Fairness of AI models

How to prevent or correct systemic biases?

Mariana DUTRA Anna JÄRVINEN Mariana OLM Felipe VICENTIN

4IM06 - Modèles génératifs, méthodes par patchs, photographie computationnelle
Télécom Paris

June 2025

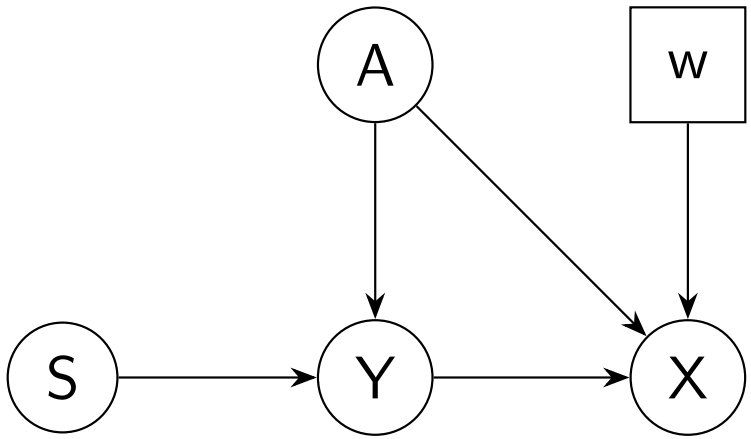
Contents

Introduction

Fairness

Statistical modeling:

- data $X \in \mathbb{R}^n$
- labels $Y \in \{0, 1\}$
- sensitive attributes $A \in \{0, 1\}$
- scenario $S \in \{\text{train}, \text{test}\}$



Fair classification:

- The predictor outputs $\hat{Y} \in \{0, 1\}$.
- We seek to learn to predict outcomes that are accurate with respect to Y and fair with respect to A.

Fairness criteria

- There are many possible criteria for group fairness.

Demographic Parity

$$P(\hat{Y} = 1 \mid A = 0) = P(\hat{Y} = 1 \mid A = 1)$$

- Enforces the rate of a positive outcome ($\hat{Y} = 1$) to be the same regardless of A.

Equalized Odds

$$\begin{cases} P(\hat{Y} \neq Y | A = 0, Y = 0) = P(\hat{Y} \neq Y | A = 1, Y = 0) \\ P(\hat{Y} \neq Y | A = 0, Y = 1) = P(\hat{Y} \neq Y | A = 1, Y = 1) \end{cases}$$

- Enforces the rate of errors to be equal across groups.
- For scenarios where $P(Y = 1|A = 0) \neq P(Y = 1|A = 1)$

Learning Adversarially Fair and Transferable Representations (LAFTR)

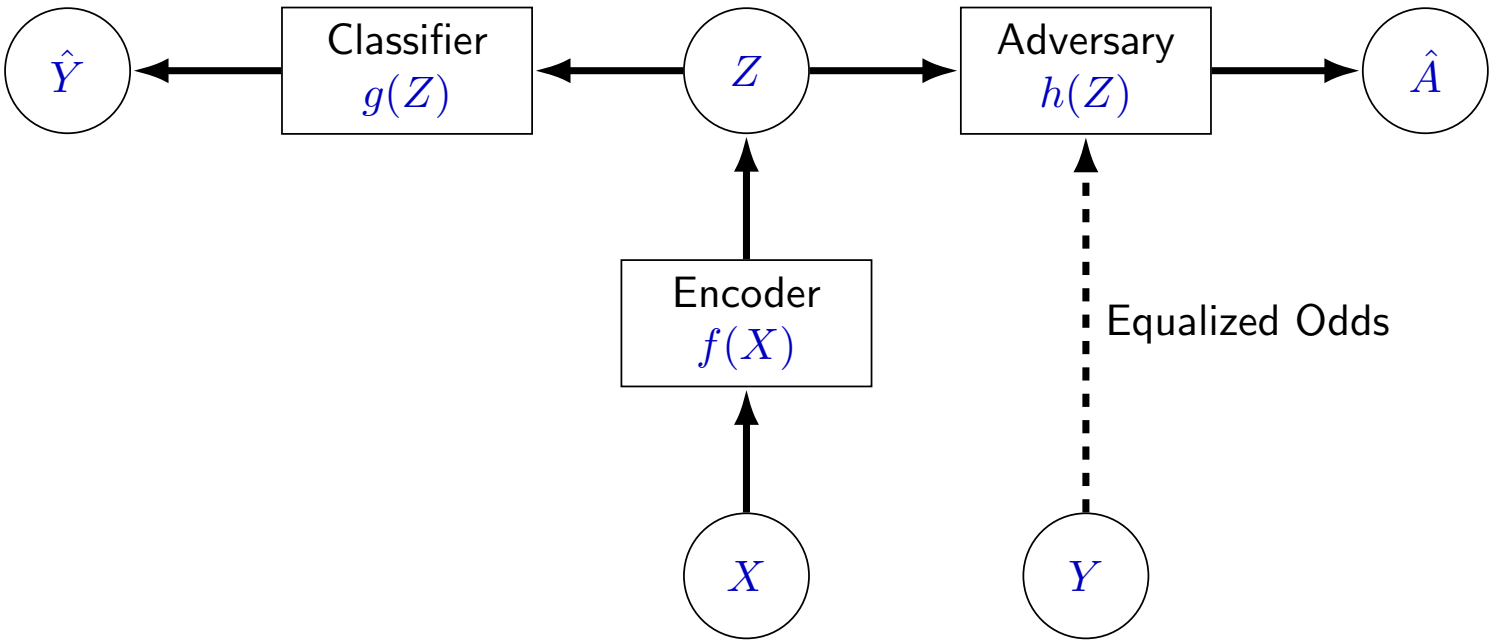


Figure: Generalized model for learning adversarially fair representations.

Data

Biased Data

- We define the bias using a conditional probability matrix:

$$M_{i,j} = P(Y = i \mid A = j), \quad \sum_{i=1}^{C} M_{i,j} = 1$$

- Biasing scheme (Controlled via β):

$$P(Y = i \mid A = j) = (1 - \beta) \cdot \frac{1}{C} + \beta \cdot \mathbf{1}_{\{i=d_j\}},$$

- Recover Attribute Distribution:

$$\mathbf{p}_y = M\mathbf{p}_a \quad \Rightarrow \mathbf{p}_a = M^{+}\mathbf{p}_y$$

- Sample Attributes Using Bayes' Rule:

$$P(A \mid Y) = \frac{P(Y \mid A) \cdot P(A)}{P(Y)},$$

Biased Binary Colored MNIST

- Binary classification:

$$Y = \begin{cases} 0 & \text{if digit is even (0,2,4,6,8)} \\ 1 & \text{if digit is odd (1,3,5,7,9)} \end{cases}$$

- Assign background based on $P(A \mid Y)$ via Bayes' rule

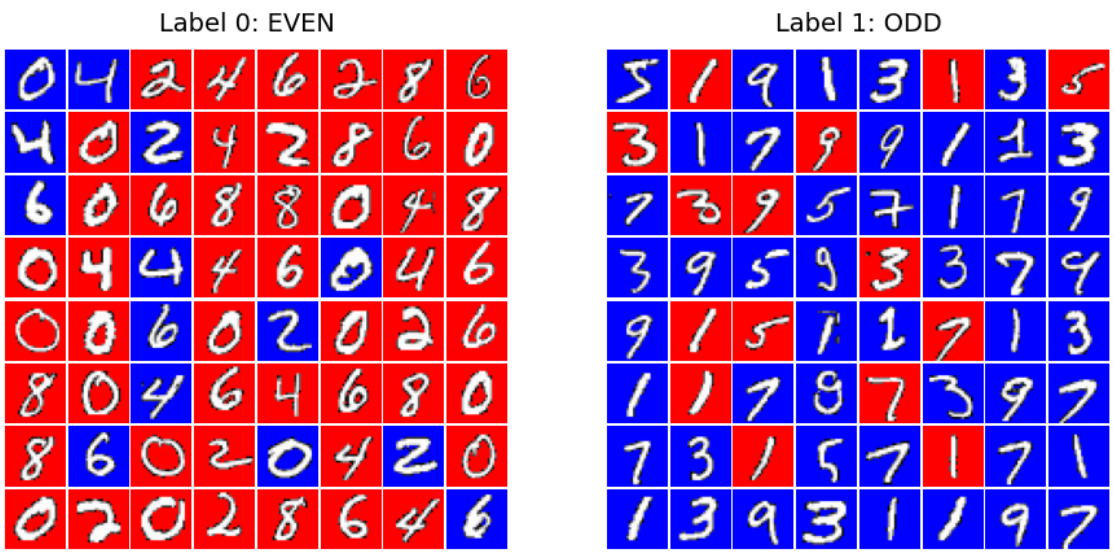


Figure: Samples of biased MNIST data by true label. ($\beta = 0.6$)

Biased CIFAR-10

- CIFAR-10: RGB images from 10 object classes

Figure: Samples of biased CIFAR-10 data.

Methodology

Architecture

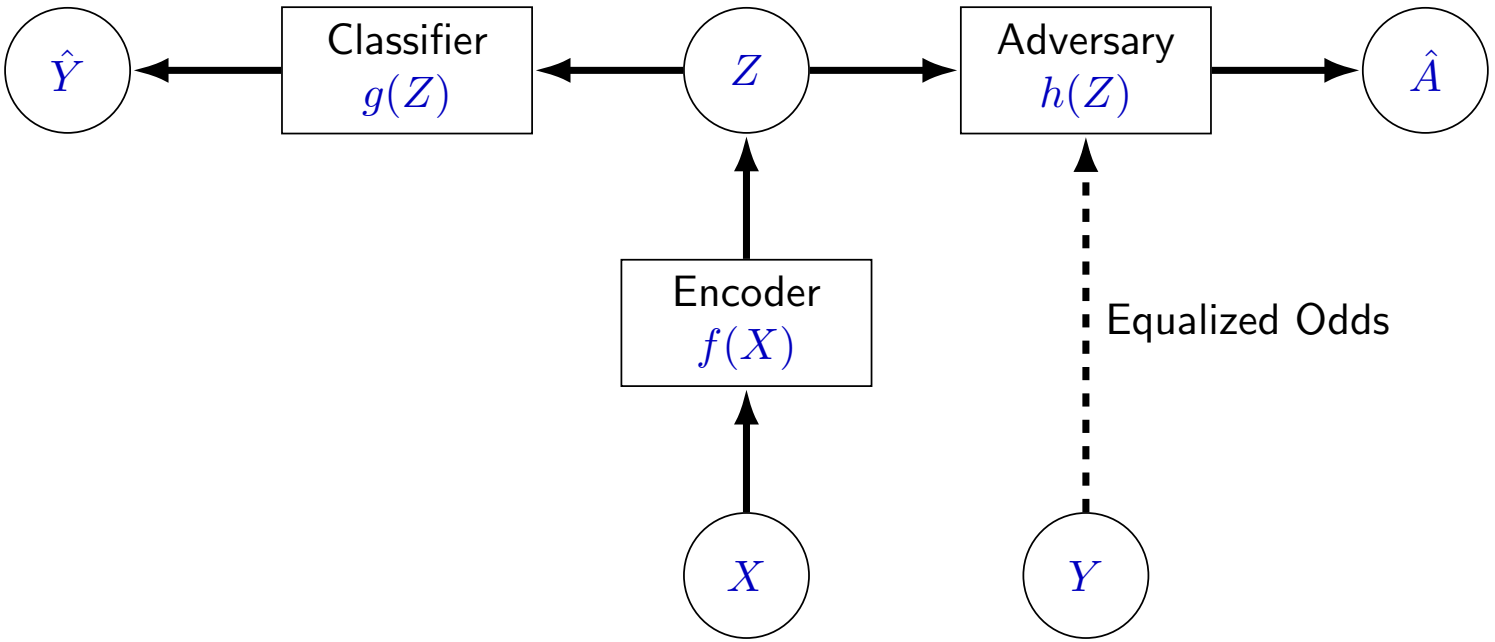
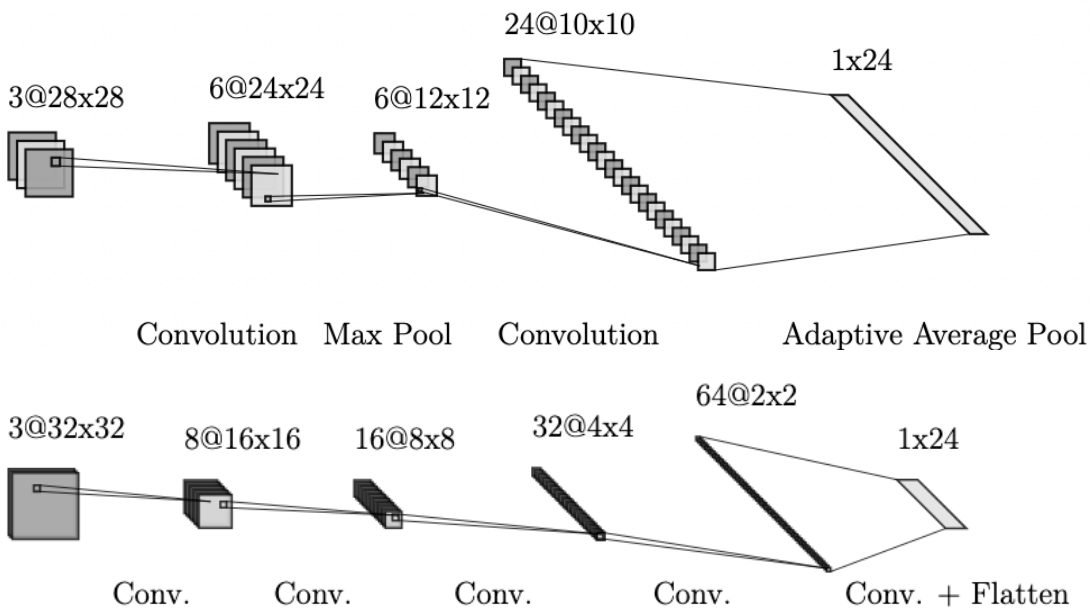


Figure: Generalized model for learning adversarially fair representations.

Encoders

- Three Encoders: MLPEncoder and ConvEncoder for MNIST, ConvEncoderCIFAR for CIFAR-10
- Encoder must extract a latent representation that is both informative for the prediction task and invariant to the sensitive attribute



Classifier and Adversary

- The Classifier and Adversary are intentionally simple

Classifier

- Single linear layer

Adversary

- Two-layered MLP with ReLU

Adversarial Loss

- We define the *sensitive groups* of the dataset $\mathcal{X}$, for $a, y \in \{0,1\}^2$:

$$\mathcal{D}_a = \{(X, A) \in \mathcal{X} \mid A = a\}$$
$$\mathcal{D}_a^y = \{(X, A) \in \mathcal{X} \mid A = a, Y = y\}.$$

Demographic Parity

$$L_{\text{Adv}}^{\text{DP}}(h) = -1 + \sum_{a \in \{0,1\}} \frac{1}{|\mathcal{D}_a|} \sum_{(X,A) \in \mathcal{D}_a} |h(f(X, A)) - a|$$

Equalized Odds

$$L_{\text{Adv}}^{\text{EO}}(h) = -2 + \sum_{a \in \{0,1\}} \sum_{y \in \{0,1\}} \frac{1}{|\mathcal{D}_a^y|} \sum_{(X,A) \in \mathcal{D}_a^y} |h(f(X, Y), Y) - a|.$$

Combined Loss

- Let L_C be some classification loss: Cross Entropy.
- We define the objective function of the model:

Combined Loss

$$L(f, g, h) = L_C(\hat{Y}, Y) - \gamma L_{\text{Adv}}(\hat{A}, A)$$

- $\gamma \geq 0$ controls how the Encoder and Classifier should punish the Adversary.

- $\gamma = 0$, Adversary has no obstacle: latent representation is biased.
- $\gamma \to \infty$, Encoder hides all information about A in the latent representation that could be learned by the Adversary.

Training Overview

- Training alternates between:
  1. **Encoder f + Classifier g:** minimize classification loss and hide sensitive info from adversary.
  2. **Adversary h:** maximize ability to infer sensitive attribute A from latent representation Z.

Training Loop

- Freeze h, update f and g to minimize $L_C - \gamma L_{\text{Adv}}$
- Freeze f, g, update h to minimize L_{Adv}

Results

Results & Discussion

- Baseline for 3 scenarios:
  - Same bias;
  - No bias;
  - Inversed bias.
- Adversarial baseline is random guessing.
- Run for multiple values of γ, starting from 0.
- For each experiment, we tested DP and EO.

Results & Discussion

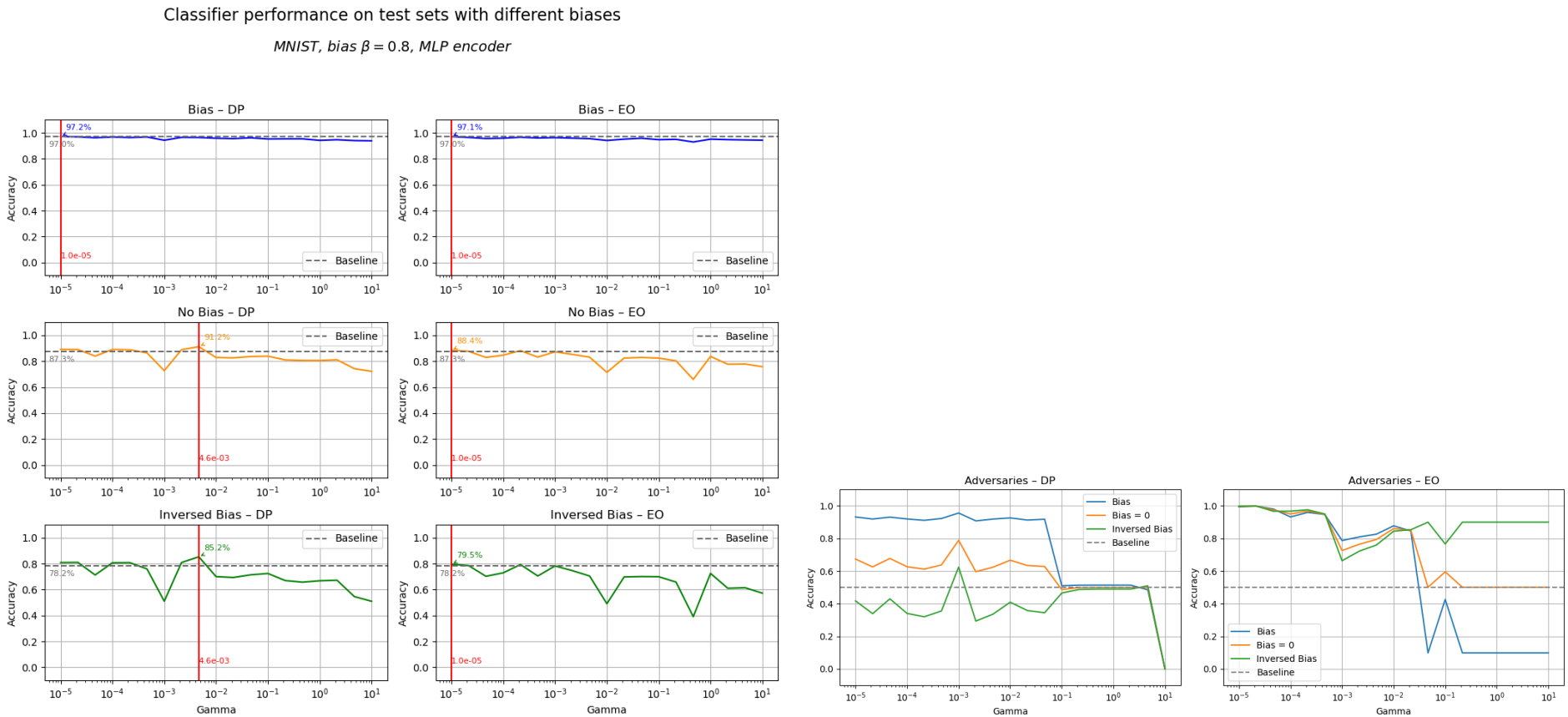


Figure: Results on MNIST with $\beta = 0.8$

Results & Discussion

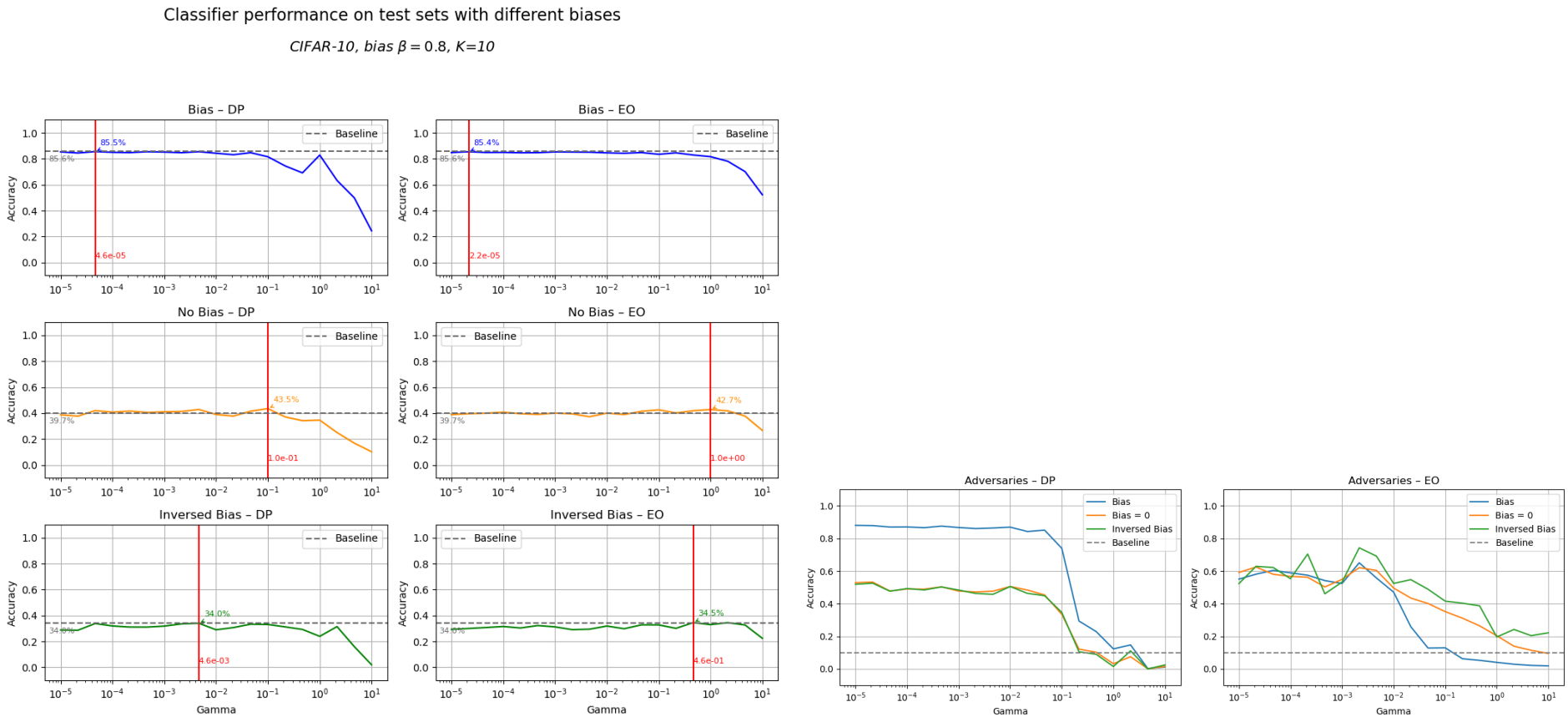


Figure: Results on CIFAR-10 with $\beta = 0.8$, $K = 10$

Results & Discussion

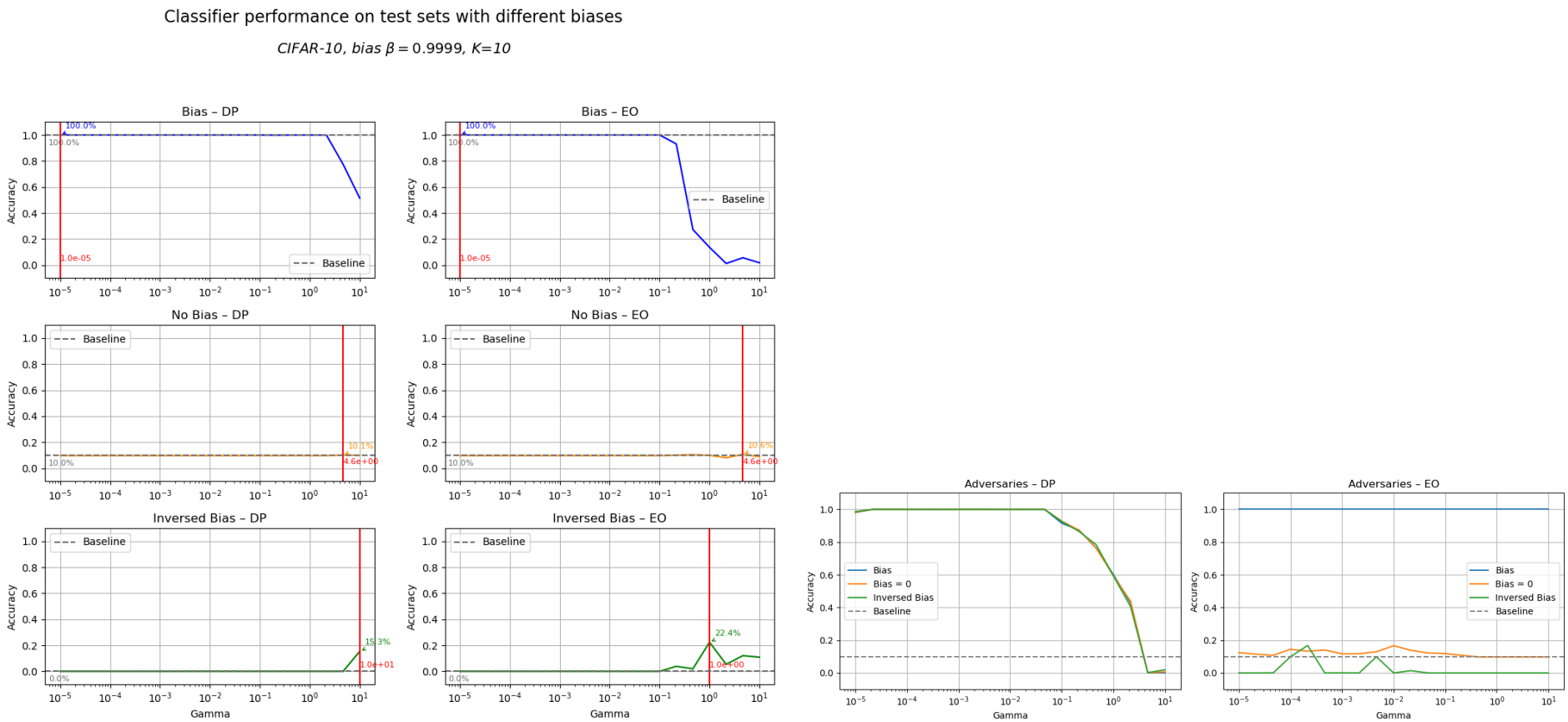


Figure: Results on CIFAR-10 with $\beta = 0.9999$, $K = 10$

Conclusion

Conclusion

- The classifier maintains stable accuracy across scenarios, while the adversary's accuracy drops to random guessing beyond a certain penalization threshold.
- Excessively high adversary penalization (γ) causes both classifier and adversary to perform no better than random, likely due to the encoder suppressing all useful information.
- The results suggest that LAFTR effectively hides biased attributes in the latent space, successfully fooling the adversary for both binary and non-binary attributes.
- However, the classifier did not improve on the unbiased test set, possibly due to limitations in latent dimension size, hyperparameter tuning, or model architecture.

Thank you !

frameProject structure verbatim LAFTR/ data/ mnist$_{d}ata/BinaryColoredMNIST.pymodels/losses/notebooks/train_{e}nc$