Mitigating noisy labels and dataset bias in machine learning

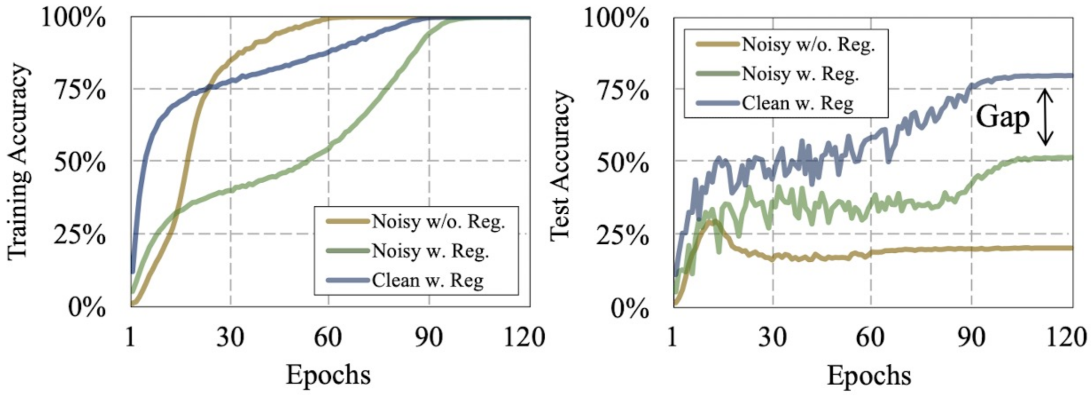
2023.08.25 (Fri.)

Superb AI Machine Learning Team

Presenter : Kyeongryeol, Go

Motivation

- For trustworthy application, robust training matters
- Model is expected to generalize well even under
  - Noisy labels
  - Dataset bias
  - Distribution shifts
  - Adversarial attacks
  - …



Typical regularizations (aug., L2, Dropout, BN) are not enough in the presence of noisy labels.

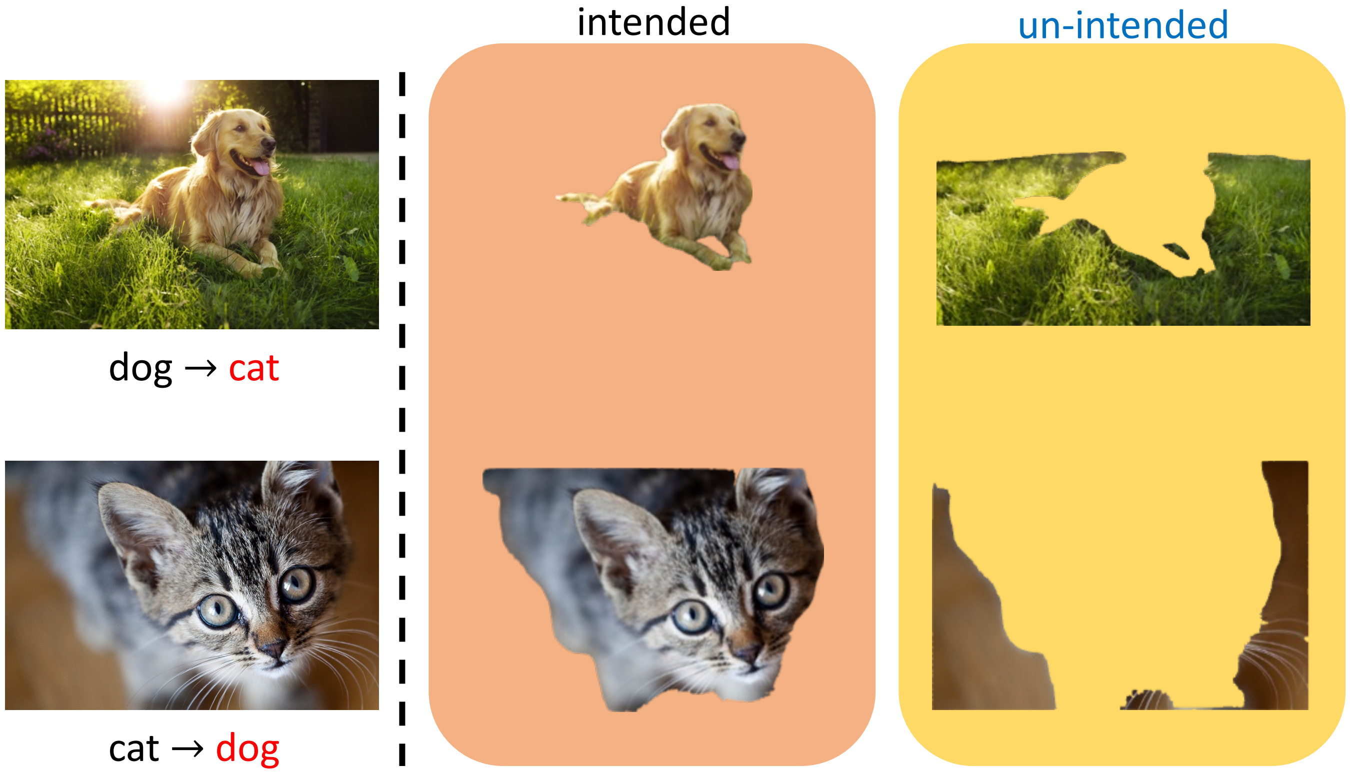


Illustration of noisy labels and dataset bias.

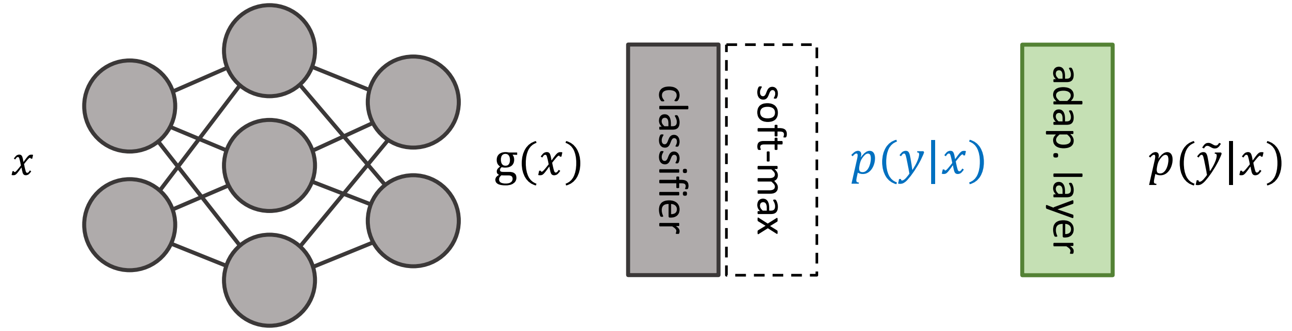
Noisy labels

1. Robust architecture

Let $T_{ij} = p(\tilde{y} = j | y = i, (x))$ be noise transition matrix

(T_{ij} is conditioned on x if label noise is input-dependent. Otherwise, it is input-independent)

- Noise type
  1. *Symmetric* : $\forall_{i=j}, T_{ij} = 1 - \tau \;\; and \;\; \forall_{i \neq j}, T_{ij} = \frac{\tau}{C-1}$
  2. *Asymmetric* : $\forall_{i=j}, T_{ij} = 1 - \tau \;\; and \;\; \exists_{i \neq j, j \neq k, i \neq k}, T_{ij} > T_{ik}$ → human annotation
- Noise adaptation layer
  - T_{ij} is trained to correct the gradient signal from the noisy label (ignored during the inference)
  - $p(\tilde{y} = j | x) = \sum_{i=1}^{C} p(\tilde{y} = j | y = i)\, p(y = i | x) = \sum_{i=1}^{C} T_{ij}\, p(y = i | x)$



(i) asymmetric
(ii) input-independent

For (i) asymmetric and (ii) input-dependent label noise, refer to heteroskedastic layers

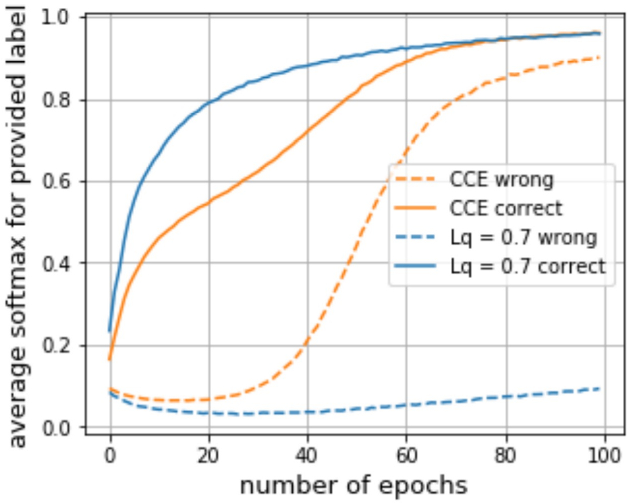
Goldberger, Jacob, and Ehud Ben-Reuven. "Training deep neural-networks using a noise adaptation layer." *International conference on learning representations*. 2016.

2. Robust loss function

- Categorical Cross Entropy (CCE) : $-\log p(y = k|x)$
  - fast convergence, poor generalization in presence of noisy labels
- Mean Absolute Error (MAE) : $|\mathrm{OneHot}(k) - p(y|x)| = 2\big(1 - p(y = k|x)\big)$
  - slow convergence, better generalization in presence of noisy labels
- Generalized Cross Entropy (GCE) : $(1 - p(y = k|x)^q)/q$
  - Consensus of CCE ($q \to 0$) and MAE ($q \to 1$)
  - up-weights the gradient of CCE for the samples of confident prediction on label $(y = k)$

$$\frac{\partial GCE(x,k)}{\partial\theta} = p(y = k|x)^q \frac{\partial CCE(x,k)}{\partial\theta}$$

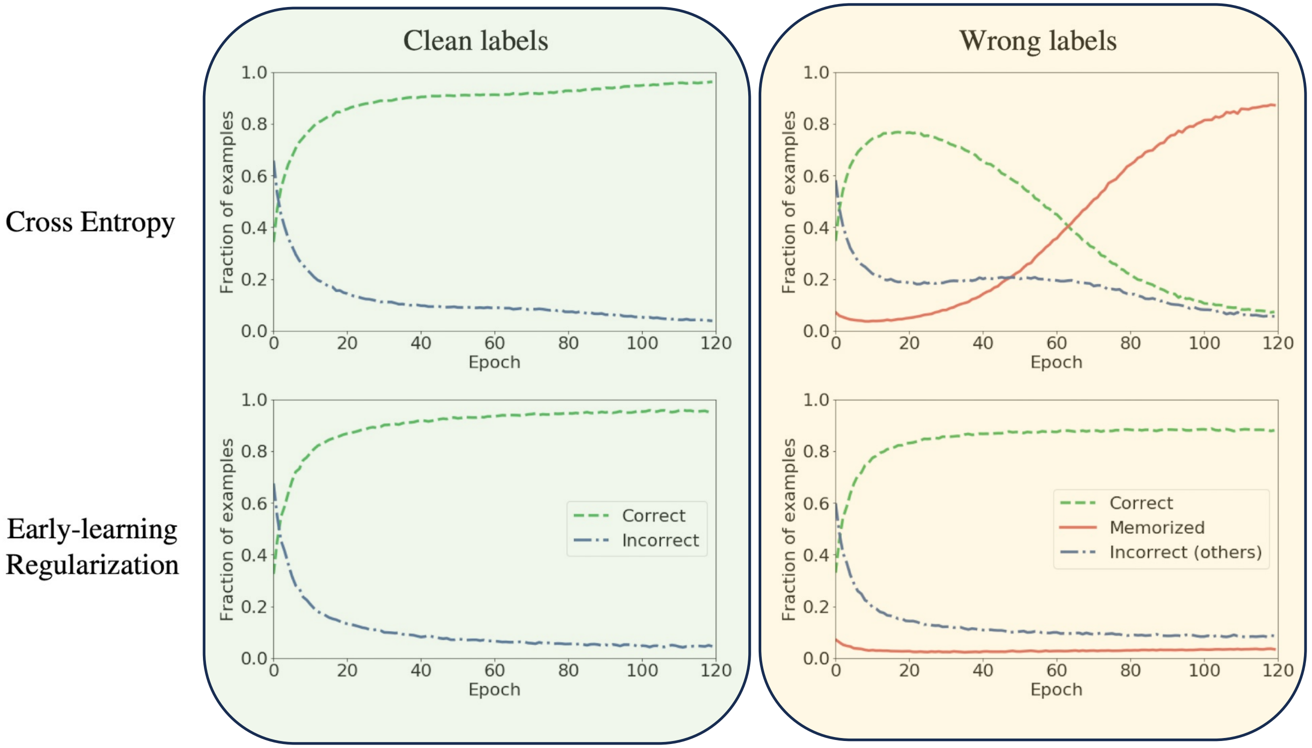
Compared to CCE, GCE compel the wrongly labeled samples to be un-confident.



Zhang, Zhilu, and Mert Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels." *Advances in neural information processing systems* 31 (2018).

3. Robust regularization

- Observation : DNNs tend to fit the clean labels first, then the noisy labels later
- Early Learning Regularization (ELR) : CCE + $\lambda \cdot \log(1 - \langle p(y|x), t(x) \rangle)$
  (for every iteration, $t(x) \leftarrow \beta \cdot t(x) + (1-\beta) \cdot p(y|x)$)
  - maximize the similarity b/t the online prediction $p(y|x)$ and the ema. prediction $t(x)$



$$\nabla S^{-1}(p(y|x))(p(y|x) - \mathrm{OneHot}(y) + \lambda \cdot \mathrm{Grad})$$

Additionally introduced gradient
(i) maintain the gradient of clean labels
(ii) neutralize the gradient of noisy labels

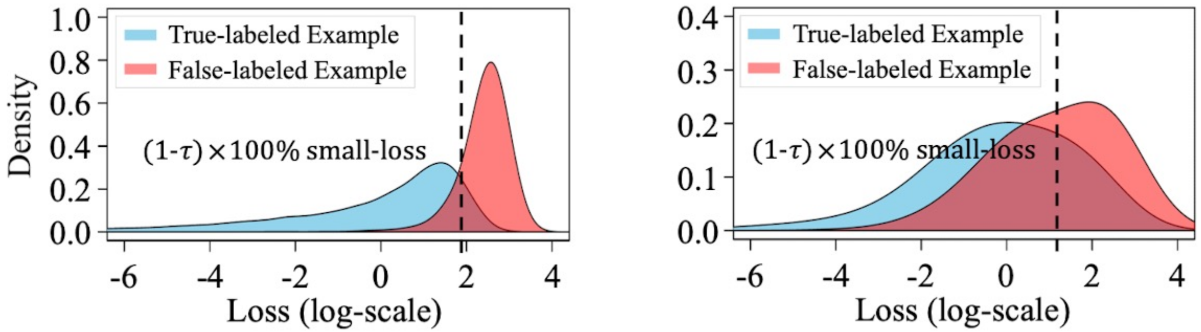
correct : predict to ground-truth label
memorized : overfit to noisy-label
Incorrect : neither correct nor memorized

Compared to CCE, ELR does not memorize the noisy labels.
(red stays low in the right column)

Liu, Sheng, et al. "Early-learning regularization prevents memorization of noisy labels." *Advances in neural information processing systems* 33 (2020): 20331-20342.

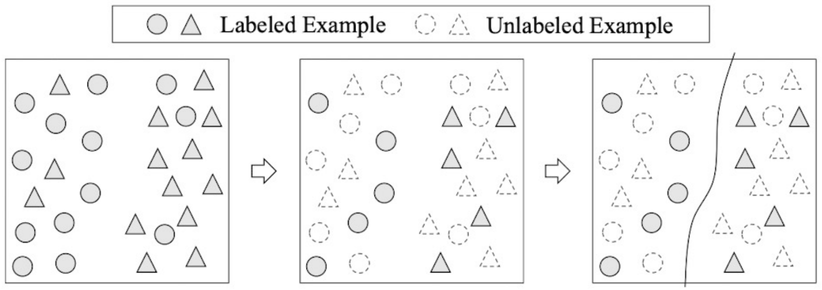
4. Sample selection

- Small loss trick : the clean label have smaller losses than the noisy label (not appropriate for the asymmetric noise)



(a) Symmetric Noise 40%. (b) Asymmetric Noise 40%.

- DivideMix
  - fit two-component Gaussian Mixture Model on loss values (small loss → clean, high loss → noise)
  - apply semi-supervised learning (mix-match) (labeled ≈ clean, unlabeled ≈ noise)



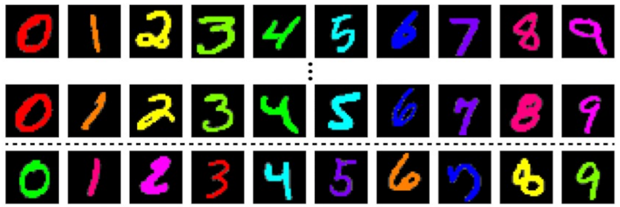
(a) Noisy Data. (b) Transformed Data. (c) SSL.

Li, Junnan, Richard Socher, and Steven CH Hoi. "Dividemix: Learning with noisy labels as semi-supervised learning." *arXiv preprint arXiv:2002.07394* (2020).

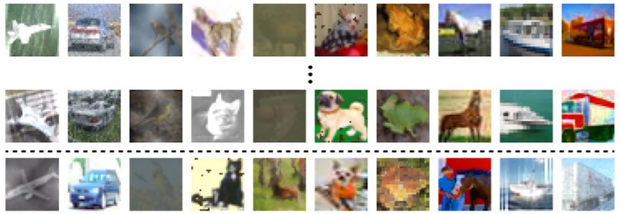
Dataset bias

Task description

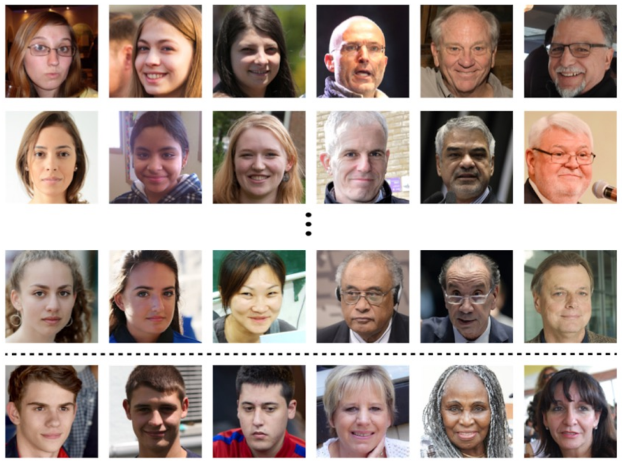
- Setting : x has many attributes (color, digit) and y is one of those (digit)
- **Def.** Dataset is “biased” if there is a highly correlated attribute that incurs bias-aligned samples
- Sample-type
    1. bias-aligned : un-intentionally, correctly predicted samples
       (e.g. camel in the desert)
    2. bias-conflicting :intentionally, in-correctly predicted samples
       (e.g. camel in the forest)



(a) Colored MNIST



(b) Corrupted CIFAR-10



(c) BFFHQ

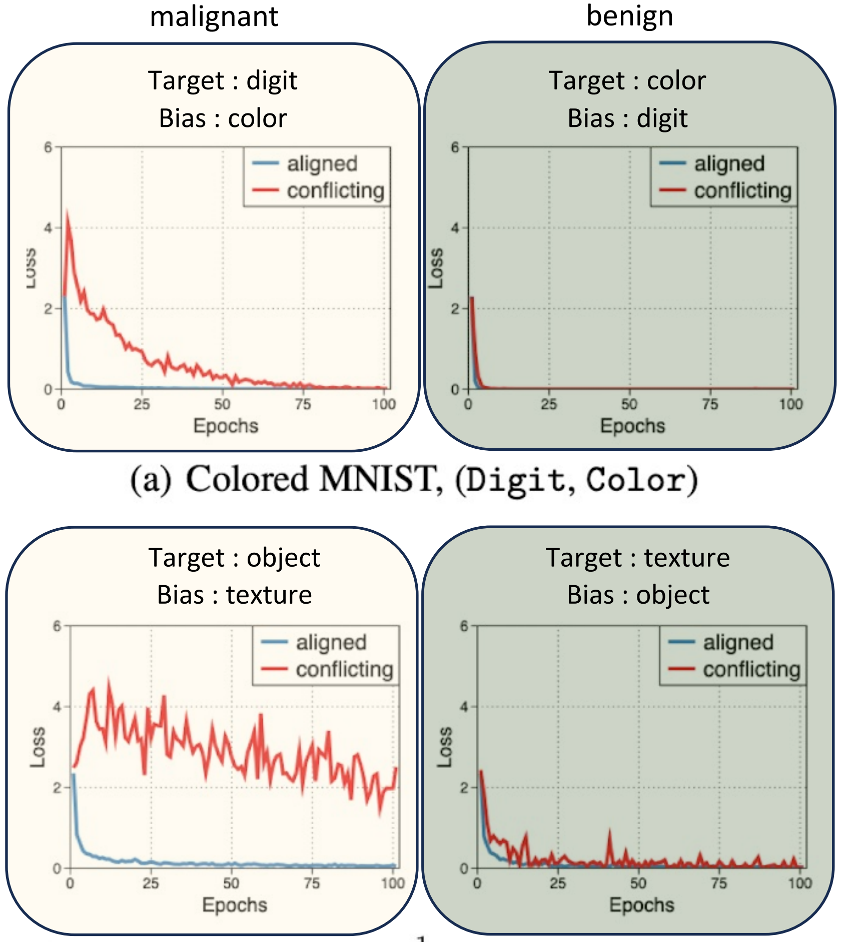
- Evaluation
    - un-biased dataset : same number for every possible combination of attributes
    - bias-conflicting dataset : remove bias-aligned from the un-biased dataset

Analogy to noisy labels

- Bias-type
  - malignant
    - bias is <u>easier</u> to learn than the target attribute
    - bias-aligned is learnt first and bias-conflicting later
  - Benign
    - bias is <u>harder</u> to learn than the target attribute
    - no difference b/t bias-aligned and bias-conflicting

- Training order of data
  - "Clean" → Noisy (noisy labels)
  - Bias-aligned → "Bias-conflicting" (dataset bias)

In contrast to noisy labels, where to focus is different



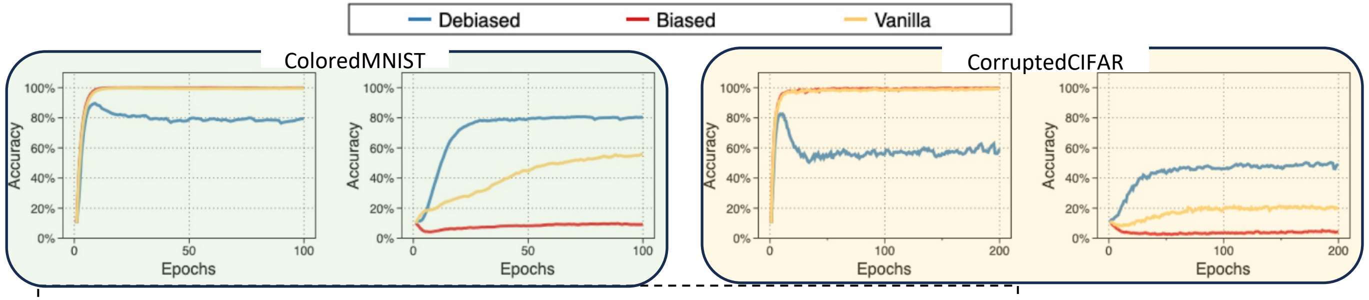
(a) Colored MNIST, (Digit, Color)

(b) Corrupted CIFAR-10[1], (Object, Corruption)

Learning from Failure (LfF)

1. Train a biased classifier (f_B) with GCE loss
   - up-weights samples of confident prediction ("clean" in noisy label, "bias-aligned" in dataset bias)
   - amplify the prejudice from the presence of dataset bias

2. Train a de-biased classifier (f_D) with re-weighted CE loss
   - relative difficulty : $\mathcal{W}(x) = \frac{CE_B(x,y)}{CE_B(x,y)+CE_D(x,y)}$ $\left(\Rightarrow \left\| \frac{\nabla_\theta CE_B(x,y)}{\sum_{(x_i,y_i)\in\mathcal{D}} \nabla_\theta CE_B(x_i,y_i)} \right\| \right)$
     - small weight to bias-aligned samples
     - large weight to "bias-conflicting" samples



Only de-biased model can achieve decent performance on both the bias-aligned (left) and the bias-conflicting (right)

Bias-supervision type

un-biased dataset

Dataset	Ratio (%)	Vanilla ○	Ours ○	HEX ◑	REPAIR ●	Group DRO ●
Colored MNIST	95.0	77.63±0.44	**85.39**±0.94	70.44±1.41	82.51±0.59	84.50±0.46
	98.0	62.29±1.47	**80.48**±0.45	62.03±0.24	72.86±1.47	76.30±1.53
	99.0	50.34±0.16	**74.01**±2.21	51.99±1.09	67.28±1.69	71.33±1.76
	99.5	35.34±0.13	**63.39**±1.97	41.38±1.31	56.40±3.74	59.67±2.73
Corrupted CIFAR-10[1]	95.0	45.24±0.22	**59.95**±0.16	21.74±0.27	48.74±0.71	53.15±0.53
	98.0	30.21±0.82	**49.43**±0.78	17.81±0.29	37.89±0.22	40.19±0.23
	99.0	22.72±0.87	**41.37**±2.34	16.62±0.80	32.42±0.35	32.11±0.83
	99.5	17.93±0.66	**31.66**±1.18	15.39±0.13	26.26±1.06	29.26±0.11
Corrupted CIFAR-10[2]	95.0	41.27±0.98	**58.57**±1.18	19.25±0.81	54.05±1.01	57.92±0.31
	98.0	28.29±0.62	**48.75**±1.68	15.55±0.84	44.22±0.84	46.12±1.11
	99.0	20.71±0.29	**41.29**±2.08	14.42±0.51	38.40±0.26	39.57±1.04
	99.5	17.37±0.31	34.11±2.39	13.63±0.42	31.03±0.42	**34.25**±0.74

Ratio of bias-aligned samples

bias-conflicting dataset

Dataset	Ratio (%)	Vanilla ○	Ours ○	HEX ◑	REPAIR ●	Group DRO ●
Colored MNIST	95.0	75.17±0.51	**85.77**±0.66	67.75±1.49	83.26±0.42	83.11±0.41
	98.0	58.13±1.63	**80.67**±0.56	58.80±0.28	73.42±1.42	74.28±1.93
	99.0	44.83±0.18	**74.19**±1.94	46.96±1.20	68.26±1.52	69.58±1.66
	99.5	28.15±1.44	**63.49**±1.94	35.05±1.46	57.27±3.92	57.07±3.60
Corrupted CIFAR-10[1]	95.0	39.42±0.20	**59.62**±0.03	14.09±0.31	49.99±0.92	49.00±0.45
	98.0	22.65±0.95	**48.69**±0.70	9.34±0.41	38.94±0.20	35.10±0.49
	99.0	14.24±1.03	**39.55**±2.56	8.37±0.56	33.05±0.36	28.04±1.18
	99.5	10.50±0.71	**28.61**±1.25	6.38±0.08	26.52±0.94	24.40±0.28
Corrupted CIFAR-10[2]	95.0	34.97±1.06	**58.64**±1.04	10.79±0.90	54.46±1.02	54.60±0.11
	98.0	20.52±0.73	**48.99**±1.61	6.60±7.23	44.63±0.75	42.71±1.24
	99.0	12.11±0.29	**40.84**±2.06	5.11±0.59	38.81±0.20	37.07±1.02
	99.5	10.01±0.01	**32.03**±2.51	4.22±0.43	31.45±0.28	30.92±0.86

Nam, Junhyun, et al. "Learning from failure: De-biasing classifier from biased classifier." *Advances in Neural Information Processing Systems* 33 (2020): 20673-20684.
Ahn, Sumyeong, Seongyoon Kim, and Se-young Yun. "Mitigating Dataset Bias by Using Per-sample Gradient." *arXiv preprint arXiv:2205.15704* (2022).

BiaSwap

- Goal : Generate bias-swapped image from the bias-aligned to the bias-conflicting (using image-to-image translation modules)

SwapAE : $swap\left(x^{(a)}, x^{(c)}\right) \rightarrow x^{(s)}$

- Encoder input : input image (x)
  (bias-aligned : $x^{(a)}$, bias-conflicting : $x^{(c)}$)
- Encoder output : content feature (z_c), style feature (z_s)
  (bias-aligned : $\left(z_c^{(a)}, z_s^{(a)}\right)$, bias-conflicting : $\left(z_c^{(c)}, z_s^{(c)}\right)$)
- Generator input : $\left(z_c^{(a)}, z_s^{(c)}\right)$
- Generator output : $x^{(s)}$

Loss function

random

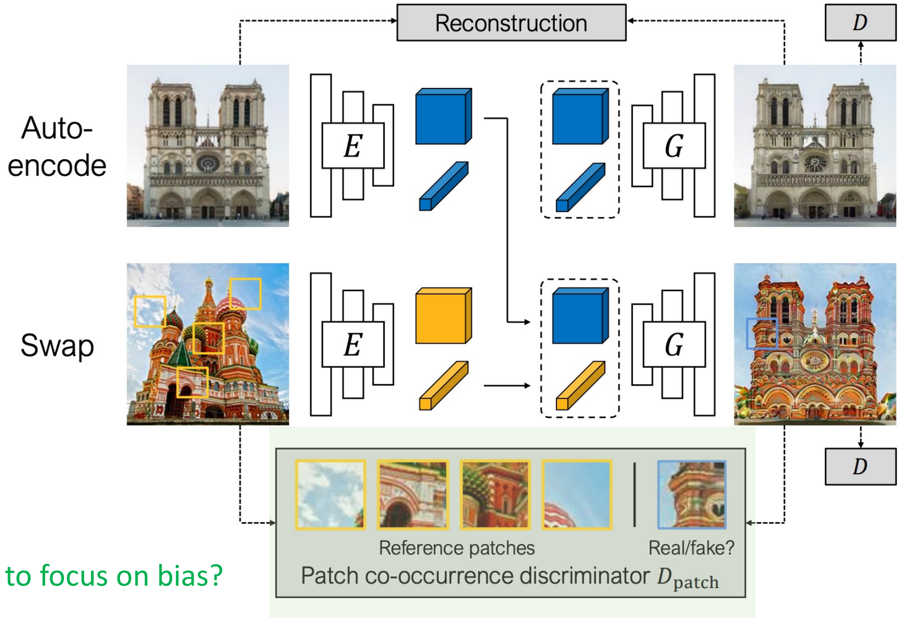
- $L_{content}(E,G) = \mathbb{E}_x\left[\left\|x - G(E(x))\right\|_2^2\right]$
- $L_{realistic1}(E,G,D) = \mathbb{E}_x\left[-\log D\left(G(E(x))\right)\right]$
- $L_{style}(E,G,D_{patch}) = \mathbb{E}_{x_1,x_2}\left[-\log D_{patch}\left(crop(swap(x_1,x_2)), crops(x_2)\right)\right]$
- $L_{realistic2}(\mathrm{E,G,D}) = \mathbb{E}_{x_1,x_2,x_1\neq x_2}\left[-\log D\left(swap(x_1,x_2)\right)\right]$

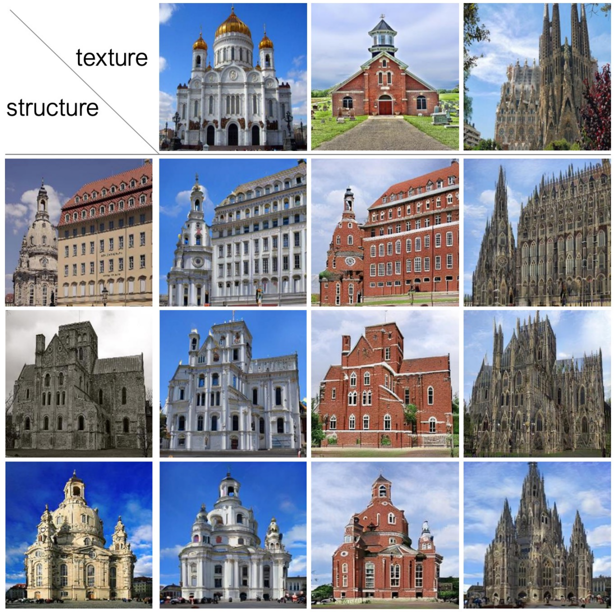
Bias-aligned

Bias-conflicting

1. How to separate?

2. How to focus on bias?





Park, Taesung, et al. "Swapping autoencoder for deep image manipulation." *Advances in Neural Information Processing Systems* 33 (2020): 7198-7211.

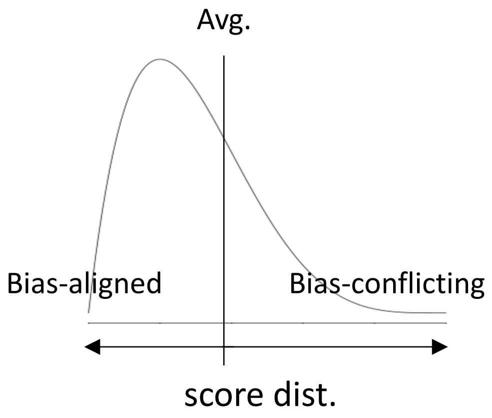
BiaSwap

How to separate the bias-aligned and the bias-conflicting?

1. Train a biased classifier (f_B) with GCE loss
2. Split the bias-aligned and the bias-conflicting

$$score(x) = \left| \underbrace{\mathbb{I}\left(\arg\max_k f_B(x)_k = y\right)}_{\text{“correct / wrong”}} - \underbrace{\max\left(\exp(f_B(x)) \,/ \sum_k \exp(f_B(x)_k)\right)}_{\text{“conf.”}} \right|$$

- bias-aligned : presumably correct → 1-conf. (≈ small value) → <u>below average</u>
- bias-conflicting : presumably wrong → conf. (≈ large value) → <u>over average</u>



Dataset	Colored MNIST	Corrupted CIFAR10	bFFHQ
Precision (%)	97.54	60.70	65.52
Recall (%)	92.12	87.28	70.62
F1 score (%)	94.74	66.13	67.70

Park, Taesung, et al. "Swapping autoencoder for deep image manipulation." *Advances in Neural Information Processing Systems* 33 (2020): 7198-7211.

BiaSwap

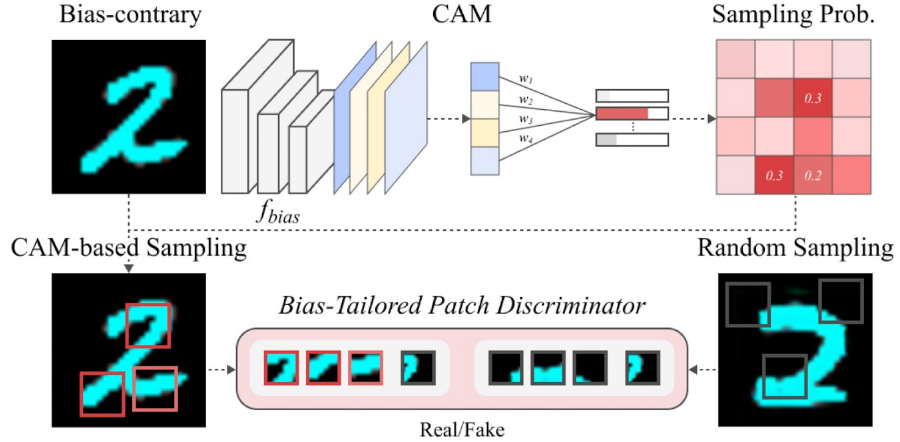
How to focus on bias?

3. Define Class Activation Map (CAM) for the target attribute

$$f_B(x)_k = \sum_c w_c^k \underbrace{\frac{1}{W\times H}\sum_{x,y} A_c(x,y)}_{\text{GAP}(c)} = \sum_{x,y} \underbrace{\sum_c \frac{1}{W\times H} w_c^k \cdot A_c(x,y)}_{\text{CAM}(x,y)}$$

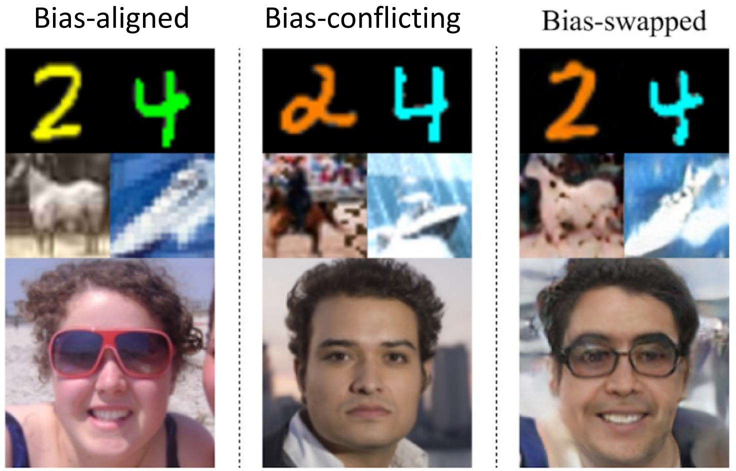
- w_c^k : the last linear layer weight from channel c to class k
- $A_c(x,y)$: (x,y)-coordinate value of the last convolutional feature map of channel c

4. Substitute the random cropping to the bias-tailored patch sampling



Sampling probability of (x,y)

$$P(x,y) = \frac{\exp(\text{CAM}(x,y))}{\sum_{w=1,h=1}^{w=W,h=H} \exp(\text{CAM}(w,h))}$$



Park, Taesung, et al. "Swapping autoencoder for deep image manipulation." *Advances in Neural Information Processing Systems* 33 (2020): 7198-7211.

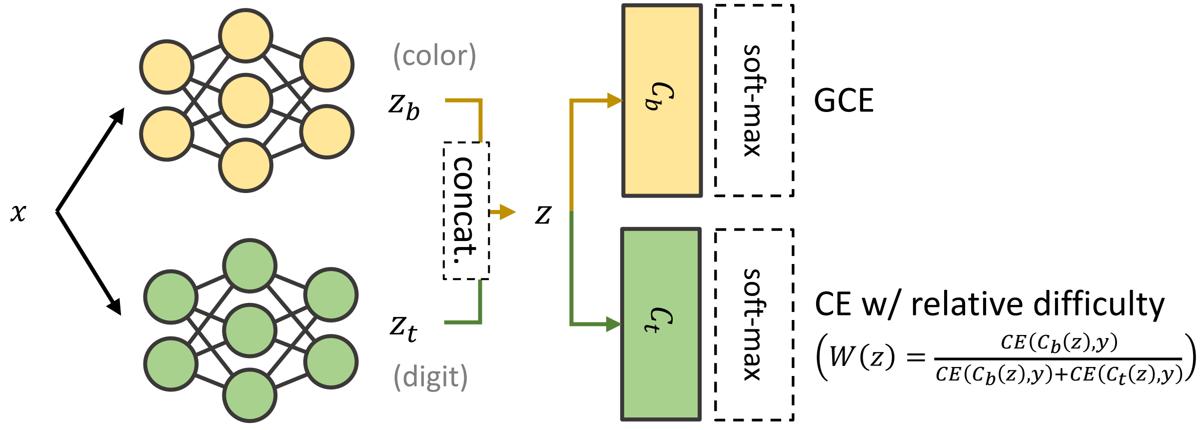
Disentangled Feature Augmentation (DFA)

- Observation
  - Diversity ratio matters more than sampling ratio
    - Diversity ratio : # of bias-conflicting / dataset
    - Sampling ratio : # of bias-conflicting / mini-batch

Dataset	Diversity ratio	Sampling ratio	Accuracy (%)
Colored MNIST	5%	50%	**83.77**$\pm$2.03
	1%	50%	67.19$\pm$1.99
	5%	1%	77.97$\pm$6.00
	1%	1%	49.91$\pm$4.22
Corrupted CIFAR-10	5%	50%	**46.99**$\pm$0.82
	1%	50%	33.08$\pm$0.80
	5%	1%	36.66$\pm$0.55
	1%	1%	23.98$\pm$0.00

- Goal : Increase diversity of bias-conflicting via feature augmentation

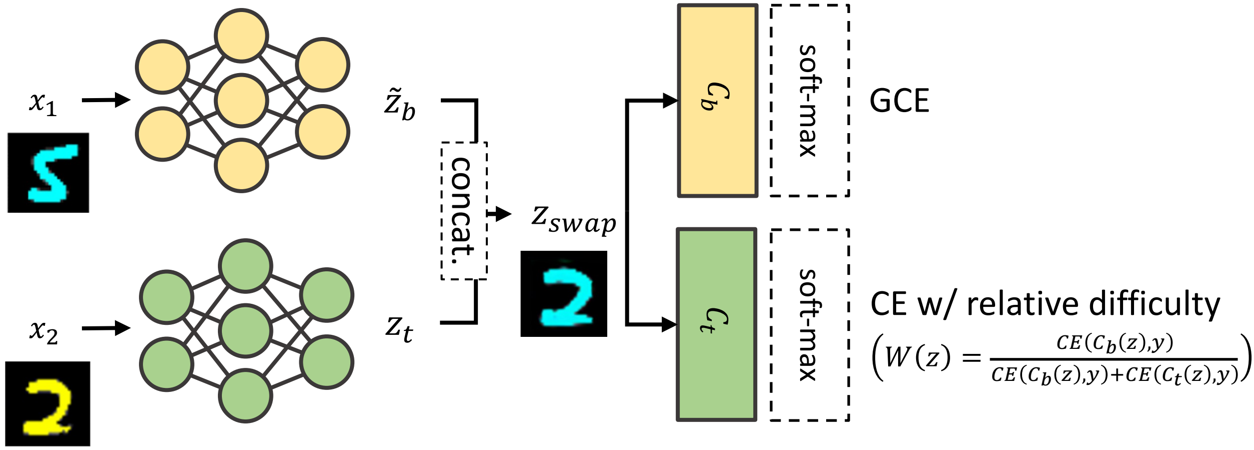
1. Train two separate encoders to obtain disentangled latent vectors (bias attribute ⟷ target attribute)



$$L_{dis} = W(z)CE(C_t(z), y) + \lambda_1 \cdot GCE(C_b(z), y)$$

(scheduling)

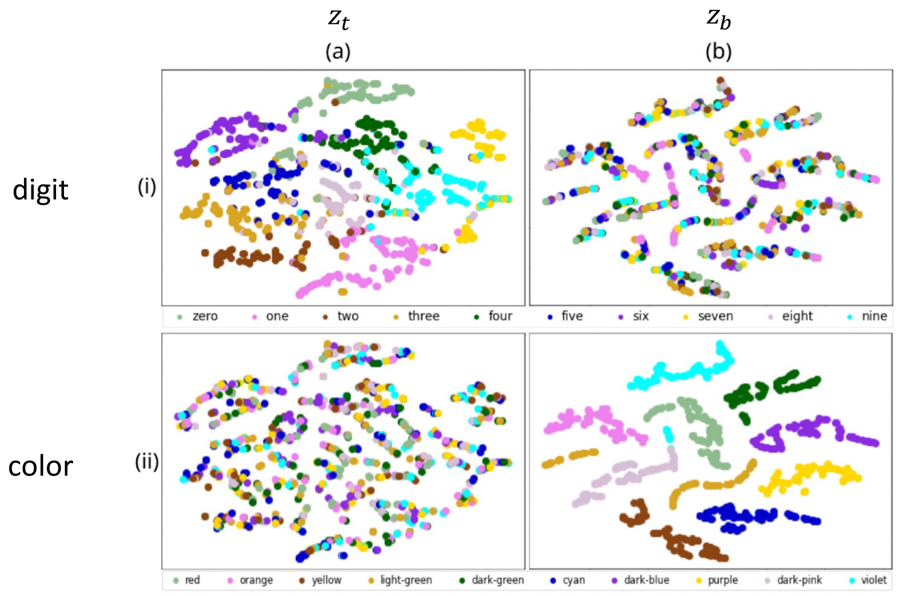
2. After some iterations, swap the latent vectors of random pairs within minibatch to increase diversity of bias-conflicting



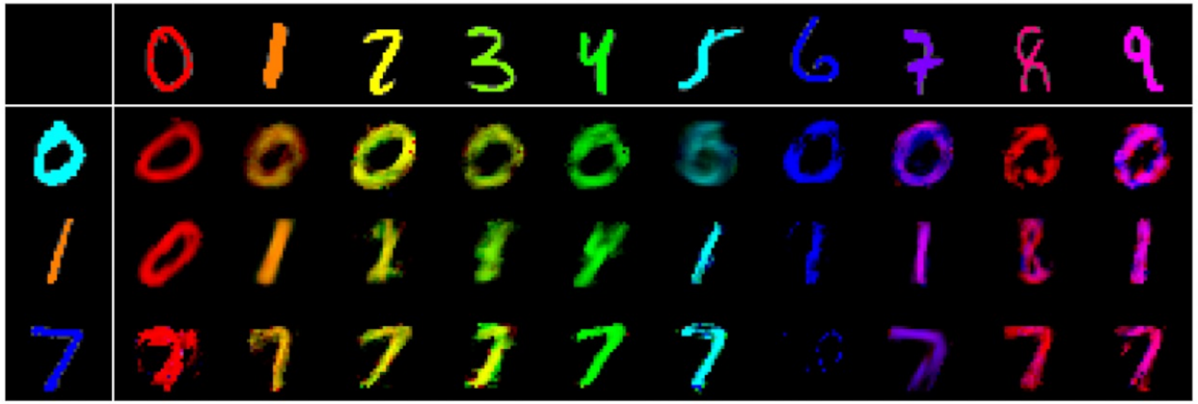
$$L_{swap} = W(z)CE\big(C_t(z_{swap})\big), y\big) + \lambda_2 \cdot GCE\big(C_b(z_{swap})\big), y\big)$$

Lee, Jungsoo, et al. "Learning debiased representation via disentangled feature augmentation." *Advances in Neural Information Processing Systems* 34 (2021): 25123-25133.

Disentangled Feature Augmentation (DFA)



T-sne embedding of z_t (left) and z_b (right) labeled by digit (top) and color (bottom).
(dataset : ColoredMNIST)



Reconstructed images based on the disentangled features
(row : maintain target attribute, column : maintain bias attribute)
(freeze the encoders and only train a decoder)

Accuracy(%)	Colored MNIST		Corrupted CIFAR10		BFFHQ	
	Target	Bias	Target	Bias	Target	Bias
Original	**76.08**	**98.07**	**35.63**	74.16	57.40	49.00
Swapping	71.40	94.29	35.14	**76.46**	**58.40**	**51.60**

With swapped latent(original → swapping), the accuracy drop is negligible.

Disentangle	Augment	Scheduled Augment	Colored MNIST	Corrupted CIFAR10	BFFHQ
–	–	–	52.09±2.88	25.82±0.33	56.87±2.69
✓	–	–	74.03±2.40	27.73±1.02	59.4±2.46
✓	✓	–	72.29±3.82	32.81±2.47	61.27±3.26
✓	✓	✓	**81.73**±2.34	**52.31**±1.00	**63.87**±0.31

Ablation study on disentangled feature, augment, scheduling

Lee, Jungsoo, et al. "Learning debiased representation via disentangled feature augmentation." *Advances in Neural Information Processing Systems* 34 (2021): 25123-25133.

E.O.D