Adversarial Removal of Demographic Attributes from Text Data

Yanai Elazar and Yoav Goldberg

Bar-Ilan University / NLP Group

November 2, 2018

Motivation

Text is used for predictions

Motivation

- F … e we predict:

Department of Linguistics &
Department of Computer Science,
Stanford University
Stanford CA 94305-2150

Education

B.A Linguistics, with honors, University of California at Berkeley, 1983
Ph.D. Computer Science, University of California at Berkeley, 1992
Postdoc, International Computer Science Institute, Berkeley, 1992-1995

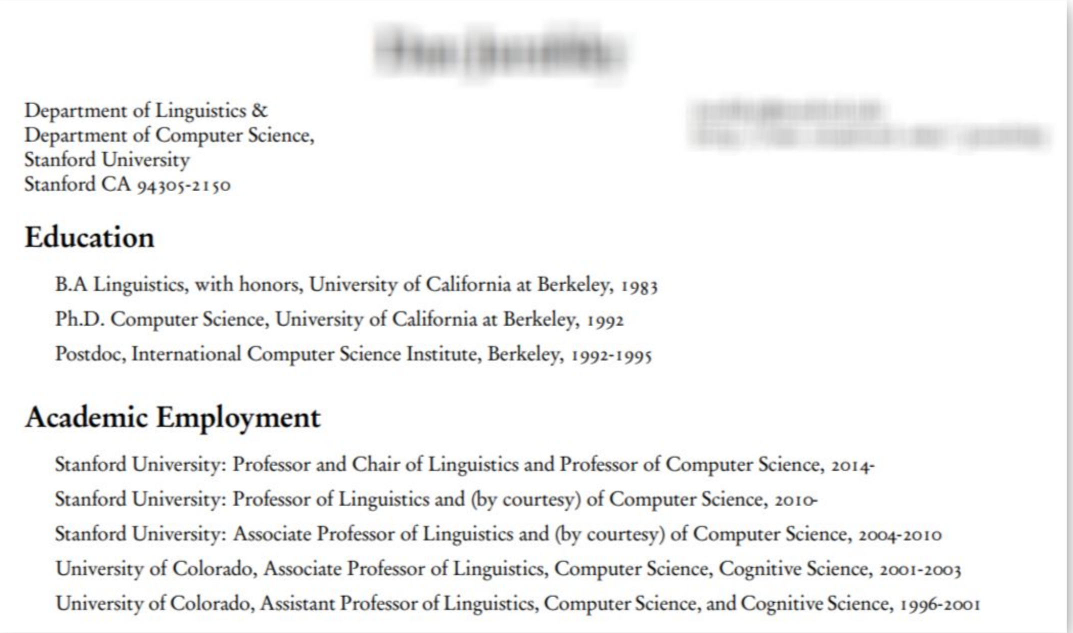
Academic Employment

Stanford University: Professor and Chair of Linguistics and Professor of Computer Science, 2014-
Stanford University: Professor of Linguistics and (by courtesy) of Computer Science, 2010-
Stanford University: Associate Professor of Linguistics and (by courtesy) of Computer Science, 2004-2010
University of Colorado, Associate Professor of Linguistics, Computer Science, Cognitive Science, 2001-2003
University of Colorado, Assistant Professor of Linguistics, Computer Science, and Cognitive Science, 1996-2001

This applicant would easily get any NLP job

Motivation

The common implementation:



Department of Linguistics &
Department of Computer Science,
Stanford University
Stanford CA 94305-2150

Education

B.A Linguistics, with honors, University of California at Berkeley, 1983
Ph.D. Computer Science, University of California at Berkeley, 1992
Postdoc, International Computer Science Institute, Berkeley, 1992-1995

Academic Employment

Stanford University: Professor and Chair of Linguistics and Professor of Computer Science, 2014-
Stanford University: Professor of Linguistics and (by courtesy) of Computer Science, 2010-
Stanford University: Associate Professor of Linguistics and (by courtesy) of Computer Science, 2004-2010
University of Colorado, Associate Professor of Linguistics, Computer Science, Cognitive Science, 2001-2003
University of Colorado, Assistant Professor of Linguistics, Computer Science, and Cognitive Science, 1996-2001

Input CV

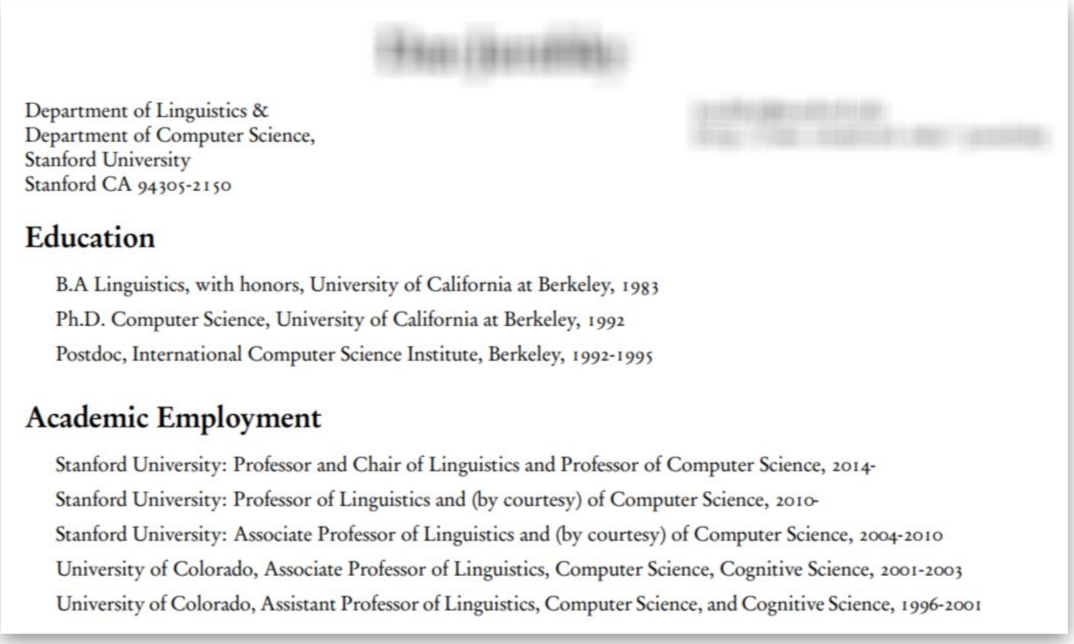
ML
Model

Hire

Don’t Hire

Motivation

The common implementation:



Department of Linguistics &
Department of Computer Science,
Stanford University
Stanford CA 94305-2150

Education

B.A Linguistics, with honors, University of California at Berkeley, 1983
Ph.D. Computer Science, University of California at Berkeley, 1992
Postdoc, International Computer Science Institute, Berkeley, 1992-1995

Academic Employment

Stanford University: Professor and Chair of Linguistics and Professor of Computer Science, 2014-
Stanford University: Professor of Linguistics and (by courtesy) of Computer Science, 2010-
Stanford University: Associate Professor of Linguistics and (by courtesy) of Computer Science, 2004-2010
University of Colorado, Associate Professor of Linguistics, Computer Science, Cognitive Science, 2001-2003
University of Colorado, Assistant Professor of Linguistics, Computer Science, and Cognitive Science, 1996-2001

Input CV

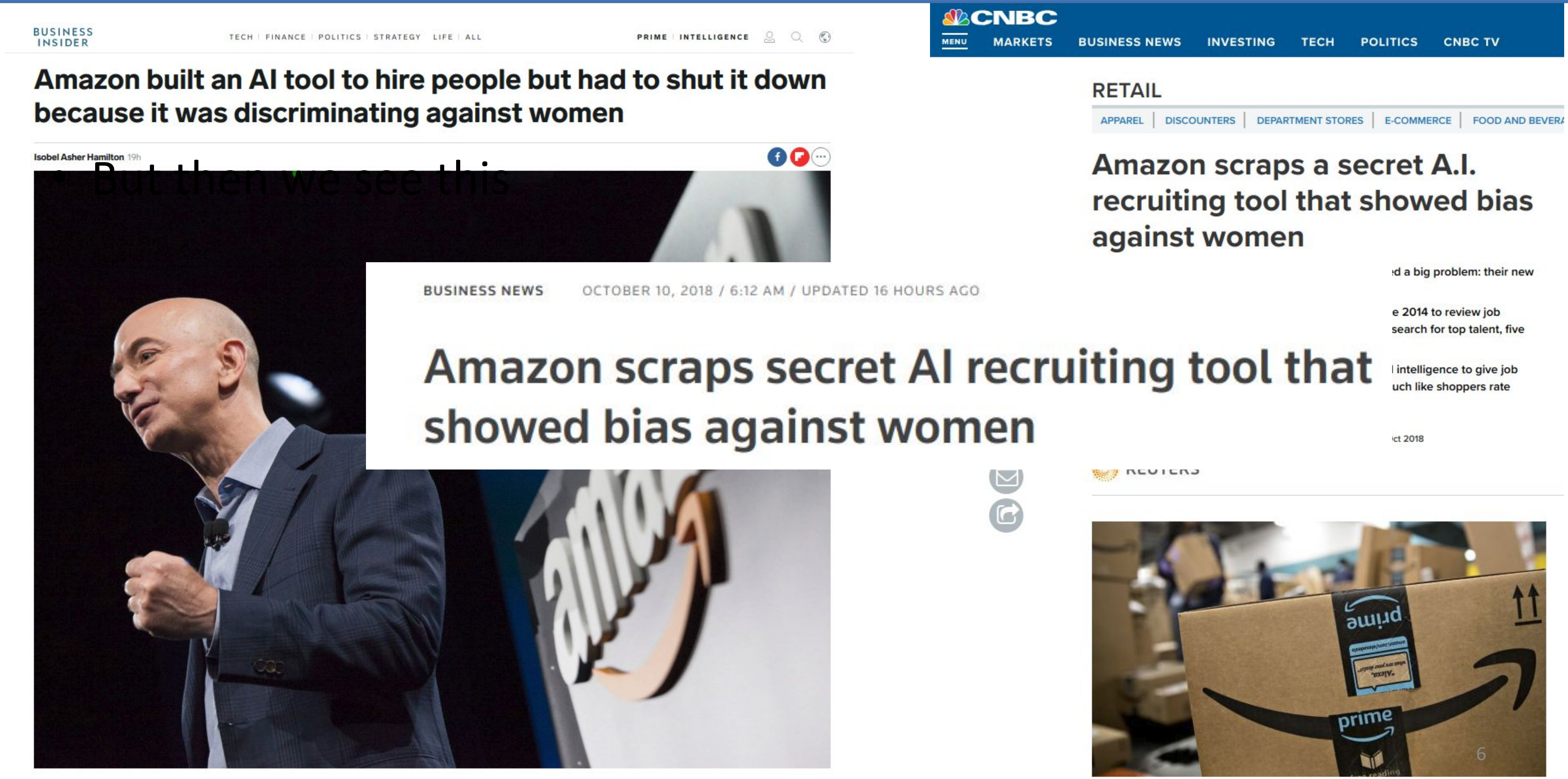
Encode → Representation → Predict

Hire

Don't Hire

Motivation

- But then we see this



Motivation

- When deciding on recruiting an applicant from his/her writings/CV
- We would like that attributes like the author's
  - Gender
  - Race
  - Age
- Won't be part of the decision
- In some places, this is even illegal

Motivation

- We seek to build models which are:
  - Predictive for some main task (e.g. Hiring decision)

  - Agnostic to irrelevant attributes (e.g. race, gender, …)

Text classification - Example

We do not have access to sensitive tasks like Resumes.

We will focus on other tasks, less sensitive

Text classification - Example

Let's predict... EMOJIS

We use DeepMoji.

DeepMoji is a model for predicting Emojis from tweets

Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm

Bjarke Felbo[1], Alan Mislove[2], Anders Søgaard[3], Iyad Rahwan[1], Sune Lehmann[4]

[1]Media Lab, Massachusetts Institute of Technology
[2]College of Computer and Information Science, Northeastern University
[3]Department of Computer Science, University of Copenhagen
[4]DTU Compute, Technical University of Denmark

Text classification - Example

Let's predict... EMOJIS

Deep Moji (Felbo et al., 2017)

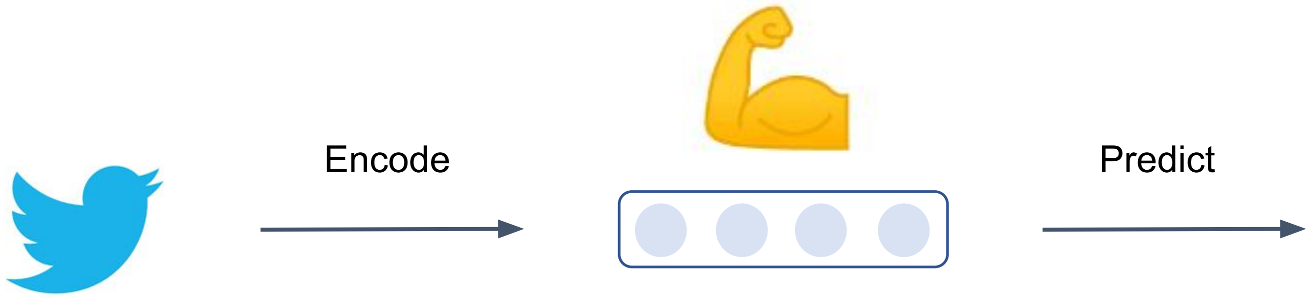
I love mom's cooking	😋 49.1%	😍 8.8%	❤️ 3.1%	☺️ 3.0%	♡ 2.9%
I love how you never reply back..	😒 14.0%	😑 8.3%	😠 6.3%	😐 5.4%	💔 5.1%
I love cruising with my homies	😎 34.0%	👌 6.6%	✌️ 5.7%	😌 4.1%	💯 3.8%
I love messing with yo mind!!	😜 17.2%	😈 11.8%	😏 8.0%	😉 6.4%	🙊 5.3%
I love you and now you're just gone..	💔 39.1%	😔 11.0%	😞 7.3%	😪 5.3%	😢 4.5%
This is shit	😠 7.0%	😡 6.4%	😞 6.0%	😒 6.0%	😤 5.8%
This is the shit	🎧 10.9%	🎶 9.7%	👌 6.5%	😎 5.7%	😏 4.8%

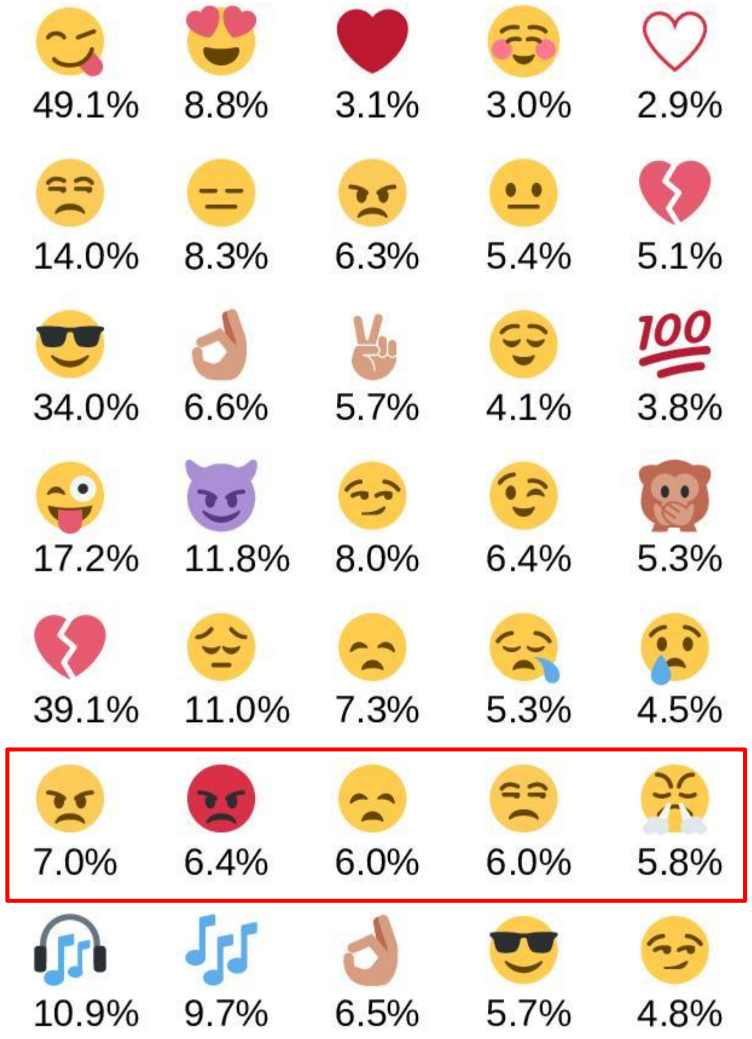
Text classification - Example

Let's predict... EMOJIS

Deep Moji (Felbo et al., 2017)

- DeepMoji is a strong and expressive model
- It also create powerful **representations**

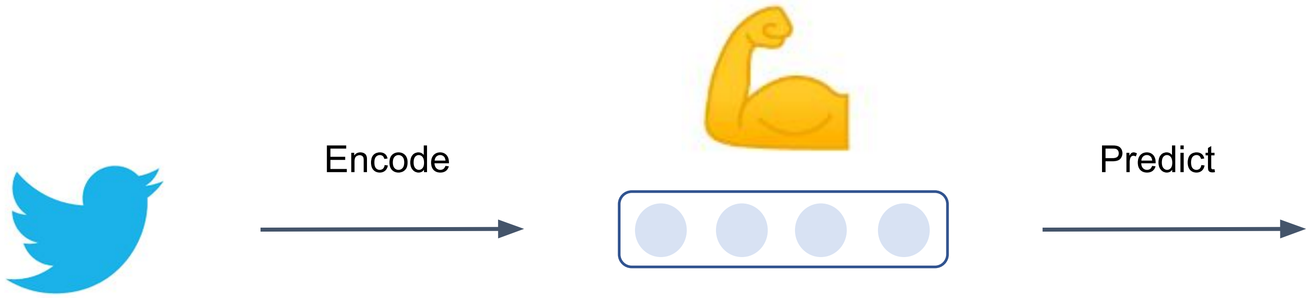




Text classification - Example

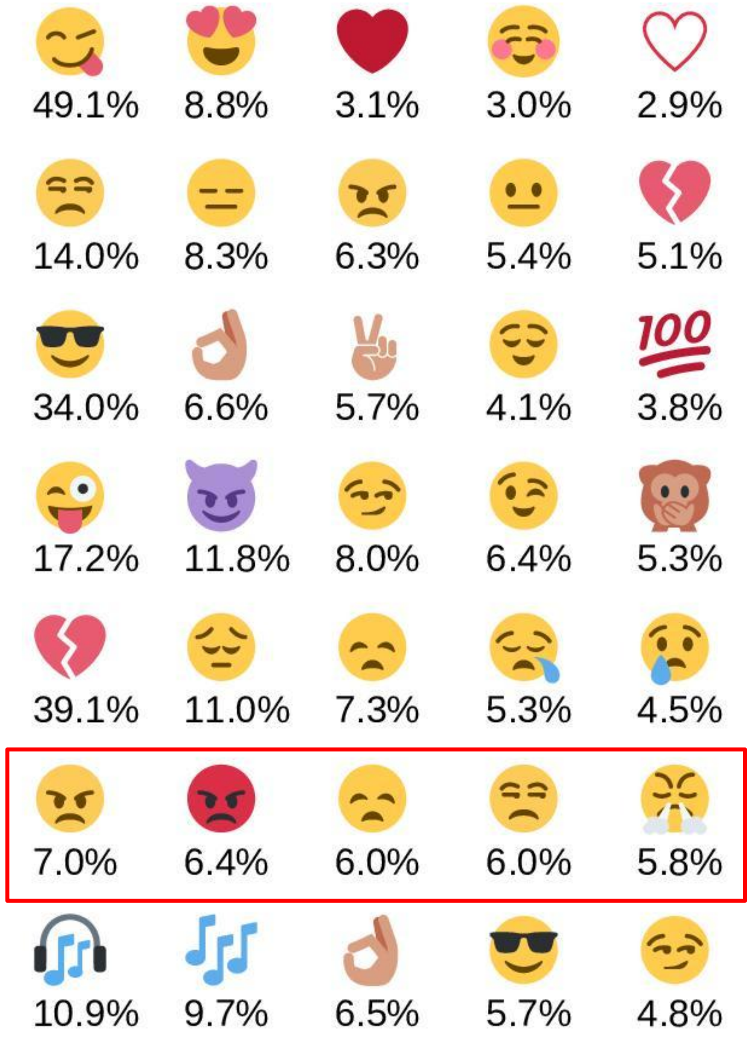
Let's predict... EMOJIS

- DeepMoji is a strong and expressive model
- It also create powerful **representations**



- Achieved several SOTA results on text classification

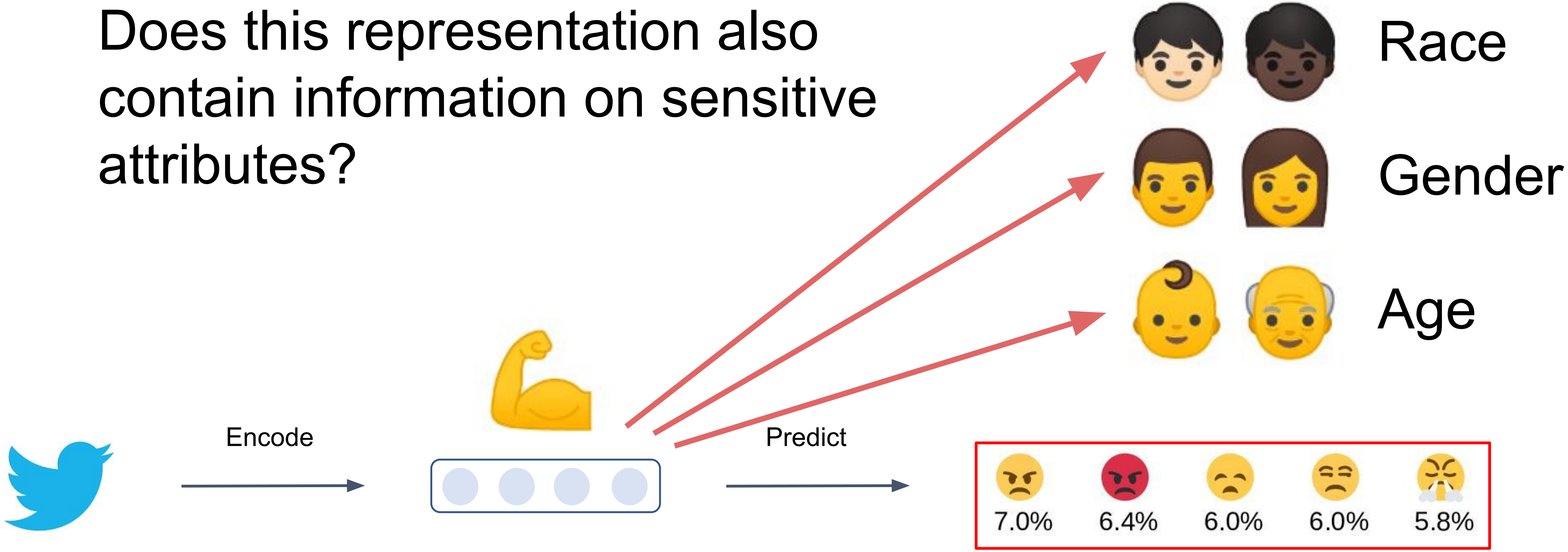
Deep Moji (Felbo et al., 2017)



Text classification - Example

Let's predict... EMOJIS

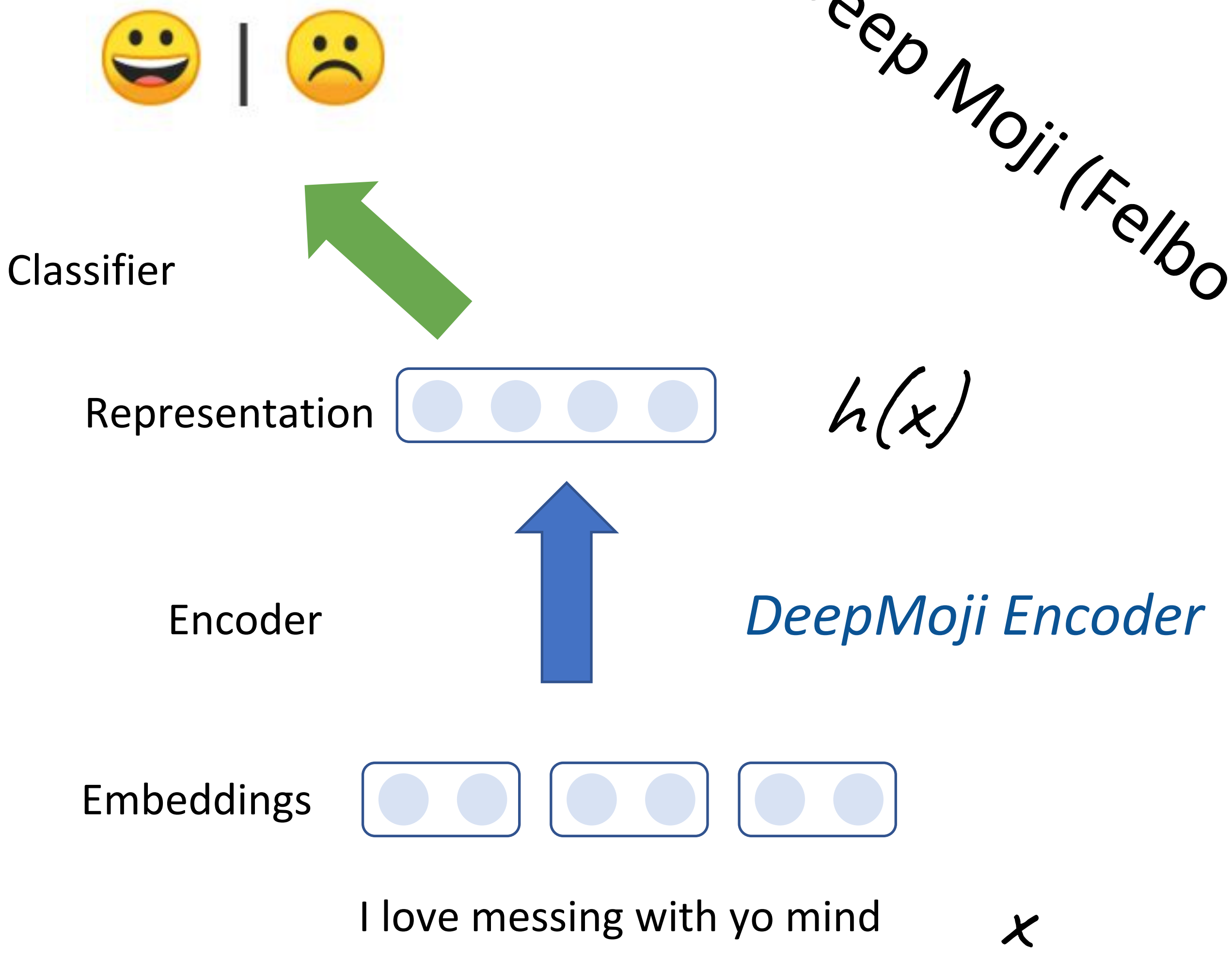
Does this representation also contain information on sensitive attributes?



Setup

Task
(Emojis)

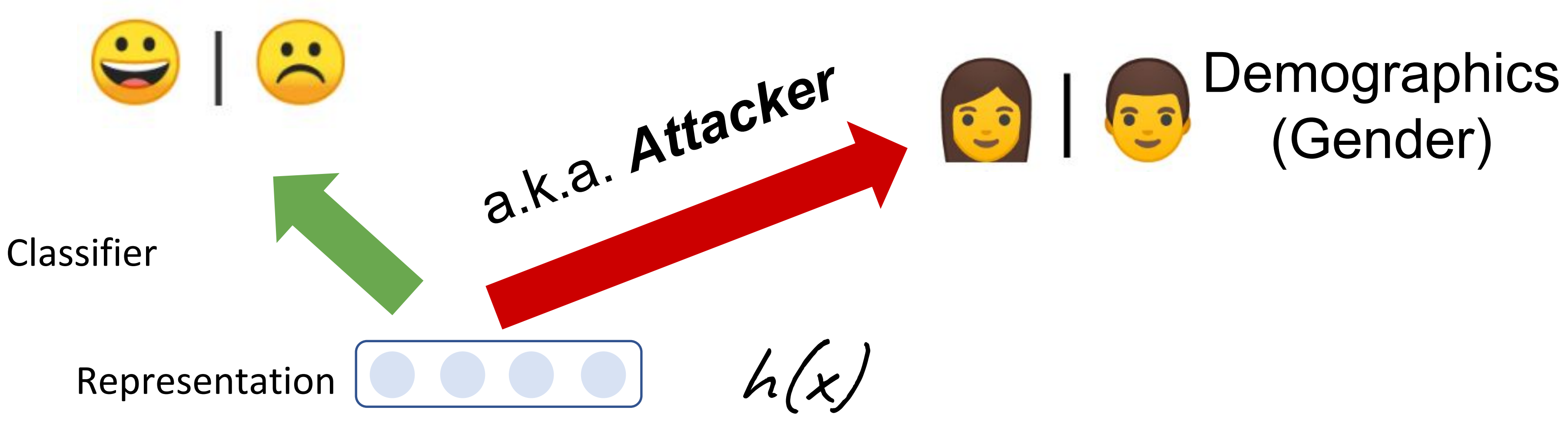
We use the representation that predict Emojis



Deep Moji (Felbo et al., 2017)

Setup

Task
(Emojis)



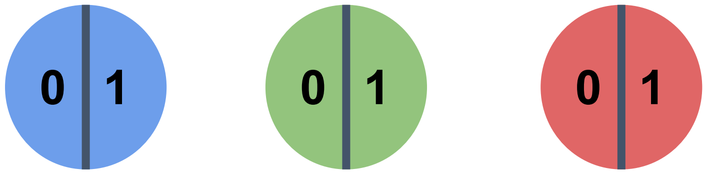
We use the representation that predict Emojis

And use them to predict demographics.

We define:
<u>leakage</u> = score above a random guess an "**Attacker**" achieves

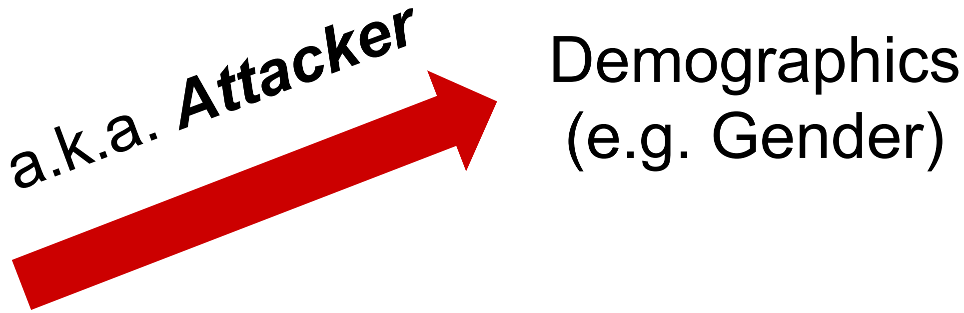
Text Leakage – Case Study

- We use DeepMoji encoder, to encode tweets, from 3 datasets, all binary and balanced



- Each dataset is tied to a different demographic label

- We then train Attackers to predict these attributes

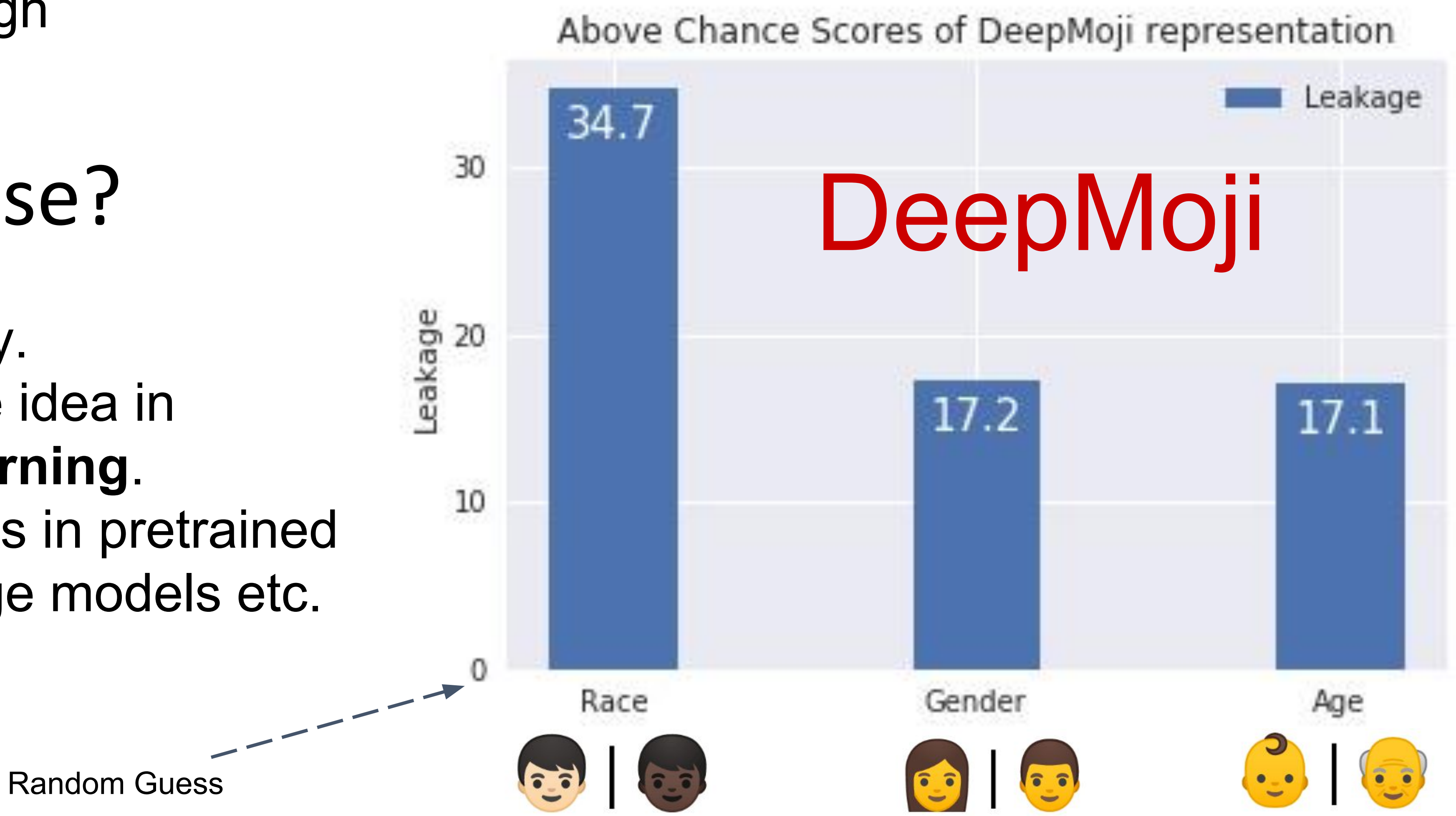


Text Leakage – Case Study

The dev-set scores above chance level are quite high

Big Surprise?

Not really.
This is the core idea in **Transfer-Learning**.
We've seen its benefits in pretrained embeddings, language models etc.



Text Leakage – Case Study

- Why do we get this major “help” in predicting other attributes than those we trained on?
- One option is the correlation between attributes in the data

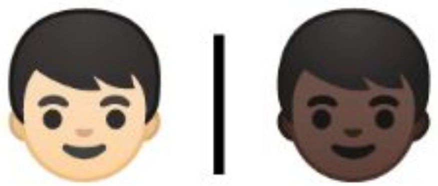
Fair enough. Let’s control it

Controlled Setup

New setup

- We use Twitter data
- We focus on sentiment prediction, emoji based

- With *Race*, *Gender* and *Age* as protected attributes



Blodgett et al., 2016

Rangel et al., 2016

Rangel et al., 2016

New setup

Balanced Dataset

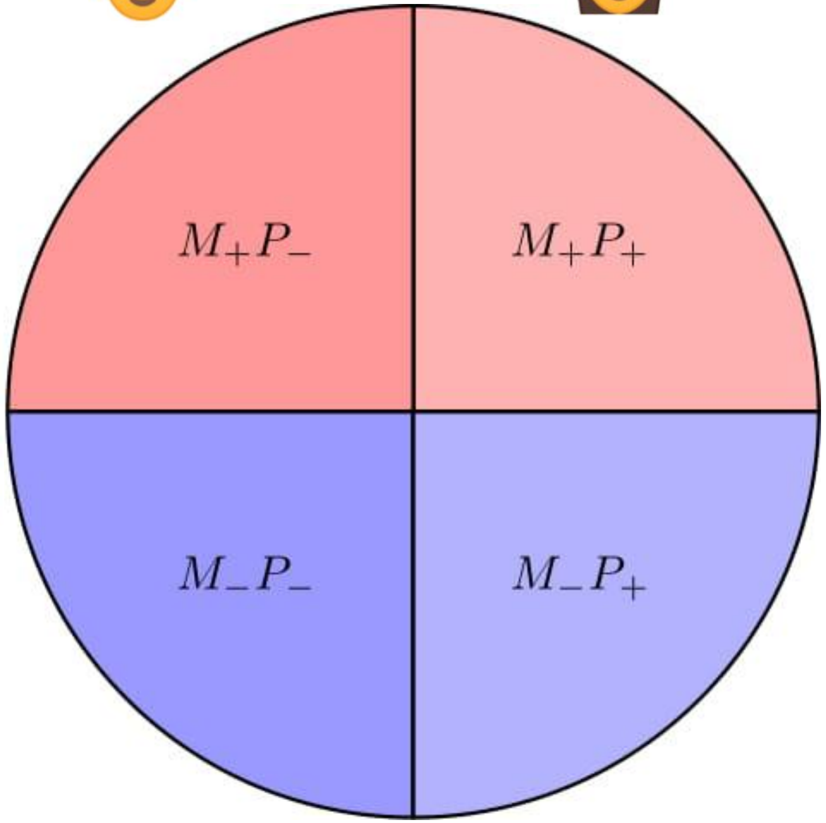
Demographics

50% Male *50% Female*

Task (Sentiment)

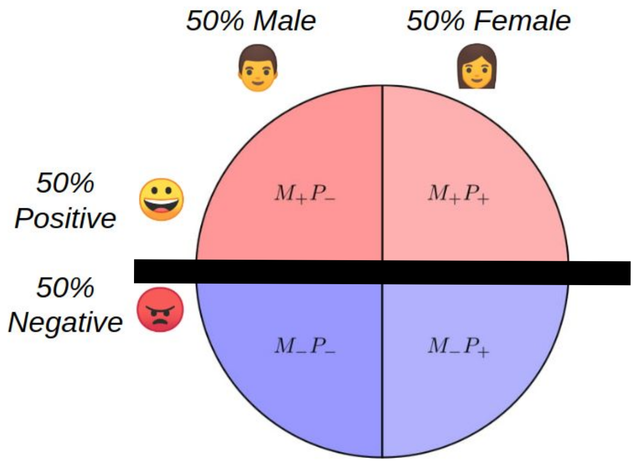
50% Positive

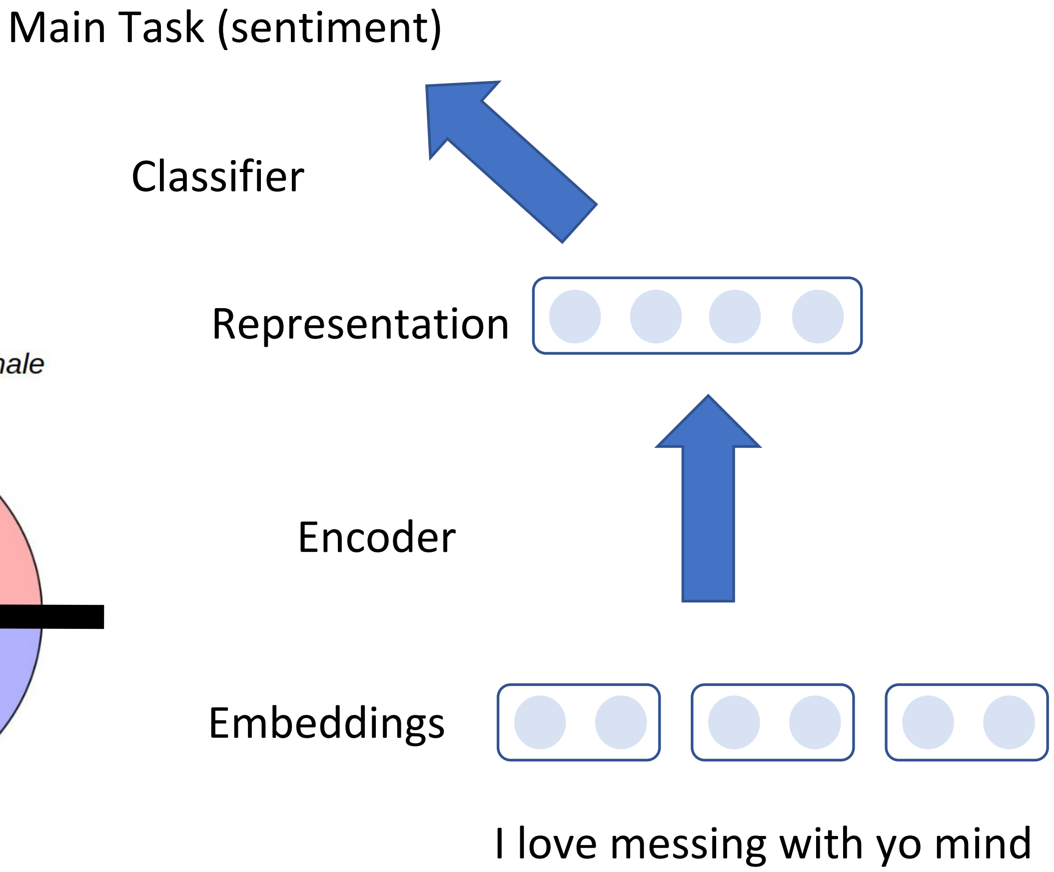
50% Negative



Balanced Training

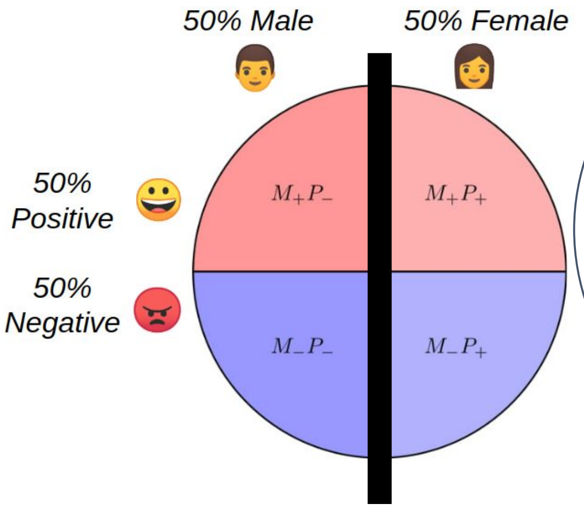
Training our own encoder on the balanced datasets

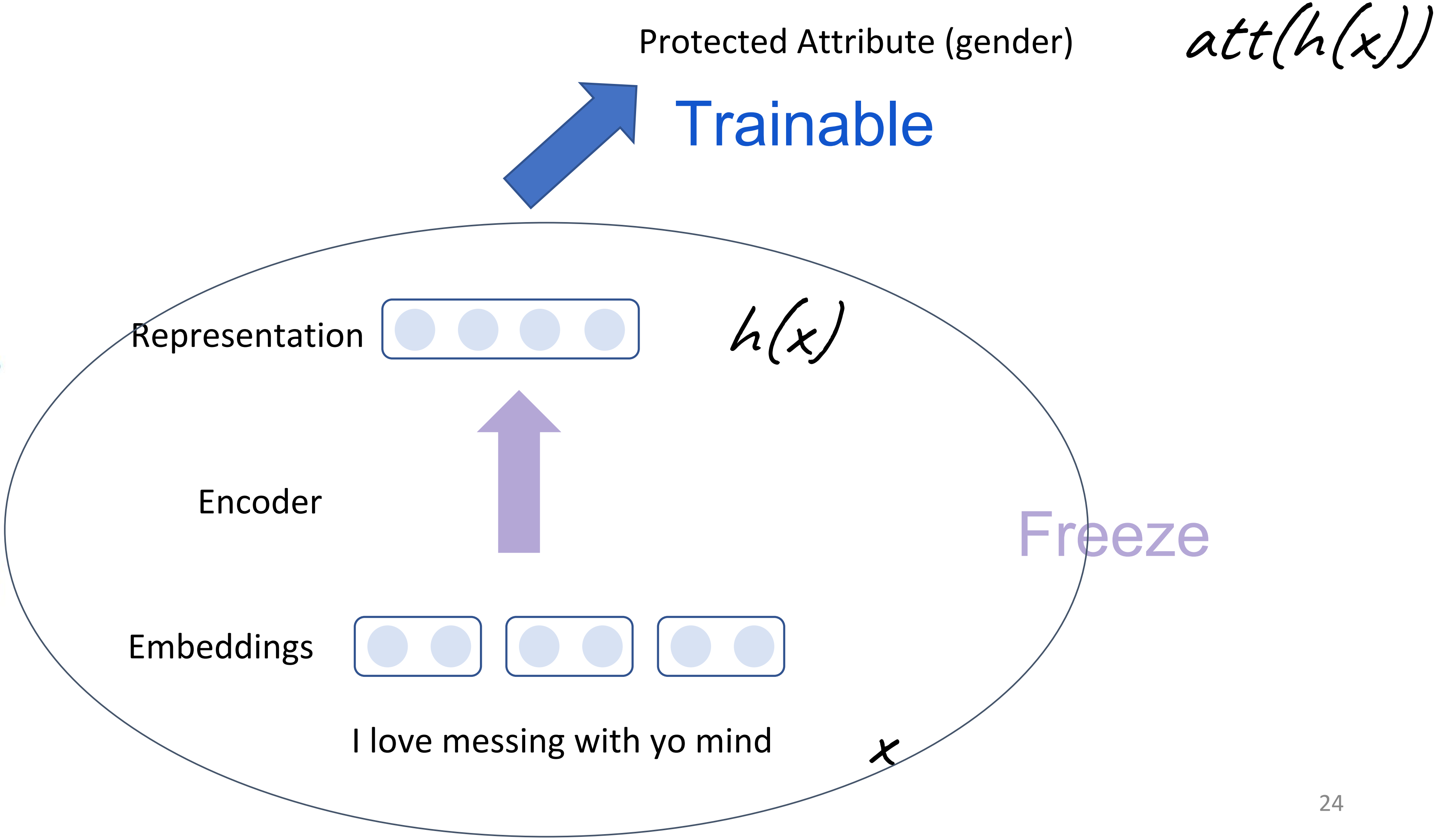




Balanced Training

And using the Attacker to check for leakage

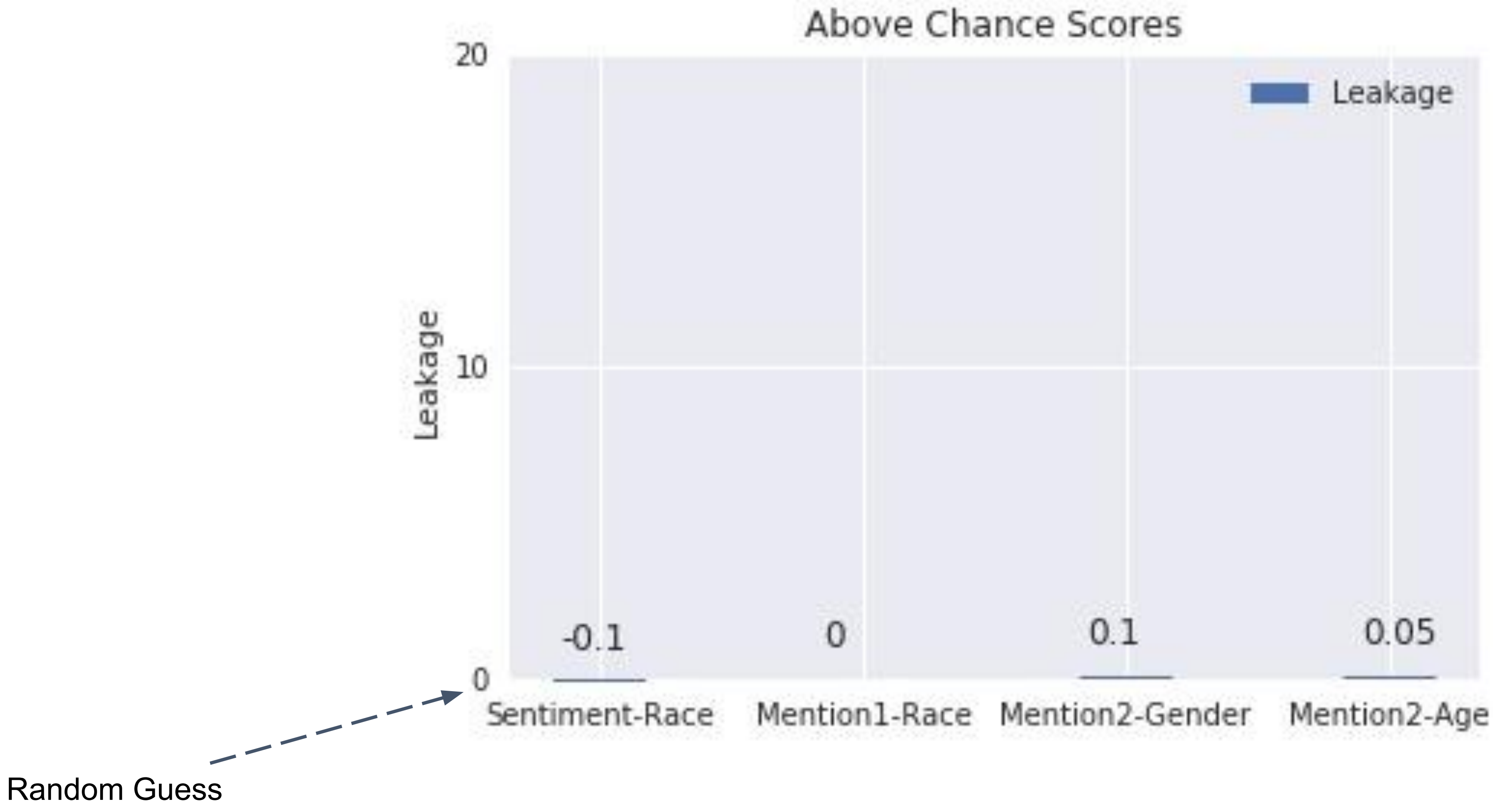




Balanced Training - Leakage

We wanted to see something like this:

But instead...



Balanced Training - Leakage

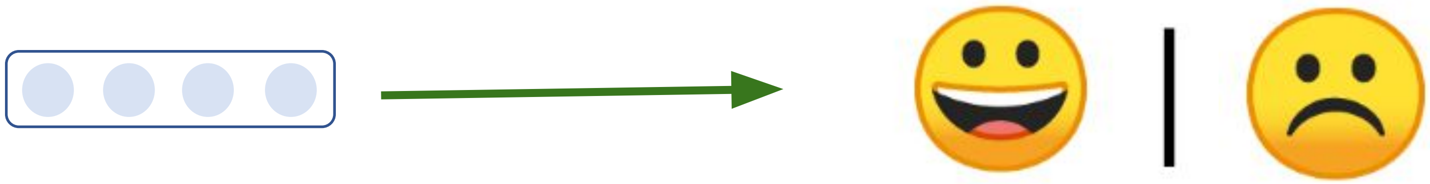
The Attacker manages to extract a substantial amount of sensitive information

Even in a balanced setup, leakage exists

Random Guess

Our objective

- Create a representation which:
  - Is predictive of the main task (e.g. sentiment)



Our objective

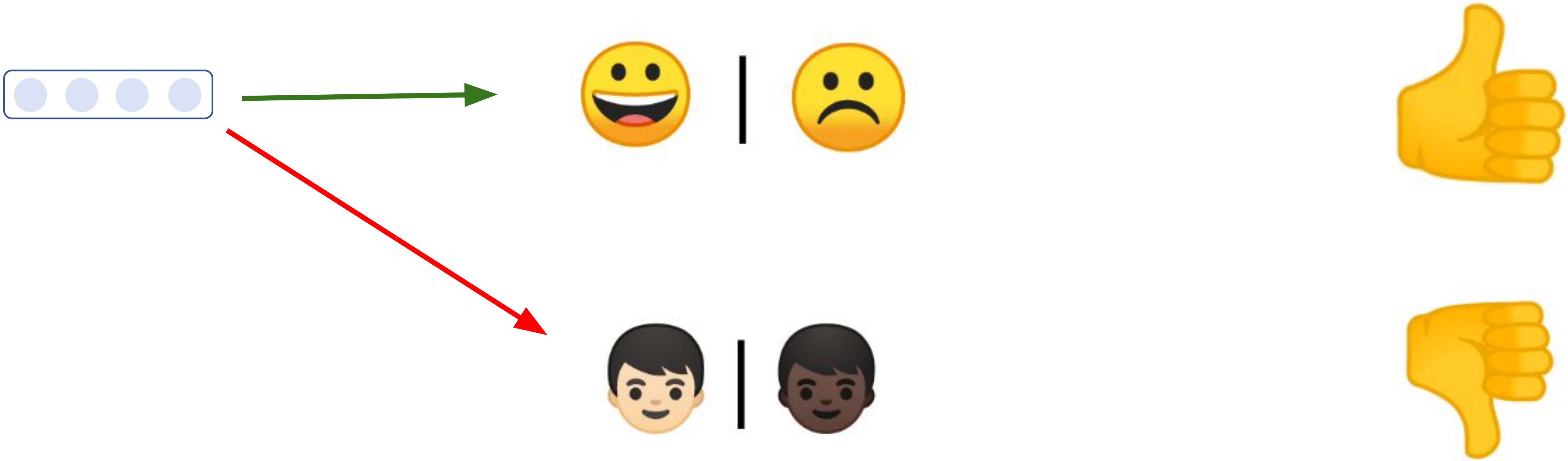
- Create a representation which:
  - Is predictive of the main task (e.g. sentiment)

and

- Is **not** predictive of protected attribute (e.g. gender, race)

Our objective

- Interesting technical problem – How to **unlearn** something?
- Interesting technical problem – Can we **unlearn** something?



Actively Reducing Leakage

Adversarial Setup

- First introduced by Goodfellow et al., 2014
  - A very active line of research
  - We will go through the details

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie[*], Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair[†], Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

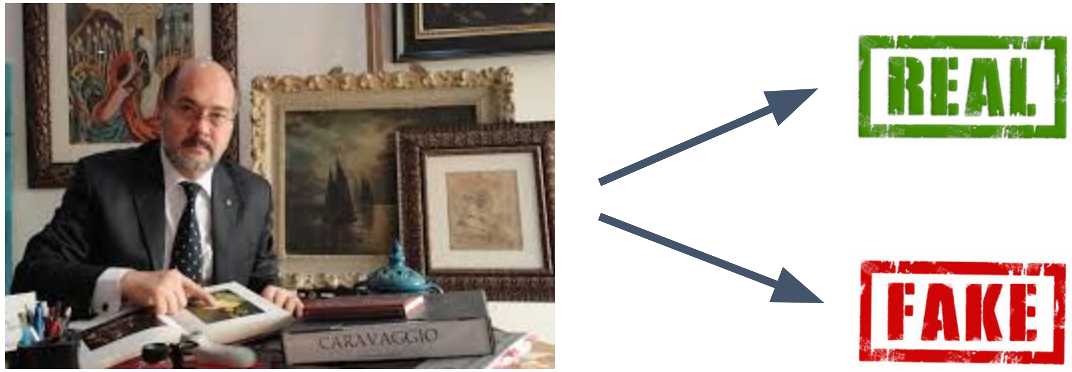
Adversarial Setup

- The motivation came from “Generative Models”
  - We would like to automatically create images
  - From… random input?

Adversarial Setup

- 2 components:
  - Generator

  - Discriminator



Adversarial Setup

$$\min_G \boxed{\max_D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\boxed{[\log D(\boldsymbol{x})]} + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))].$$

A good Discriminator

(real data gets a high score, meaning it’s real)

Adversarial Setup

$$\boxed{\min_G} \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}.$$

A good Generator

(fake data gets a high score,
for maximizing *D*’s probability)

Adversarial Setup

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\boxed{[\log D(\boldsymbol{x})]} + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}.$$

- 2 competing objectives.
- We don't know how to solve this

Adversarial Setup

$$\min_G \max_D V(D, G) = \boxed{\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \boxed{\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}.$$

Goodfellow et al. solution:
iterate training between the Generator and Discriminator

- Update the discriminator by ascending its stochastic gradient:

- Update the generator by descending its stochastic gradient:

Adversarial Setup

- The Adversarial setup was invented to create an “output”
- Which can’t (or seem hard) to separate real from fake
- What if we want to create an intermediate representation?

Adversarial Setup

- The Adversarial setup was invented to create an "output"
- Which can't (or seem hard) to separate real from fake
- What if we want to create an intermediate representation…
  - Which is indistinguishable for some feature or attribute?

Adversarial Setup

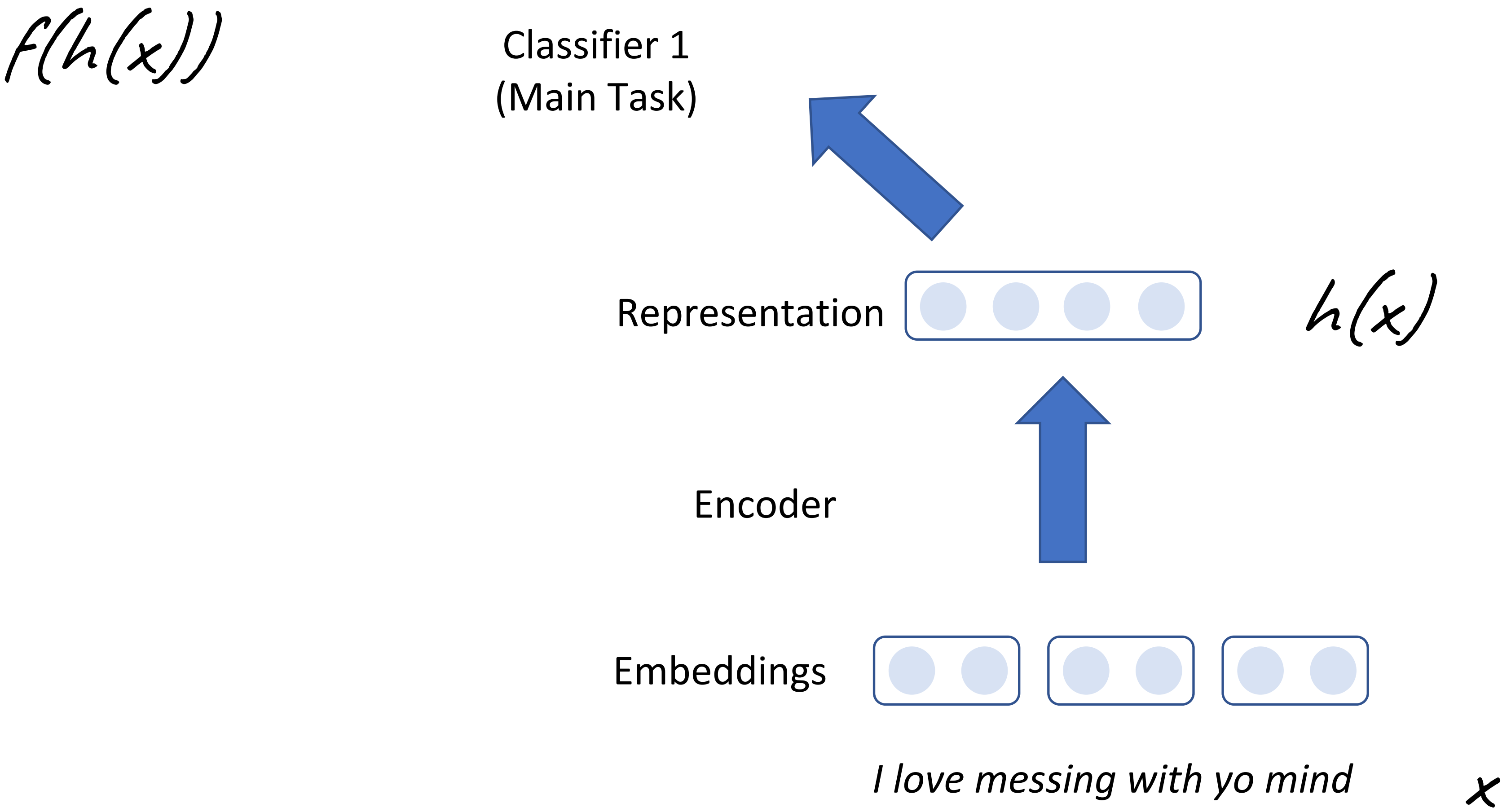
- Ganin and Lempitsky, 2015
- Application: Domain Adaptation
- New trick for adversary train: Gradient Reversal Layer (GRL)

Unsupervised Domain Adaptation by Backpropagation

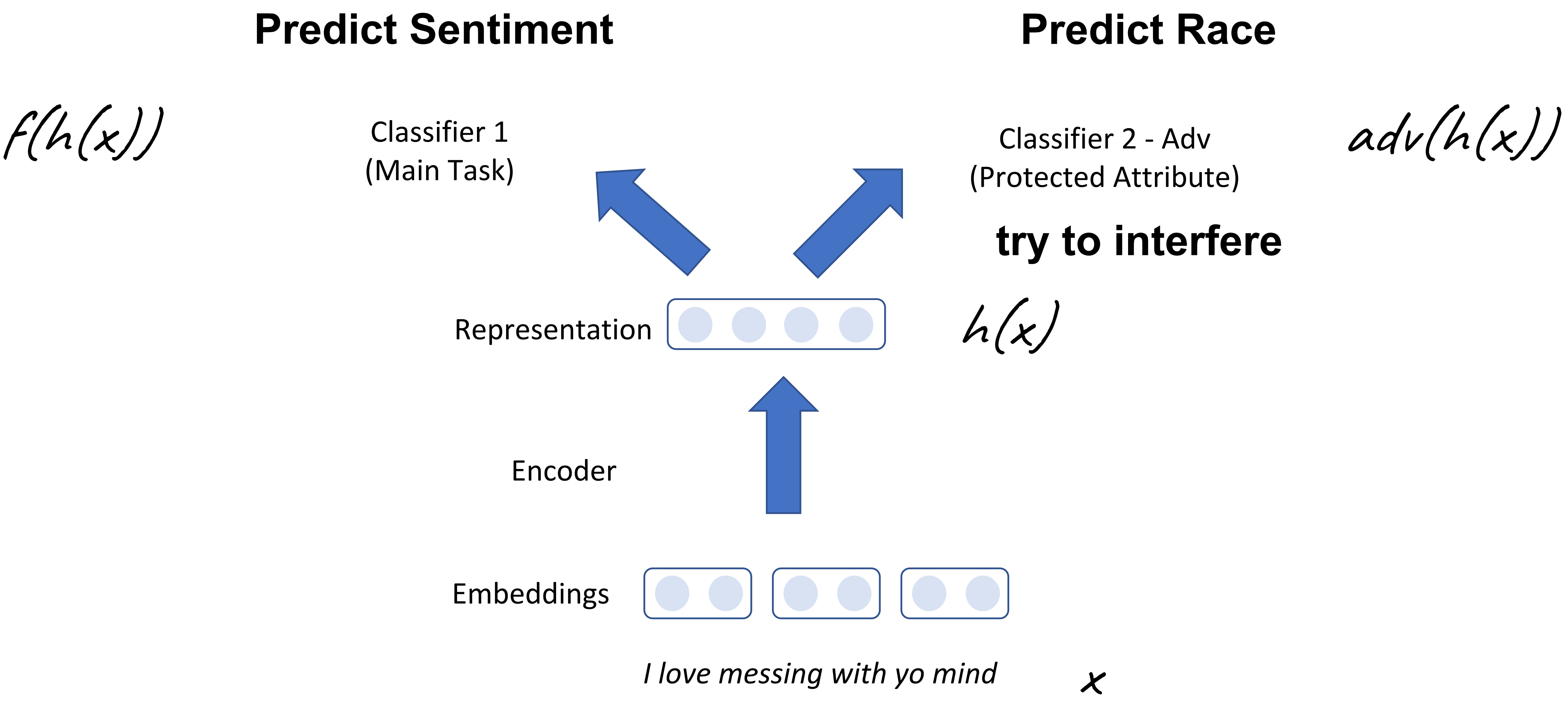
Yaroslav Ganin GANIN@SKOLTECH.RU
Victor Lempitsky LEMPITSKY@SKOLTECH.RU
Skolkovo Institute of Science and Technology (Skoltech)

Adversarial Setup (Ganin and Lempitsky, 2015)

Predict Sentiment

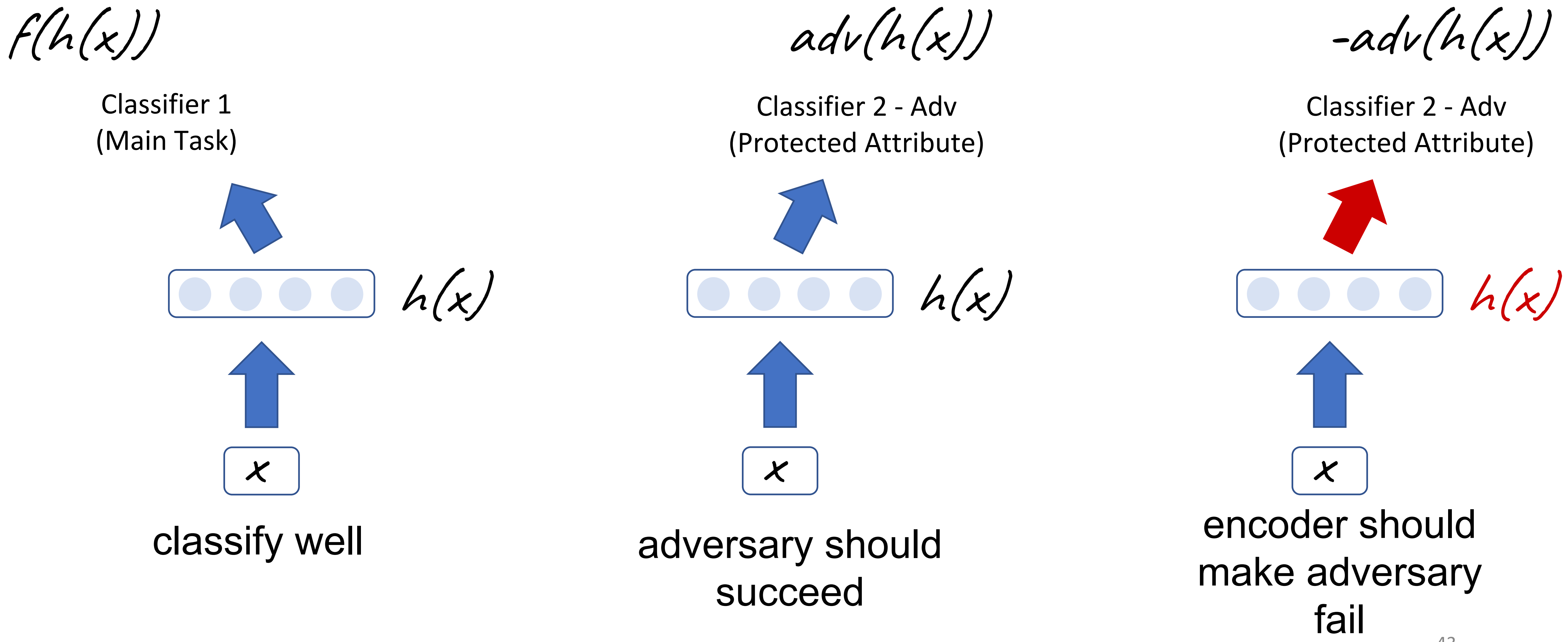


Adversarial Setup (Ganin and Lempitsky, 2015)

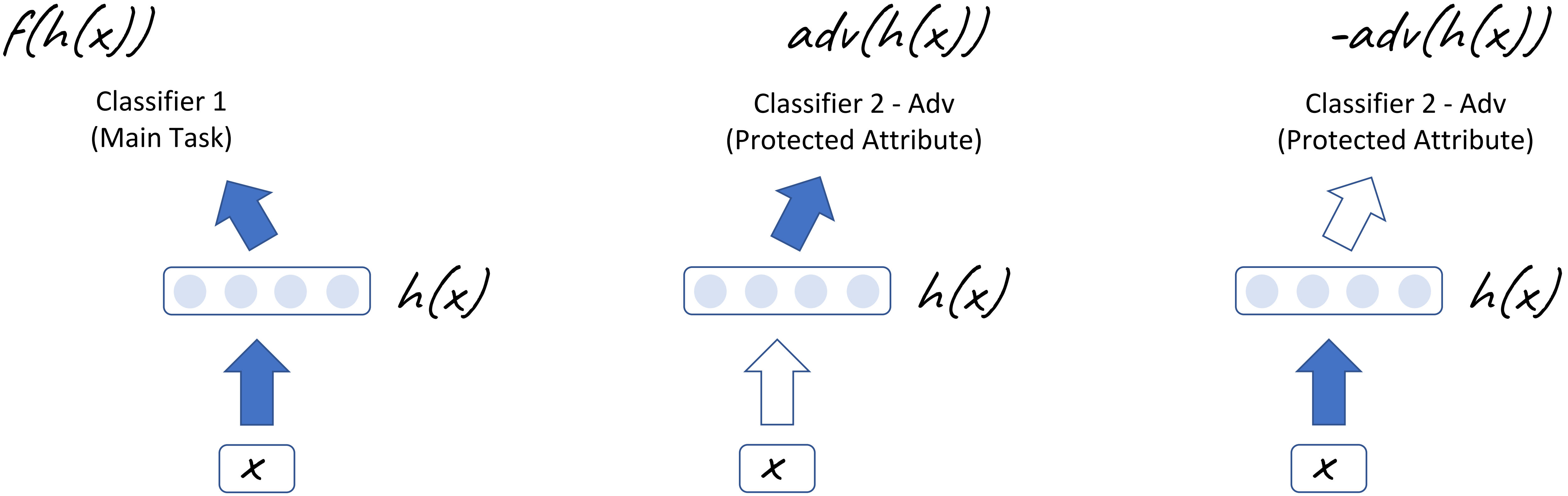


Adversarial Setup (Ganin and Lempitsky, 2015)

3 different sub-objectives

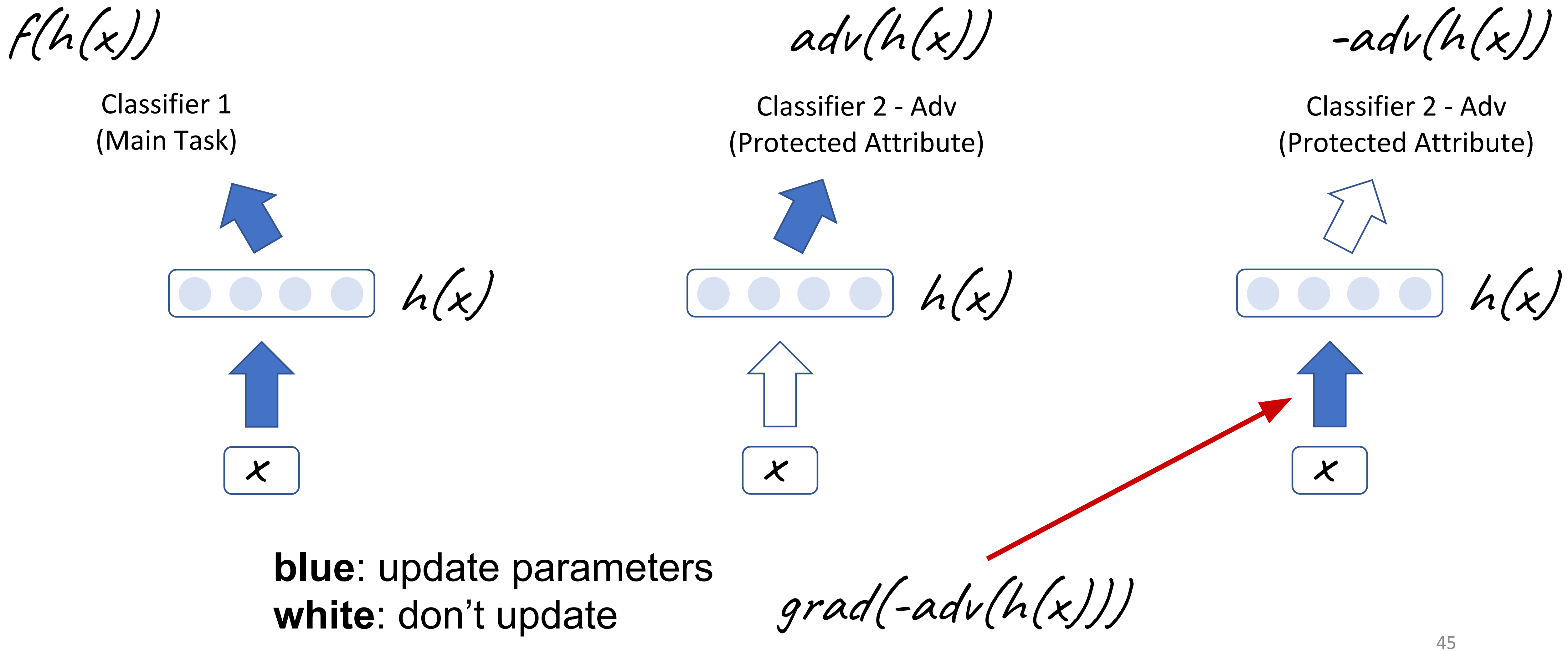


Adversarial Setup (Ganin and Lempitsky, 2015)



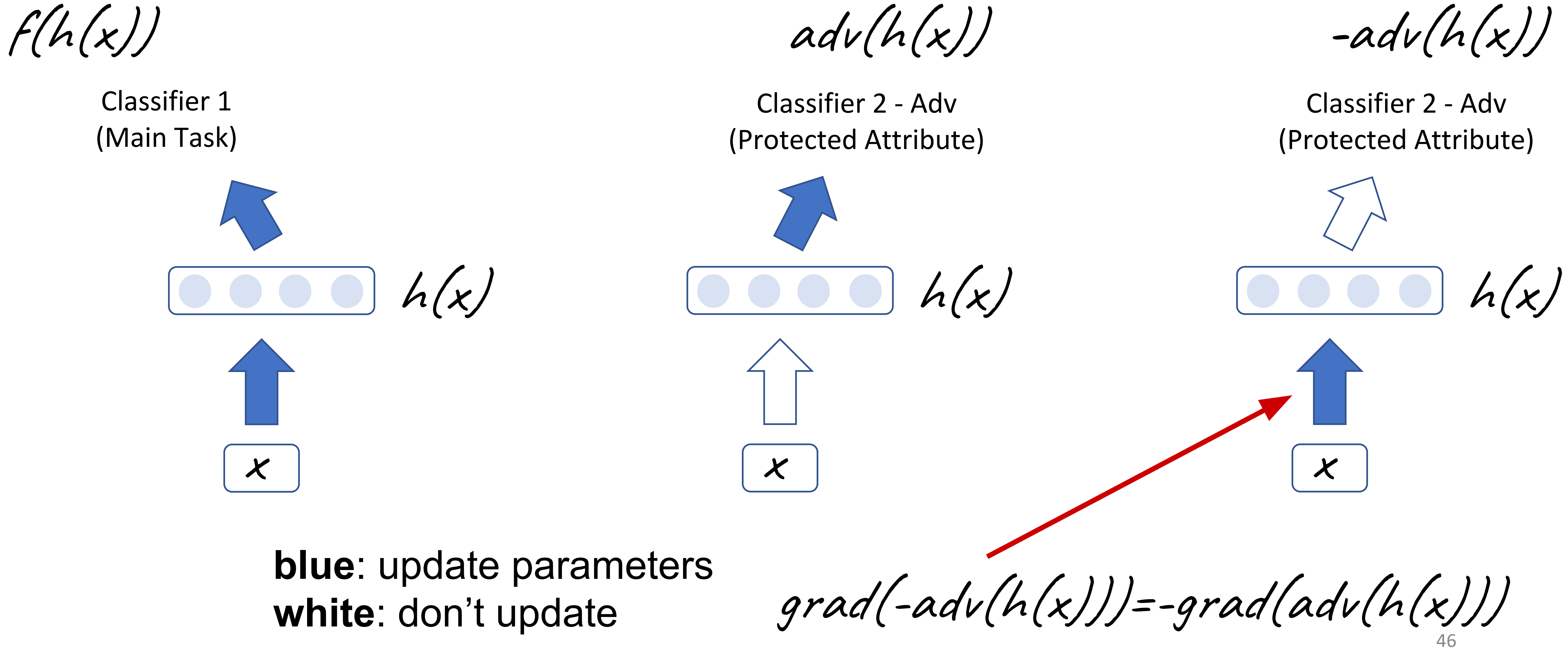
blue: update parameters
white: don't update

Adversarial Setup (Ganin and Lempitsky, 2015)



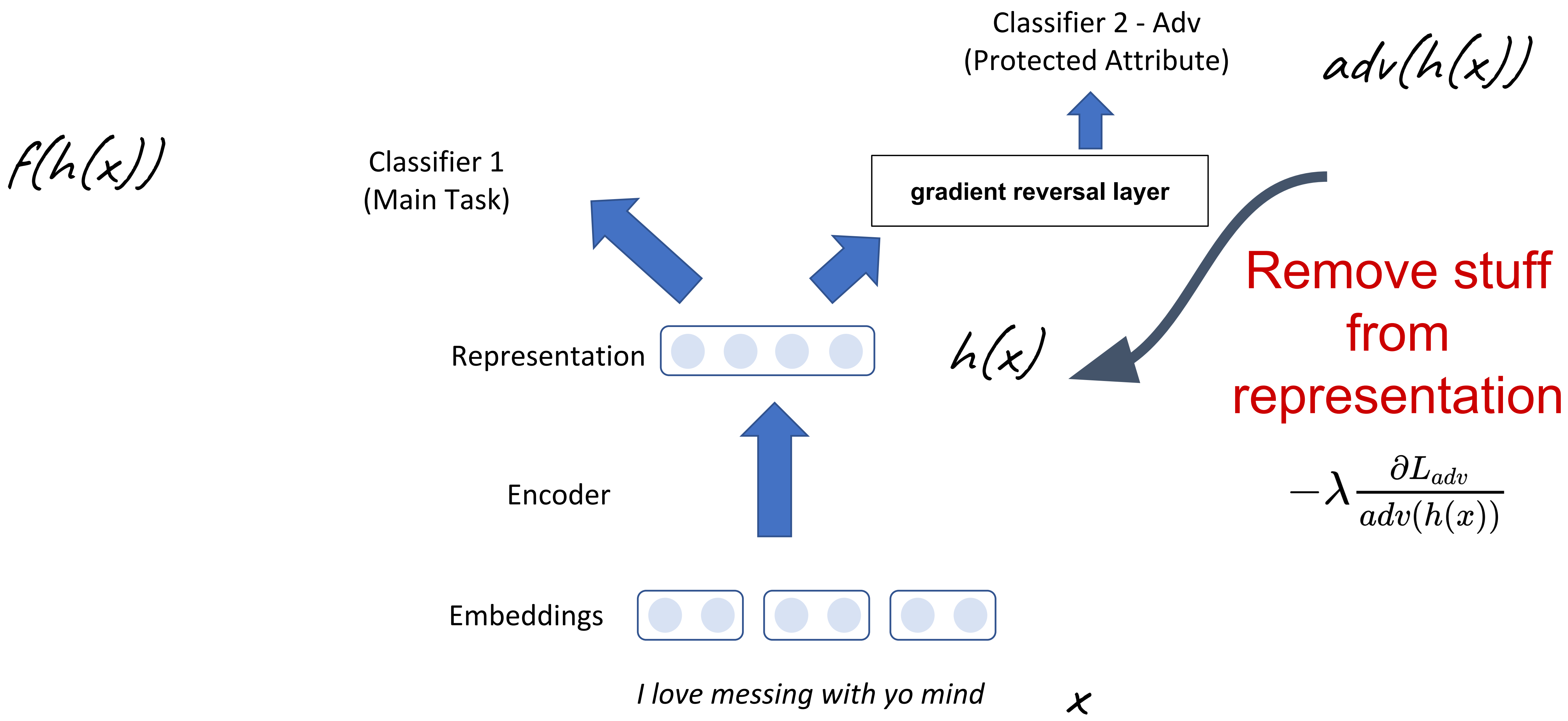
blue: update parameters
white: don't update

Adversarial Setup (Ganin and Lempitsky, 2015)



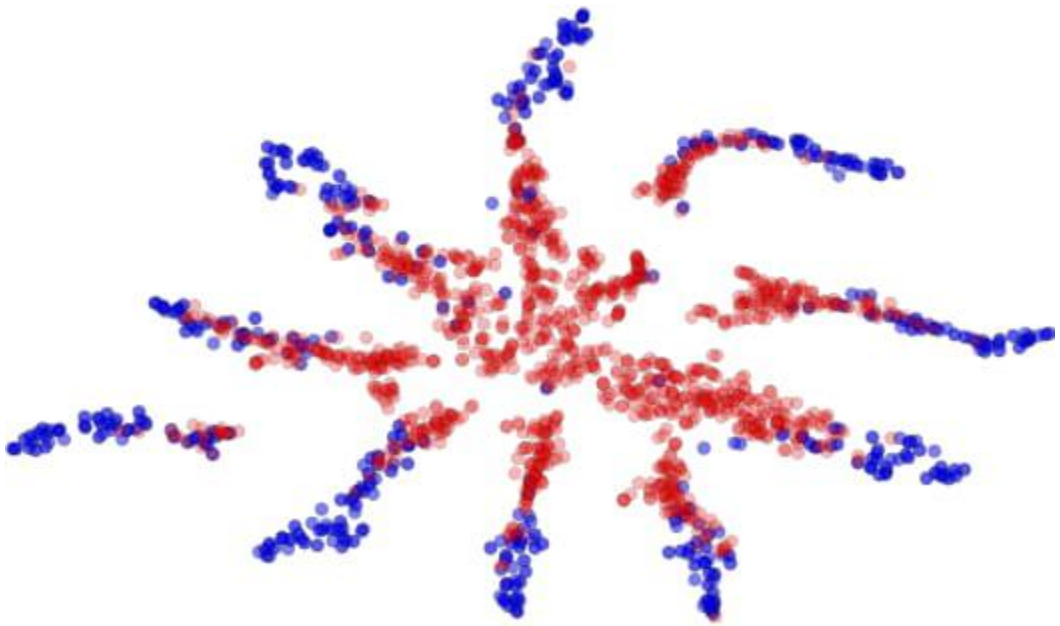
blue: update parameters
white: don't update

Adversarial Setup (Ganin and Lempitsky, 2015)

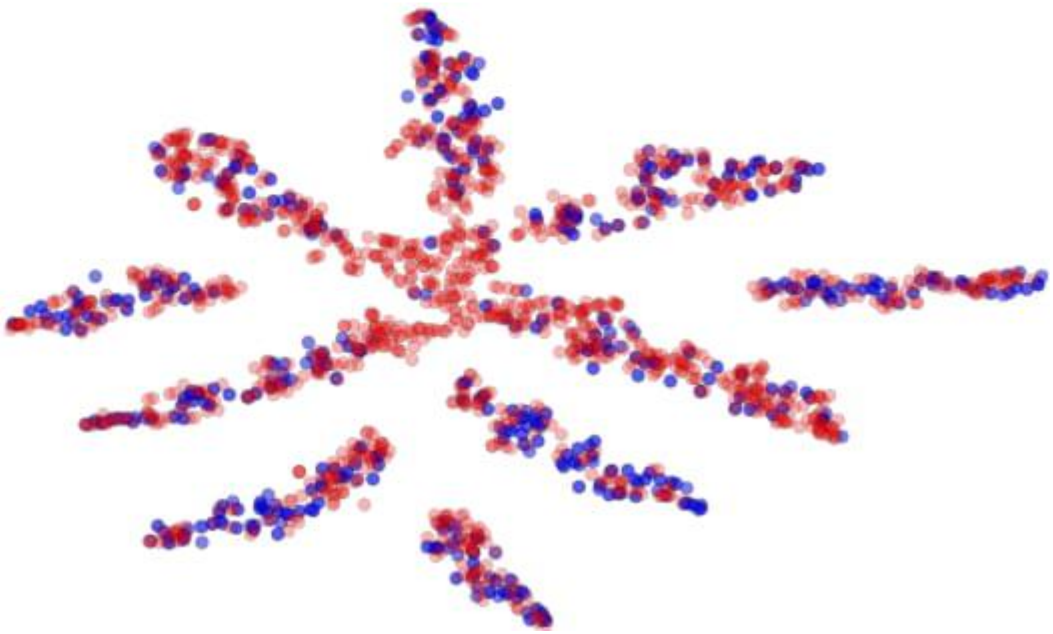


Adversarial Setup (Ganin and Lempitsky, 2015)

- In their paper, the representation after the adversarial training seems invariant to the domain

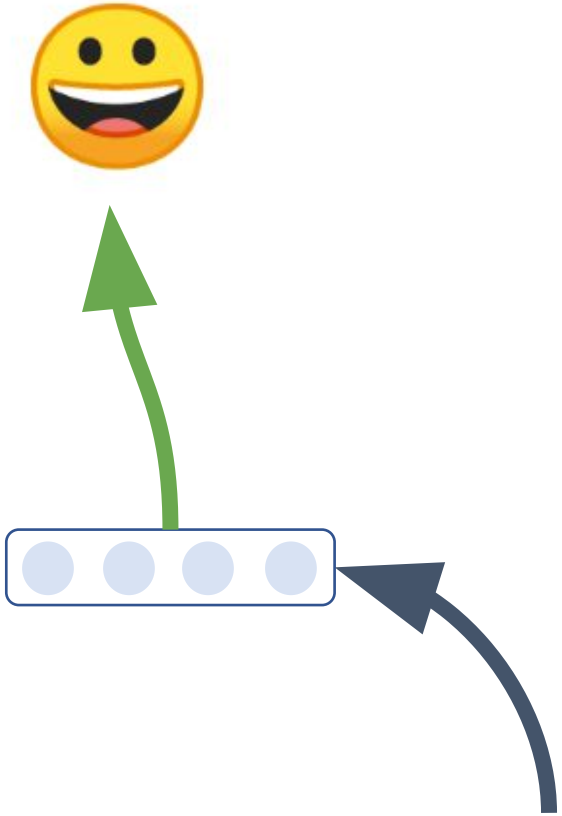


before



after

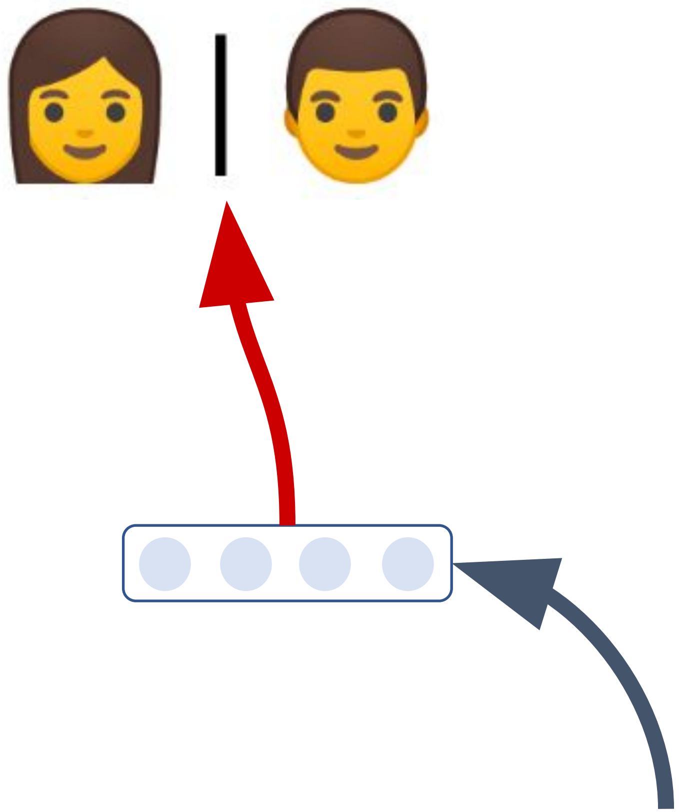
Does it work?



Successfully predicting sentiment

"I love mom's cooking"

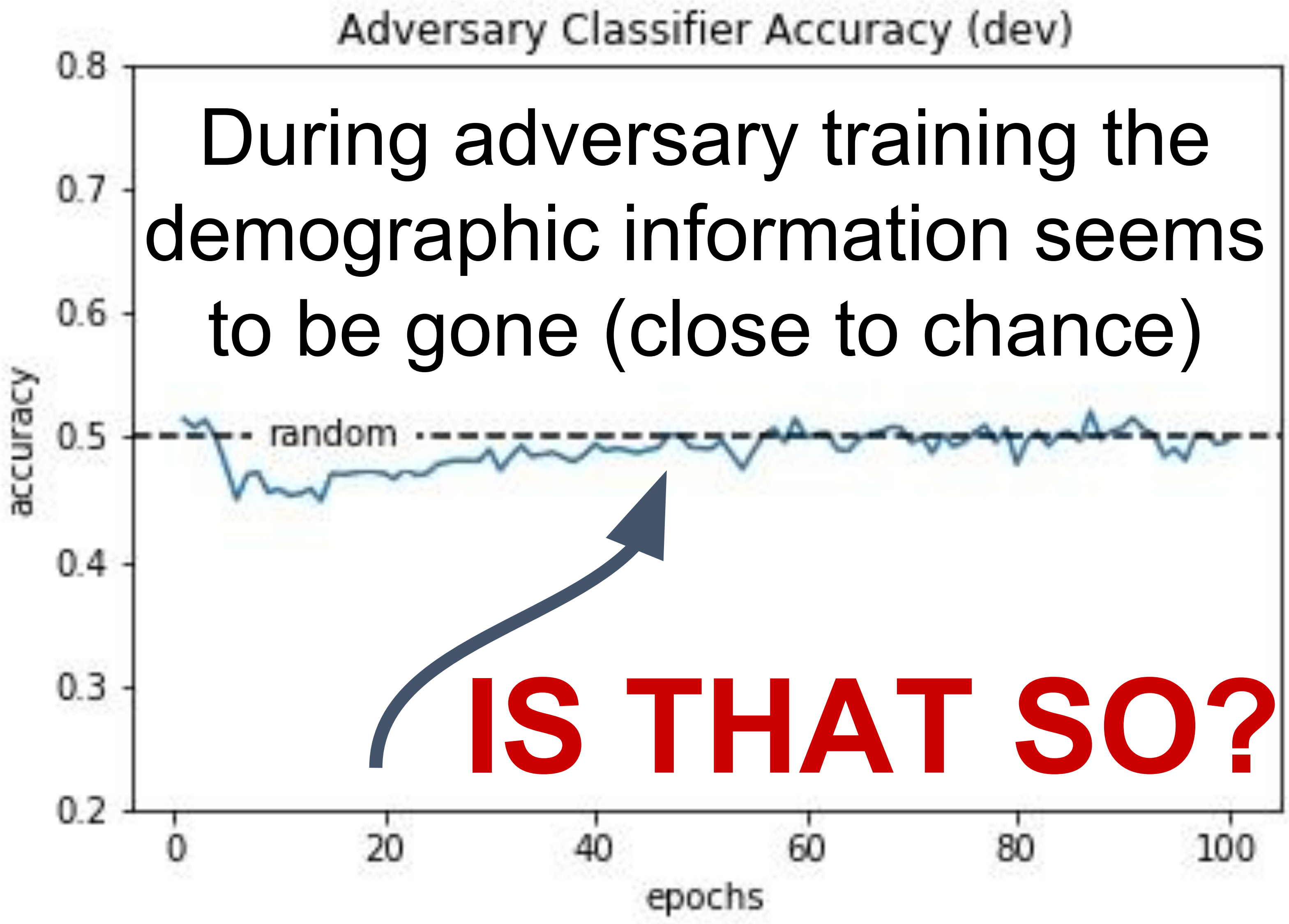
Does it work?



Successfully removed demographics?

“I love mom’s cooking”

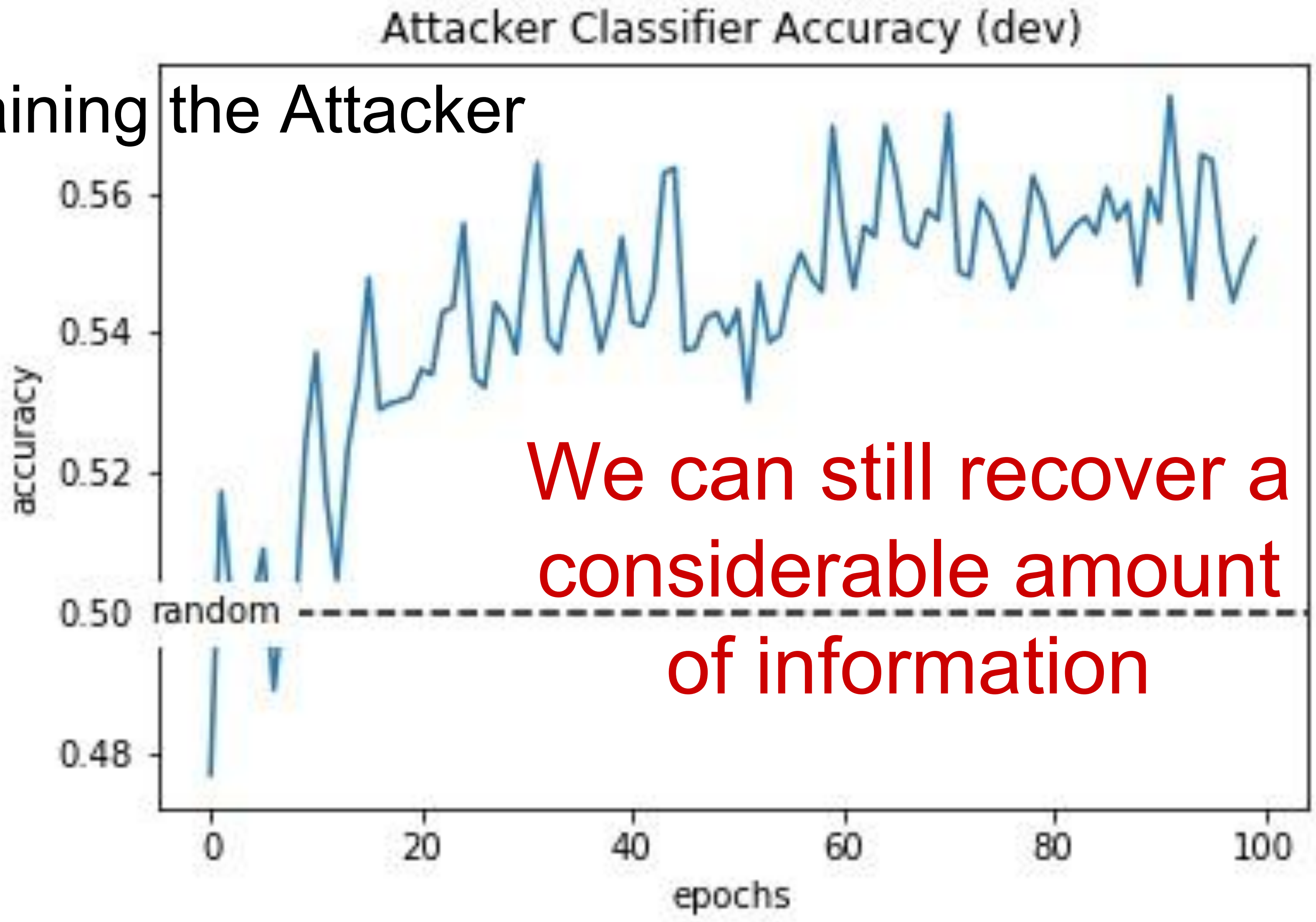
Does it work?



Adversary Classifier Accuracy (dev)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
accuracy
random
0
20
40
60
80
100
epochs
During adversary training the demographic information seems to be gone (close to chance)
IS THAT SO?

Does it work? Not so quickly…

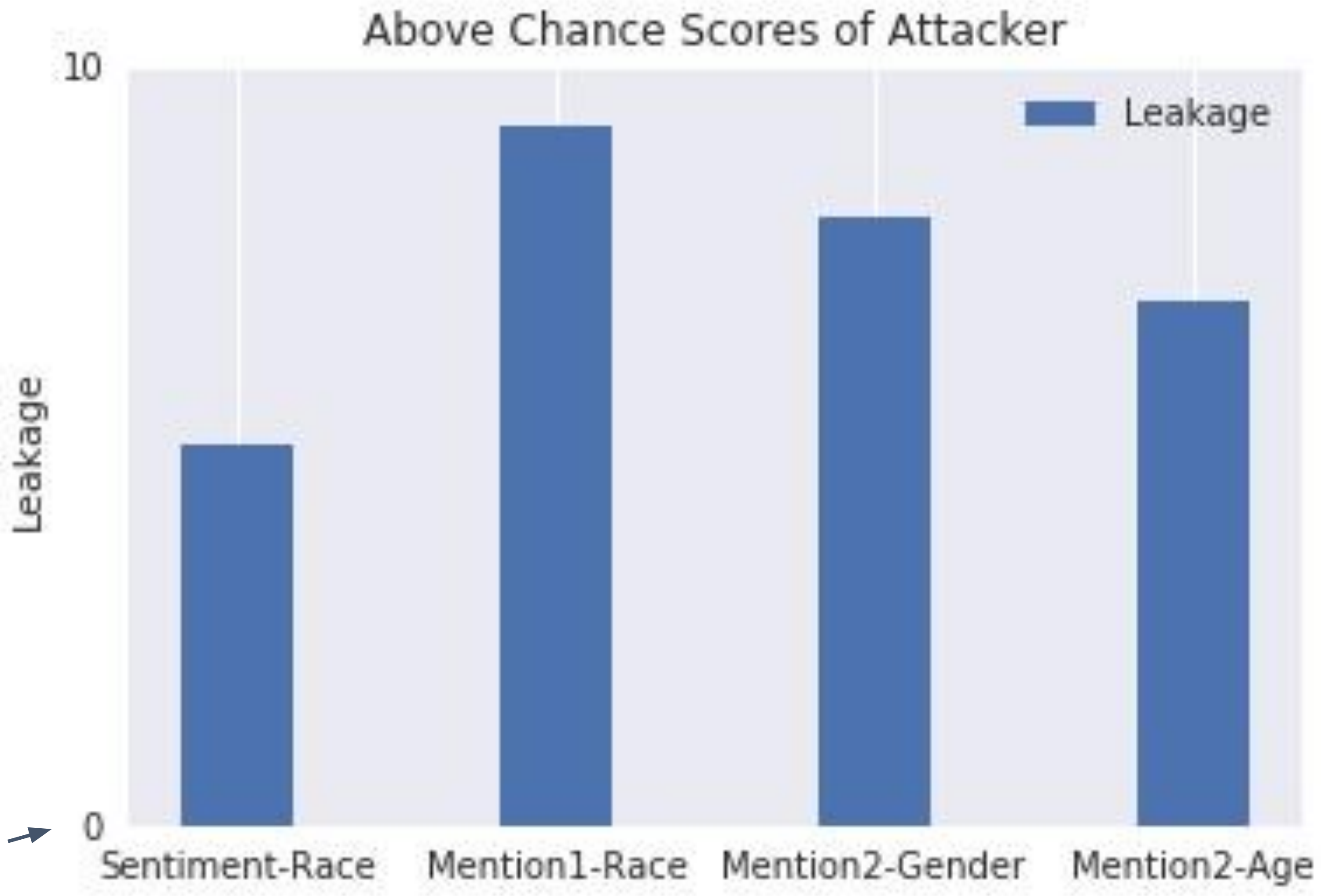
When training the Attacker



We can still recover a considerable amount of information

Does it work? Not so quickly...

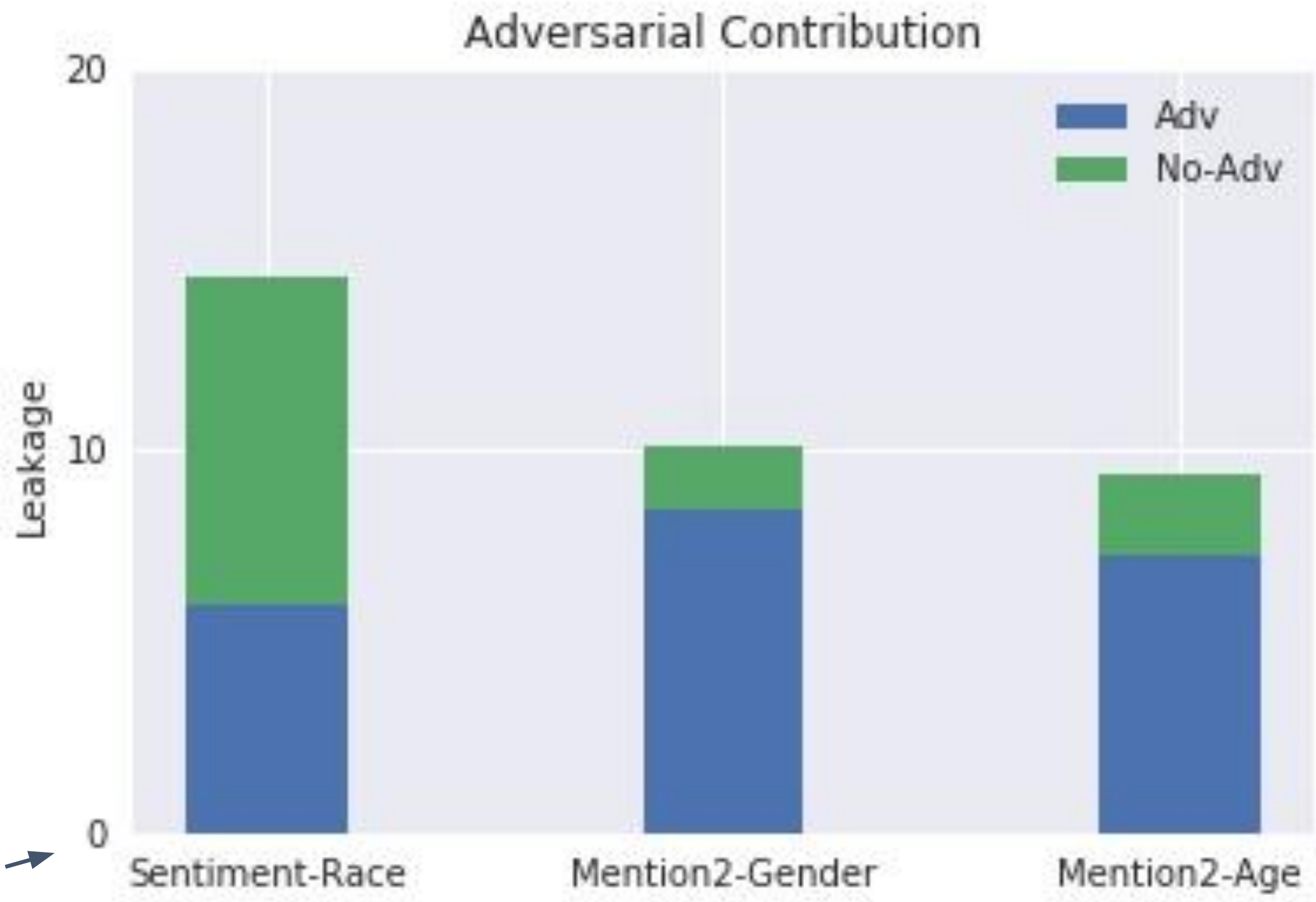
Consistent across tasks and protected attributes



Random Guess

Does it work? more or less

Well, the adversarial method does help.
But not enough



Random Guess

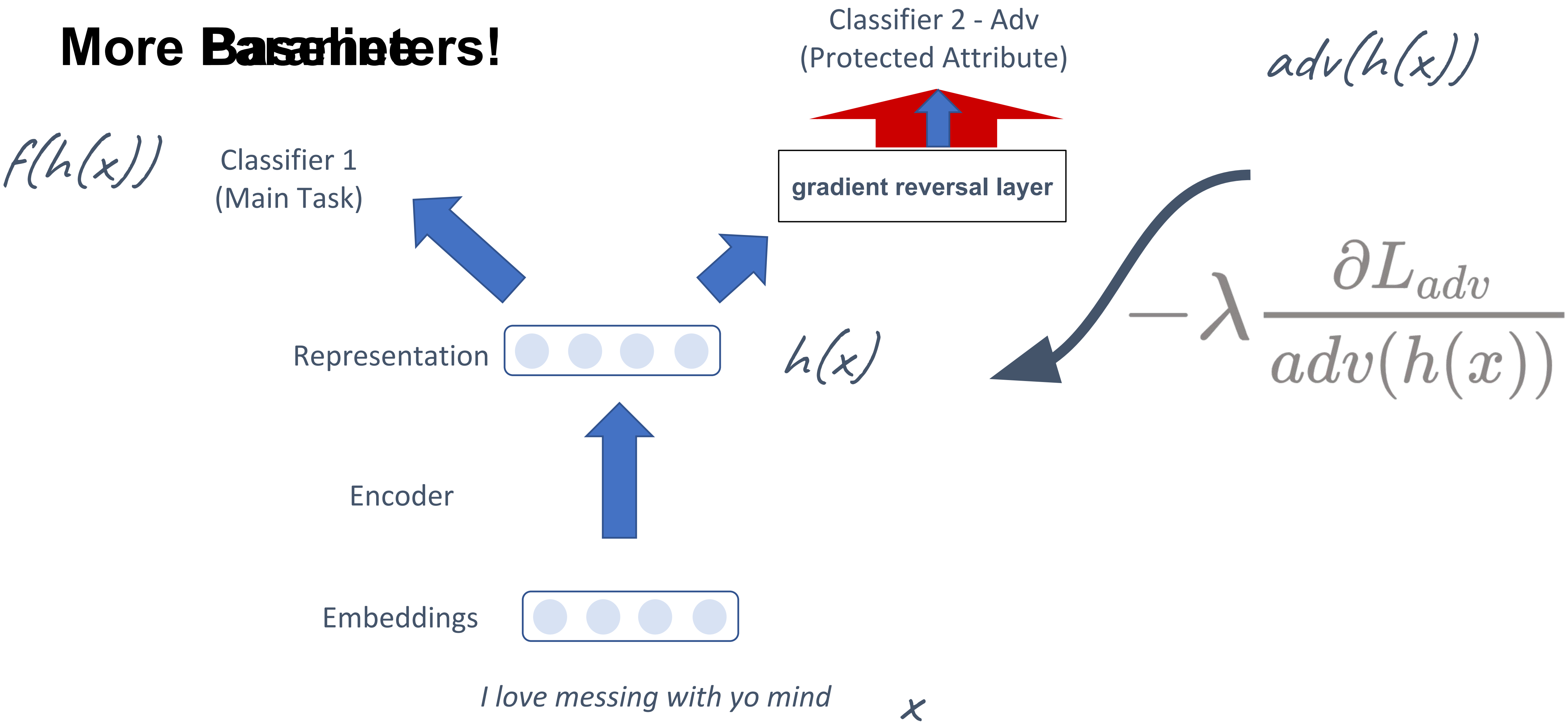
While effective during training, in test time, the adversarial do not remove all the protected information

Stronger, Better, Bigger???

Can we make stronger adversaries?

Stronger, Better, Bigger???

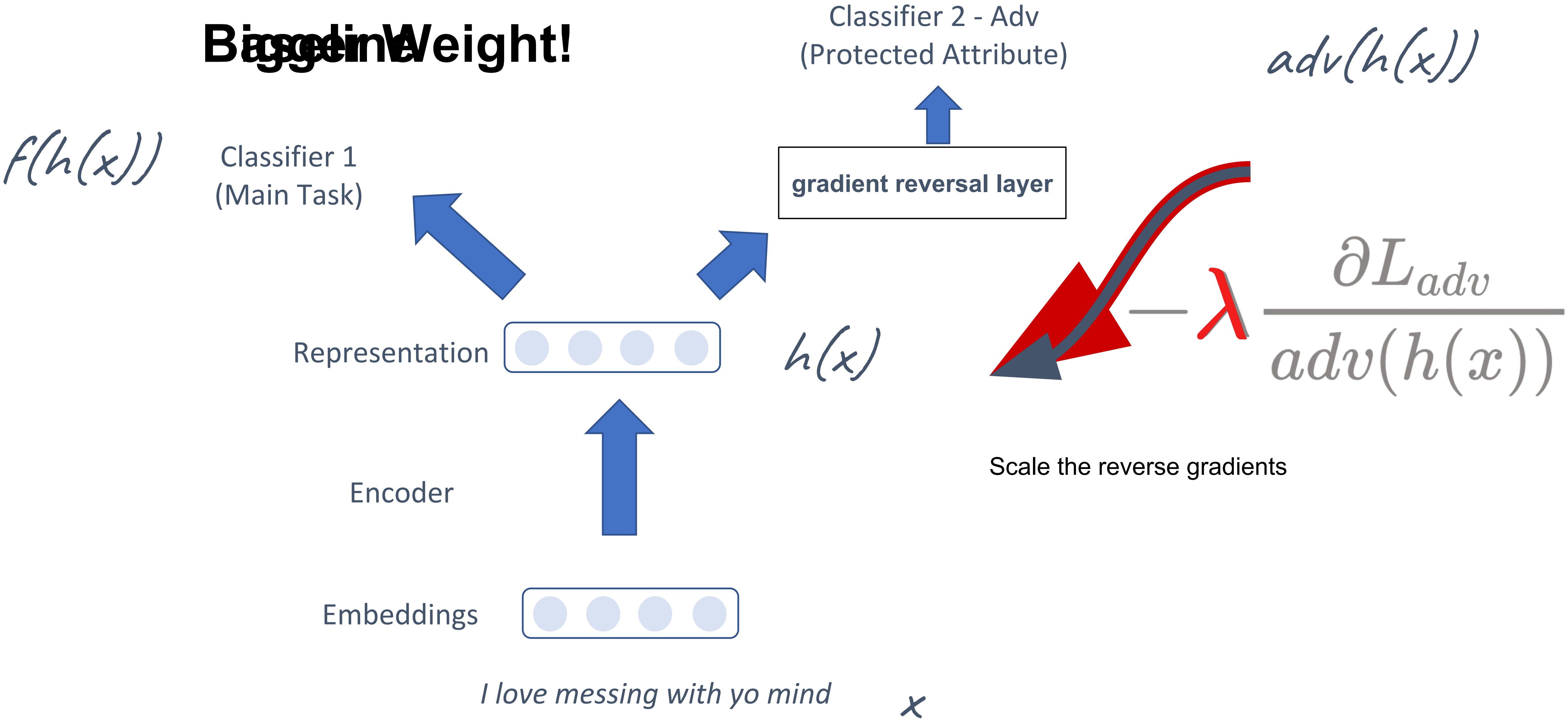
More Parameters!



Stronger, Better, Bigger???

Baseline

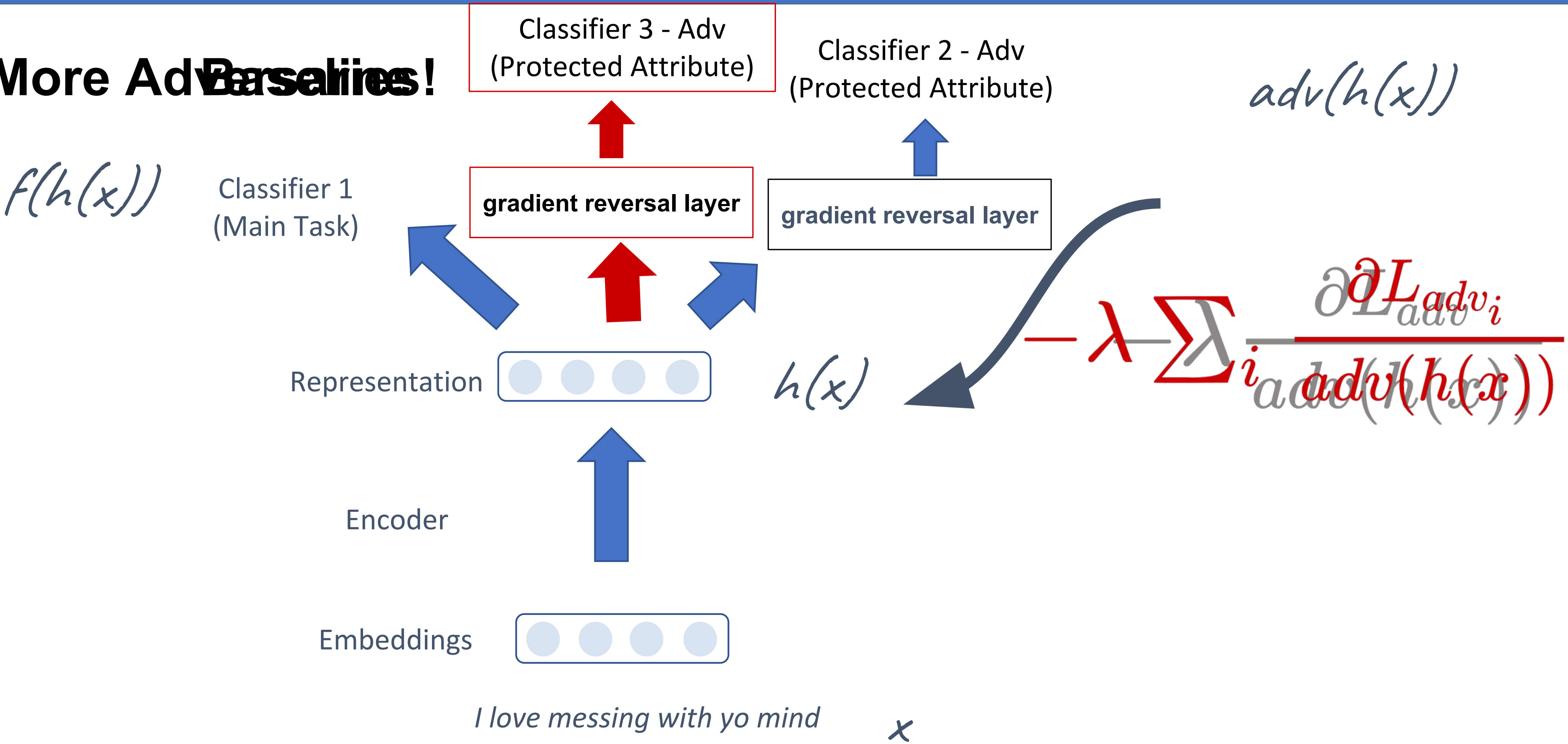
Bigger Weight!



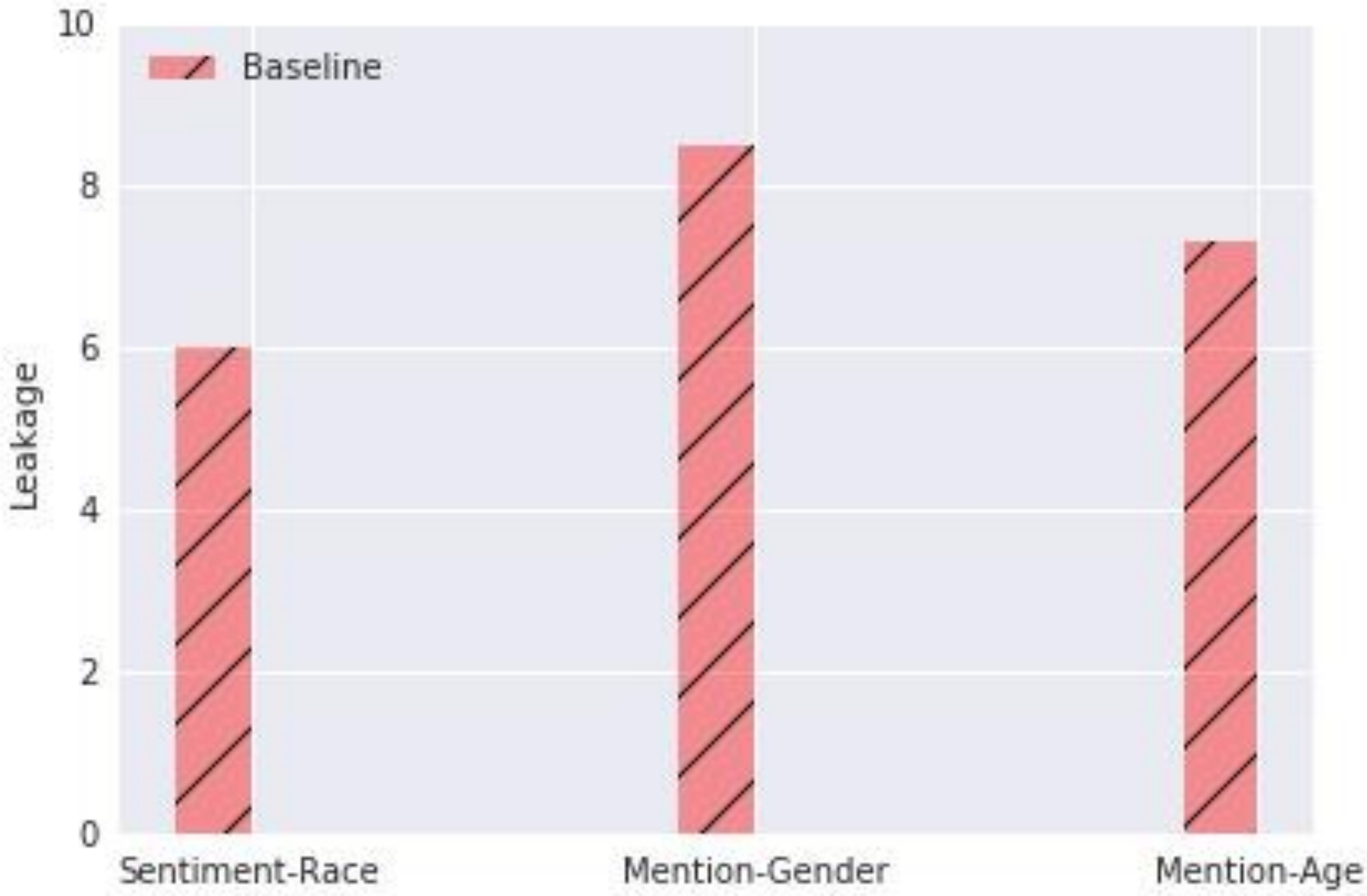
Scale the reverse gradients

Stronger, Better, Bigger???

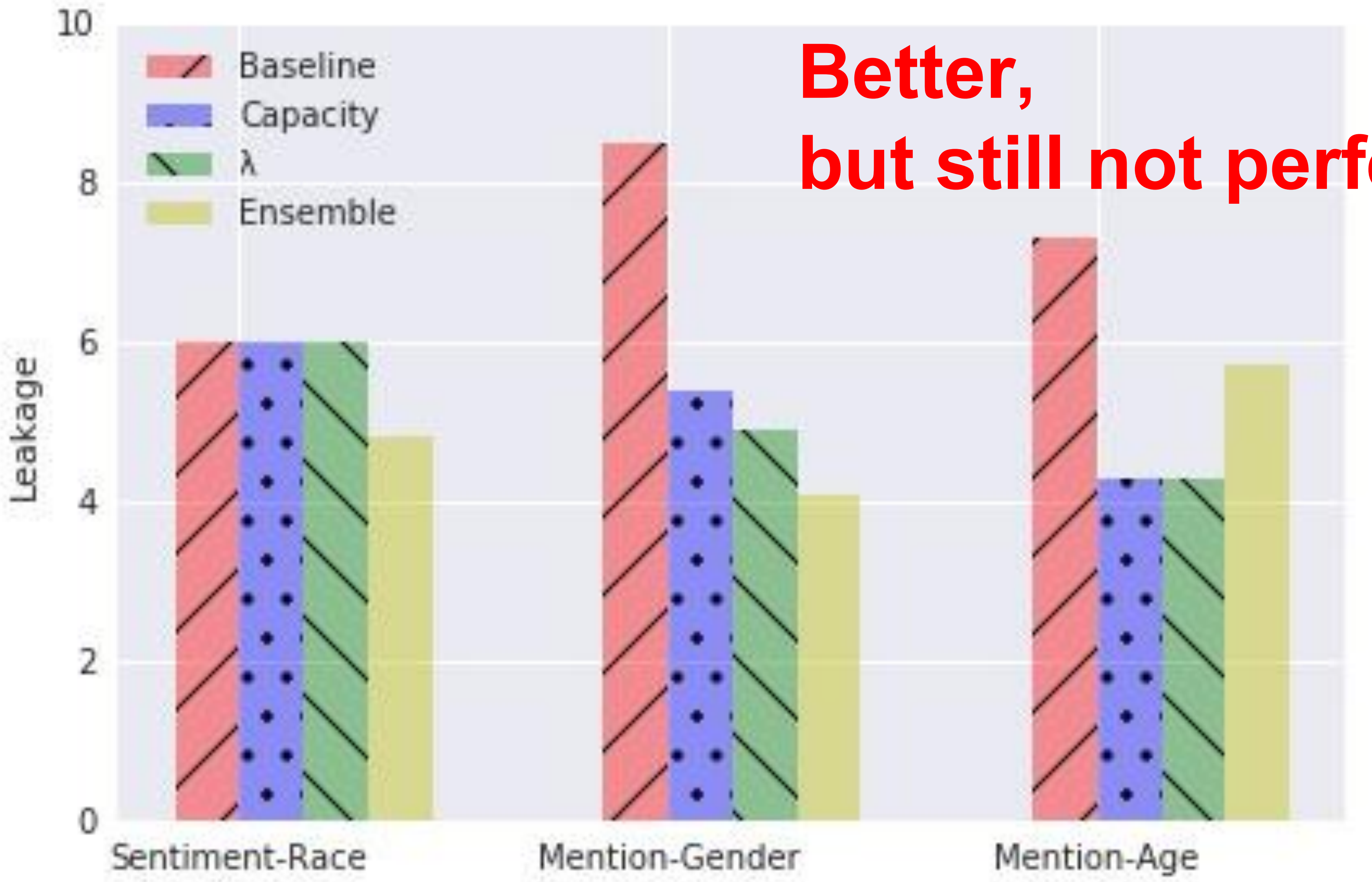
More Adversaries!



Stronger, Better, Bigger???



Stronger, Better, Bigger???



Better, but still not perfect

Error Analysis

Wait. I remember this thing called Overfitting

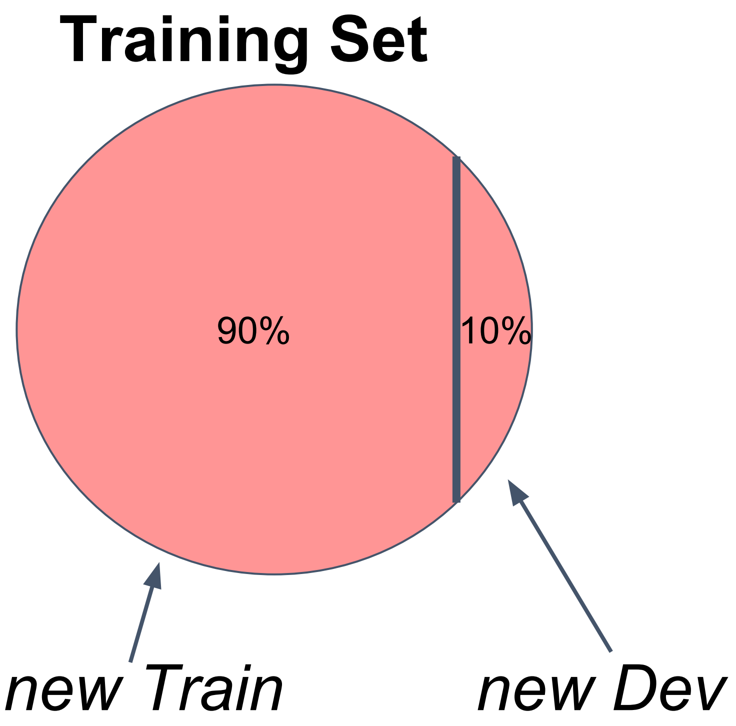
- We still have a problem
  - During training it seems that the information was removed
  - But the Attacker tells us another story
- Everything we reported was on the dev-set
- Is it possible that we just overfitted on the training-set?

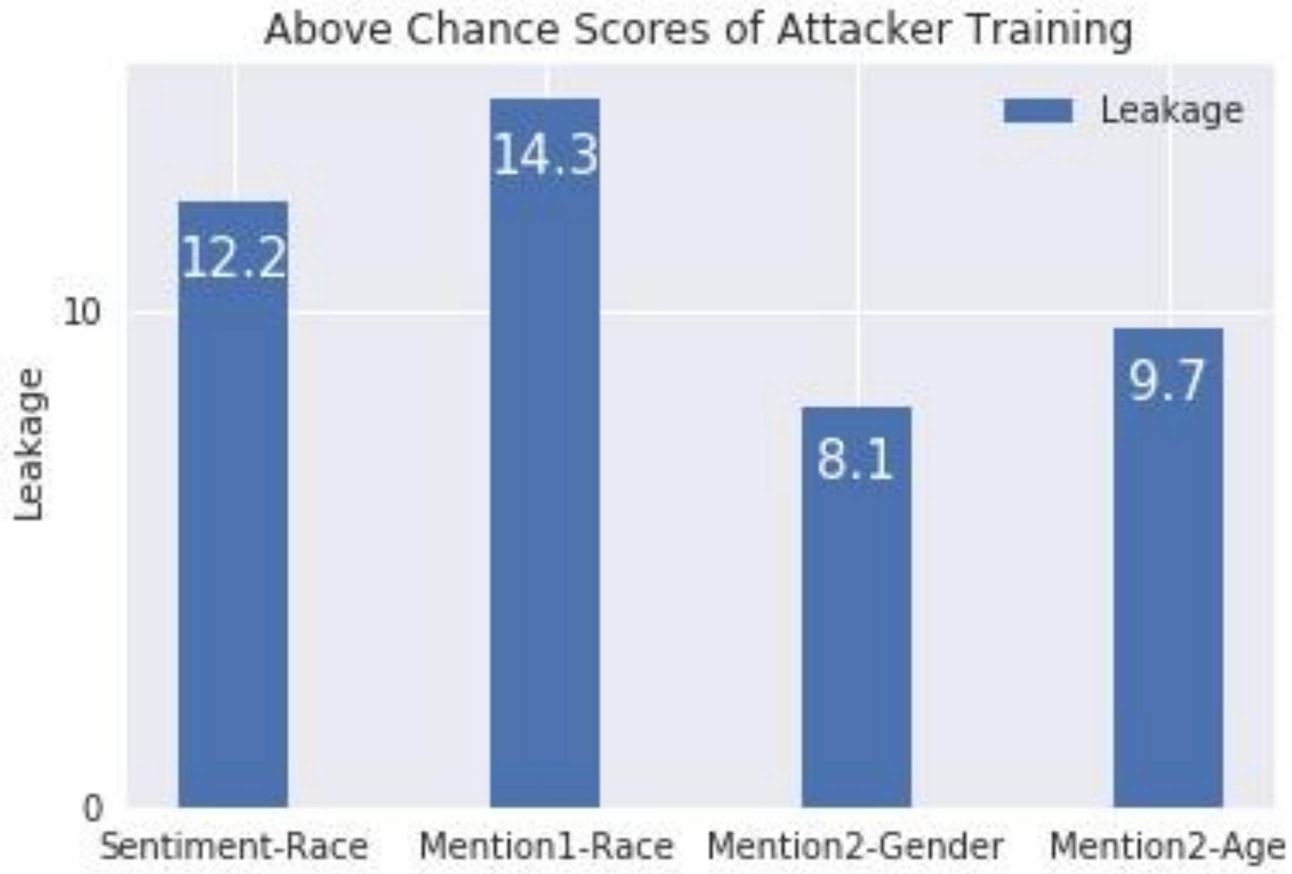
Wait. I remember this thing called Overfitting

- “Adversary overfitting”:
  - Memorizing the training data
  - By removing all its sensitive information
  - While leaking in test time

Wait. I remember this thing called Overfitting

We trained on 90% on the "overfitted" training set, and tested the remaining 10%





It is more than that

Persistent Examples

- What are the hard cases, which slip the adversary?
  - We trained the adversarial model 10 times (with random seeds)
  - then, trained the Attacker on each model
  - We collected all examples, which were consistently labeled correctly

Persistent Examples

AAE("non-hispanic blacks")
Enoy yall day
_ Naw im cool
My Brew Eatting
My momma Bestfrand died
Tonoght was cool

SAE ("non-hispanic whites")
I want to be tan again
Why is it so hot in the house?!
I want to move to california
I wish I was still in Spain
Ahhhh so much homework.

More about the leakage origin can be found in the paper

Few words about fairness

- Throughout this work, we aimed in achieving zero leakage, or in other words: *fairness by blindness*

- Many other definitions for “fairness” (>20)
- With 3 popular
  - *Demographic parity*
  - *Equality of Odds*
  - *Equality of Opportunity*

In the paper, we prove that in out setup (balanced data) these definitions are identical

Summary

- When training a text encoder for some task
  - Encoded vectors are still useful for predicting various things (“transfer learning”)
  - Including things we did not want to encode (“leakage”)
- **It is hard to completely prevent such leakage**
  - **Do not blindly trust adversarial training**
  - **Recheck your model using an “Attacker”**

Thank you