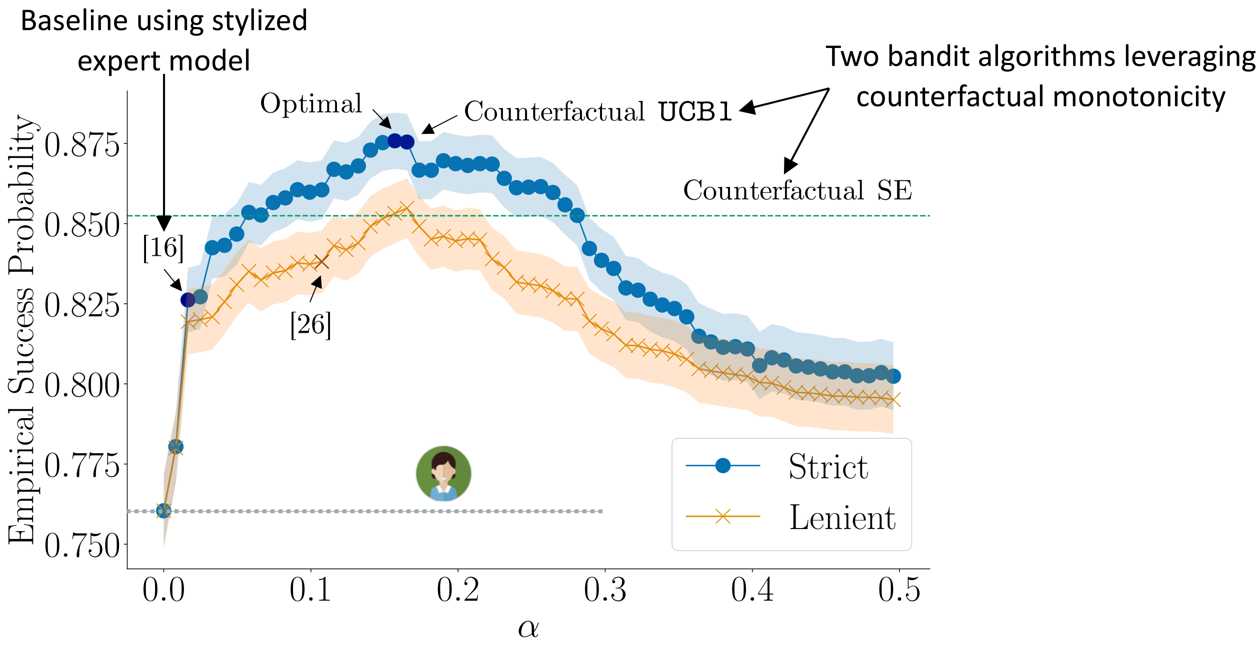
1. If you chose 'Other' above, please choose a category:

Choose ▾

Lenient implementation

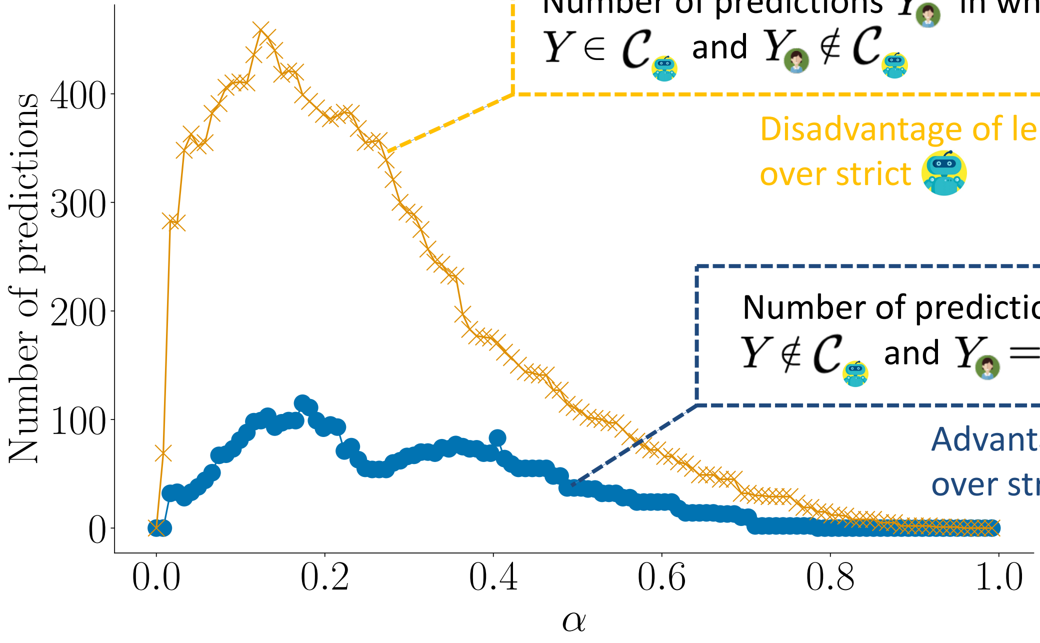
Human experts are allowed to pick $Y \in \mathcal{Y}$

Limiting expert's level of agency offers greater performance



The **strict implementation**, which **adaptively limits experts' agency**, **beats** the **lenient implementation**, which **allows experts** to **always exercise their agency**

Allowing experts to exercise their own agency does not pay off



Number of predictions Y in which $Y \in \mathcal{C}$ and $Y \notin \mathcal{C}$

Disadvantage of lenient over strict

Number of predictions Y in which $Y \notin \mathcal{C}$ and $Y = Y$

Advantage of lenient over strict

Moving beyond classification tasks

There are **many decision making processes** where one does not need to solve **classification tasks** but **other types of tasks.**

Huge amount of excitement about the possibility of using **sophisticated LLMs (e.g., ChatGPT)** to improve **decision making.**

→ However, human experts still need to **understand when to trust** the answers provided by LLMs.

Developing **trustworthy decision support systems** using **LLMs** is **highly non trivial.**

Thanks!

Improving Expert Predictions with Conformal Prediction, ICML 2023
https://arxiv.org/abs/2201.12006
https://github.com/Networks-Learning/improve-expert-predictions-conformal-prediction

Designing Decision Support Systems Using Counterfactual Prediction Sets, Arxiv 2023
https://arxiv.org/abs/2306.03928
https://github.com/Networks-Learning/counterfactual-prediction-sets

Eleni

Nastaran

Luke

Learn more about our research at
learning.mpi-sws.org