Conference on Computer and Communications Security (CCS ‘21)
Privacy and Anonymity Track – November 2021

Honest-but-Curious Nets:
Sensitive Attributes of Private Inputs can be Secretly Coded into the Classifiers’ Outputs

Mohammad Malekzadeh, Anastasia Borovykh and Deniz Gündüz

mmalekzadeh.github.io

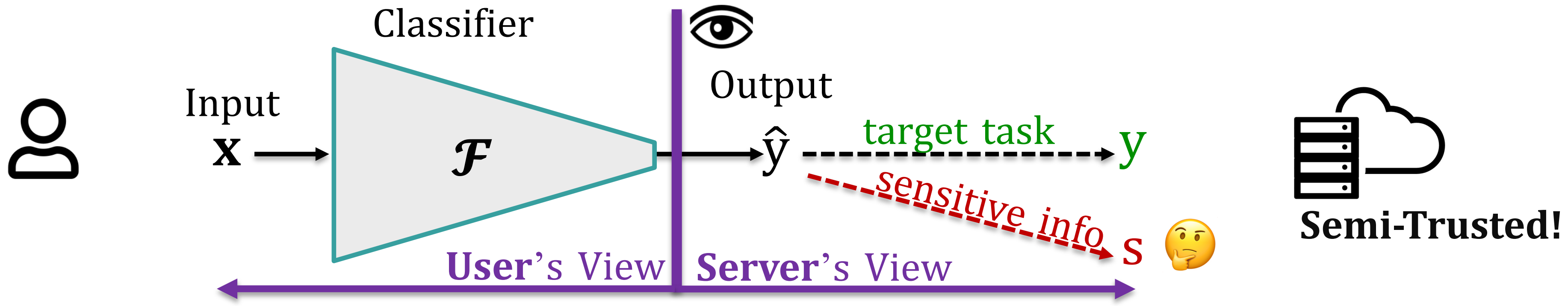
abrvkh.github.io

imperial.ac.uk/people/d.gunduz

Imperial College London

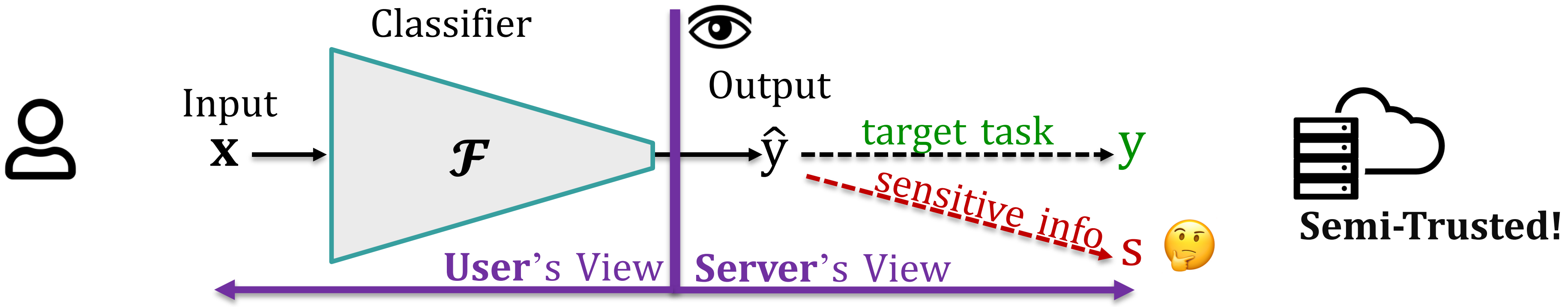
Acknowledgments:
- European Research Council (ERC) Starting Grant BEACON (no. 677854)
- UK EPSRC Grant within the CHIST-ERA program (no. EP/T023600/1)
- J.P. Morgan A.I. Research Award 2019

Overview



Classifier
Input
x
$\mathcal{F}$
Output
ŷ
target task
y
sensitive info
s
User's View
Server's View
Semi-Trusted!

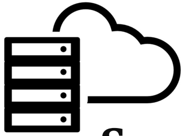
Overview



We show **how** the **output** of a classifier (e.g., a neural network) can secretly carry **sensitive** information about its **input.**

Our work challenges the privacy protection offered via **edge/on-device** or **encrypted/multi-party** approaches that **hide the input** and only **release the output**.

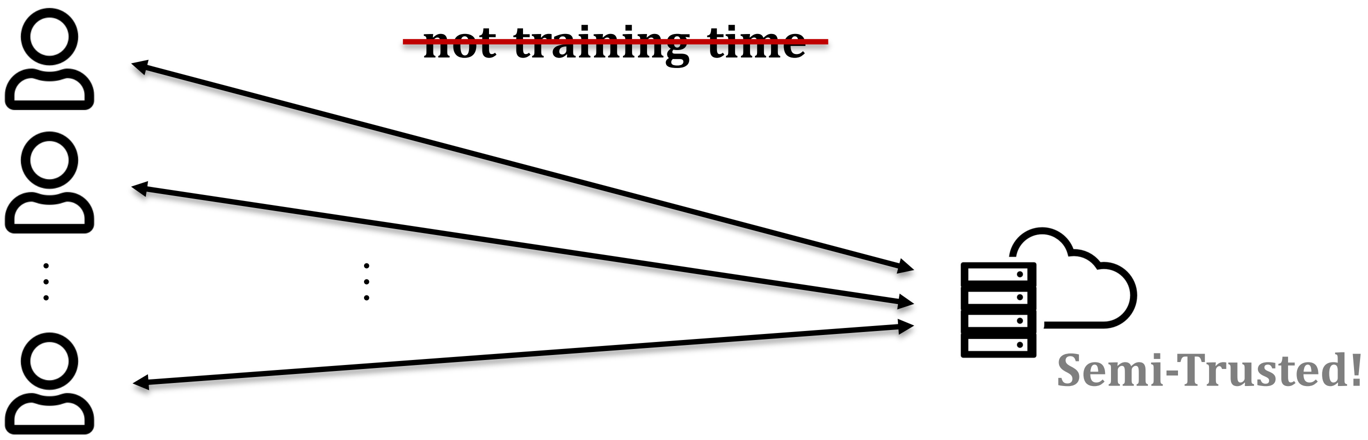
The Problem Setting



Semi-Trusted!

Server
(Machine Learning Service Provider)

At inference time (aka test time)



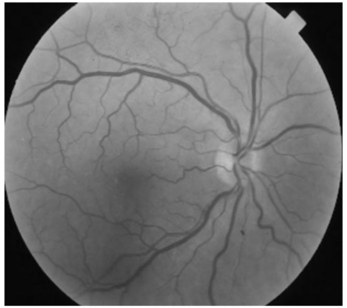
Users
(Private Data Owners)

Server
(Machine Learning Service Provider)

- $\mathbf{x}$: user's private sample

Face image

wikipedia.org/wiki/Roya_Mahboob



Retinal Vessel

Coyner, Aaron S., et al. (2021) arXiv:2109.13845

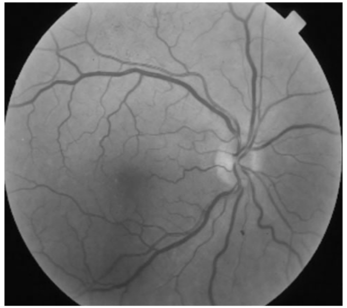
Chest X-Ray

Banerjee, Imon, et al. (2021) arXiv:2107.10356

, speech, text, sensors, …

- $\mathbf{x}$: user's private sample
- f and g are two unknown **ground-truth** functions.
- $y = f(\mathbf{x})$: a target attribute … users wish to **release** to the server
- $s = g(\mathbf{x})$: a sensitive attribute … users wish to **keep** it private

Face image
wikipedia.org/wiki/Roya_Mahboob



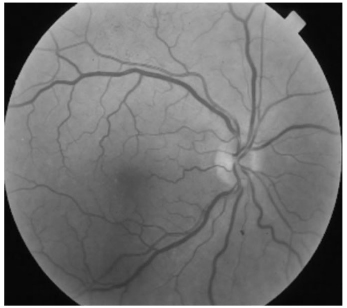
Retinal Vessel
Coyner, Aaron S., et al. (2021)
arXiv:2109.13845

Chest X-Ray
Banerjee, Imon, et al. (2021)
arXiv:2107.10356

, speech, text, sensors, …

- $\mathbf{x}$: user's private sample
- f and g are two unknown **ground-truth** functions.
- $y = f(\mathbf{x})$: a target attribute … users wish to **release** to the server
- $s = g(\mathbf{x})$: a sensitive attribute … users wish to **keep** it private

y : **age**
s : **race**

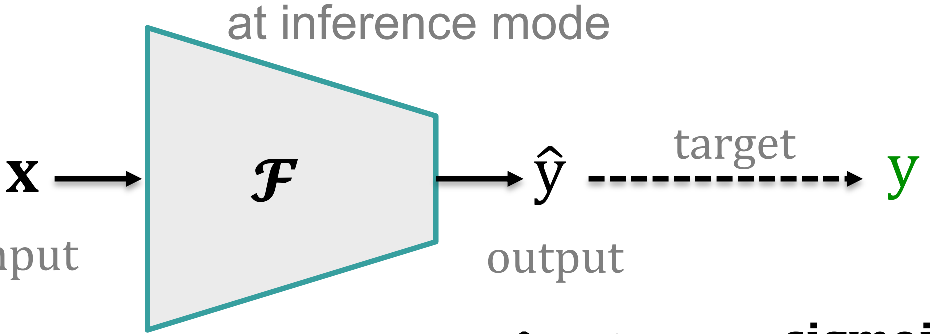


y : **disorder**
s : **race**

y : **disease**
s : **race**

, speech, text, sensors, …

- **Server** provides a pre-trained **classifier**
- $\mathcal{F}(\mathbf{x})$: a classifier that approximates $\mathrm{y} = \mathrm{f}(\mathbf{x})$



e.g.: face image

f : **age ≥18**
$\mathbf{x} \rightarrow \{\text{No}, \text{Yes}\}$

$\hat{y} = 1.7$

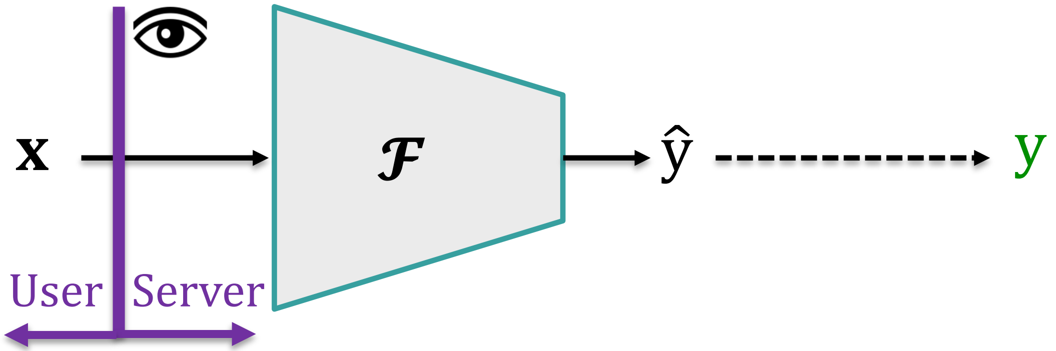
$$\mathbf{sigmoid}(\hat{y}) = \frac{\mathbf{1}}{\mathbf{1} + e^{-\hat{y}}} = 0.85$$

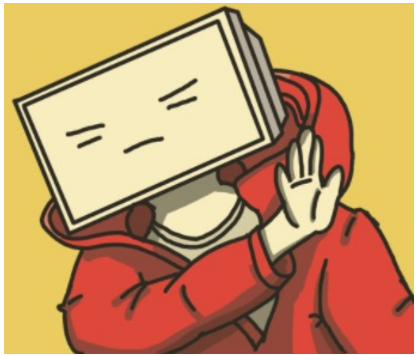
(Soft output ≈ probability of "Yes")

User-Server Interaction Models

Non-Encrypted Cloud Computing

- $\mathcal{F}$ is hosted in the cloud
- Server observes the private data



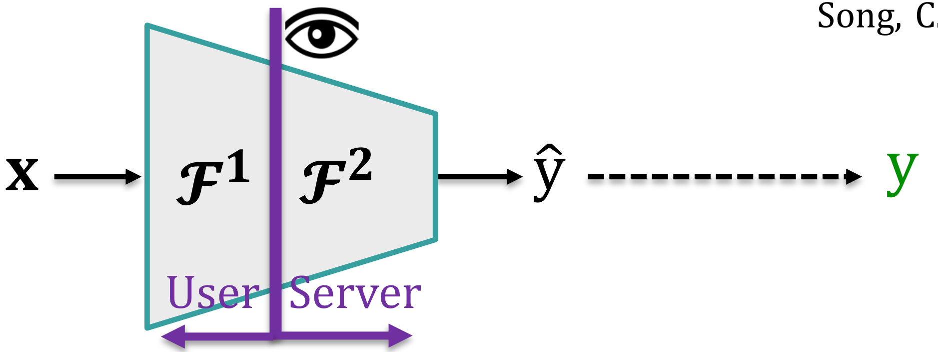


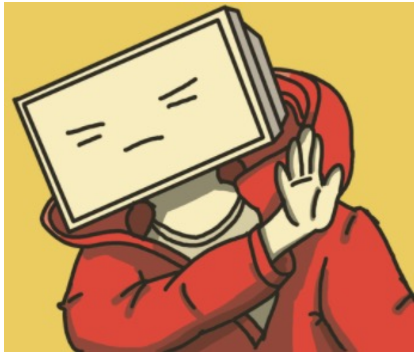
System32Comics

Split Computing

- $\mathcal{F} = \mathcal{F}^2\left(\mathcal{F}^1(\mathbf{x})\right)$
- Server only observes the output of $\mathcal{F}^1$.
- But Server still can infer sensitive attributes.

Song, C., & Shmatikov, V., ICLR 2020

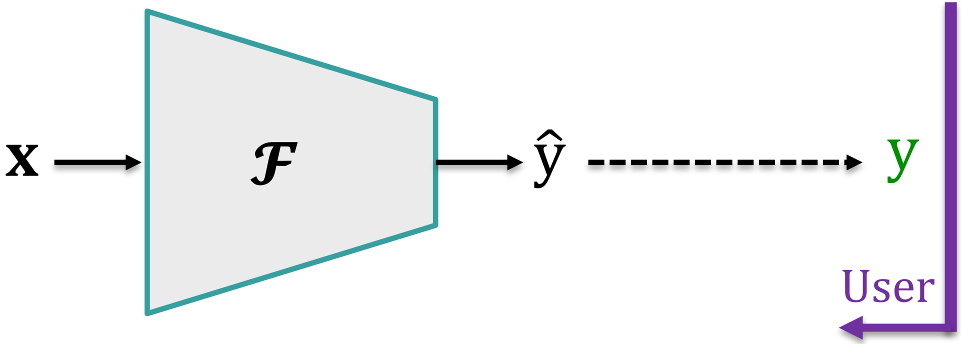


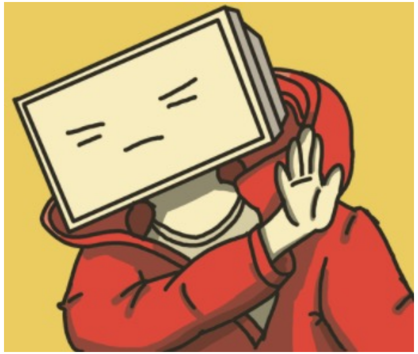


System32Comics

No Server?

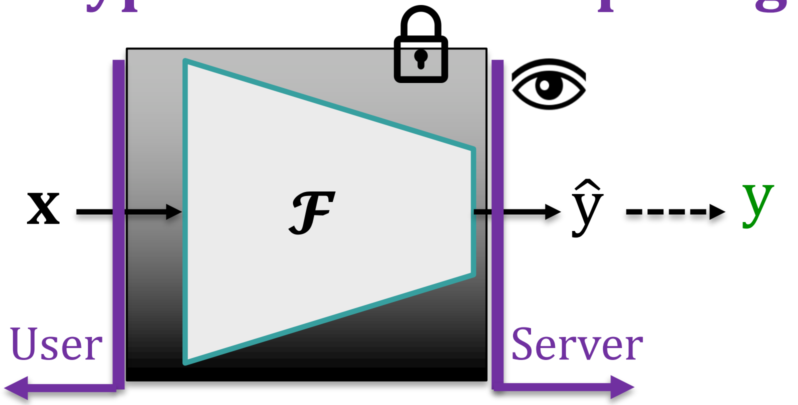
- Then who gives us the service!



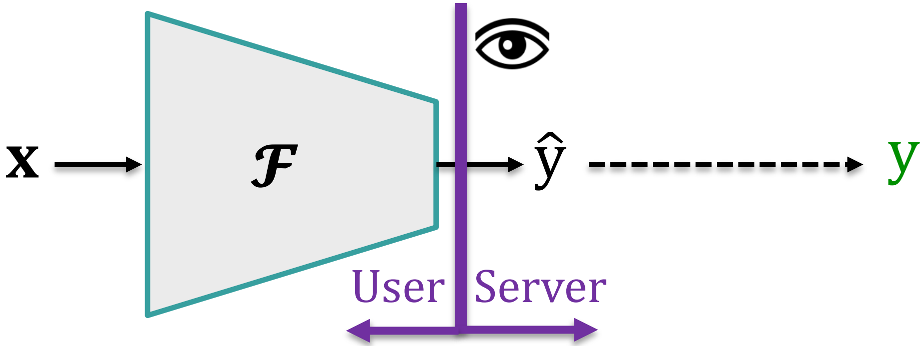


System32Comics

- Encrypted Cloud Computing



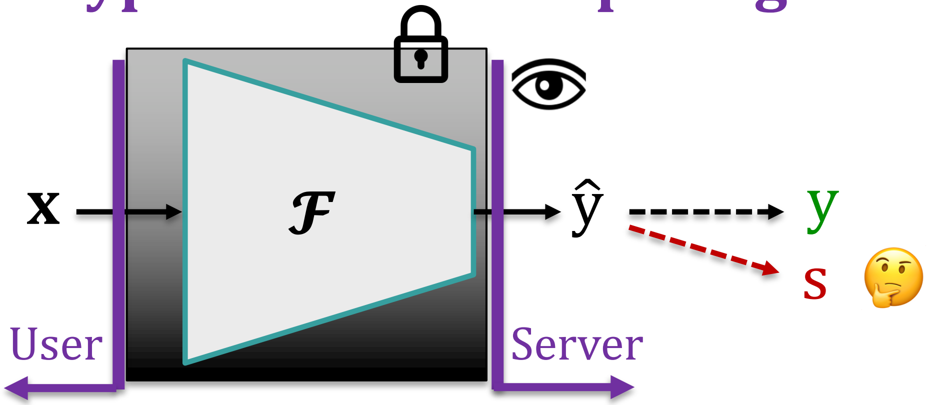
- On-Device/Edge Computing



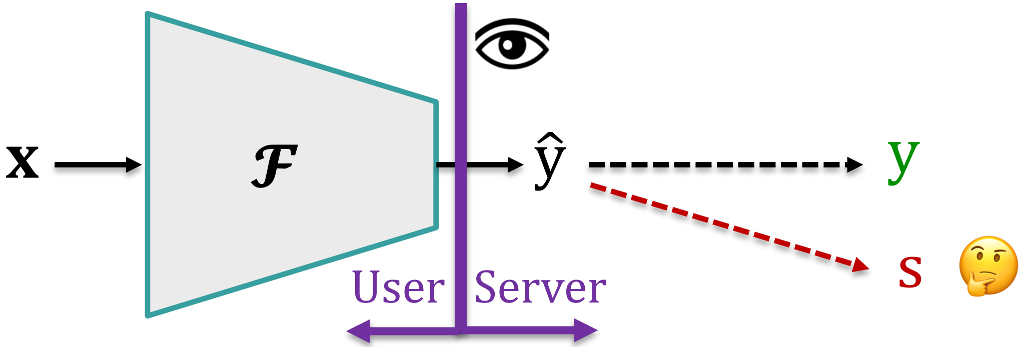
- **Current solutions**
- **Server only observes the output**

System32Comics

- **Encrypted Cloud Computing**



- **On-Device/Edge Computing**



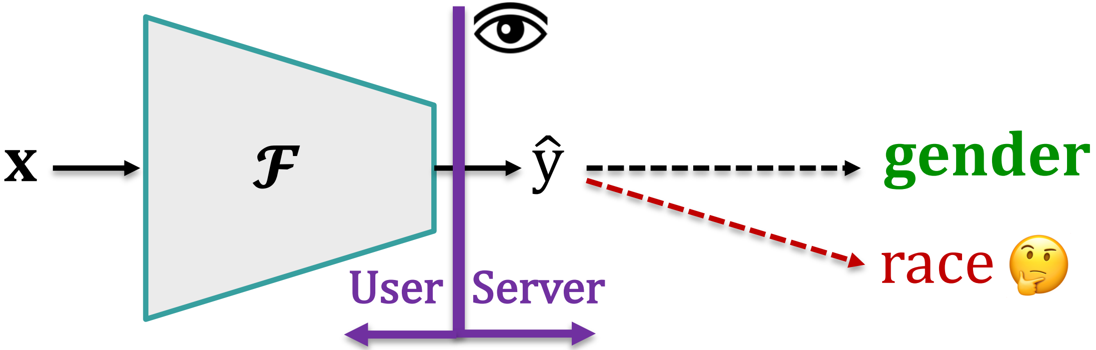
Our Main Question

Can Server infer a **sensitive** attribute of private input from the **target** output?

Especially, for **uncorrelated** attributes
e.g., **gender** and **race**

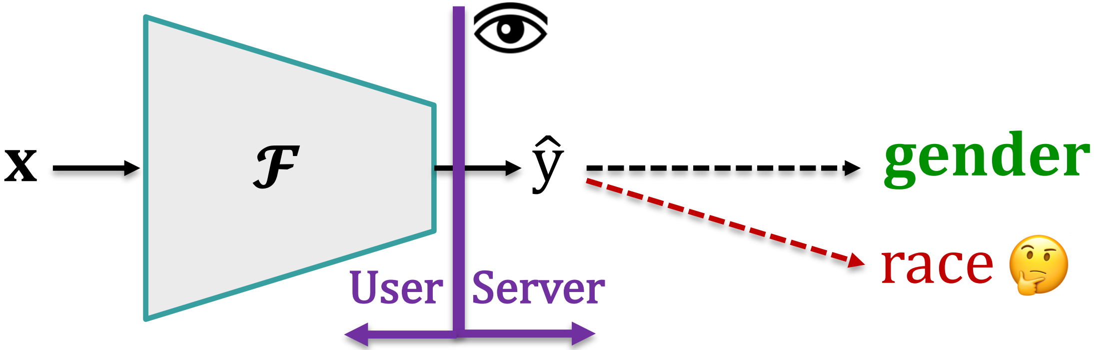
An Example

Two **Uncorrelated** Attributes



To predict the **race** from the **output** of a binary **gender** classifier

Two **Uncorrelated** Attributes



To predict the **race** from the **output** of a binary **gender** classifier

UTKFace Dataset
(Zhang, Zhifei, et al. CVPR 2017)

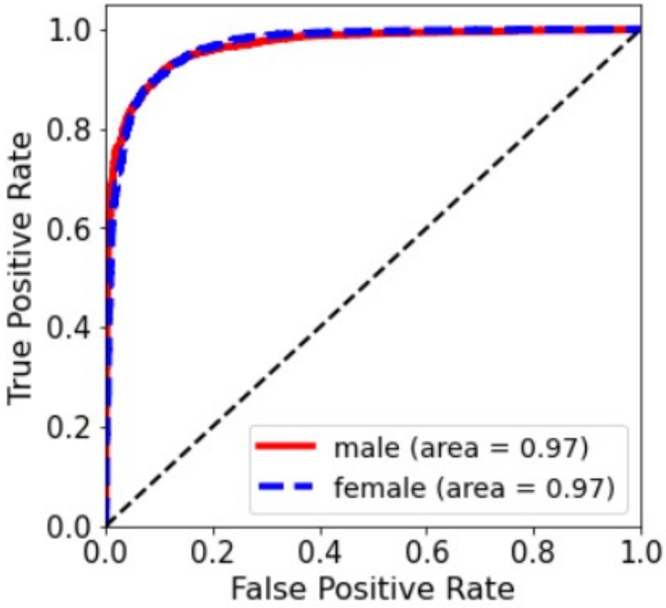
random guess

y : gender {male 52%, female 48% }   s : race { white 42% , non-white 58%}

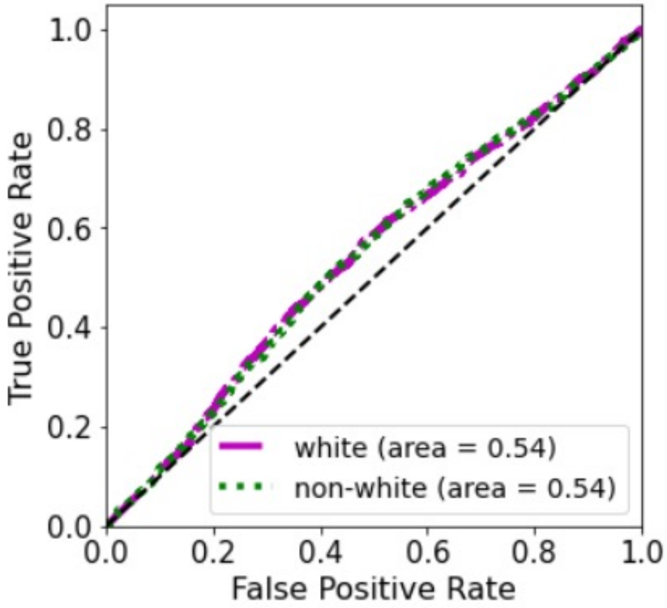
ROC curves of a “**standard**” classifier

[four convolutional layers + two fully-connected layers]

y : gender {male 52%, female 48% }



s : race { white 42% , non-white 58%}

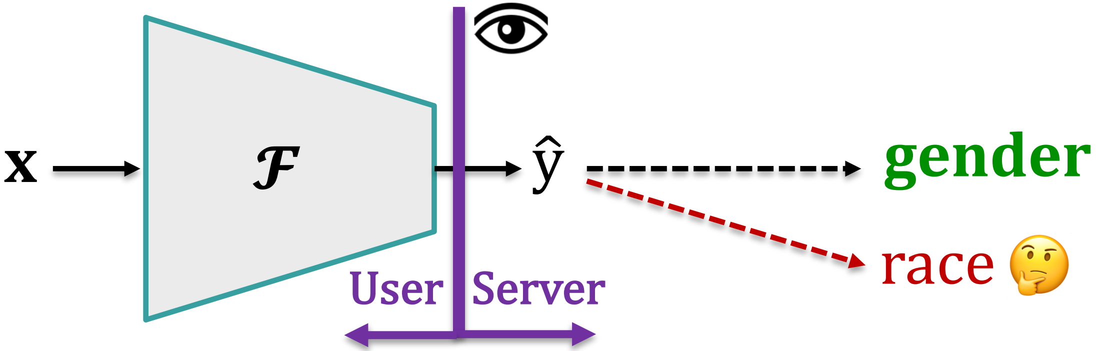


* a classifier that is trained only using the “cross-entropy loss function” for the target attribute.

Our Question

To predict the **race** from the **output** of a binary **gender** classifier

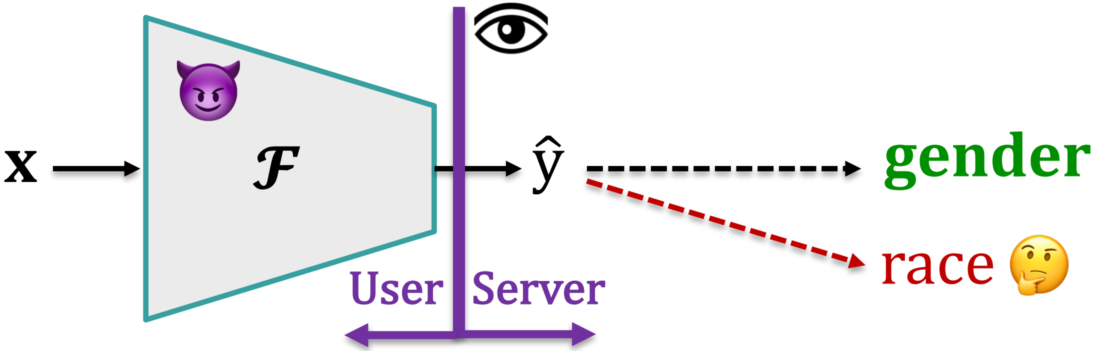
Initial Answer**:**
If $\mathcal{F}$ is a **standard*** classifier, then "**No**".



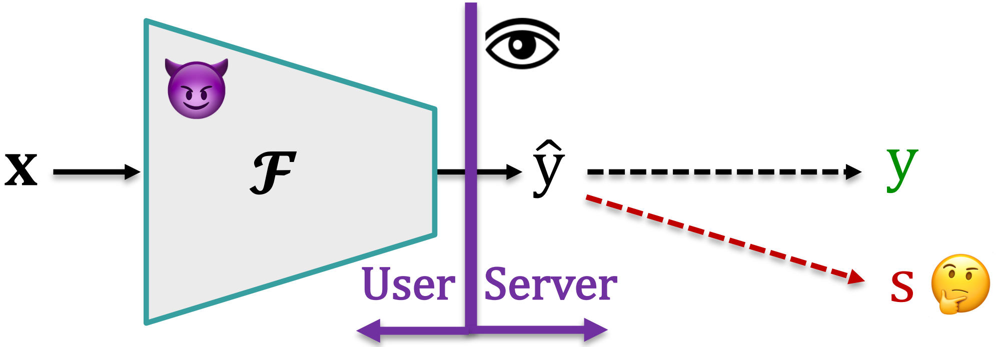
Our **new** Question

To predict the **race** from the **output** of a binary **gender** classifier

What if the classifier is not "standard"?



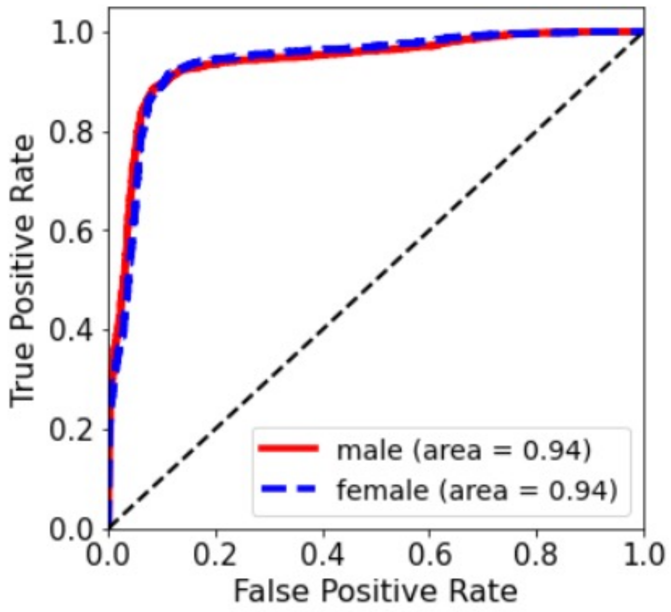
Honest But Curious (HBC) Classifiers



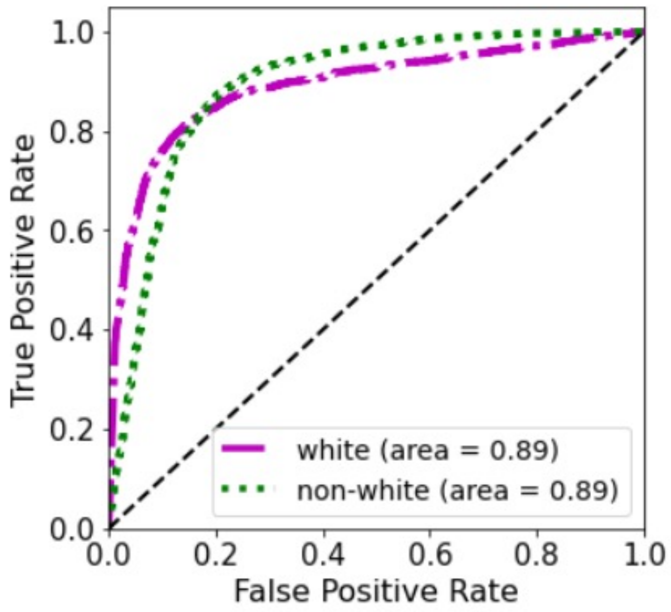
- **Honesty:** $\hat{y}$ accurately estimates the target attribute.
- **Curiosity**: $\hat{y}$ also reveals a sensitive attribute.

ROC curves of a "**HBC**" classifier

y : gender {male 52%, female 48% }



s : race { white 42% , non-white 58%}

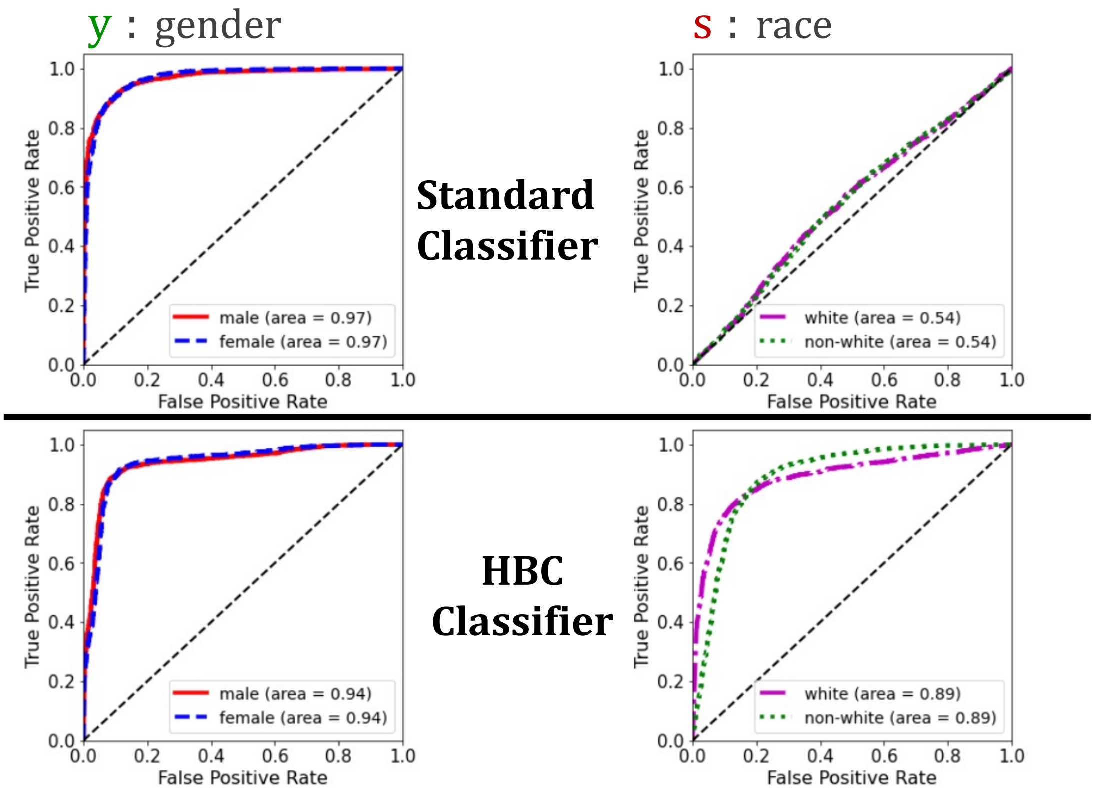


* a classifier that is trained only using the "cross-entropy loss function" for the target attribute.

“Standard” vs. “HBC” classifiers

- **Same**
  - Model
  - Dataset
  - Initialization
  - Hyperparameters

- **The only difference is the training procedure!**



Methodology

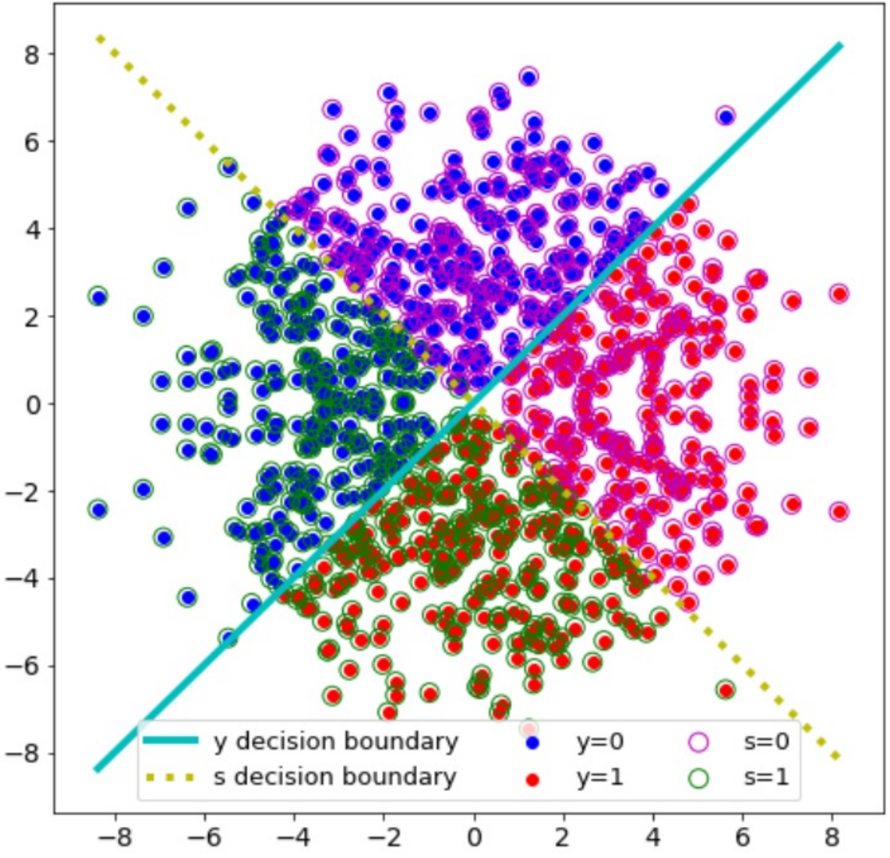
Notation

$(\delta^{\mathbf{y}} \text{——} \ \delta^{\mathbf{s}})$-HBC

- $\delta^{\mathbf{y}} \in [0,1]$: the accuracy for the **target** attribute
- $\delta^{\mathbf{s}} \in [0,1]$: the accuracy for the **sensitive** attribute

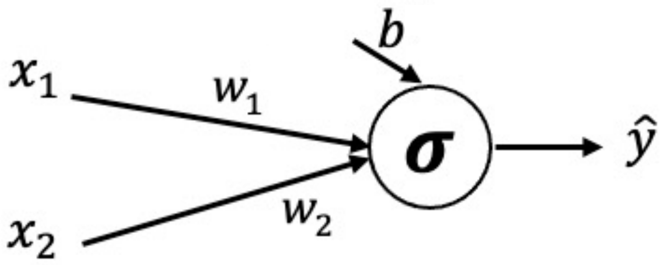
A Synthetic 2-d Dataset

- Two **uncorrelated** labels each having two classes: {0, 1}
  - y
  - s
- Samples of each label are **linearly** separable



Linear Classifier 1

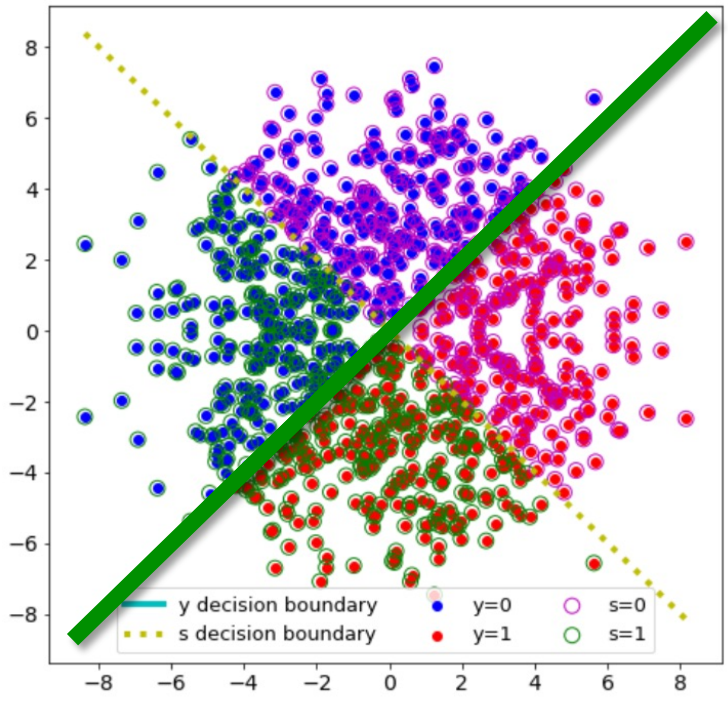
$(\delta^{y}\text{——}\ \delta^{s})$-HBC Logistic Regression



$$\delta^{y} = 1$$
$$\delta^{s} = 0.5$$

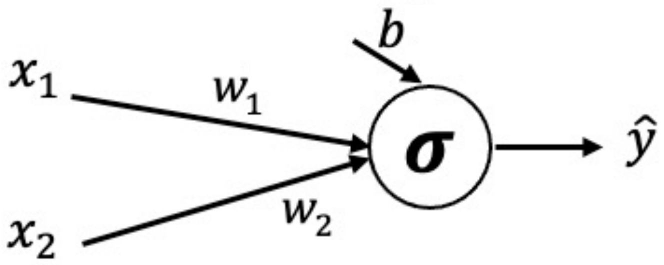
Perfect Honesty

No Curiosity



Linear Classifier 2

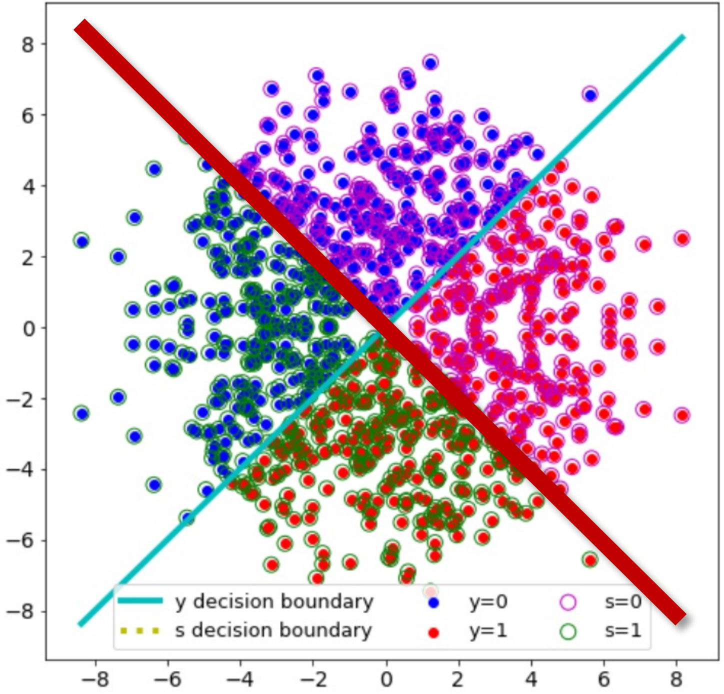
(δ^{y}—— δ^{s})-HBC Logistic Regression



$$\delta^{y} = 0.5$$
$$\delta^{s} = 1$$

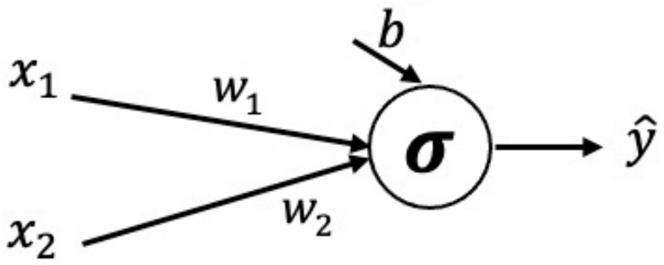
No Honesty

Perfect Curiosity



Linear Classifier 3

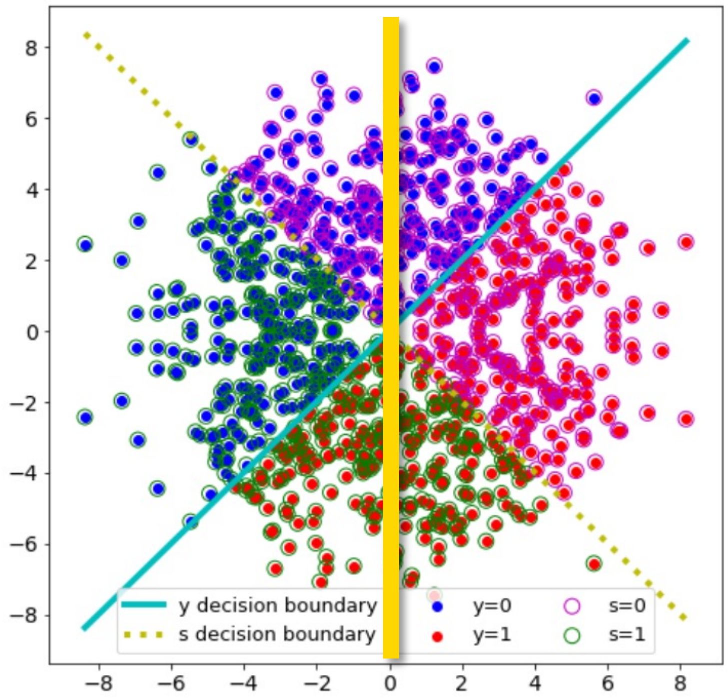
$(\delta^{y} —— \delta^{s})$-HBC Logistic Regression



$$\delta^{y} = 0.75$$
$$\delta^{s} = 0.75$$

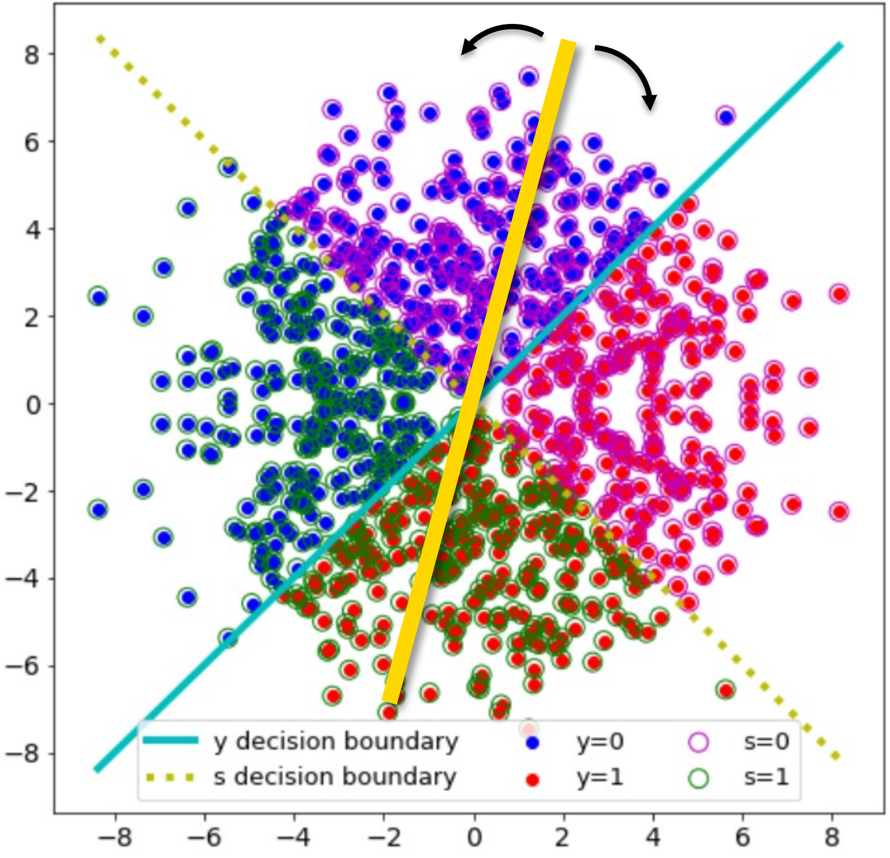
Weak Honesty

Weak Curiosity

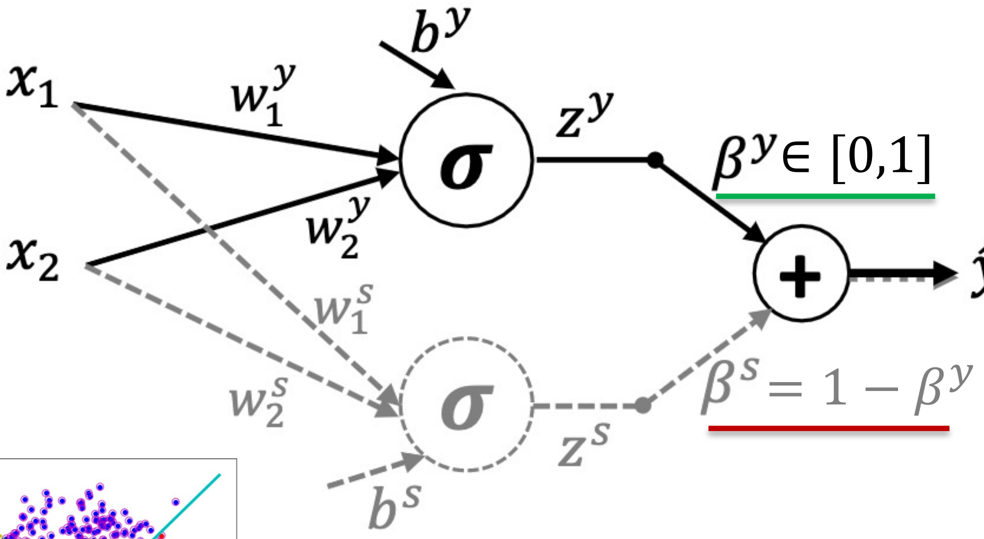


Initial Observation

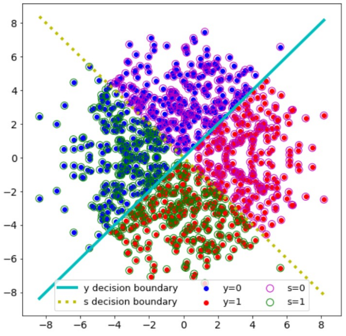
A **linear** classifier cannot become both **honest** and **curious** at the same time.



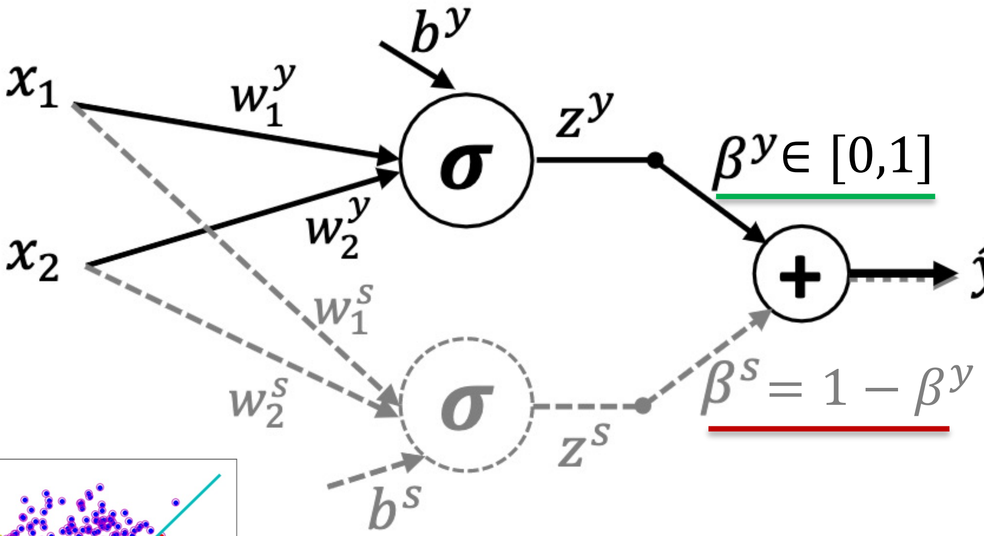
What if we double the capacity?



$$\hat{y} = \beta^y \lfloor z^y \rceil + \beta^s \lfloor z^s \rceil$$

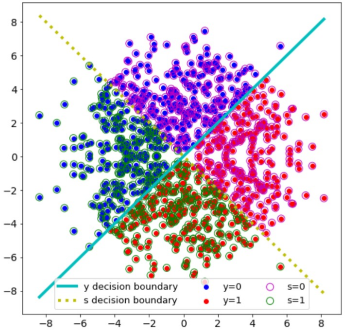


What if we double the capacity?



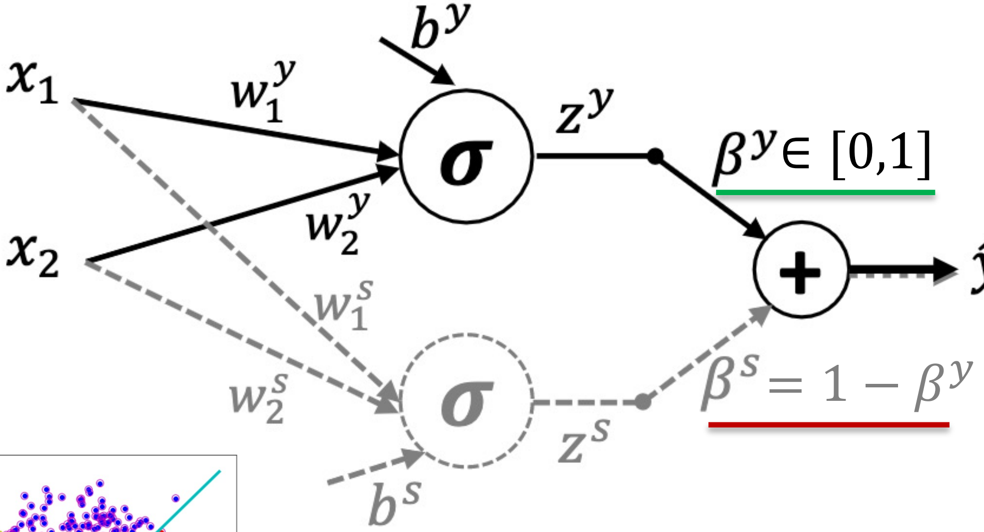
$$\hat{y} = \beta^y \lfloor z^y \rceil + \beta^s \lfloor z^s \rceil$$

Round to the nearest integer



No linear anymore!

We get the Perfect Honesty & Curiosity

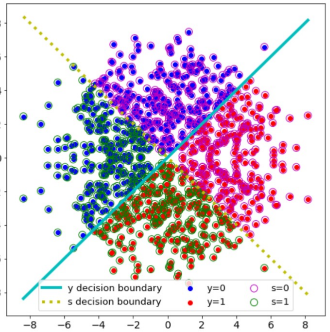


Works for any $\beta^y \neq 0.5$

$$\hat{y} = \beta^y \lfloor z^y \rceil + \beta^s \lfloor z^s \rceil$$

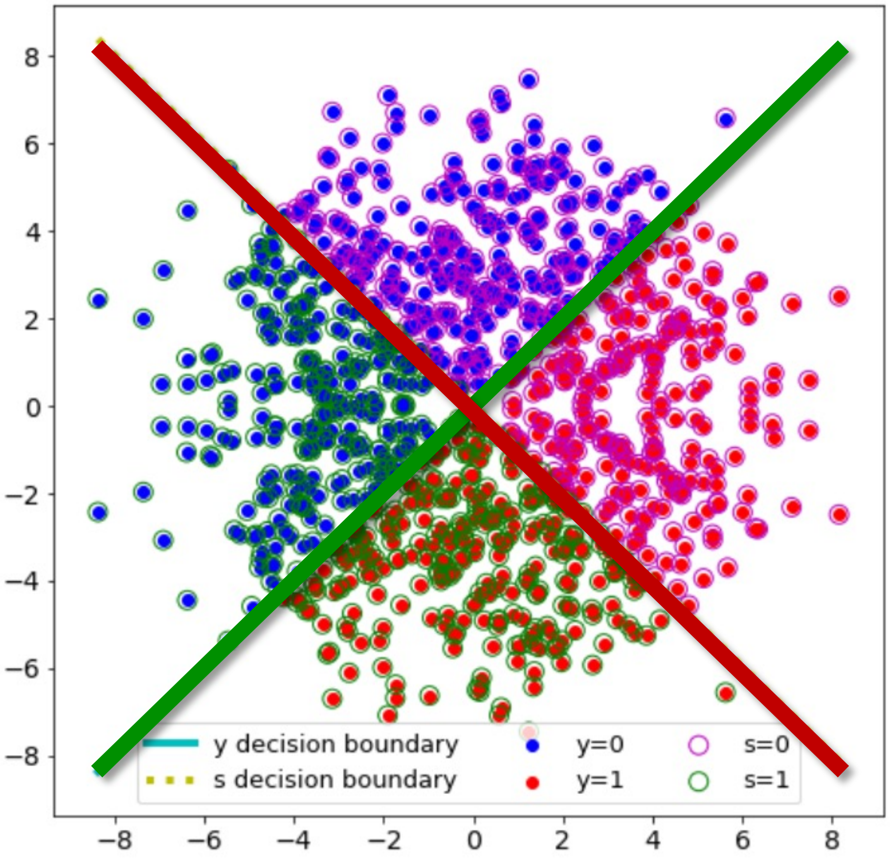
Round to the nearest integer

$$(\delta^y = 1 \text{ -- } \delta^s = 1)\text{-HBC} = \begin{cases} 0 & \text{if } y = 0, s = 0 \\ \beta^s & \text{if } y = 0, s = 1 \\ \beta^y & \text{if } y = 1, s = 0 \\ 1 & \text{if } y = 1, s = 1 \end{cases}$$



New Observation

The combination of **two linear** classifiers can become both **honest** and **curious** at the same time.



Expanding the Idea

- What if we cannot use an arbitrary **architecture** (like prev. example)?

- Given any classifier $\mathcal{F}$, can we **train** it such that at **inference** time:
  1. $\mathcal{F}$ be **honest**, like a standard classifier for y,
  2. $\mathcal{F}$ be **curious**, like as a standard classifier for s,
  3. $\mathcal{F}$ cannot be easily distinguished from a **standard** classifier?

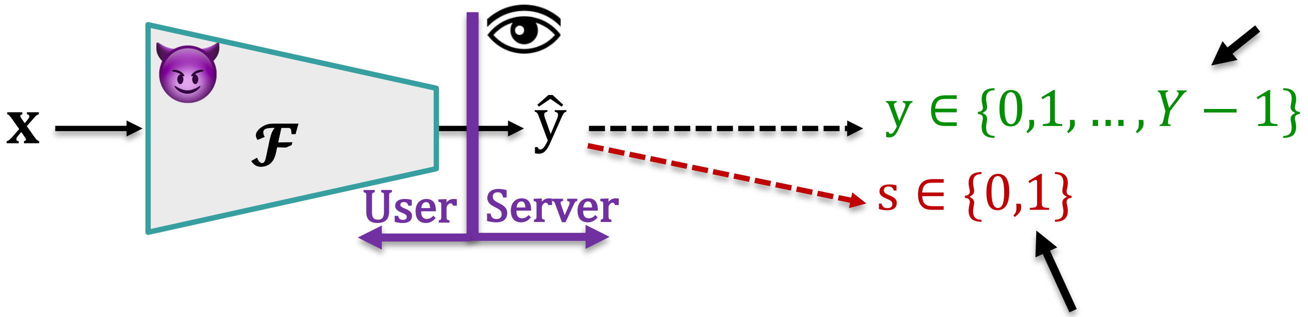
  Even if users have full-access white-box view of $\mathcal{F}$.

We Introduce Two Solutions

(1) **Regularized**: for binary s

(2) **Parameterized**: for categorical s

1) **Regularized**: for binary attributes



ArgMax vs. Entopy

$A.\ \ \hat{y} = [0.95, 0.05]:$ argmax($\hat{y}$) = 0 , H($\hat{y}$) = 0.29

$B.\ \ \hat{y} = [0.75, 0.25]:$ argmax($\hat{y}$) = 0 , H($\hat{y}$) = 0.81

$$\text{entropy} \rightarrow \mathrm{H}(\hat{y}) = \sum_i \widehat{y_i}\ log\, \widehat{y_i}$$

Exploiting the Entropy

- **Training Loss**

Maximize/minimize the entropy for s

$$\mathcal{L}^b = -\beta^y \boxed{\sum_{i=0}^{Y-1} y_i \log \hat{y}_i} - \beta^s \boxed{\left(\mathbb{I}_{(s=0)}\left(\sum_{i=0}^{Y-1} \hat{y}_i \log \hat{y}_i\right) - \mathbb{I}_{(s=1)}\left(\sum_{i=0}^{Y-1} \hat{y}_i \log \hat{y}_i\right)\right)}$$

Typical cross-entropy for y

Exploiting the Entropy

- **Training Loss**

Maximize/minimize the entropy for s

$$\mathcal{L}^b = -\beta^y \boxed{\sum_{i=0}^{Y-1} y_i \log \hat{y}_i} - \beta^s \boxed{\left(\mathbb{I}_{(s=0)}\left(\sum_{i=0}^{Y-1} \hat{y}_i \log \hat{y}_i\right) - \mathbb{I}_{(s=1)}\left(\sum_{i=0}^{Y-1} \hat{y}_i \log \hat{y}_i\right)\right)}$$

Typical cross-entropy for y

- **Inference**

The threshold is decided via a validation set

y = argmax(ŷ)

$$s = \begin{cases} 0, & \text{if } \mathsf{H}(\hat{\mathbf{y}}) \leq \tau \\ 1, & \text{otherwise} \end{cases}$$

Exploiting the Entropy

- **Training Loss**

Maximize/minimize the entropy for s

$$\mathcal{L}^b = -\beta^y \boxed{\sum_{i=0}^{Y-1} y_i \log \hat{y}_i} - \beta^s \boxed{\left(\mathbb{I}_{(s=0)}\left(\sum_{i=0}^{Y-1} \hat{y}_i \log \hat{y}_i\right) - \mathbb{I}_{(s=1)}\left(\sum_{i=0}^{Y-1} \hat{y}_i \log \hat{y}_i\right)\right)}$$

Typical cross-entropy for y

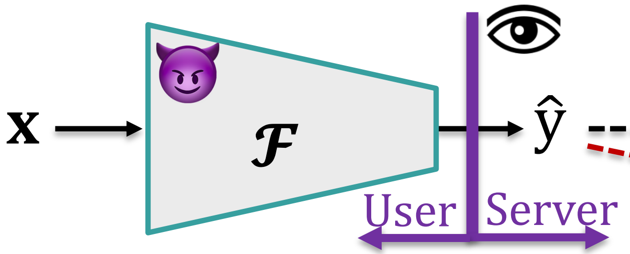
- **Inference**

y = argmax($\hat{\mathrm{y}}$)

The threshold is decided via a validation set

$$s = \begin{cases} 0, & \text{if } \mathrm{H}(\hat{\mathrm{y}}) \leq \tau \\ 1, & \text{otherwise} \end{cases}$$

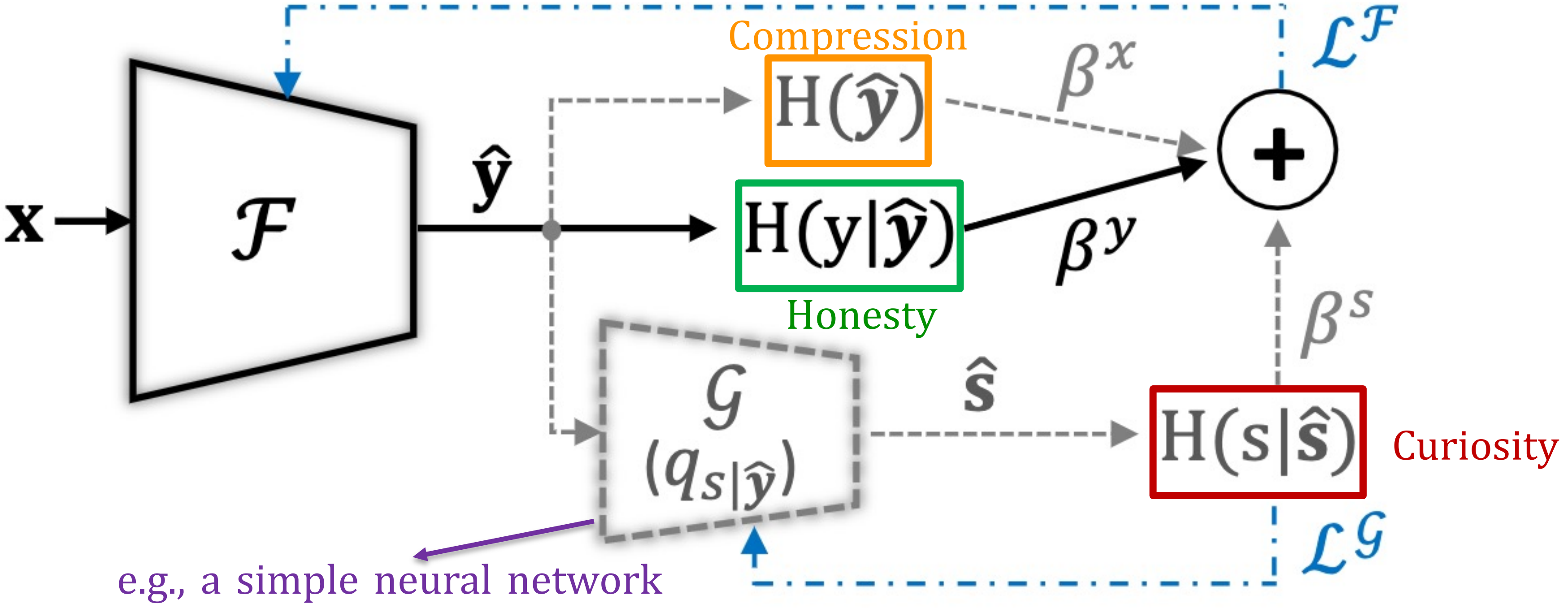
2) **Parameterized**: for general attributes



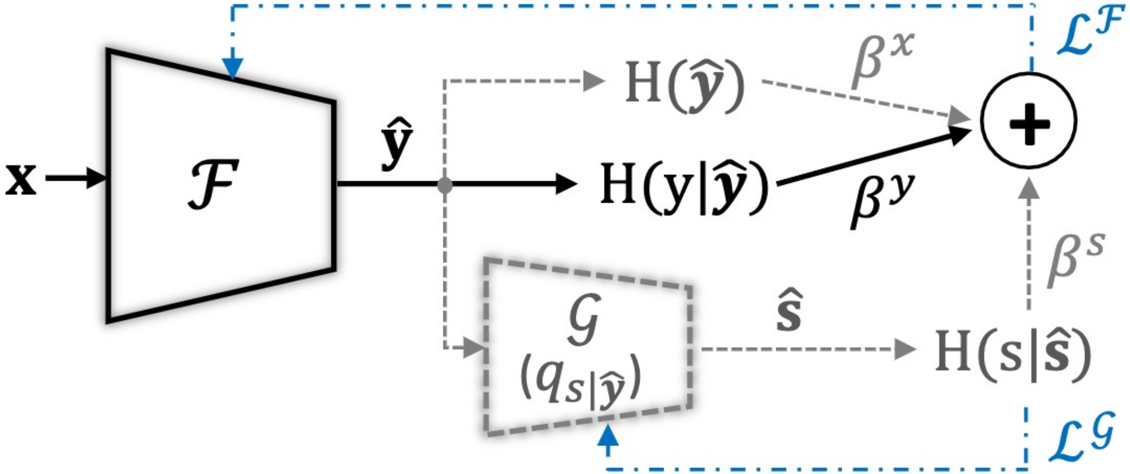
$y \in \{0,1,\ldots,Y-1\}$

$s \in \{0,1,\ldots,S-1\}$

Based on **information bottleneck principle** and using **variational approximation** of conditional entropies.



Based on **information bottleneck principle** and using **variational approximation** of conditional entropies.



$$\min_{(\mathbf{x},y,s)\leftarrow\mathcal{D},\ \mathcal{F}\in\mathbb{F},\ \hat{\mathbf{y}}\leftarrow\mathcal{F}(\mathbf{x}),\hat{\mathbf{s}}=\mathcal{G}(\hat{\mathbf{y}})} \left[\mathcal{H} = \beta^x \mathrm{H}(\hat{\mathbf{y}}) + \beta^y \mathrm{H}(y|\hat{\mathbf{y}}) + \beta^s \mathrm{H}(s|\hat{\mathbf{y}})\right]$$

Experimental Results

CelebA Dataset

(Liu, Ziwei, et al. ICCV 2015)

y	Smile			
s	**MouthOpen**	**Makeup**	**Male**	**WavyHair**
MI(y, s)	0.231	0.024	0.015	0.003
Easiness(s)	93.4%	89.0%	97.7%	77.3%

CelebA Dataset

(Liu, Ziwei, et al. ICCV 2015)

y	Smile							
s	**MouthOpen**		**Makeup**		**Male**		**WavyHair**	
MI(y, s)	0.231		0.024		0.015		0.003	
Easiness(s)	93.4%		89.0%		97.7%		77.3%	
Model	δ^y%	δ^s%	δ^y%	δ^s%	δ^y%	δ^s%	δ^y%	δ^s%
standard	92.1	79.6	92.1	68.2	92.1	61.0	92.1	57.9
HBC (R)	91.7	91.2	91.6	96.0	91.6	96.0	91.7	73.5
HBC (P)	91.8	93.4	92.3	97.2	92.3	97.2	92.1	76.7

UTKFace Dataset

(Zhang, Zhifei, et al. CVPR 2017)

Parameterized Method

$S = 2$ gender; age

Model	$Y=3$: $\delta^y\%$	$Y=3$: $\delta^s\%$	$Y=4$: $\delta^y\%$	$Y=4$: $\delta^s\%$	$Y=5$: $\delta^y\%$	$Y=5$: $\delta^s\%$
standard	85.5	56.0	81.2	58.9	80.7	61.1
HBC (Raw)	85.8	89.1	82.1	89.2	81.4	89.0
HBC (Soft)	85.7	83.5	81.1	83.6	80.5	88.4

$$\textbf{softmax}(\hat{y}) = \boldsymbol{softmax}(\hat{y} + a) \textbf{ for all } \boldsymbol{a}$$

Vulnerability to Knowledge Distillation

$$\mathcal{L}^{KL} = \sum_{i=1}^{Y} \hat{y}_i^{Teacher} \log\left(\hat{y}_i^{Teacher} / \hat{y}_i^{Student}\right)$$

CelebA Dataset
(Liu, Ziwei, et al. ICCV 2015)

y	Smile							
s	MouthOpen		Makeup		Male		WavyHair	
Model	δ^y%	δ^s%	δ^y%	δ^s%	δ^y%	δ^s%	δ^y%	δ^s%
Teacher	90.1	89.1	90.7	85.2	90.2	92.7	92.0	61.1
Student	90.3	88.1	91.0	82.6	90.2	91.6	91.8	59.7

Examining HBC models

The average entropy of HBC models

y: Age 3-classes: ≤ 20 & 21-35 & >35

s: Race 3-classes: White & Asian & Others

Model	δ^{y}	δ^{s}	Avg. Entropy
standard	81.43	58.14	0.48
HBC (raw)	81.32	82.97	0.63

Without Compression $\boldsymbol{\beta^x = 0}$

$$\min_{(\mathbf{x},y,s)\leftarrow\mathcal{D},\ \mathcal{F}\in\mathbb{F},\ \hat{\mathbf{y}}\leftarrow\mathcal{F}(\mathbf{x}),\hat{\mathbf{s}}=\mathcal{G}(\hat{\mathbf{y}})} \left[\mathcal{H} = \boxed{\beta^x \mathrm{H}(\hat{\mathbf{y}})} + \boxed{\beta^y \mathrm{H}(y|\hat{\mathbf{y}})} + \boxed{\beta^s \mathrm{H}(s|\hat{\mathbf{y}})}\right]$$

0.7 **0.3**

The average entropy of HBC models

y: Age 3-classes: $\leq$ 20 & 21-35 & >35

s: Race 3-classes: White & Asian & Others

Model	δ^y	δ^s	Avg. Entropy	δ^y	δ^s	Avg. Entropy
standard	81.43	58.14	0.48	80.90	58.60	0.40
HBC (raw)	81.32	82.97	0.63	80.95	83.86	0.39

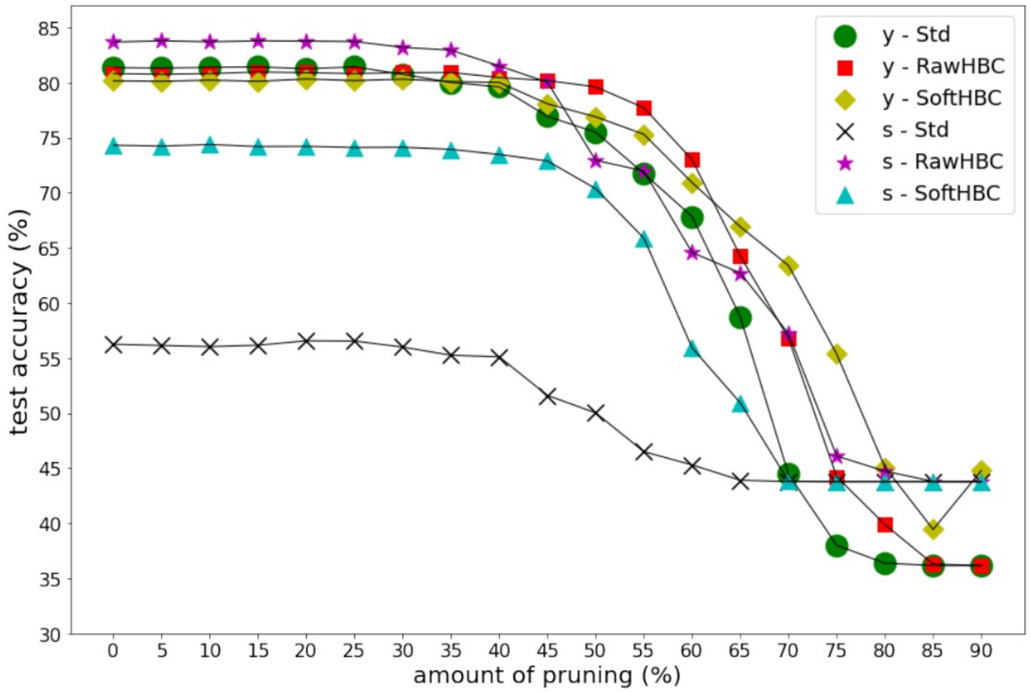
Without Compression $\boldsymbol{\beta^x = 0}$ With Compression $\boldsymbol{\beta^x = 0.4}$

$$\min_{(\mathbf{x},y,s)\leftarrow\mathcal{D},\ \mathcal{F}\in\mathbb{F},\ \hat{\mathbf{y}}\leftarrow\mathcal{F}(\mathbf{x}),\hat{\mathbf{s}}=\mathcal{G}(\hat{\mathbf{y}})} \left[\mathcal{H} = \boxed{\beta^x \mathrm{H}(\hat{\mathbf{y}})} + \underbrace{\boxed{\beta^y \mathrm{H}(y|\hat{\mathbf{y}})}}_{\mathbf{0.7}} + \underbrace{\boxed{\beta^s \mathrm{H}(s|\hat{\mathbf{y}})}}_{\mathbf{0.3}}\right]$$

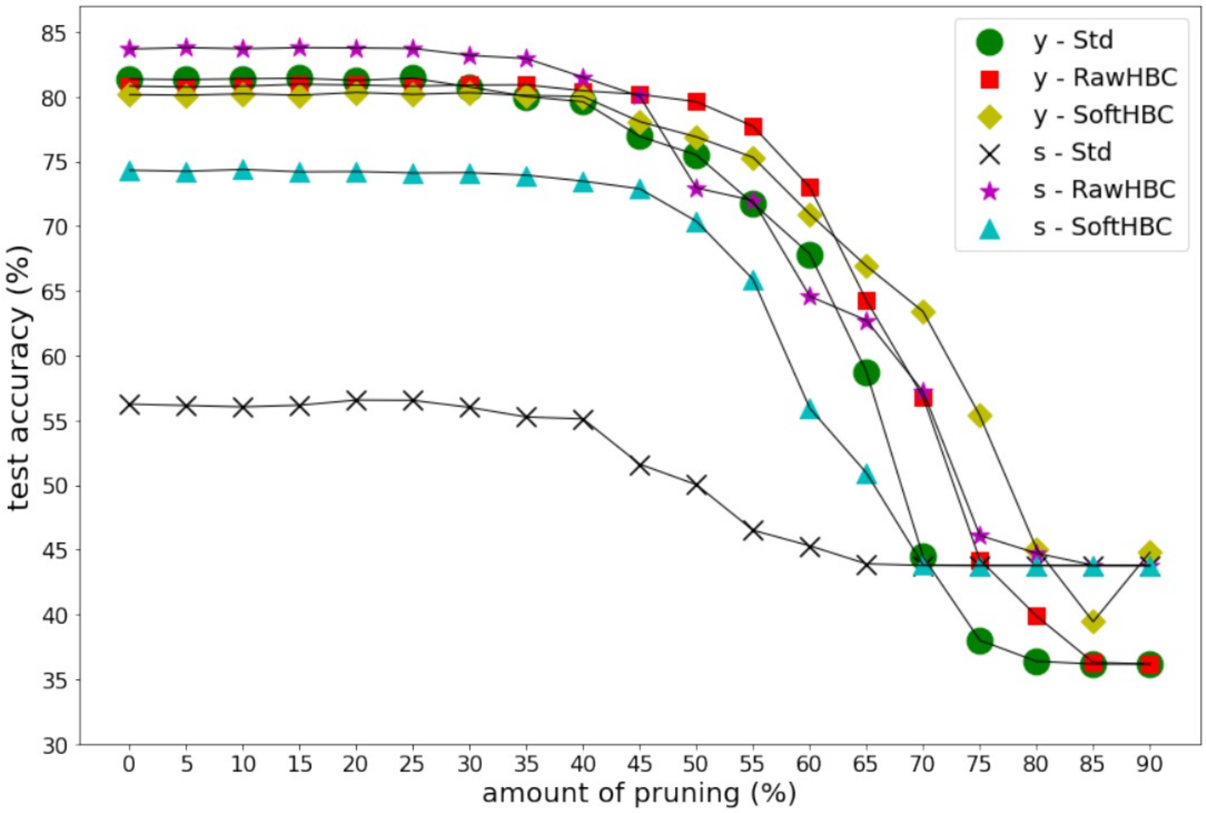
Pruning hurts both Honesty and Curiosity

y: Age 3-classes: ≤ 20 & 21-35 & >35

s: Race 3-classes: White & Asian & Others



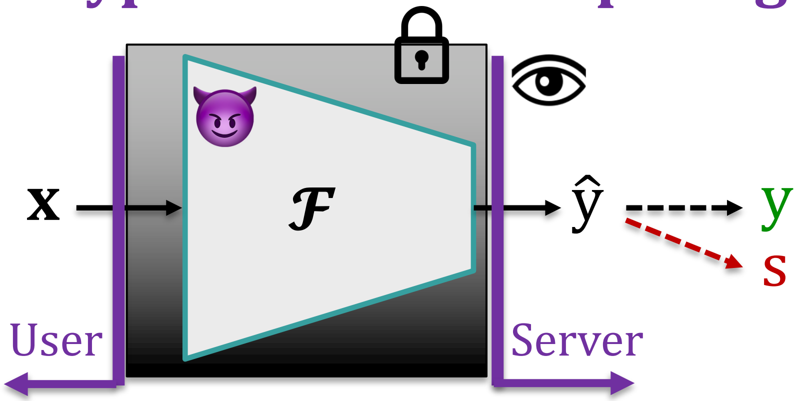
Pruning hurts both Honesty and Curiosity



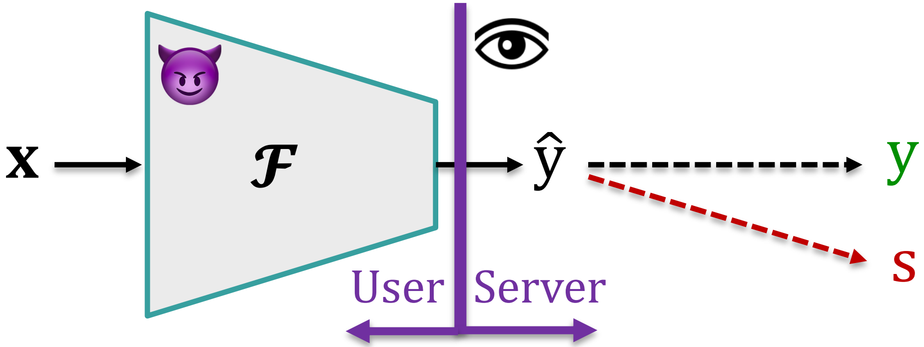
- ***N***% of parameters with the lowest **L1-norm** are set to zero at inference time.
- Thus, most of the parameters capture information related to both y and s.

Conclusion

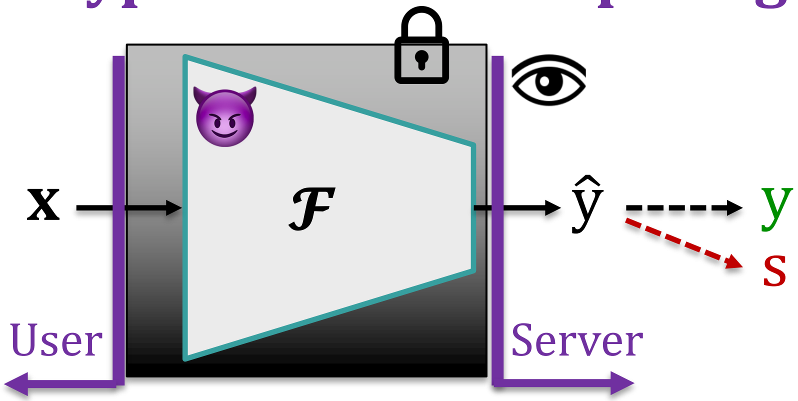
- Encrypted Cloud Computing



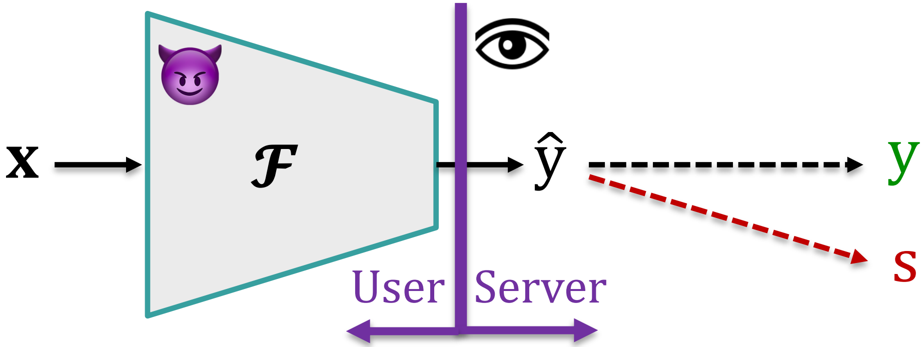
- On-Device/Edge Computing

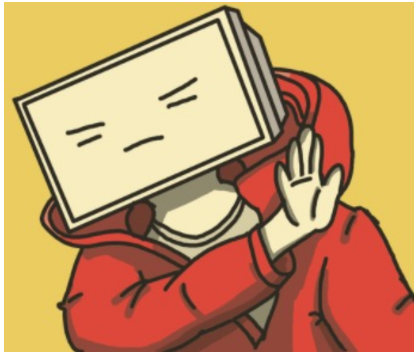


- Encrypted Cloud Computing



- On-Device/Edge Computing





System32Comics

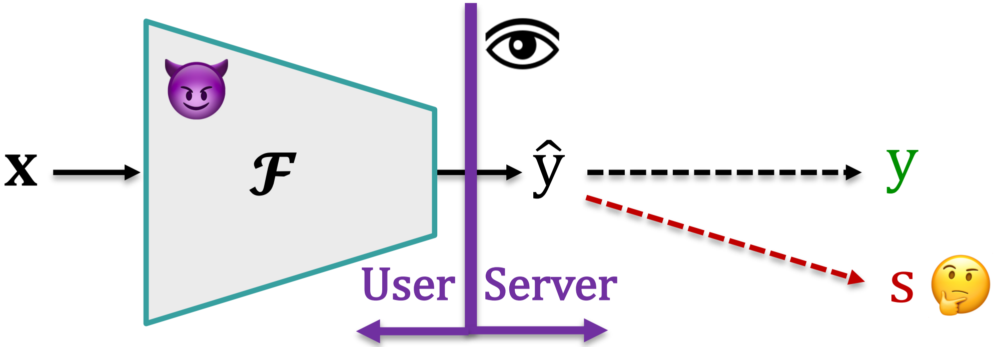
If we do not trust
a service provider, then
even releasing the output
is not completely safe!

Other Observations

- ☑ **Overparameterized** deep neural nets enable **HBC nets**.
- ☑ Releasing **Sigmoid** or **Softmax** is not sufficient
- ☑ Not easy to identify whether a model is **HBC** or not.
  - no general defense mechanism!
  - sensitive attribute must be known & a labelled dataset is required

Open Directions

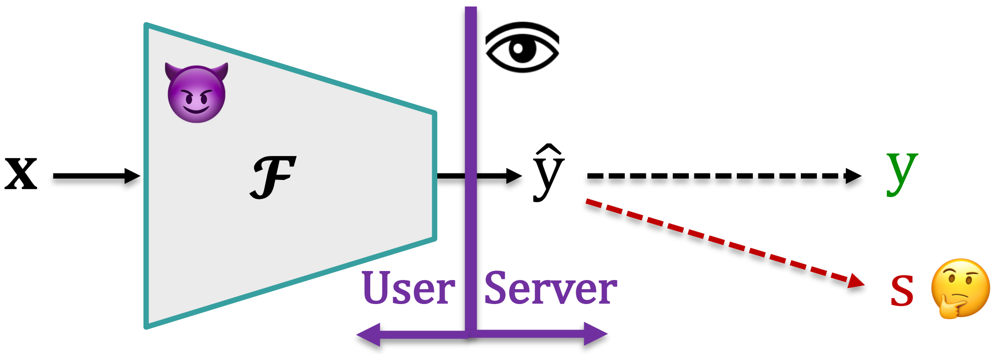
- To improve or extend the **attack**:
  1. Encoding more than **one** sensitive attribute
  2. Applying the attack in **collaborative or federated learning**



Open Directions

- To improve or extend the **attack**:
  1. Inferring more than **one** sensitive attribute
  2. Applying the attack in **collaborative learning** settings

- To propose efficient **defences**:
  1. **Examination** methods finding **“HBC”** classifiers
  2. **Inference-time** methods for removing potential curiosities



Thank You

Honest-but-Curious Nets: Sensitive Attributes of Private Inputs can be Secretly Coded into the Classifiers' Outputs

Mohammad Malekzadeh, Anastasia Borovykh and Deniz Gündüz

Code: https://github.com/mmalekzadeh/honest-but-curious-nets

Happy to hear from you: m.malekzadeh@imperial.ac.uk

Paper's Webpage

Imperial College London

Acknowledgments:
- European Research Council (ERC) Starting Grant BEACON (no. 677854)
- UK EPSRC Grant within the CHIST-ERA program (no. EP/T023600/1)
- J.P. Morgan A.I. Research Award 2019