# 2021.03.09
# 미팅

**신승재**
Phd Candidate
Applied Artificial Intelligence Laboratory
Department of Industrial and Systems Engineering, KAIST

# Bias 스터디

- **이전의 진행 방식**
  - 분야 : Dataset Bias, Invariant Learning, Out-of-distribution Generalization, Noisy Label

  ⇔ Noisy label이 들어간 이유는 Dataset Bias 분야의 방법론과 매우 긴밀한 관계를 가지고 있기 때문에...!
  - 차주에 하나의 논문을 소개하는 정도의 로드로 진행했음.
  - +) 참여자 중 일부 인원의 개인 연구에 대한 Discussion 포함했었음.

- **새롭게 바뀌는 점**
  - Bias 스터디 / 개인연구 Discussion 분리
  - Bias 스터디 참여자
    - (현) 신승재, 김혜미, 송경우, 장준호
    - 추가 참여 가능. 가능 일정 및 시간대의 경우 최종 참여자끼리 협의
    - 스터디 주제 및 범위 또한 최종 참여자끼리 협의. => 오늘 Discussion 끝나구 잠시 이야기하시죠!

  - 승재 개인연구 Discussion 참여자
    - 신승재, 김혜미, 송경우, 장준호, 배희선, 나병후, 조수현
    - 차후 논문화할 경우에, 모든 인원 공저자로 배치 가능.
    - 허나 그만큼 본인이 후순위 (최대 7저자) 저자로 갈 확률이 있으므로, 이에 대한 Risk는 본인이 판단해주었으면 좋겠음.
    - 매주 Discussion을 여는 것을 목표로 하고 있으나, 실험이나 방법론적으로 Discussion 거리가 더 없을 경우 해당 주엔 넘기는 방안도 있을 듯.

For accurate and unbiased recognition, the network should learn about a category by relying more on its **corresponding pixel regions (Object)** than those of its **context. (Bias)**

- From Don’t Judge an Object by Its Context: Learning to Overcome Contextual Bias (CVPR 2020)

# DATASET BIAS IN ML

# Dataset Bias in Machine Learning

- Why Bias can be dangerous for image classification?
  - Model seems to recognize simple-cues for the accurate recognition.
  - In the training and test sets, the short-cuts for predictions can change into different context.
  - The biased models fail to generalize when the bias shifts to a different class.

- Example cases

**Training Data**

Car on road

Frog in swamp

Model

"Car" have road patterns
"Frog" have swamp patterns

Use this patterns to maximize the accuracy

**Deployment**

Car in Swamp

Frog on road

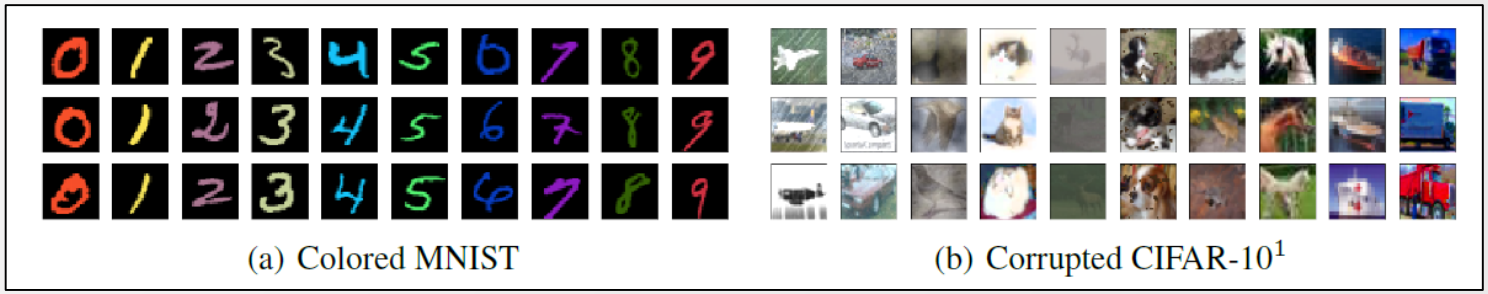
Model

Left thing is "Frog"
Right thing is "Car"

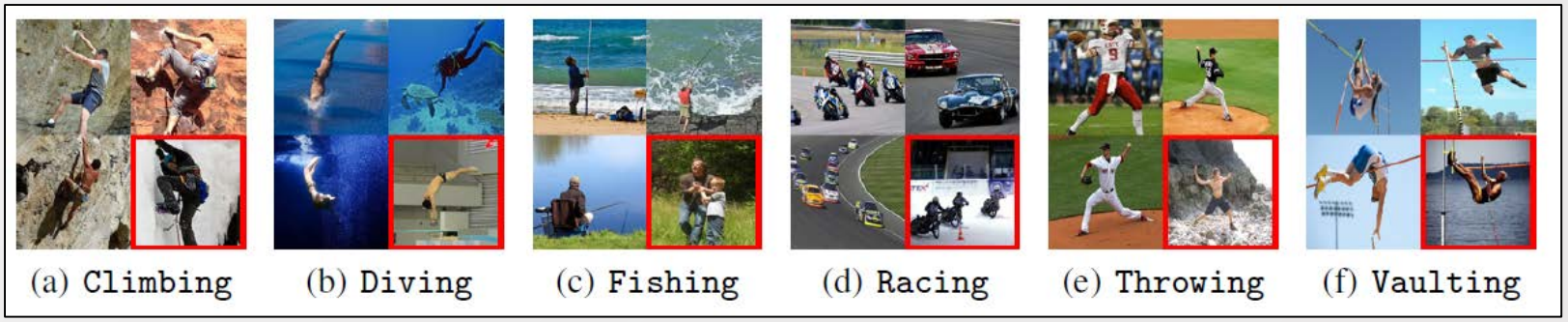
Biased prediction, which is huge barrier for generalization.

# Example of the biased data

- Colored MNIST
  - Bias : Color, Label : digit
- Cifar-10
  - Bias : Corruption (Snow, Frost, Fog, Brightness, Contrast, Spatter, Label : object
- Biased Action Recognition
  - Bias : Place (Rockwall, Underwater, Water-Surface), Label : Action (Climbing, Fishing, Racing)



(a) Colored MNIST (b) Corrupted CIFAR-10[1]



(a) Climbing (b) Diving (c) Fishing (d) Racing (e) Throwing (f) Vaulting

# Cross-bias generalization Problem

- In definition, This is Cross-bias generalization Problem.
  - First, Bias and label in the training distribution is correlated. However, the dependency changes across training and test distributions. 즉, Training dataset에서의 Correlation만 절대적으로 신뢰하면, 큰 코 다칠 수 있다.
    - Signal S **(개구리)** : The cues essential for the recognition of X as Y -> Train, Test 데이터에 상관없이 Signal과의 관계는 유지됨.
    - Bias B **(늪)** : The cues not essential for the recognition but correlated with the target Y -> 데이터에 따라 Bias와 Y (Label)간의 상관관계가 달라질 수 있다.

In training distribution, biases and labels are correlated 1) $p(B\text{^}tr)$ dependent on $p(\text{Y}\text{^}tr)$ the dependency changes across test distribution 2) $p(B\text{^}tr, Y\text{^}tr) \neq p(B\text{^}\text{te}, Y\text{^}\text{te})$,

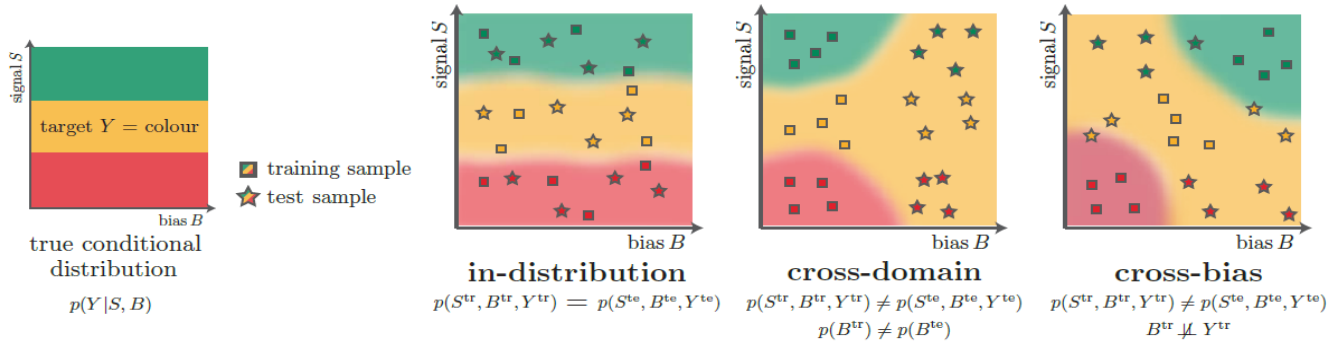
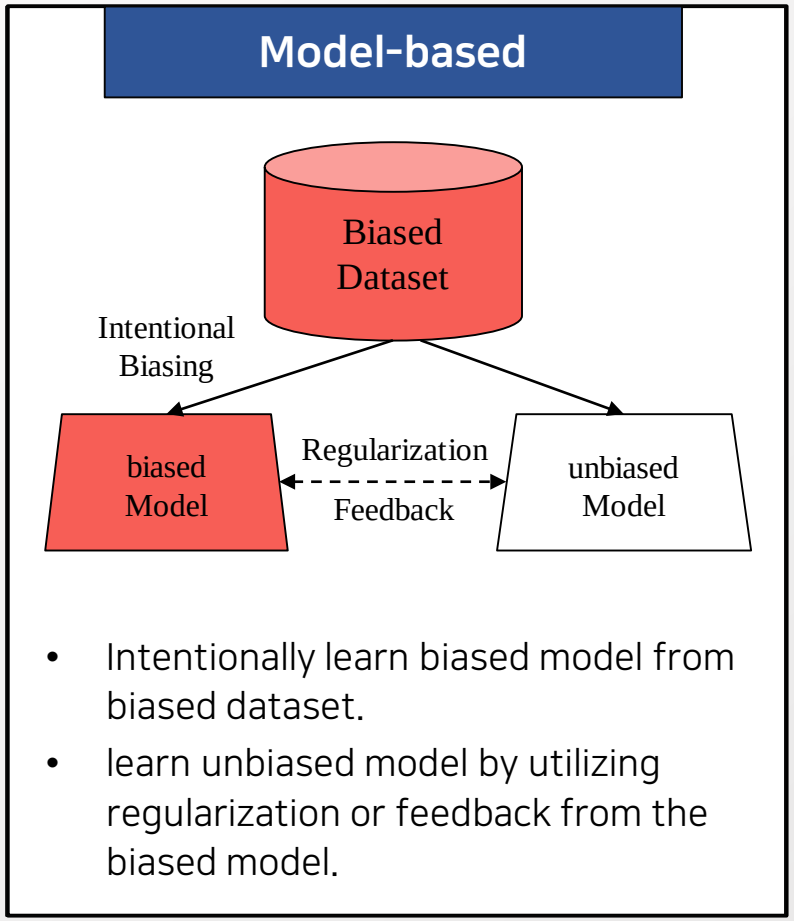



*Figure 1.* **Learning scenarios**. Different distributional gaps may take place between training and test distributions. Our work addresses the *cross-bias generalisation* problem. Background colours on the right three figures indicate the decision boundaries of models trained on given training data.

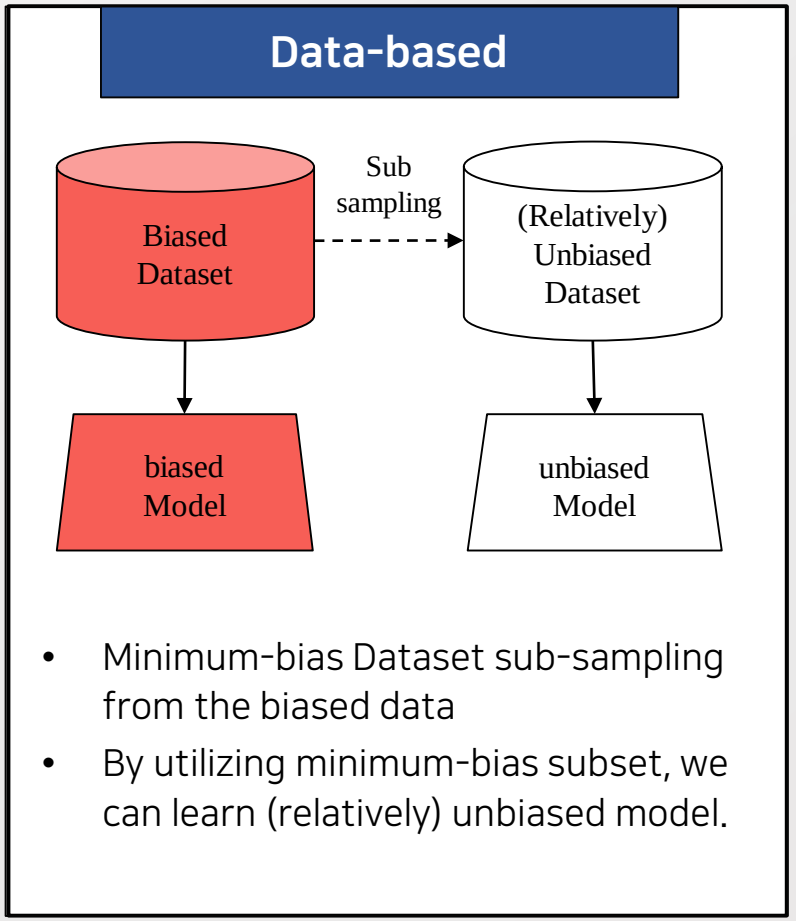
# MAIN BASELINE

# De-biasing mechanisms

- **De-biasing mechanism for unknown biases**
  - It can be divided into 1) model-based de-biasing and 2) Data-based de-biasing.



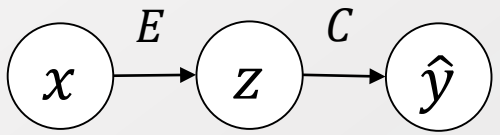

- Intentionally learn biased model from biased dataset.
- learn unbiased model by utilizing regularization or feedback from the biased model.



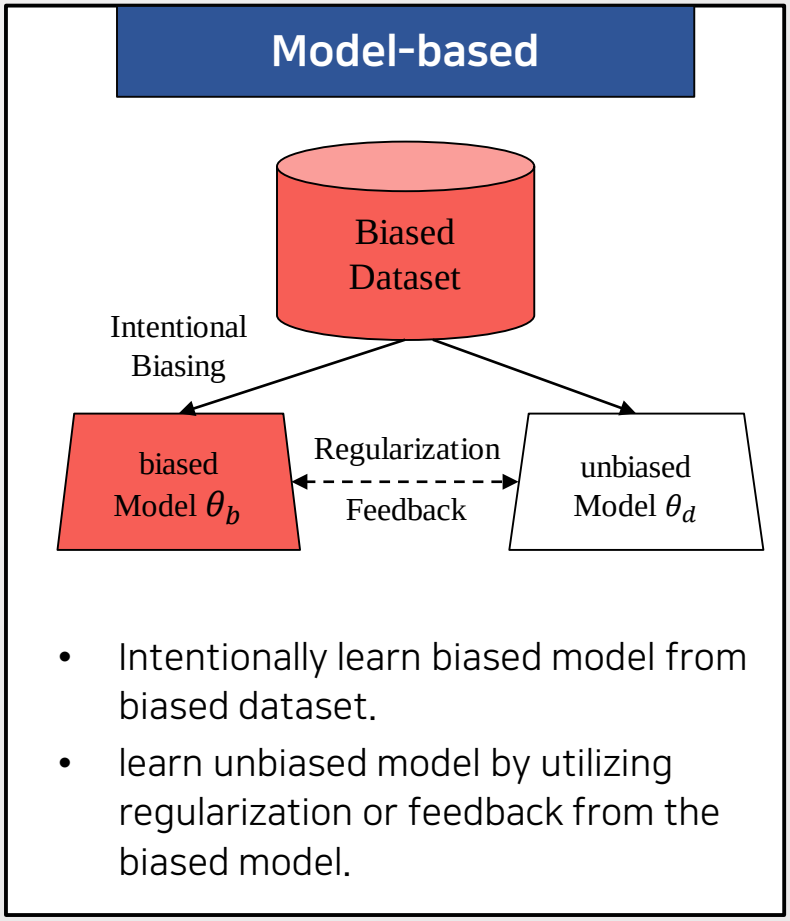

- Minimum-bias Dataset sub-sampling from the biased data
- By utilizing minimum-bias subset, we can learn (relatively) unbiased model.

# Objective & Notation for model-based debiasing

- Main Objective
  - Get generalizable performances on the unbiased test dataset



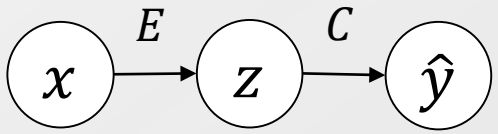

- Model parameter
  - Intentionally Biased model $\theta_b$
  - (Hope to be) de-biased model $\theta_d$
  - Let feature generator $E$ and classifier $C$




- Intentionally learn biased model from biased dataset.
- learn unbiased model by utilizing regularization or feedback from the biased model.

# Learning de-biased representation (ICML 2020)

- Key Point
  - Learning feature independence between biased model parameter $\theta_b$ and de-biased model parameter $\theta_d$
  - Let feature generator $E$ and classifier $C$
  - Let $L_{\theta_d}(X,Y) = L_{\theta_d}(C(E(X)),Y)$




- Whole Objective function
  - Model structure of $\theta_d$ and $\theta_b$ different.
  - $\theta_d$ : debiased model with high capacity, $\theta_b$ : biased model with lower capacity (small parameter)

$$min_{\theta_d} \sum_{(x,y)\in(X,Y)} \left\{ L_{\theta_d}(x,y) + max_{\theta_b} \left( HSIC\left(E_{\theta_b}(x), E_{\theta_d}(x)\right) - L_{\theta_b}(x,y) \right) \right\}$$

- Objective for each model
  - Biased Model $\theta_b$

$$\theta_b^* = argmin_{\theta_b} \sum_{(x,y)\in(X,Y)} \left\{ L_{\theta_b}(x,y) - HSIC\left(E_{\theta_b}(x), E_{\theta_d}(x)\right) \right\}$$

  - Debiased Model $\theta_d$

$$\theta_d^* = argmin_{\theta_d} \sum_{(x,y)\in(X,Y)} \left\{ L_{\theta_d}(x,y) + HSIC\left(E_{\theta_b}(x), E_{\theta_d}(x)\right) \right\}$$

# Learning from Failure (NeurIPS 2020)

- Key Point
  - Intentionally train biased model $\boldsymbol{\theta_b}$. Let de-biased model parameter $\theta_d$ to focus on the difficult data sample from model $\theta_b$
  - There're no interaction feedback from $\boldsymbol{\theta_d}$ to $\theta_b$

- Objective for each model
  - Biased Model $\theta_b$
    - update model parameter by generalized cross entropy. It acquires more gradient for heavily assigned class probability.

$$\theta_b^* = argmin_{\theta_b} \sum_{(x,y)\in(X,Y)} GCE_{\theta_b}(X,Y)$$

$$\text{Update } \theta_b \rightarrow \theta_b - \nabla_{\theta_b} \sum_{(x,y)\in(X,Y)} GCE_{\theta_b}(x,y)$$

$$\frac{\partial \mathrm{GCE}(p,y)}{\partial \theta} = p_y^q \frac{\partial \mathrm{CE}(p,y)}{\partial \theta},$$

  - Debiased Model $\theta_d$
    - Re-weight gradient of corresponding data by relative difficulty scores.

⇔ biased model이 어려워하는 데이터일수록 gradient 값을 크게 주겠다.

$$\text{Update } \theta_d \rightarrow \theta_d - \nabla_{\theta_d} \sum_{(x,y)\in(X,Y)} W(x) CE_{\theta_d}(X,Y)$$

$$W(x) = \frac{CE_{\theta_b}(x,y)}{CE_{\theta_b}(x,y) + CE_{\theta_d}(x,y)}$$

# Problem of previous work

- Repair (CVPR 2019) 요약
  - It only utilize the bias-minimum subset of whole datasets.
  - The downside of under-sampling is that it discards a large portion of the data.
    - From Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss, NeurIPS 2019

- Rebias (ICML 2020) 요약
  - Biased information 과의 independency를 최소화하는 방향.

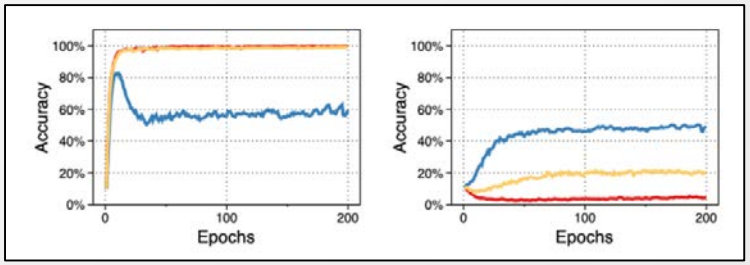
  ⇔ Bias 정보를 최대한 안 보도록 하자.
  - Feature간의 independency를 모델링하는 과정에서 important feature leakage에 대한 문제 가능성.

- Learning from Failure (NeurIPS 2020) 요약
  - Biased 된 데이터에 대한 Gradient를 작게 받고, Unbiased 데이터에 대한 Gradient를 크게 받자.

  ⇔ Biasedness에 따라 Gradient 크기를 조절하자.
  - weighted gradient update tends to cause unstable training.
    - Balanced Meta-Softmax for Long-Tailed Visual Recognition, NeurIPS 2020
  - By inducing small gradient for biased data, it conveys same problem as Repair. (similar effect as discarding a large portion of the data)

# Motivation

- Toward regularizing bias, not removing bias.
  - 이전의 연구는 1) Bias 정보를 removing 하거나 2) Bias가 있는 데이터에 대한 활용을 제한하거나 (ex : removing the biased data, decreasing the gradient of the biased data) 하는 식으로 문제를 해결함.
  - Test data의 분포에 따라, Bias 정보는 recognition에 도움을 주는 존재일 수도 있다. (except for the social biases.)



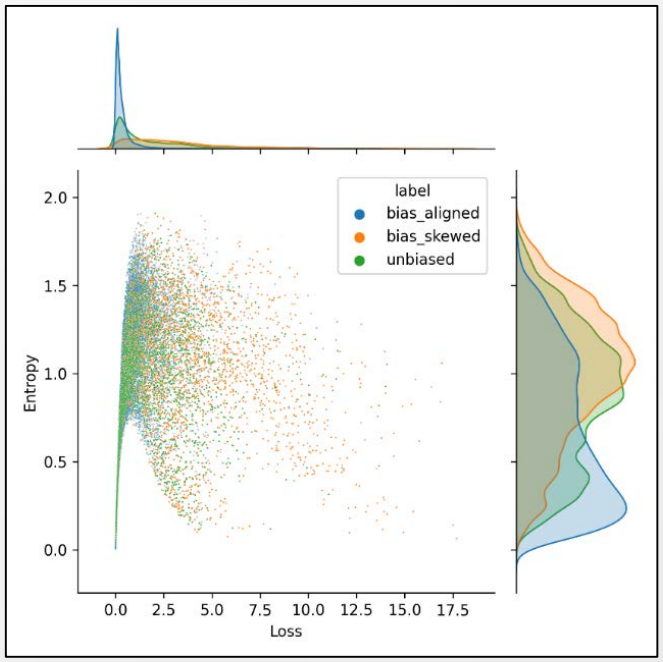

biased data unbiased data

  - Vanilla (주황)이 그냥 모델, LfF (블루)가 bias를 위한 모델일 때 LfF는 unbiased data에 대한 예측력을 높이는 과정에서 biased data에 대한 예측력이 크게 손실됨.
    - Robust to unbiased data, however, ironically it is not robust to biased data.

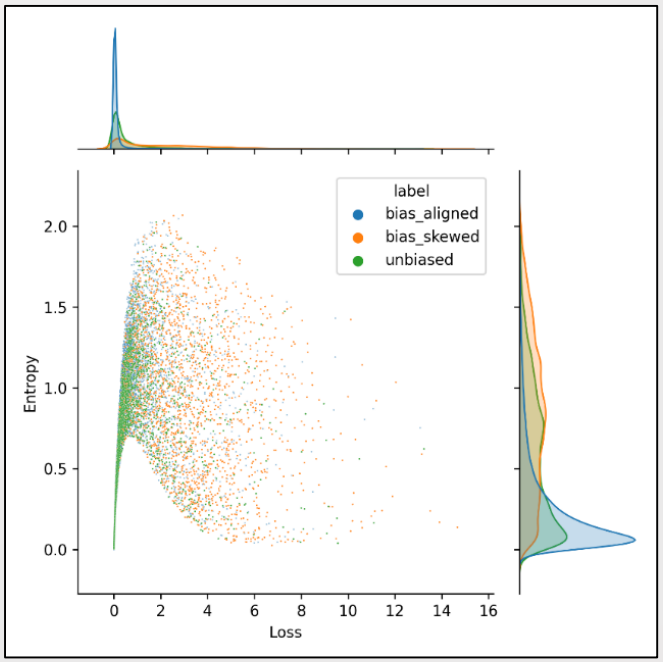
- Direction
  - Bias를 없애지 않고 적절한 규제를 하는 것 만으로도 Biased / Unbiased 둘 다 잘하는 모델을 만들 수 있지 않을까?

# 무엇을 규제할 것인가?

- **Biased data tend to be too easy (low loss) and over-confident (low entropy) than unbiased data**
  - Colored MNIST with Multi-Layered Perceptron (MLP) model
  - Bias-aligned : Biased data
  - Bias-skewed, unbiased : Unbiased data




From model of Epoch 2




From model of Epoch 200

# IDEA_SKETCH 1 : Grouped w.r.t bias

- Relative Regularization
  - Bias를 무조건적으로 배제하기 보다는, 상대적으로 Unbiased된 녀석들이 biased 된 녀석들보다 학습이 잘 되도록 규제를 해줄 순 없을까?

- If we can partition each group with respect to the biasedness, how about regularizing the loss and confidence of each group in relative manner?

- IDEA SKETCH : Group-based loss and confidence regularization
  - Let the whole dataset $D = \sum_{m=1}^{M} D_m$
    - Assume that whole groups are sorted based on the biasedness : $D_1 \geq D_2 \ldots \geq D_M$
    - Let the confidence of each group $m$ as $E_{(x,y)\in D_m}[std(p(y|x;\theta)]$
      - Standard deviation of softmax output
      - Or we can define it as entropy or target class probability.

$$min_\theta \sum_m \sum_{(x,y)\in D_m} l(f_\theta(x), y)$$
$$\text{s.t. } E_{(x,y)\in D_1} l(f_\theta(x), y) \geq E_{(x,y)\in D_2} l(f_\theta(x), y) \geq \cdots \geq E_{(x,y)\in D_M} l(f_\theta(x), y)$$
$$\text{s.t. } E_{(x,y)\in D_1}[std(p(y|x;\theta)] \geq E_{(x,y)\in D_2}[std(p(y|x;\theta)] \geq \cdots \geq E_{(x,y)\in D_M}[std(p(y|x;\theta)]$$

# IDEA SKETCH 1

- IDEA SKETCH : Group-based loss and confidence relative regularization
  - Let the whole dataset $D = \sum_{m=1}^{M} D_m$
    - Assume that whole groups are sorted based on the biasedness : $D_1 \geq D_2 \ldots \geq D_M$
    - Let $E_{(x,y)\in D_m} l(f_\theta(x), y)$ as $D_m(loss)$

$$min_\theta \sum_m \sum_{(x,y)\in D_m} l(f_\theta(x), y)$$
$$\text{s.t. } E_{(x,y)\in D_1} l(f_\theta(x), y) \geq E_{(x,y)\in D_2} l(f_\theta(x), y) \geq \cdots \geq E_{(x,y)\in D_M} l(f_\theta(x), y)$$
$$\text{s.t. } E_{(x,y)\in D_1} [std(p(y|x;\theta)] \geq E_{(x,y)\in D_2} [std(p(y|x;\theta)] \geq \cdots \geq E_{(x,y)\in D_M} [std(p(y|x;\theta)]$$

1. First constraint : Loss Sorting ⇔ margin-based variance minimization
2. Second constraint : Confidence Sorting ⇔ group-dependent label smoothing

$$min_\theta \sum_m \sum_{(x,y)\in D_m} l(f_\theta(x), y^m) + \lambda \boldsymbol{Var(D_1(loss) - \beta_1, D_2(loss) - \beta_2, \ldots, D_M(loss) - \beta_M)}$$

When $\boldsymbol{\beta_1 \geq \beta_2 \geq \cdots \geq \beta_M}$ and

$$\boldsymbol{y_k^m = \begin{pmatrix} y_k(1-\alpha_m) + \frac{\alpha_m}{k} & if\ k = True\ class\ index \\ \frac{\alpha_m}{k} & else \end{pmatrix}} \quad \boldsymbol{\text{with } \alpha_1 \geq \alpha_2 \geq \alpha_3 \ldots \geq \alpha_M}$$

# Hyperparameter 너무 많은데, 어떻게 설정함?

- Original Loss

$$min_\theta \sum_m \sum_{(x,y)\in D_m} l(f_\theta(x), y^m) + \lambda Var(D_1(loss) - \beta_1, D_2(loss) - \beta_2, \dots, D_M(loss) - \beta_M)$$

When $\boldsymbol{\beta_1 \geq \beta_2 \geq \cdots \geq \beta_M}$ and

$$y_k^m = \begin{pmatrix} y_k(1-\alpha_m) + \frac{\alpha_m}{k} & if\ k = True\ class\ index \\ \frac{\alpha_m}{k} & else \end{pmatrix} \text{ with } \boldsymbol{\alpha_1 \geq \alpha_2 \geq \alpha_3 \dots \geq \alpha_M}$$

- Label Smoothed loss as a lower bound of Original Loss with cross-entropy
  - Only when we focus on true class

Let $k$ be the true class index and assume that $f_\theta(x)_k \leq \boldsymbol{y_k^m}$ for $\forall\ (x,y) \in D_m$,
Then $E_{(x,y)\in D_m} l(f_\theta(x)_k, y_k^m) \leq E_{(x,y)\in D_m} l(f_\theta(x)_k, y_k)$

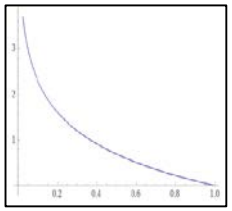
let $\boldsymbol{\beta_m = E_{(x,y)\in D_n} l(f_\theta(x)_k, y_k) - E_{(x,y)\in D_n} l(f_\theta(x)_k, y_k^m)}$
$l(f_\theta(x)_k, y_k) - l(f_\theta(x)_k, y_k^m) = -\log f_\theta(x)_k\,(1 - y_k^m)$ when loss function is cross-entropy.

We need to show the validation of first assumption
$\beta_m = E_{(x,y)\in D_m}[-\log f_\theta(x)_k\,(1-y_k^m)] \geq E_{(x,y)\in D_{m+1}}[-\log f_\theta(x)_k\,(1-y_k^{m+1})]$
$= \beta_{m+1}$

그러한 경향성은 확실히 있으나, 증명 필요한 상태.

$0 \leq -\log f_\theta(x)_k$

lower $f_\theta(x)_k$,
higher $-\log f_\theta(x)_k$



when $\alpha_m \geq \alpha_{m+1}$

$$y_k^m = y_k(1-\alpha_m) + \frac{\alpha_m}{k}$$
$$\leq y_k(1-\alpha_{m+1}) + \frac{\alpha_{m+1}}{k} = y_k^{m+1}$$

Then,
$1 - y_k^m \geq 1 - y_k^{m+1}$

# IDEA SKETCH with lower numbers of hyper-parameter

- Variance Minimization by approximated label smoothing loss

Let $k$ be the true class index and assume that $f_\theta(x)_k \leq y_k^m$ for $\forall\ (x,y) \in D_m$,
Then $E_{(x,y)\in D_m} l(f_\theta(x)_k, y_k^m) \leq E_{(x,y)\in D_m} l(f_\theta(x)_k, y_k)$
let $\beta_m = E_{(x,y)\in D_n} l(f_\theta(x)_k, y_k) - E_{(x,y)\in D_n} l(f_\theta(x)_k, y_k^m)$

  - By utilizing the above, we just minimize the variance from true class label smoothed losses from each group. We don't have to calculate the original loss for variance minimization and don't have to set the margin for each group.

$$Var\left(E_{(x,y)\in D_1} l(f_\theta(x)_k, y_k) - \beta_1, E_{(x,y)\in D_2} l(f_\theta(x)_k, y_k) - \beta_2, \dots, E_{(x,y)\in D_M} l(f_\theta(x)_k, y_k) - \beta_M\right)$$
$$\Leftrightarrow$$
$$Var\left(E_{(x,y)\in D_1} l\left(f_\theta(x)_k, y_k^1\right), E_{(x,y)\in D_2} l\left(f_\theta(x)_k, y_k^2\right), \dots, E_{(x,y)\in D_M} l\left(f_\theta(x)_k, y_k^M\right)\right)$$

- Final Loss based on Group
  - The hyperparameter is $\lambda$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_M]$

$$min_\theta \sum_m \sum_{(x,y)\in D_m} l(f_\theta(x), y^m) + \lambda Var\left(E_{(x,y)\in D_1} l\left(f_\theta(x)_k, y_k^1\right), E_{(x,y)\in D_2} l\left(f_\theta(x)_k, y_k^2\right), \dots, E_{(x,y)\in D_m} l(f_\theta(x)_k, y_k^m)\right)$$

$$\text{When } y_k^m = \begin{pmatrix} y_k(1-\alpha_m) + \frac{\alpha_m}{k} & if\ k = True\ class\ index \\ \frac{\alpha_m}{k} & else \end{pmatrix} \text{ with } \alpha_1 \geq \alpha_2 \geq \alpha_3 \dots \geq \alpha_M$$

# 여기까지...

$$min_\theta \sum_m \sum_{(x,y)\in D_m} l(f_\theta(x), y^m) + \lambda Var\left(E_{(x,y)\in D_1} l(f_\theta(x)_k, y_k^1), E_{(x,y)\in D_2} l(f_\theta(x)_k, y_k^2), \dots, E_{(x,y)\in D_m} l(f_\theta(x)_k, y_k^m)\right)$$

$$\text{When } y_k^m = \begin{pmatrix} y_k(1-\alpha_m) + \frac{\alpha_m}{k} & if\ k = True\ class\ index \\ \frac{\alpha_m}{k} & else \end{pmatrix} \text{ with } \alpha_1 \geq \alpha_2 \geq \alpha_3 \dots \geq \alpha_M$$

- What is the novelty?
  - 위의 Loss를 통해 각 Group별 Loss 및 Confidence에 대한 상대적인 고저 regularization이 가능하다. (In approximated way)
  - ⇔ Group별 Relative Regularization이 가능하다.

- 문제점 #1. Group 어떻게 만들지?
  - 그룹을 무슨 기준으로 나눌래? / 그룹을 몇 개로 나눌래? / 그룹을 어떤 비율로 나눌래?
  - 그룹별 hyper-parameter $[\alpha_1, \alpha_2, \dots, \alpha_M]$ 어떻게 설정해줄건데?
  - 그렇다면 그룹을 안 만들고 위의 방법론을 활용할 수 있는 방법은?

"Group 만드는 과정에서 문제점이 너무 많아...
Group 안 만드는 방향으로 고민해보자."

# mini-batch based relative regularization

- 똑같은 방식을 Group이 아닌 Mini-batch내의 데이터 각각에 해줄 수는 없을까? (Not Group-based, Data instance-based)
  - Group이 아닌 각각의 data에 대한 formulation도 본래 식에 입각하여 어렵지 않게 구성할 수 있다.
  - How about class-wise biasedness computation?

⇔ 클래스 별로 각기 다른 biasedness를 가질 것이다. 모든 데이터를 같은 Scale에서 비교하는 것이 올바른가?

- Mini-batch based relative regularization
  - Let the mini-batch = $\sum_c \sum_{i=1}^{M_c} \{x_i, y_i\}$ decomposed by class
  - Then for each class-wise mini-batch data,

$y_i$ be the original label of $x_i$ and Let $\hat{y}_i$ be smoothed label by input-dependent smoothing factor $\alpha_i$

$$min_\theta \sum_{i=1}^{M_c} (f_\theta(x_i), \hat{y}_i) + \lambda Var\left(l\left(f_\theta(x_1)_k, \hat{y}_{1,k}\right), l\left(f_\theta(x)_k, \hat{y}_{2,k}\right), \dots, l\left(f_\theta(x)_k, \hat{y}_{M_c,k}\right)\right)$$

$$\text{When } \hat{y}_{i,k} = \begin{pmatrix} y_{i,k}(1-\alpha_i) + \frac{\alpha_i}{k} & if\ k = True\ class\ index \\ \frac{\alpha_i}{k} & else \end{pmatrix} \text{ with } \alpha_1, \alpha_2, \alpha_3 \dots, \alpha_{M_c}$$

- In this formulation, we still need to compute
  - Input-dependent label smoothing factor $\alpha_1, \alpha_2, \alpha_3 \dots, \alpha_{M_c}$

# Input dependent label smoothing factor

- **Determine the class-wise biasedness based on the response of biased model**
  - The response can be loss or entropy.
  - Like LfF, utilize the generalized cross entropy loss.
  - One more assumption : The model early-learns biased data rather than the unbiased data
    - Early-Learning Regularization Prevents Memorization of Noisy Labels (NeurIPS 2020)
    - Learning from Failure (NeurIPS 2020)

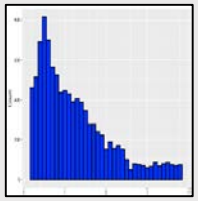
⇔ For biasedness computation, utilize EMA (Exponential Mean Average).

If we choose 'loss' to be biasedness response,

$$B(x_i)_t = \alpha B(x_i)_{t-1} + (1-\alpha) l(f_{\theta_t}(x_i), y_i)$$

depending on the biasedness, get Input-dependent label smoothing factor

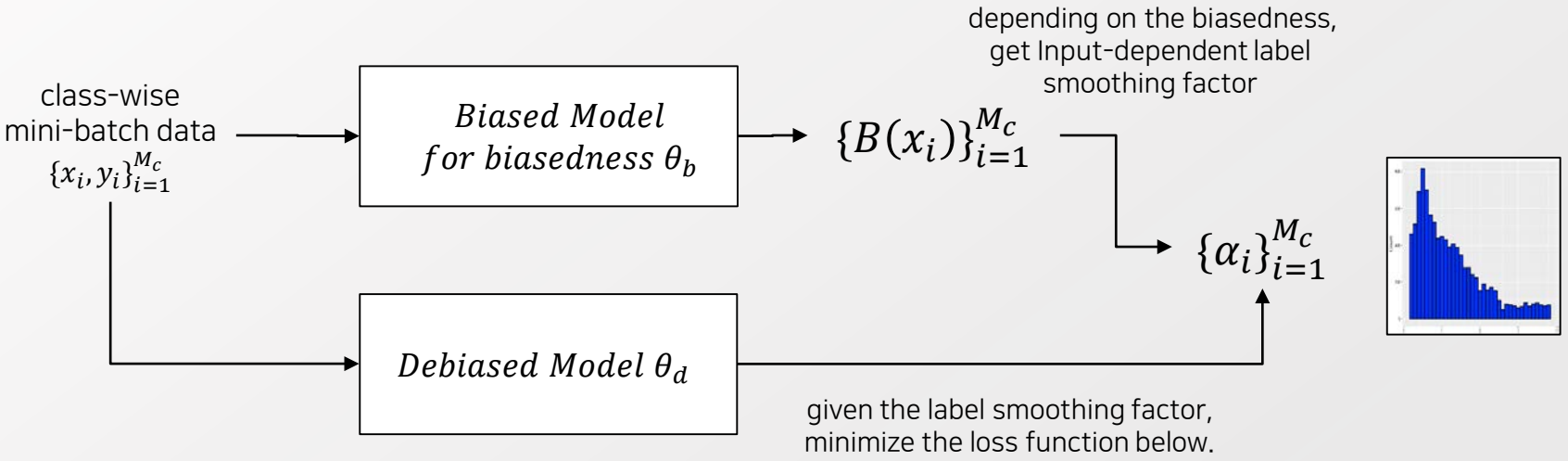
class-wise mini-batch data $\{x_i, y_i\}_{i=1}^{M_c}$ → *Biased Model for biasedness* $\theta_b$ → $\{B(x_i)\}_{i=1}^{M_c}$ → $\{\alpha_i\}_{i=1}^{M_c}$



- **To get input-dependent label smoothing factor in class-wise manner,**
  - Normalize the $\{B(x_i)\}_{i=1}^{D_c}$ into [0,1], get $\{\hat{B}(x_i)\}_{i=1}^{D_c}$
  - Let $\alpha_i = \max(\alpha) - \max(\alpha) * \hat{B}(x_i)$
    - Loss가 클수록 Unbiased 되어 있기 때문!

# Whole Structure from mini-batch setting

- Graphical Figure in mini-batch setting



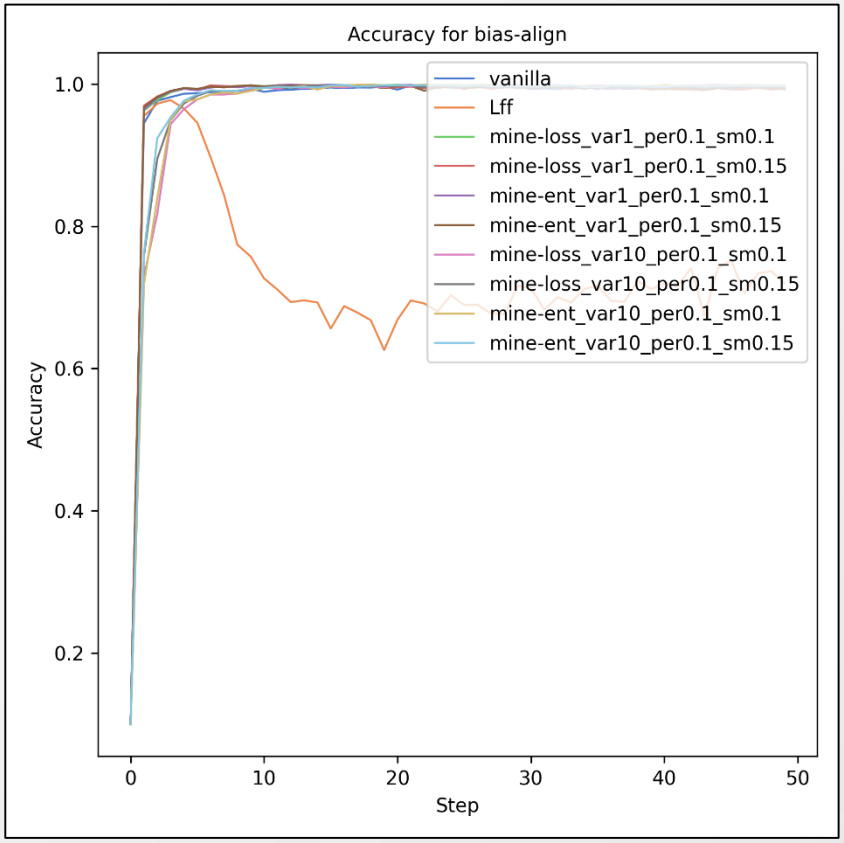

$y_i$ be the original label of $x_i$ and Let $\hat{y}_i$ be smoothed label by input specific smoothing factor $\alpha_i$

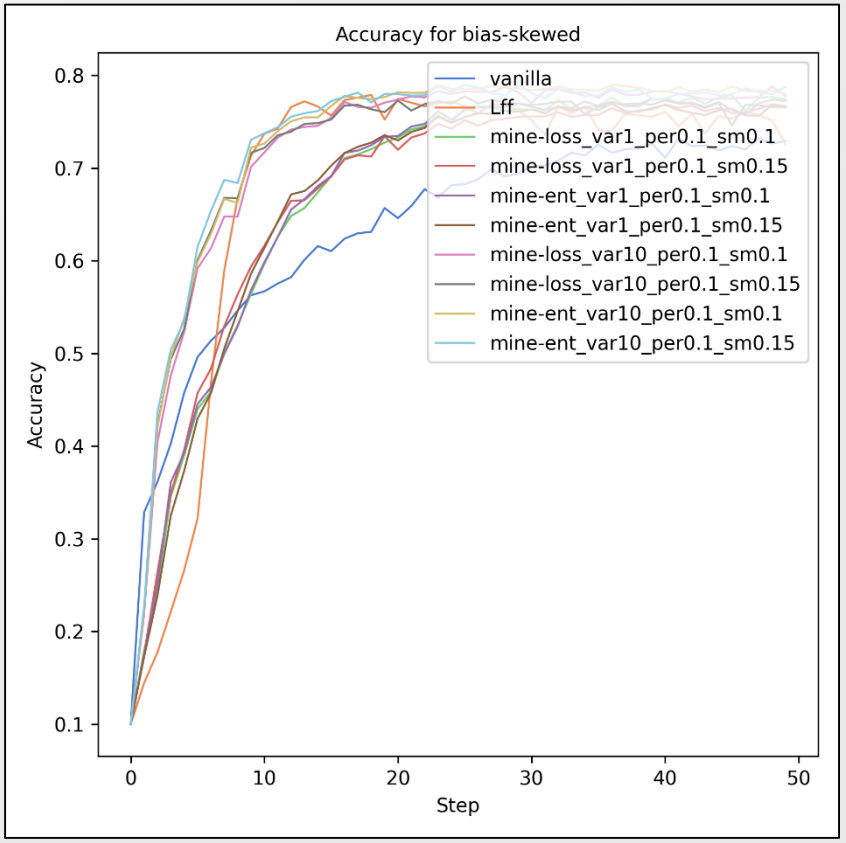
$$min_\theta \sum_c \sum_{i=1}^{M_c} (f_\theta(x_i), \hat{y}_i) + \lambda Var\left(l\left(f_\theta(x_1)_k, \hat{y}_{1,k}\right), l\left(f_\theta(x)_k, \hat{y}_{2,k}\right), \dots, l\left(f_\theta(x)_k, \hat{y}_{M_c,k}\right)\right)$$

$$\text{When } \hat{y}_{i,k} = \begin{pmatrix} y_{i,k}(1-\alpha_i) + \frac{\alpha_i}{k} & if\ k = True\ class\ index \\ \frac{\alpha_i}{k} & else \end{pmatrix} \text{ with } \alpha_1, \alpha_2, \alpha_3 \dots, \alpha_{M_c}$$

# Results (Colored MNIST)

- Colored MNIST with 0.05 unbiased ratio
  - mine-{biasedness}_var(reg_hyperparam)_per(percentile)_sm(maximum smoothing factor)
  - Vaniila : original model / LfF : Main strong baseline (NeurIPS 2020)



# 현재 실험 진행 상황

- Mini-batch based Regularization
  - Unbiased 비율이 높을 때는 괜찮은 성능

  ⇔ Unbiased 의 비율이 줄어들수록 (최소 0.5%까지) 성능에 그리 좋지 못함.

  ⇔ 어쩌면 당연할수도?

- (Previous) Group-based Regularization
  - 전체 데이터에 대해 Loss 기준으로 정렬, Unbiased ratio 안다고 가정하고 두 Group으로 자름.
  - label smoothing 등 hyper-parameter로 세팅해야할 것들이 많았다. 임의로 세팅했을 때

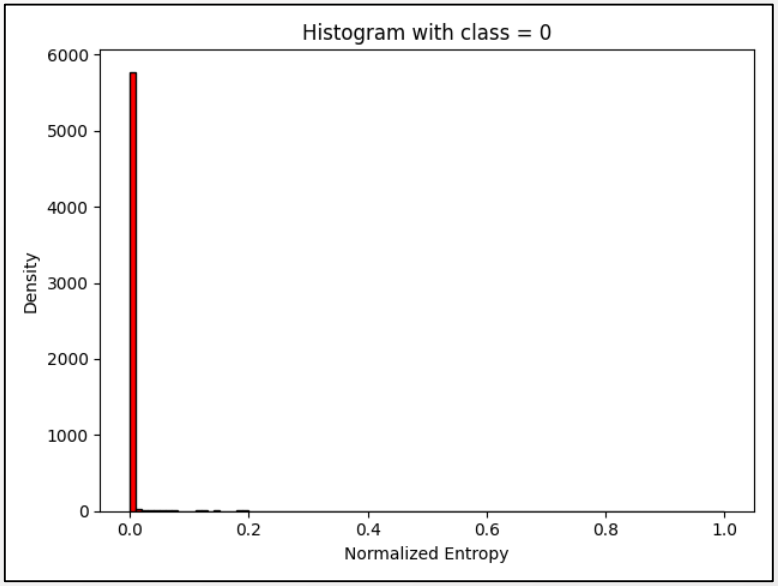
  ⇔ **성능 잘 나옴.**

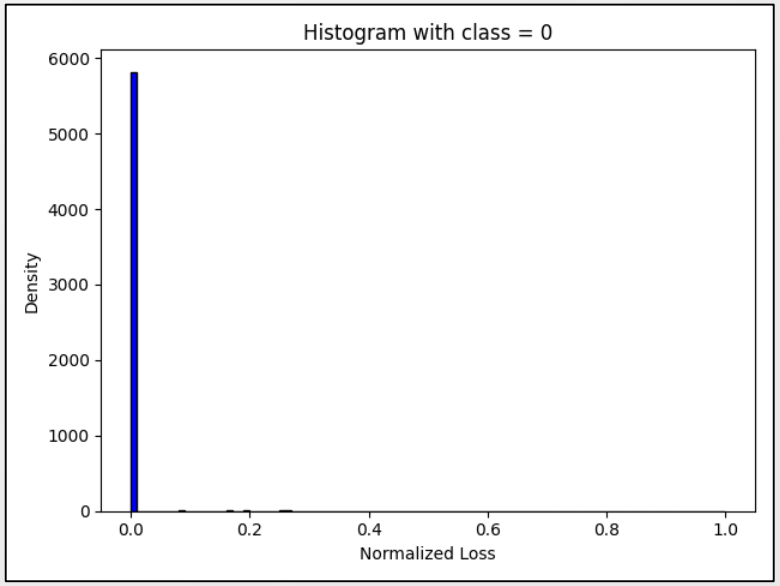
- (Present) Group-based Regularization
  - Class별로 Biased된 정도가 다르다는 가정하에, 각 class별로 적절한 threshold 를 찾아서 잘라줄 수 있다면?
  - Group-specific label smoothing factor 를 적절히 잘 선택해줄 수 있다면?

  ⇔ **성능을 기대해볼 수 있지 않을까?**

# Group Partitioning threshold

- 현 문제는, biased 데이터의 비율이 너무나도 작은 Setting이라는 것
  - 이로 인해, 실제 Biasedness plotting에서도 너무 큰 skewedness를 보임.
  - Unbiased ratio를 모르는 상태에서 어디서 Partition을 해줘야하지...?ㅋㅋㅋ



# Previous Experiments (예전 실험 결과)

- 일단 Grouping 한 상태에 대한 실험
- Experimental Setting
  - Assume that we don't know the biased / unbiased group exactly. However, we assume that we know the unbiased ratio.
  - Learn general model first, partition the whole training dataset into two group based on unbiased ratio only based on the loss. (biased 일수록 loss가 낮다는 것을 가정하여, loss 기준에 따라 일단 두 그룹으로만 나눔.)

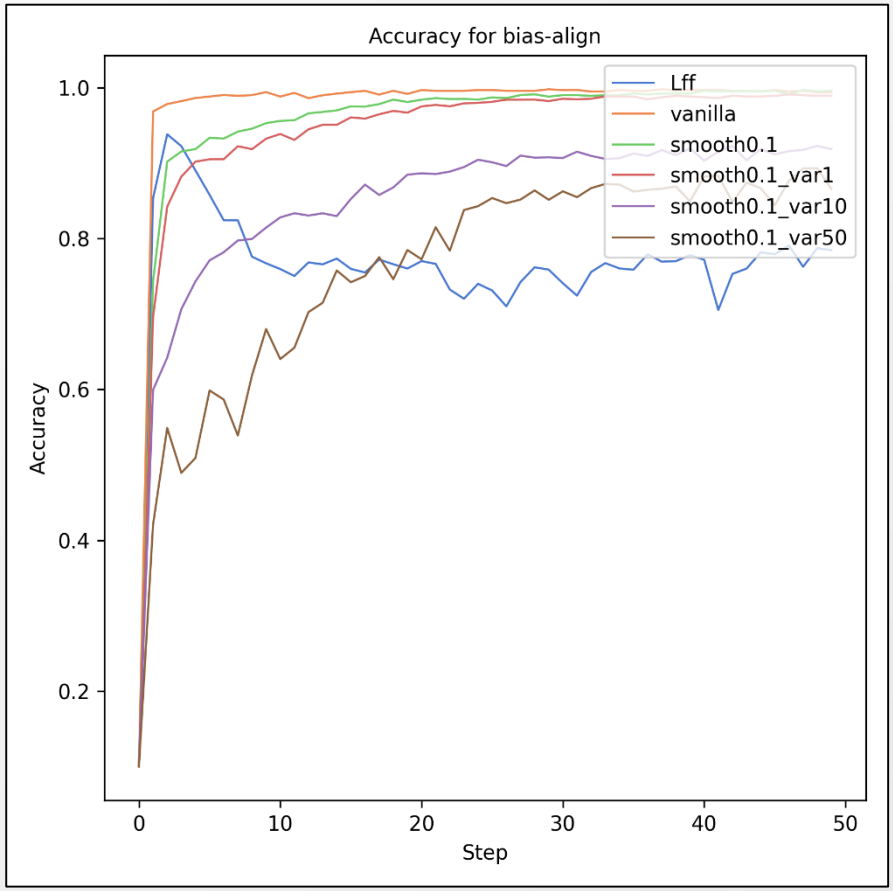
  ⇔ If unbiased ratio = 0.5% -> we divide whole dataset into 99.5% group / 0.5% group only based on loss.
  - Utilize the label smoothing only on the minor group.

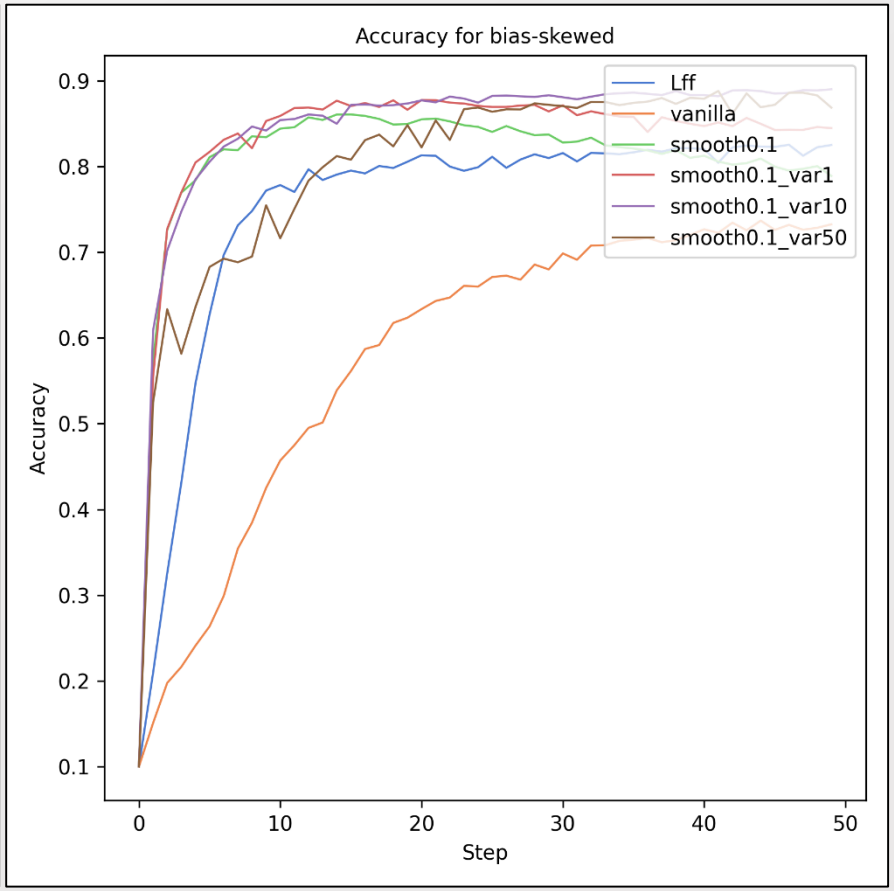
- Dataset
  - Colored MNIST with varying unbiased ratio (0.005 / 0.05)
  - Corrupted Cifar10 with varying unbiased ratio (0.005 / 0.05)

- 목표
  - Unbiased data에 대한 예측력을 높이면서, Biased data에 대한 성능을 최대한 떨어뜨리지 않는 것.
  - 이전 모델에 비해 좋은 성능을 보이는 것.

# Results (Colored MNIST)

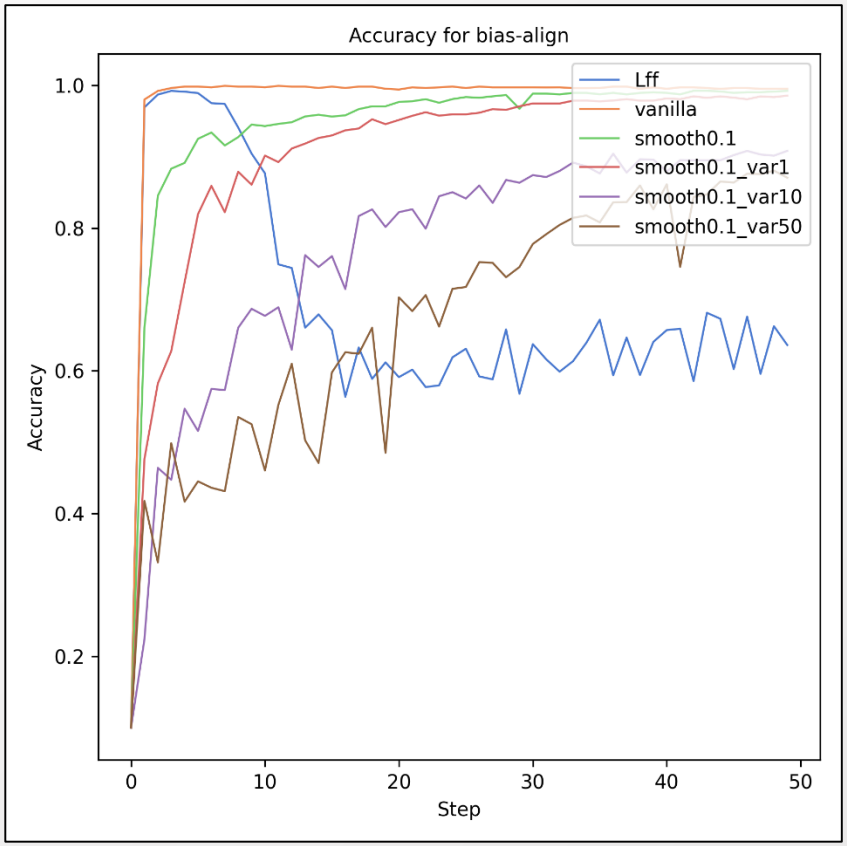
- Colored MNIST with 0.05 unbiased ratio
  - Smooth(label smoothing)_var(reg_hyperparam)
  - Vaniila : original model / LfF : Main strong baseline (NeurIPS 2020)

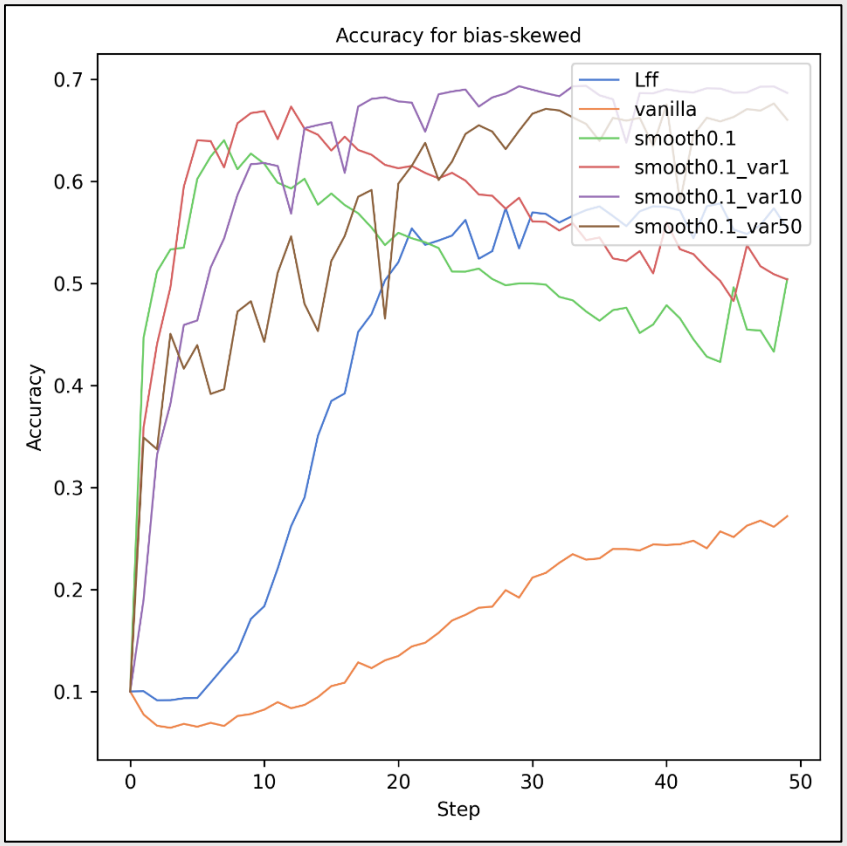


# Results (Colored MNIST)

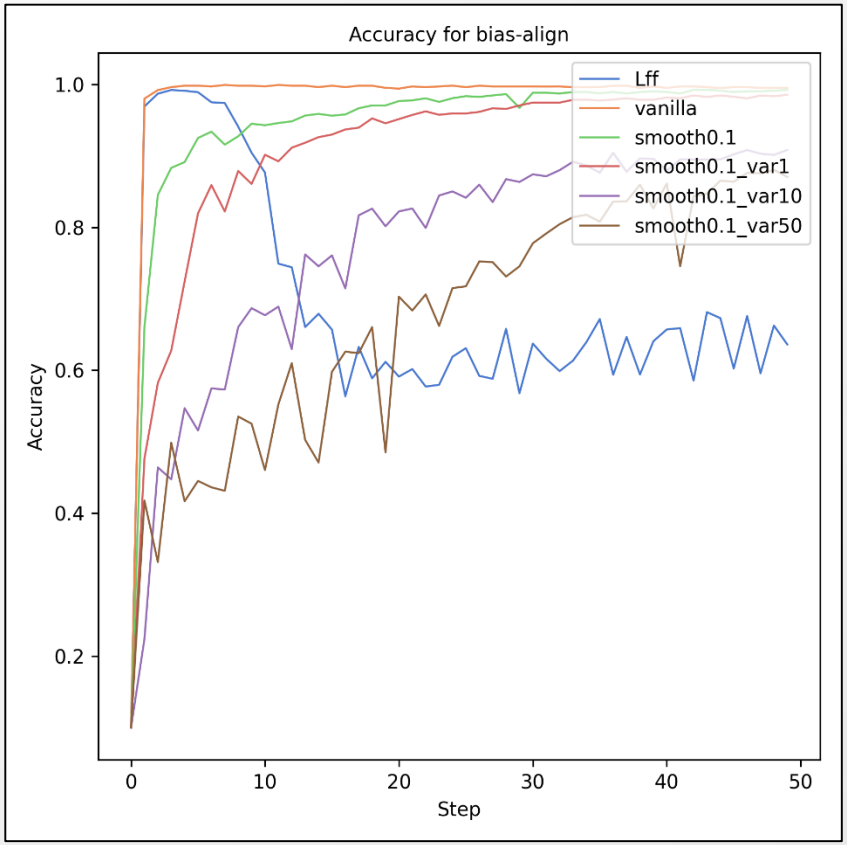
- **Colored MNIST with 0.005 unbiased ratio**
  - Smooth(label smoothing)_var(reg_hyperparam)
  - Vaniila : original model / LfF : Main strong baseline (NeurIPS 2020)
  - Learning from failure 보다 biased / unbiased 둘 다 잘할 수 있다. (bias를 없애지 않고, regularize함으로써)

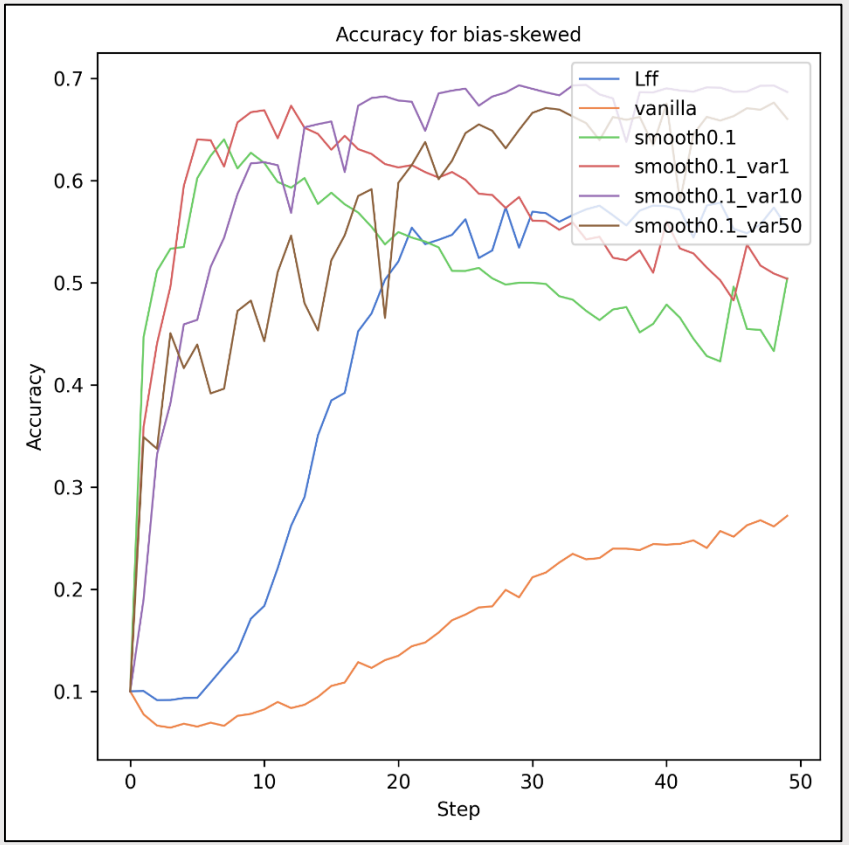


# Results (Corrupted Cifar10)

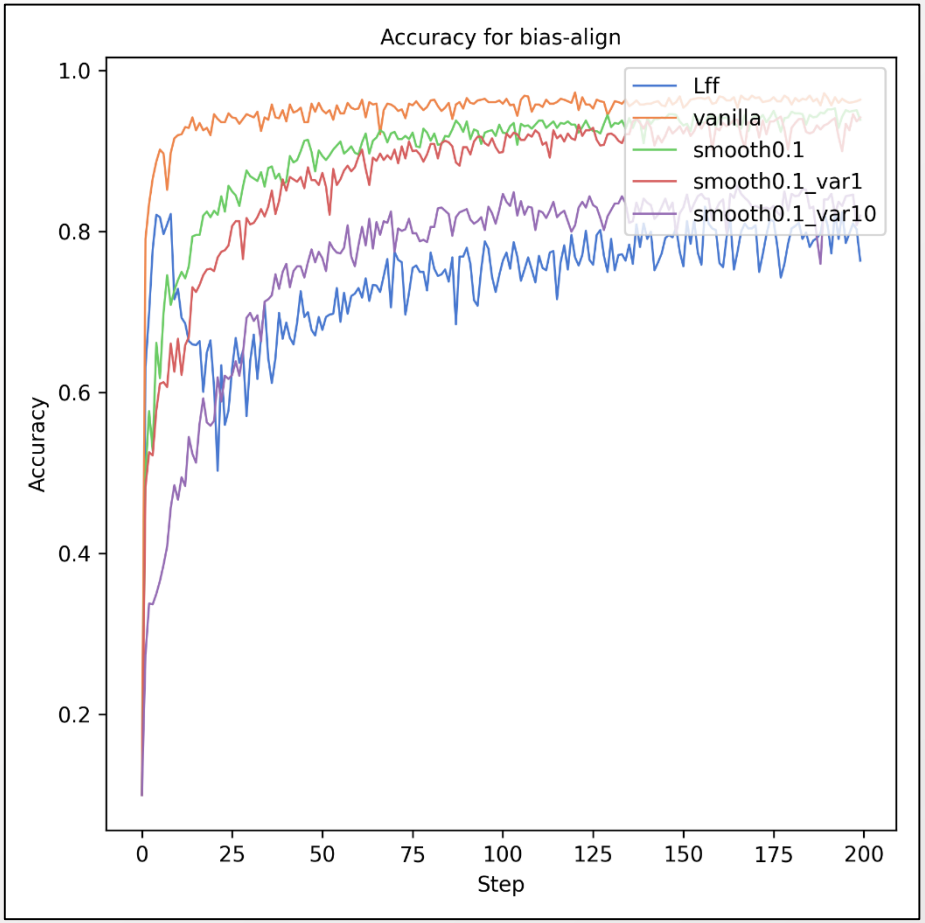
- **Corrupted Cifar10 with 0.005 unbiased ratio**
  - Smooth(label smoothing)_var(reg_hyperparam)
  - Vaniila : original model / LfF : Main strong baseline (NeurIPS 2020)

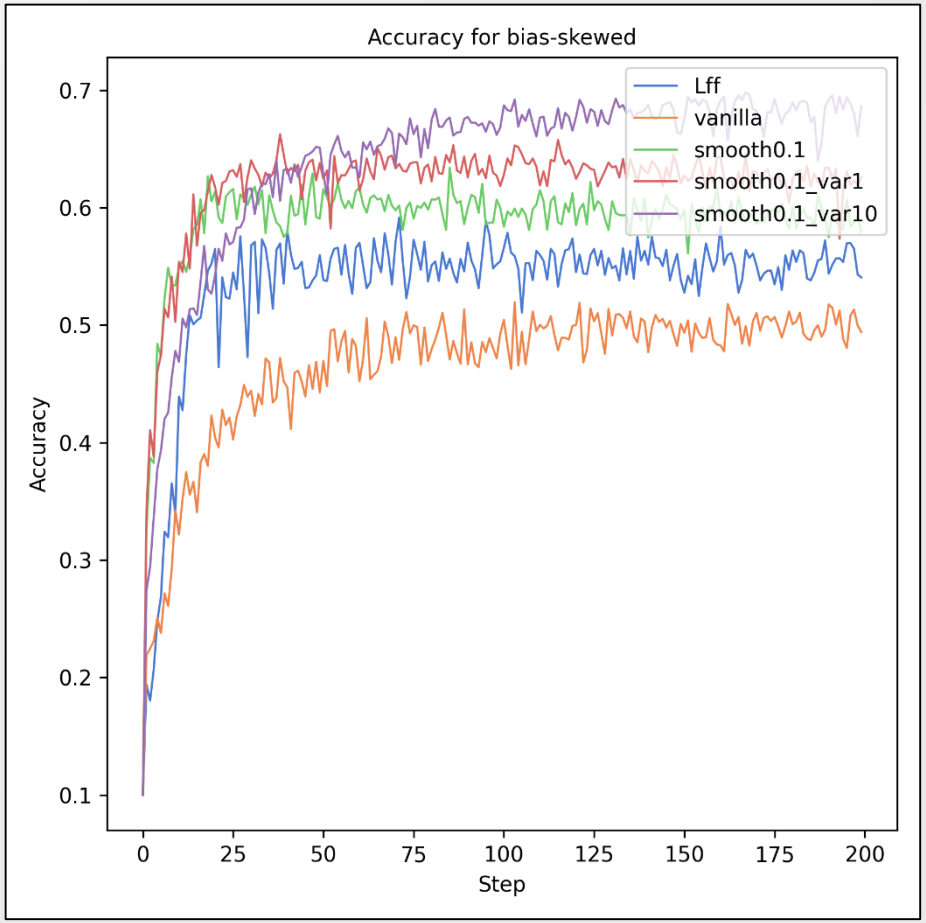


# Results (Corrupted Cifar10)

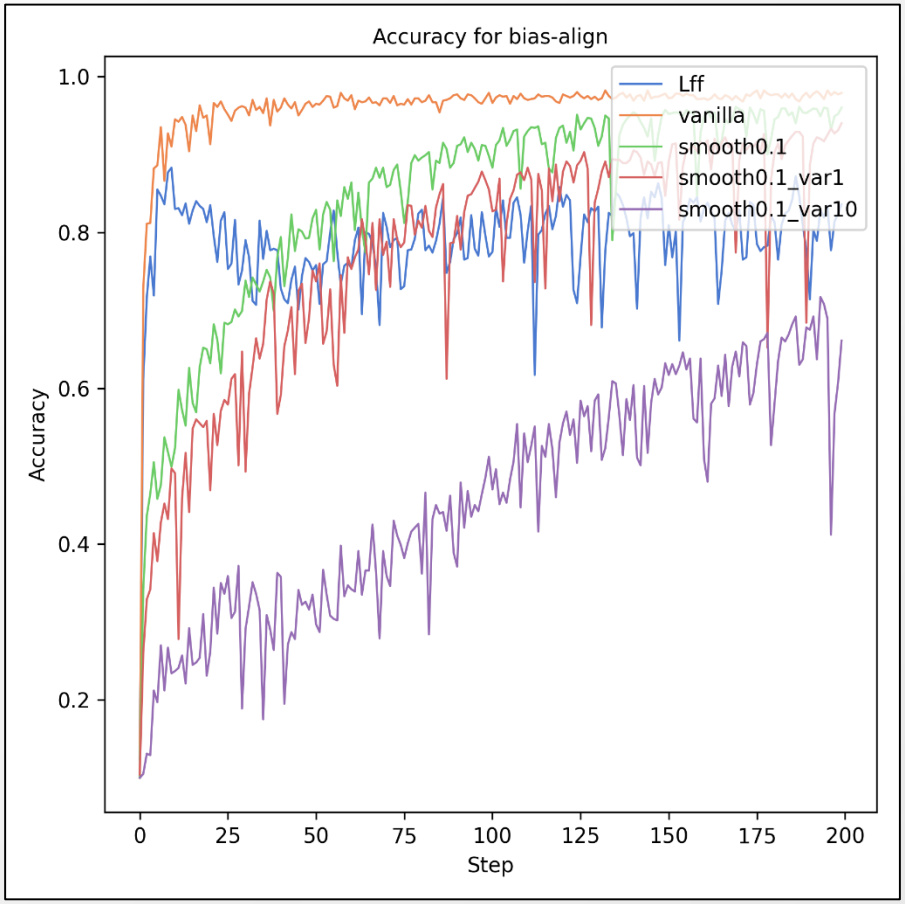
- Corrupted Cifar10 with 0.05 unbiased ratio
  - Smooth(label smoothing)_var(reg_hyperparam)
  - Vaniila : original model / LfF : Main strong baseline (NeurIPS 2020)

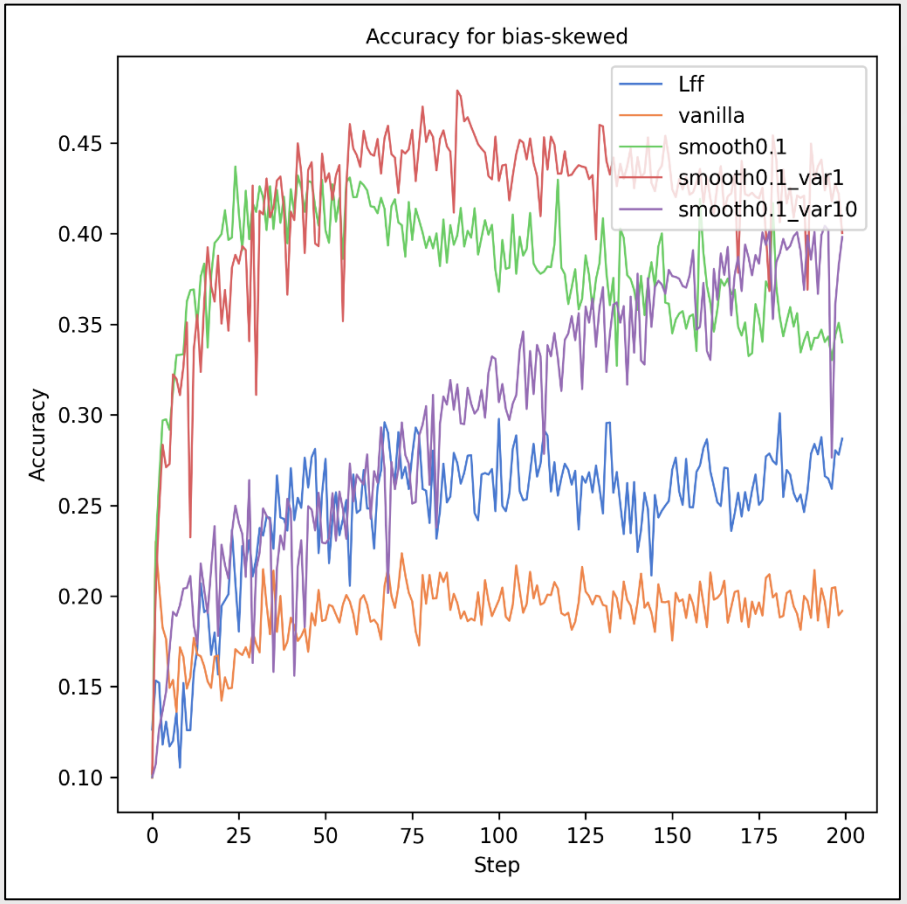


# Results (Corrupted Cifar10)

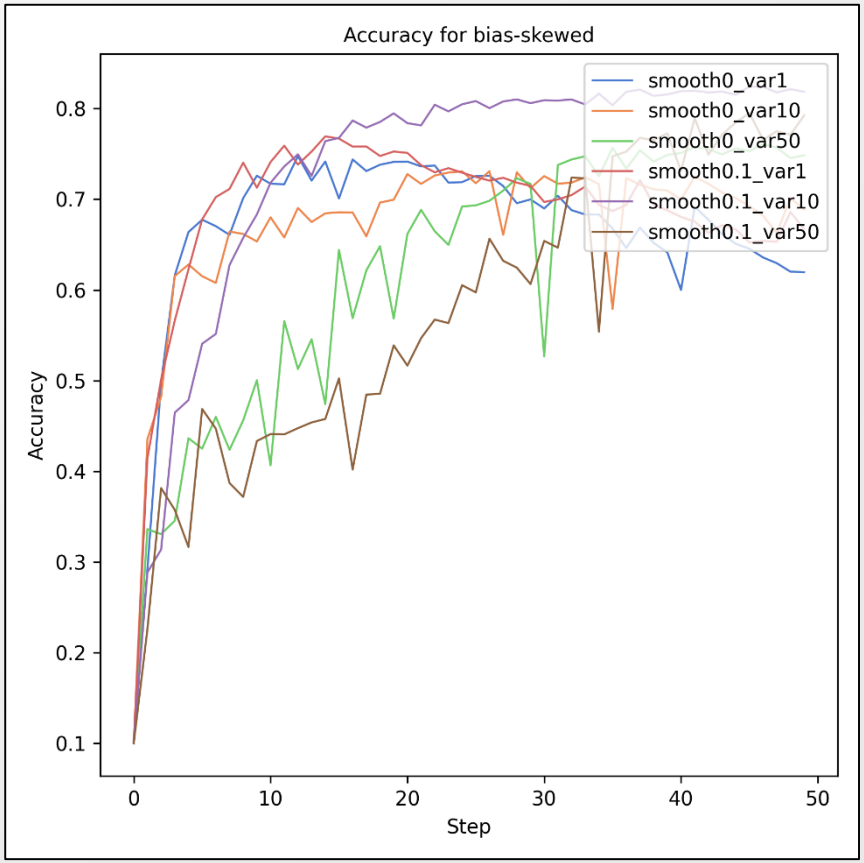
- Corrupted Cifar10 with 0.005 unbiased ratio
  - Smooth(label smoothing)_var(reg_hyperparam)
  - Vaniila : original model / LfF : Main strong baseline (NeurIPS 2020)



# Label smoothing도 효과가 있어?

- with / without label smoothing for confidence regularization
  - 동일한 variance hyperparameter에 대하여, smoothing 을 실시한 결과가 더 좋은 양상을 띔.
  - loss와 confidence 를 함께 regularization해주는 건 각각의 효과가 있다.

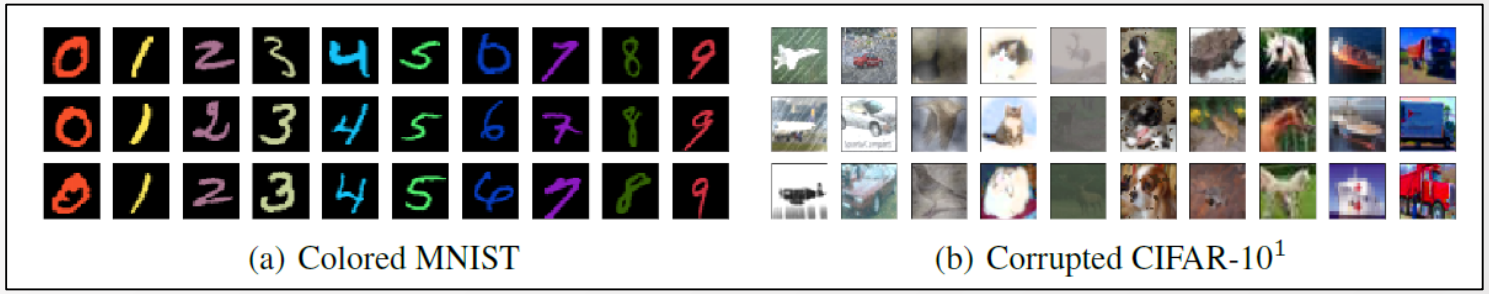


# Experiment Plan

- Baseline
  - Vanilla
  - Hex (ICLR 2019) – 구현 완료 & 비슷한 수준 성능 재현 완료
    - Projecting the model's representation orthogonal to the texture bias by utilizing gray-level co-occurrence matrix
  - GroupDRO (ICLR 2020) - 구현 완료 : 그런데 성능이 재현 안되는 상황
    - Minimizing the worst-case risk from the pre-defined groups
  - Invariant Risk Minimization - 구현중
    - Minimizing the invariant risk from the pre-defined groups
  - Repair (CVPR 2019) - 구현 완료 & 비슷한 수준 성능 재현 완료
    - Learn biasedness of each data and only utilizing minimum biased subset
  - Rebias (ICML 2020) – 구현 완료 & 성능 재현 완료
    - Learning bias-independent feature by modeling statistical independence between biased / unbiased model
  - LfF (NeurIPS 2020) – 구현 완료 & 성능 재현 완료
    - Learning bias-independent feature by modeling statistical independence between biased / unbiased model

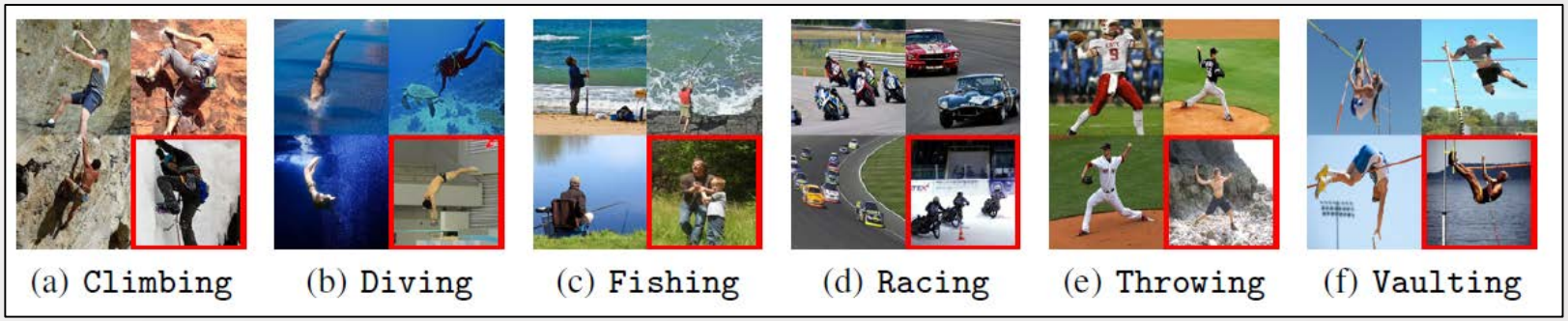
# Experiment Plan

- Artificially Biased Dataset
  - Colored MNIST – 구현 완료
  - Corrupted Cifar10 – 구현 완료



(a) Colored MNIST (b) Corrupted CIFAR-10[1]

- RealWorld Dataset
  - CelebA – Socially biased dataset – 구현 완료
  - Biased Action Recognition – 구현 완료



(a) Climbing (b) Diving (c) Fishing (d) Racing (e) Throwing (f) Vaulting

# Bias 스터디

- **진행 방향**
  - 현 : 김혜미, 송경우 교수님, 신승재, 장준호
  - 주제 및 진행방향
    - Dataset Bias
    - Invariant Representation Learning
    - Out-of-Distribution Generalization
    - Noisy-Label
  - 위의 다양한 분야를 Cover하며 논문 읽어 나가기. 너무 분야가 넓어진다고 느껴질 경우, 일부 Track 삭제 가능.
  - Study 시작 전 범위는 정확히 Define하고 가면 좋을 듯.
  - 논문 리스트 정리
    - Spreadsheet로 정리하는 것이 좋을 듯 합니다.
  - 로드에 무리가 되지 않도록 2주 ~ 3주에 한번 Turn이 오도록 합시다.