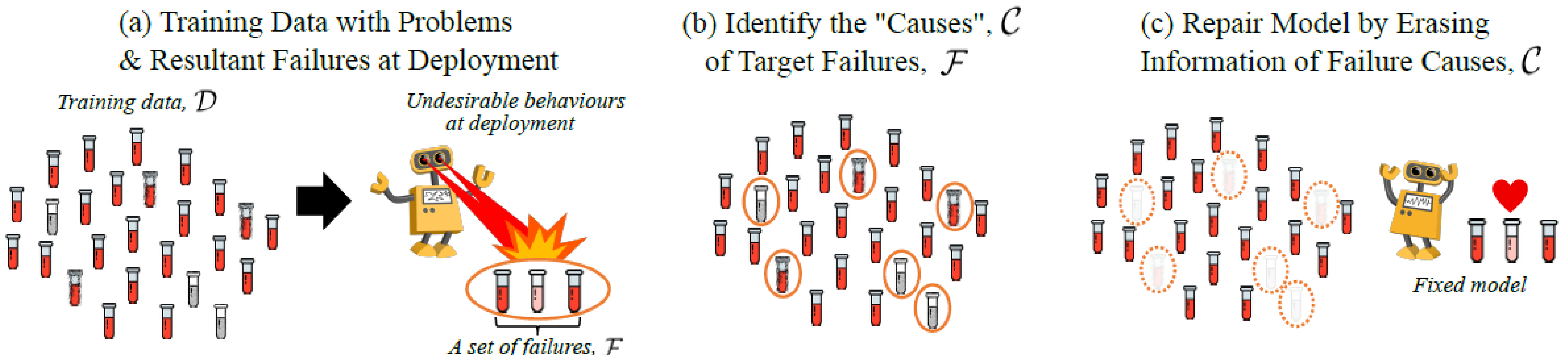
Mix-up based on data valuation score (2)

-Summary-

Mixup data valuation [Review]

- Based on [Repairing NN by Leaving the Right Past Behind, NeurIPS 2022], we can try similar method for diagnosing which mixup samples are conflicting with test samples.



- Notation for algorithms:
  - $\mathcal{F}$: failure set (wrong test samples after training)
  - $\mathcal{C}$: failure cause (training samples which contributes model to make failure set $\mathcal{F}$ after training) $\subset \mathcal{D}$
  - $\mathcal{D}$: training set / $\mathcal{D}_{test}$: test set

Mixup data valuation [Review]

- Step 1 : Failure set $\mathcal{F}$ identification
  - Train the model and check which test samples get wrong → set these test set as $\mathcal{F}$

- Step 2 : Failure cause $\mathcal{C}$ identification (when mixup is not used, follow original paper)
  - We want to observe the impact on model's prediction on failure set $\mathcal{F}$ by deleting a subset of training set $\mathcal{C} \subset \mathcal{D}$: → **observe the change of $\boldsymbol{r(\mathcal{C})}$**

$$r(\mathcal{C}) \coloneqq \log p(\mathcal{F}|\mathcal{D} - \mathcal{C}) - \log p(\mathcal{F}|\mathcal{D})$$

where $p(\mathcal{F}|\mathcal{D}) = \int p(\mathcal{F}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}, \; p(\mathcal{F}|\mathcal{D} \backslash \mathcal{C}) = \int p(\mathcal{F}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D} \backslash \mathcal{C}) d\boldsymbol{\theta}, \; p(\mathcal{F}|\boldsymbol{\theta}) = \prod_{(\boldsymbol{x},y) \in \mathcal{F}} p(y|\boldsymbol{x},\boldsymbol{\theta})$

Mixup data valuation [Review]

- Step 2 : Failure cause $\mathcal{C}$ identification [detailed descriptions are skipped]
  - Using i.i.d modeling assumption and Bayes' rule, we get following:

$$r(\mathcal{C}) = \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})}[p(\mathcal{C}|\boldsymbol{\theta})^{-1}] - \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[p(\mathcal{C}|\boldsymbol{\theta})^{-1}]$$

Note : $\boldsymbol{r(\mathcal{C}) = F\big(1, p(\theta|\mathcal{D},\mathcal{F})\big) - F\big(1, p(\theta|\mathcal{D})\big)}$**, where** $\boldsymbol{F(\epsilon, g(\theta)) = log \int g(\theta) e^{-\epsilon log p(\mathcal{C}|\theta)} d\theta}$

  - By using Taylor expansion : (Apply 1st order Taylor approx. on $F\big(\epsilon, g(\theta)\big)$ around $\epsilon = 0$)

$$\hat{r}(\mathcal{C}) \coloneqq \mathbb{E}_{p(\theta|\mathcal{D})}[\log p(\mathcal{C}|\theta)] - \mathbb{E}_{p(\theta|\mathcal{D},\mathcal{F})}[\log p(\mathcal{C}|\theta)]$$

  - Assume that data are i.i.d sampled and define $\boldsymbol{z} = (\boldsymbol{x}, y)$, then

$$\hat{r}(\mathcal{C}) = \sum_{\boldsymbol{z} \in \mathcal{C}} \hat{r}(\boldsymbol{z})$$

where $\hat{r}(\boldsymbol{z}) = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\log p(\boldsymbol{z}|\boldsymbol{\theta})] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})}[\log p(\boldsymbol{z}|\boldsymbol{\theta})], \quad p(\boldsymbol{z}|\boldsymbol{\theta}) = p(y|\boldsymbol{x},\boldsymbol{\theta})$

Mixup data valuation [Review]

- Step 2 : Failure cause $\mathcal{C}$ identification [detailed descriptions are skipped]
  - Note that $\hat{r}(\boldsymbol{z}) = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\log p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})}[\log p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})]$ (# $\boldsymbol{z} = (\boldsymbol{x}, y)$) whose computation is only valid when there is no mixup (mixed OH encoding vector)
  - But, the fundamental idea is to observe the difference of **log-prediction** at each sample $\boldsymbol{z}$ before and after the training the failure set $\mathcal{F}$.

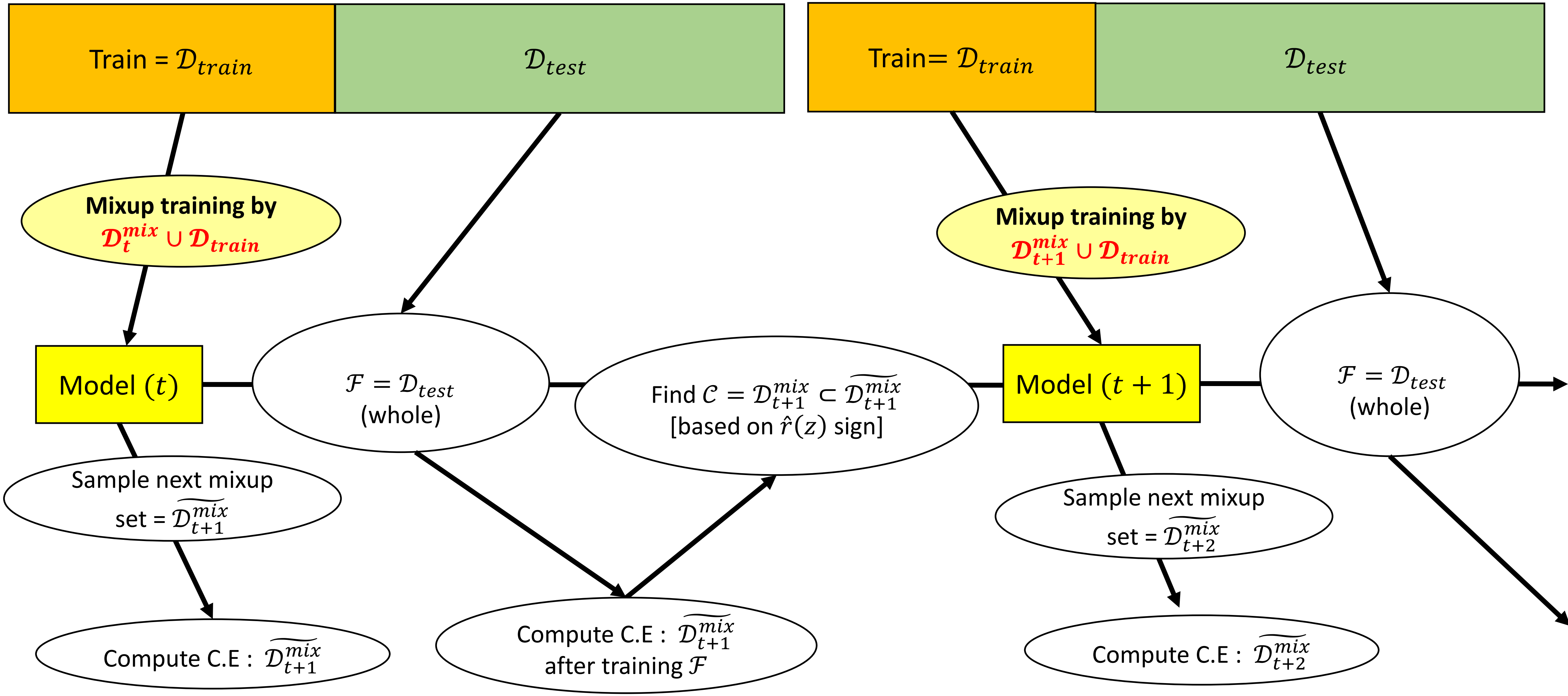
Note : $H(\boldsymbol{y}^{\boldsymbol{mix}}, p(\boldsymbol{x}^{\boldsymbol{mix}}|\boldsymbol{\theta}))$ = -$\lambda logp(\boldsymbol{y}_i|\boldsymbol{\theta},\boldsymbol{z}^{\boldsymbol{mix}}) - (1-\lambda) logp(\boldsymbol{y}_j|\boldsymbol{\theta},\boldsymbol{z}^{\boldsymbol{mix}})$

  - For mixup, we change the metric $\log p(\boldsymbol{z}|\boldsymbol{\theta}) \rightarrow H(\boldsymbol{y}^{\boldsymbol{mix}}, p(\boldsymbol{x}^{\boldsymbol{mix}}|\boldsymbol{\theta}))$ [cross entropy]

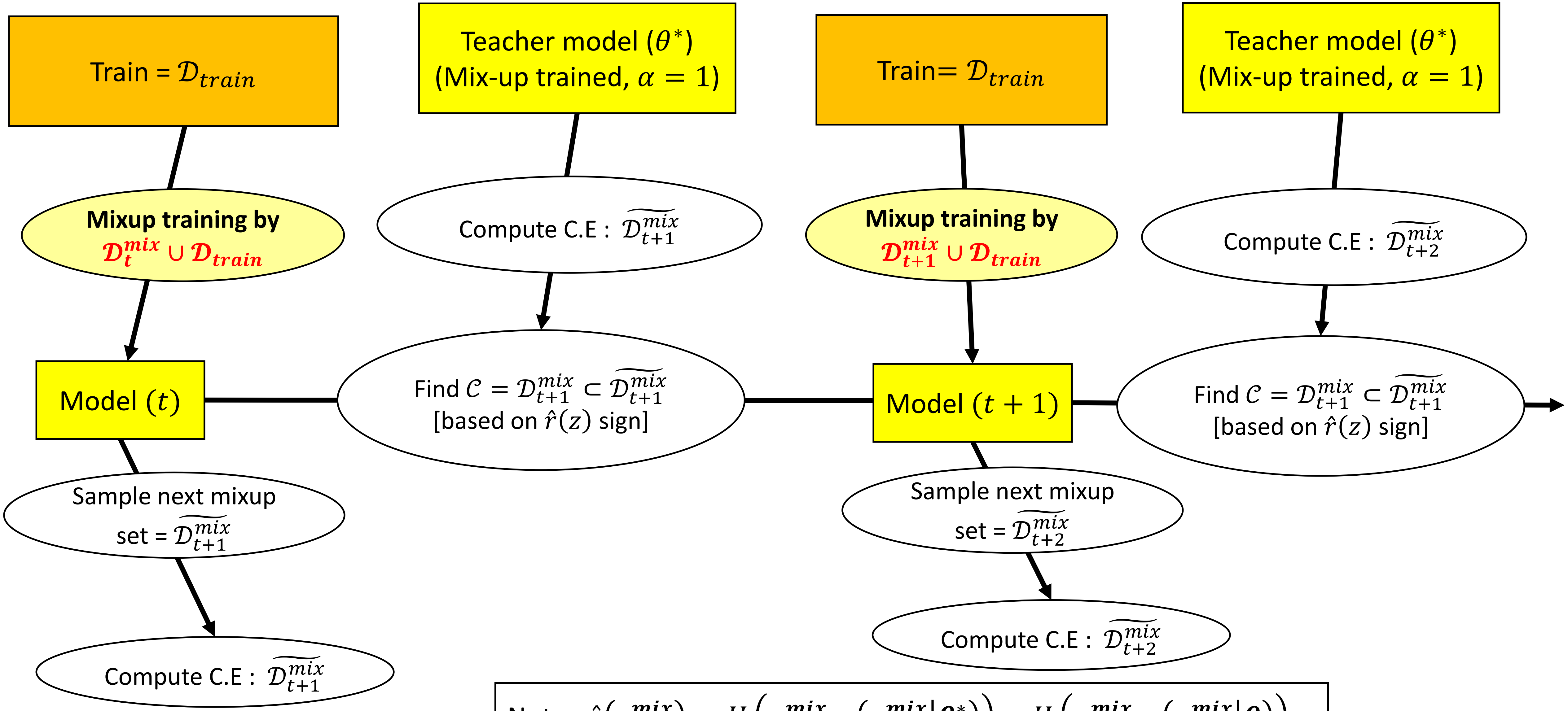
$$\hat{r}(\boldsymbol{z}^{\boldsymbol{mix}}) = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})}[H(\boldsymbol{y}^{\boldsymbol{mix}}, p(\boldsymbol{x}^{\boldsymbol{mix}}|\boldsymbol{\theta}))] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[H(\boldsymbol{y}^{\boldsymbol{mix}}, p(\boldsymbol{x}^{\boldsymbol{mix}}|\boldsymbol{\theta}))]$$

    - When the additional training of failure set $\mathcal{F}$ con flicts the prediction of $\boldsymbol{x}^{\boldsymbol{mix}}$, then $\hat{r}^{mix}(\boldsymbol{z}^{\boldsymbol{mix}})$ should be high.
    - If the training does not conflict much, then $\hat{r}^{mix}(\boldsymbol{z}^{\boldsymbol{mix}})$ should be low.

Mixup data valuation – Algorithm (Whole)



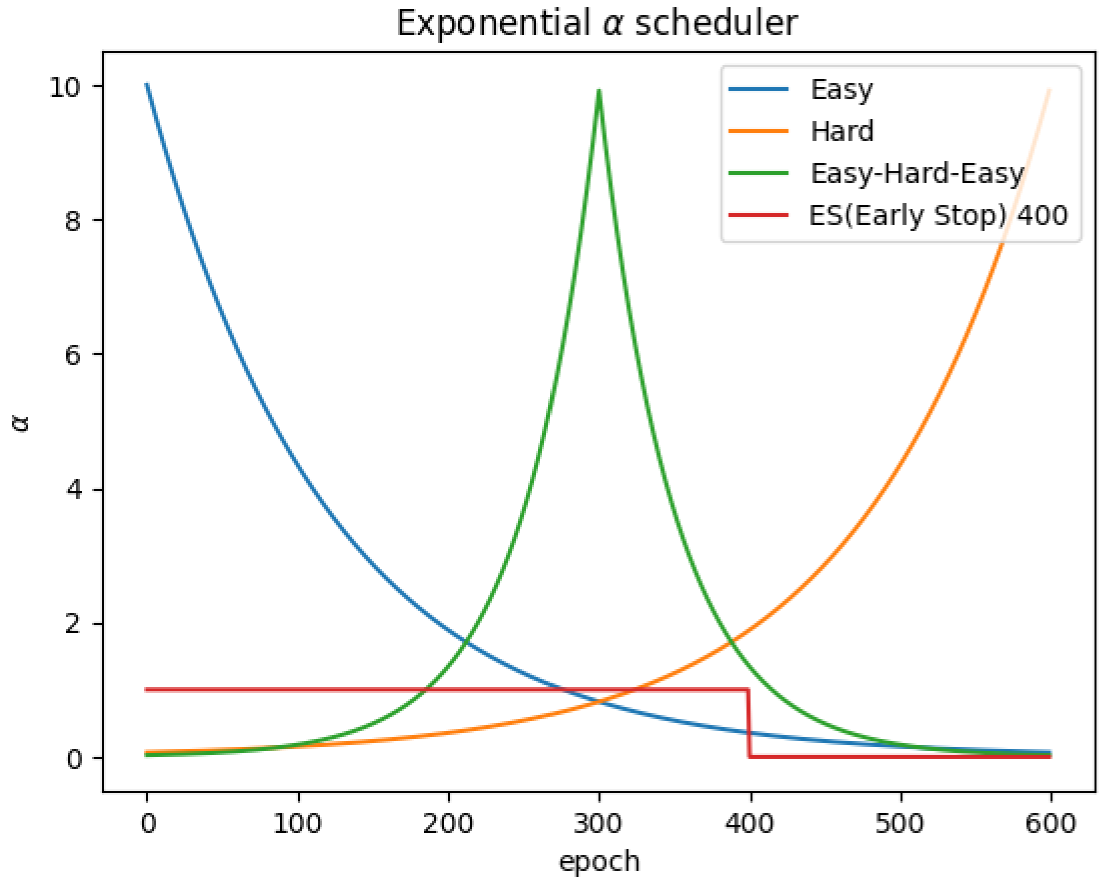
Mixup data valuation – Algorithm (Train)



Note : $\hat{r}(\boldsymbol{z^{mix}}) = H\left(\boldsymbol{y^{mix}}, p(\boldsymbol{x^{mix}}|\boldsymbol{\theta}^*)\right) - H\left(\boldsymbol{y^{mix}}, p(\boldsymbol{x^{mix}}|\boldsymbol{\theta})\right)$

Mixup data valuation – Observation (real data)

- Exponential α - scheduler : $(\alpha = 1$ at the half of the epochs, where $\alpha \in (0, 10))$



Note

1. Easy / Hard scheduler can be fine-tuned by observing the validation accuracy and modify the shape manually.

2. Early stopping is suggested to resolve intrinsic sub-optimal optimization of mix-up training.

Mixup data valuation – Observation (real data)

- Test accuracy for each dataset (using scheduler) :

Epoch = 600 / 400 (Reg)	CIFAR-10	CIFAR-100	STL-10	Caltech-101	DTD	Aircraft	Tiny-Imagenet
Baseline	95.22	78.61	77.43	79.60	21.38	79.52	59.53
Mixup ($\alpha = 1$)	96.41	80.08	86.6	82.63	28.40	83.80	60.86
Regmixup ($\alpha = 20$)	96.61	80.45	85.58	83.22	24.83	83.23	62.51
Easy scheduler	95.92	80.07	86.2	85.03	30.85	83.65	61.22
Hard scheduler	96.11	78.92	84.01	82.09	24.31	79.48	59.75
Easy-Hard-Easy scheduler	95.93	80.21	85.20	84.48	30.31	83.20	60.78
ES at epoch 400	95.20	79.63	86.48	86.81	30.53	84.87	60.14
$+\hat{r}(z)$ (whole, test)	95.68 (95.87)	79.38 (79.38)	86.81 (87.10)	85.59 (85.82)	24.73 (26.06)	82.93 (83.74)	-
$-\hat{r}(z)$ (whole, test)	96.44 (96.88)	81.37 (82.56)	87.73 (88.40)	86.44 (89.01)	30.21 (31.65)	84.61 (85.87)	61.94 (64.0)
$+\hat{r}(z)$ (TR/$\alpha = 1$)	95.33	77.7	78.24	83.60	20.05	77.78	-
$-\hat{r}(z)$ (TR/$\alpha = 1$)	96.62	**81.55**	**89.02**	84.59	31.43	85.93	62.04
$-\hat{r}(z)$ (TR/α =Hard)	**96.76**	81.21	88.68	**86.82**	**32.23**	**85.98**	-
whole $\hat{r}(z)$ (TR/$\alpha = 1$)	96.12	80.43	88.13	83.56	30.71	85.68	-

Mixup data valuation – Observation (real data)

- Test accuracy for each dataset (using scheduler) :

All results are averaged values by 2 trials

Epoch = 600 / 400 (Reg)	CIFAR-10	CIFAR-100	STL-10	Caltech-101	DTD	Aircraft	Tiny-Imagenet
Baseline	95.22	78.61	77.43	79.60	21.38	79.52	59.53
Mixup ($\alpha = 1$)	96.41	80.08	86.6	82.63	28.40	83.80	60.86
Mixup ($\alpha = 1$, 1200ep)	96.34	79.54	85.78	84.32	32.18	84.58	-
Regmixup ($\alpha = 20$)	96.61	80.45	85.58	83.22	24.83	83.23	62.51
Regmixup ($\alpha = 20$, 1200ep)	**96.60**	79.84	88.78	86.35	33.67	85.92	-
Easy scheduler	95.92	80.07	86.2	85.03	30.85	83.65	61.22
Hard scheduler	96.11	78.92	84.01	82.09	24.31	79.48	59.75
ES at epoch 400	95.20	79.63	86.48	86.81	30.53	84.87	60.14
$-\hat{r}(z)$ (TR/α =Easy)	96.34	**81.07**	89.85	**86.43**	**35.21**	85.68	-
$-\hat{r}(z)$ (TR/$\alpha = 1$) T: Regmix (1200ep)	96.29	80.64	88.56	86.43	29.09	85.53	-
$-\hat{r}(z)$ (TR/$\alpha = 20$) T: Regmix (1200ep)	96.59	80.80	**90.56**	85.59	31.80	85.77	-

Easy scheduler : scheduler parameter is optimized manually by observing validation accuracy with trials and errors

Mixup data valuation

- Algorithm goal : reduce the (mix-up) training loss per sample lower than the teacher model.
  - When $\hat{r}\left(z^{mix}\right) \leq 0 : z^{mix}$ is included / When $\hat{r}\left(z^{mix}\right) > 0 : z^{mix}$ is excluded

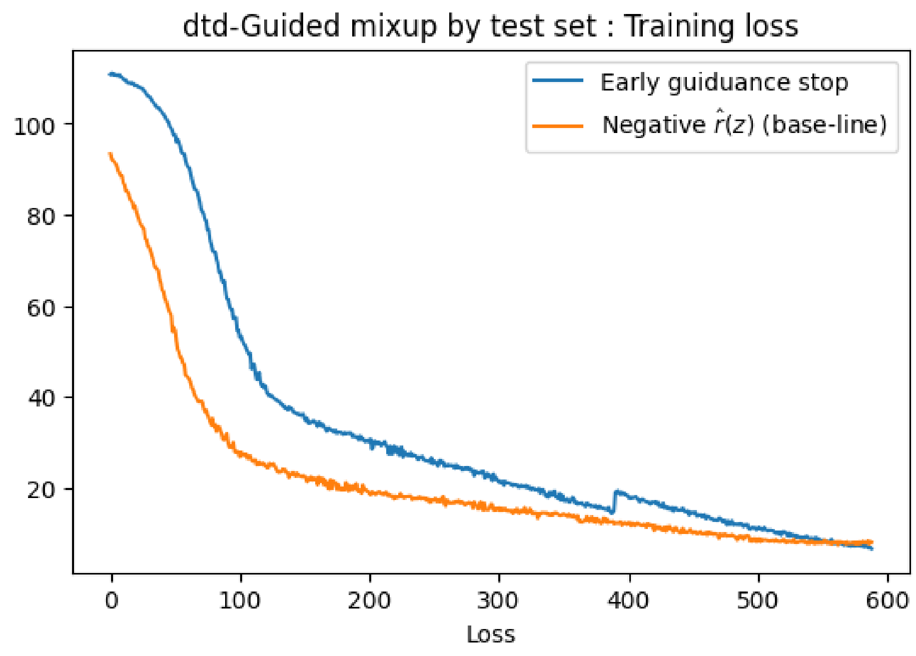
- Problems:
  - Q1: Around at 500 epoch (out of 600 epoch), **the student model starts to perform better than the teacher model** ⇒ Is it reasonable for the student to be guided from teacher?
  - Q2 : It is known that mix-up training leads to regularize input gradient $||\nabla_x f_\theta(x)||_2$ [Zhang, 2020], and **eventually leads to wrong optimal $\boldsymbol{\theta}$ if the model is overtrained with mix-up** (at least in regression problem) → The goal of our algorithm is desirable?

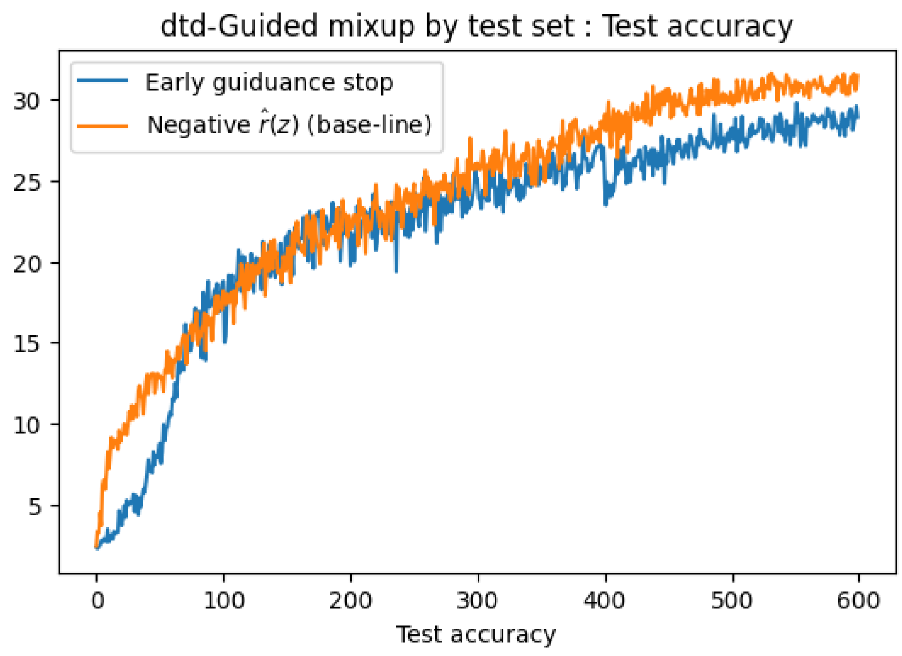
Mixup data valuation – Q1

- Q1 : Is it reasonable for the student to be guided from teacher (at the end tail of epochs)?

- 1^{st} solution : Stop guiding around epoch 400.

  → Unfortunately, this method leads to performance drop (Ex : DTD : 31% -> 29%) + The sudden drop of test accuracy is observed when guiding is stopped.

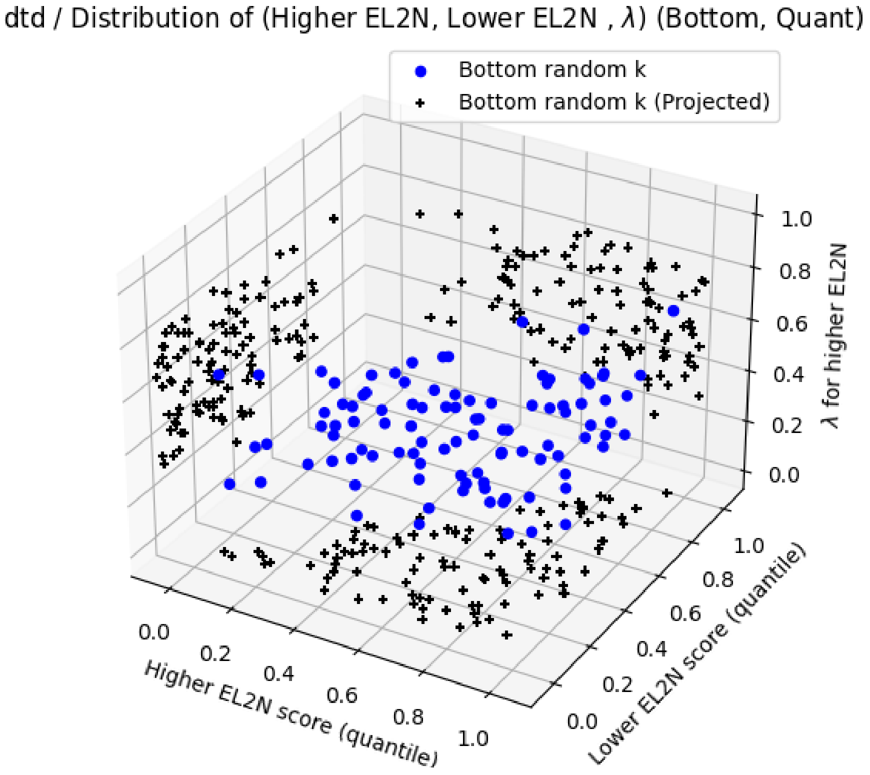


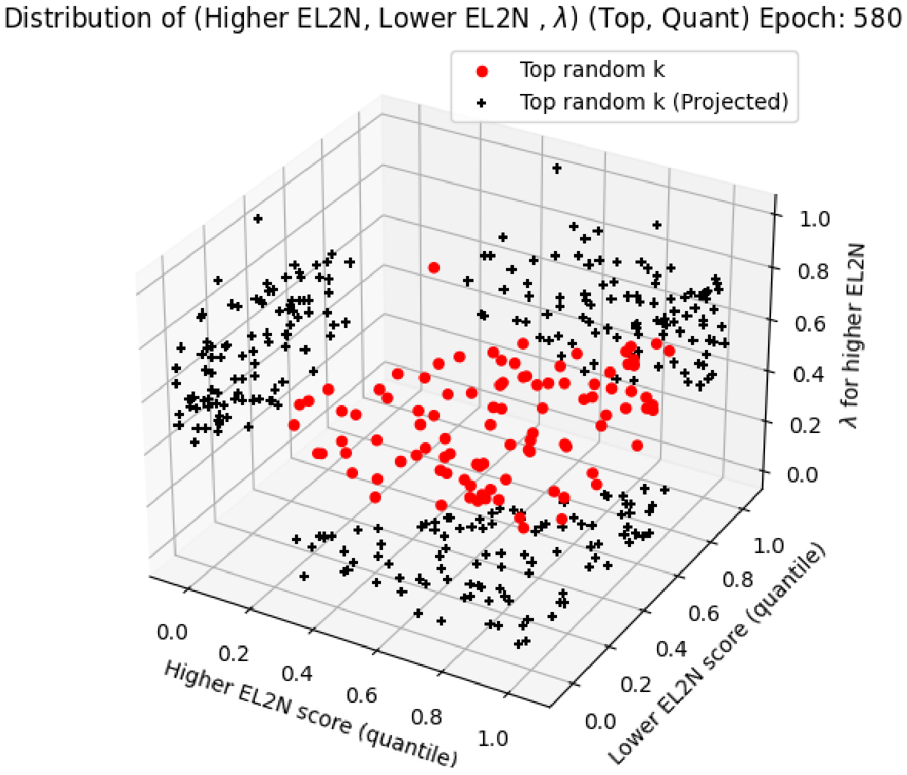


Mixup data valuation – Q1

Note : Unfortunately, there seems no meaningful correlation between EL2N score and $\hat{r}(z^{mix})$ values.

- 2^{nd} solution: Use easy α scheduler to effectively use the guide training from Teacher model.
  - Empirically, it turned out that $|\hat{r}(z^{mix})|$ is usually bigger as $\lambda \to 0.5$



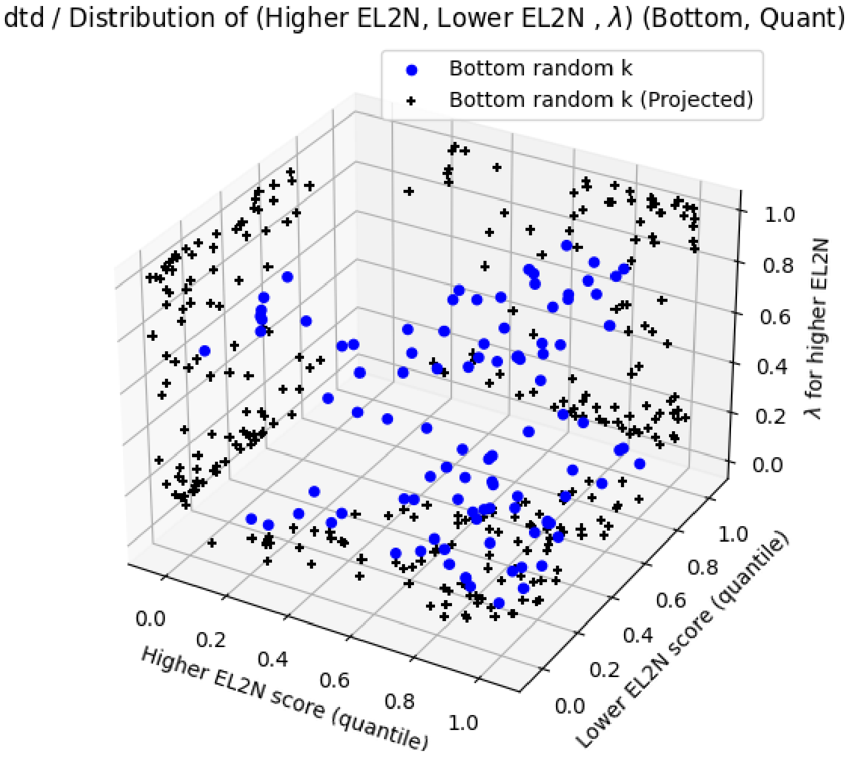


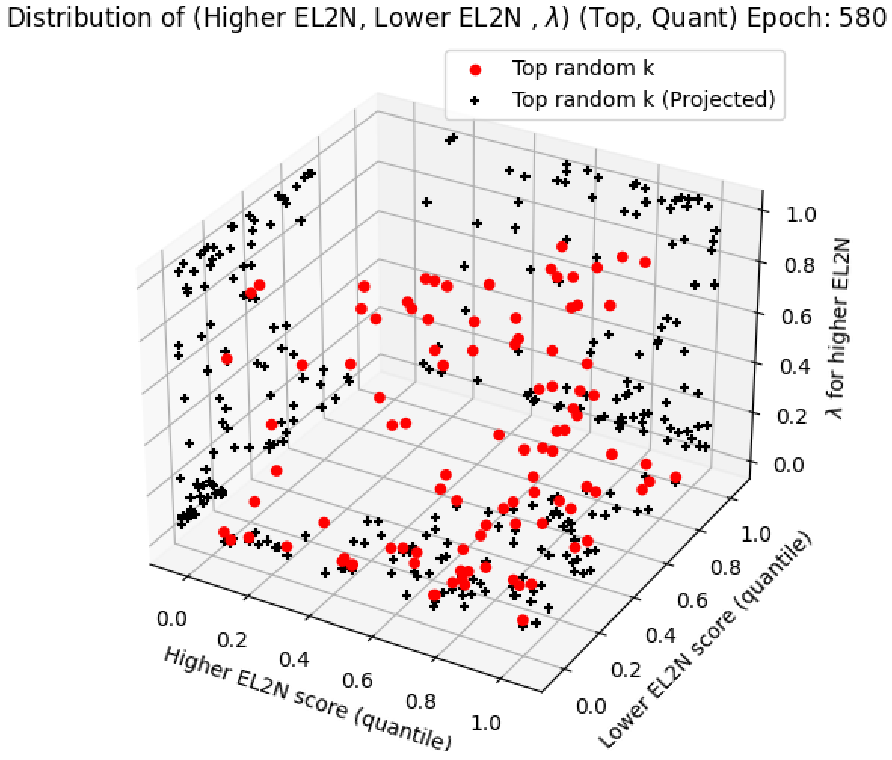
Tail 100 of each negative(Bottom) / positive (Top) samples' λ, EL2N distribution

Mixup data valuation – Q1

Note : Unfortunately, there seems no meaningful correlation between EL2N score and $\hat{r}(z^{mix})$ values.

- 2nd solution: Use easy α scheduler to effectively use the guide training from Teacher model.
    - Empirically, it turned out that $|\hat{r}(z^{mix})|$ is usually bigger as $\lambda \to 0.5$





Center 100 of each negative(Bottom) / positive (Top) samples' λ, EL2N distribution

Mixup data valuation – Q1

- 2nd solution: Use easy α scheduler to effectively use the guide training from Teacher model.
  - This implies there are more 'loss difference' (decision difference) between Student and Teacher model about r^{mix} with $\lambda \cong 0.5$.
  - To amplify the effect of guidance by teacher, it is intuitive **to suggest mix-up samples with $\boldsymbol{\lambda} \rightarrow \mathbf{0.5}$ (or $\boldsymbol{\alpha} = \mathbf{20}$),** which will be filtered out based on loss difference.
  - But, the problem is that we do not trust the guidance after 500 epoch (due to higher performance of student) $\rightarrow$ reduce $\lambda \rightarrow 0$ to minimize the effect of guidance.

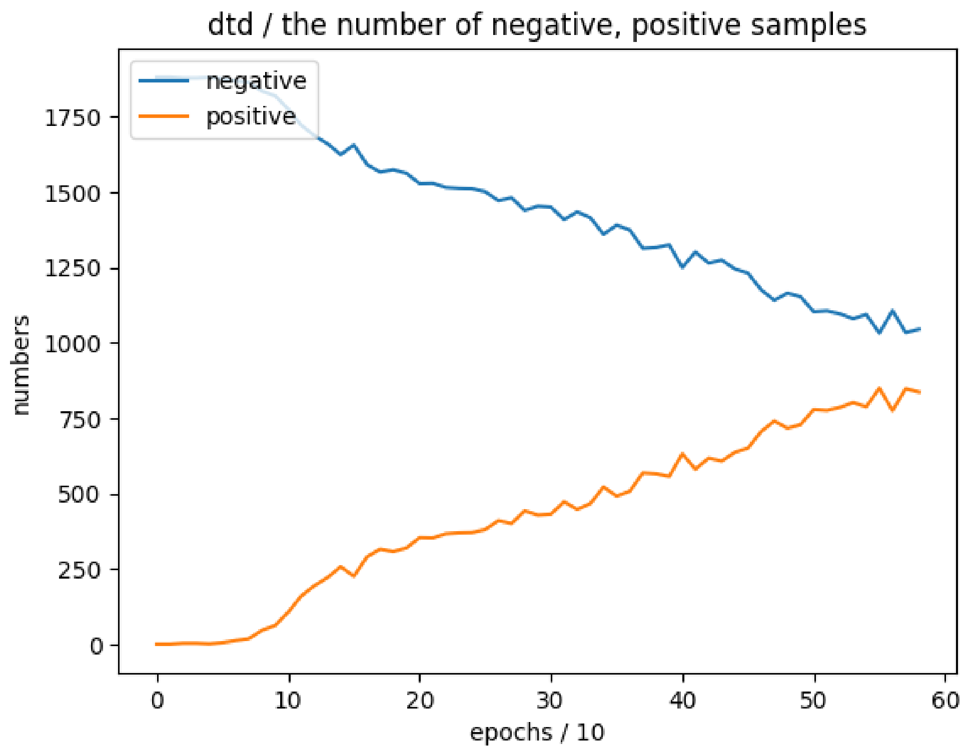
Mixup data valuation – Q1

- 2nd solution: Use easy α scheduler to effectively use the guide training from Teacher model.
  - Another problem to be resolved :
    - We figured out that there are certain types of dataset which favors High λ mix-up training. (Such as CIFAR-10, Tiny-Imagenet)
    - In this case, the easy tail of α scheduler can degrade the generalization performance at the end.

Epoch = 600 / 400 (Reg)	CIFAR-10	Epoch = 600 / 400 (Reg)	CIFAR-10
Baseline	95.22	Easy scheduler	95.92
Mixup ($\alpha = 1$)	96.41	Hard scheduler	96.11
Mixup ($\alpha = 1$, 1200ep)	96.34	$+\hat{r}(z)$ (TR/$\alpha = 1$)	95.33
Regmixup ($\alpha = 20$)	96.61	$-\hat{r}(z)$ (TR/$\alpha = 1$)	96.62
Regmixup ($\alpha = 20$, 1200ep)	96.60	$-\hat{r}(z)$ (TR/$\alpha =$Hard)	**96.76**
$-\hat{r}(z)$ (TR/$\alpha =$Easy)	96.34	whole $\hat{r}(z)$ (TR/$\alpha = 1$)	96.12

Mixup data valuation – Q1-SideNote

- (Appendix) : How can we grasp the intensity of guidance effect from Teacher?
  - One idea is to check the cardinality of $-\hat{r}(z^{mix})$ along epochs



Intuitive way:

- When the # of $-\hat{r}(z^{mix})$ < the # of $+\hat{r}(z^{mix})$ (*):
  We can treat the Student model as a more generalized one.

Experimental results:

- However, the Student model outperforms even before the condition (*) satisfied.

- In this experiment, the Student outperforms Teacher around 500 epoch.

(Appendix) : What happen if we change the Student into more wider one?

- If we change the Student model → EfficientNet V2 (S), the test accuracy gap was 5% on DTD dataset.

Mix-up ($\alpha = 1$)	20.69	$-\hat{r}(z)$ (TR/$\alpha = 1$)	25.32

- But, this result might not be followed from the guidance effect, rather from increased # of iterations on our algorithm.

Mixup data valuation – Q2

- Q2: The goal of our algorithm is desirable? (framework from [Z. Liu, 2023])
  - Consider simple least square regression problem with data (X, Y), and let $f: \mathcal{X} \rightarrow \mathcal{Y}$ be the ground-truth labelling function.
  - Let $(\tilde{X}, \tilde{Y})$ be a synthetic pair obtained by mixing (X, Y) and (X', Y'), and set synthesized training dataset $\tilde{S} = \{(\tilde{X}_i, \tilde{Y}_i)\}_{i=1}^{m}$
  - Consider a random feature model: $\theta^T \phi(X)$

    where $\phi: \mathcal{X} \rightarrow \mathbb{R}^d$ and $\theta \in \mathbb{R}^d$ (Note that ϕ is fixed and only θ is learned by SGD)
  - Define MSE loss as follows:

$$\hat{R}_{\hat{S}}(\theta) = \frac{1}{2m} \left\| \theta^T \widetilde{\Phi} - \tilde{Y}^T \right\|_2^2$$

where $\widetilde{\Phi} = \left[\phi(\tilde{X}_1), \ldots, \phi(\tilde{X}_m)\right] \in \mathbb{R}^{d \times m}$ and $\tilde{Y} = \left[\tilde{Y}_1, \tilde{Y}_2, \ldots, \tilde{Y}_m\right] \in \mathbb{R}^m$

Mixup data valuation – Q2

- Q2: The goal of our algorithm is desirable? (framework from [Z. Liu, 2023])
  - Our SGD update rule is as follows:

$$\dot{\theta} = -\eta \nabla \hat{R}_{\tilde{S}}(\theta) = \frac{\eta}{m} \widetilde{\Phi}\widetilde{\Phi}^T \left(\widetilde{\Phi}^{\dagger}\tilde{Y} - \theta\right)$$

where η is learning rate, and $\widetilde{\Phi}^{\dagger}$ is pseudo inverse of $\widetilde{\Phi}^T$.

Lemma 5.1 from [Z. Liu, 2023]

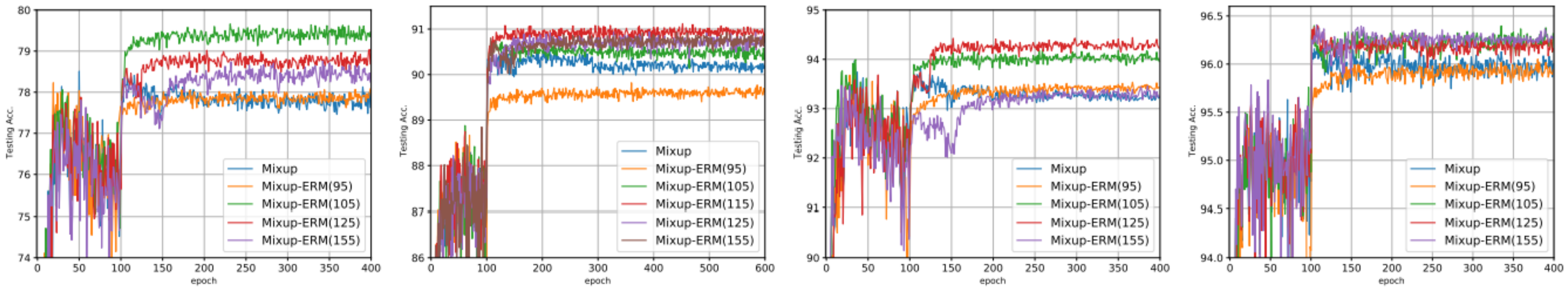
Let $\theta^* = \widetilde{\Phi}^{\dagger}\tilde{Y}$ and $\theta^{noise} = \widetilde{\Phi}^{\dagger}Z$, where $Z = [Z_1, \ldots Z_m] \in \mathbb{R}^m$ (where $Z \coloneqq \tilde{Y} - \tilde{Y}^*$ and $\tilde{Y}^* = f(\tilde{X})$), the above ODE has the following closed form solution:

$$\theta_t - \theta^* = (\theta_0 - \theta^*)e^{-\frac{\eta}{m}\widetilde{\Phi}\widetilde{\Phi}^T t} + \left(I_d - e^{-\frac{\eta}{m}\widetilde{\Phi}\widetilde{\Phi}^T t}\right)\theta^{noise}$$

- Hence, as $t \to \infty$, $\theta_\infty = \theta^* + \theta^{noise}$, which leads to wrong solution under mix-up.

Mixup data valuation – Q2

- To resolve this problem, two paper [Z. Liu, 2023], [D.Zou, 2023] suggest the early-stop of mix-up



(a) CIFAR10 (30%) (b) CIFAR10 (100%) (c) SVHN (30%) (d) SVHN (100%)

Figure 6: Switching from Mixup training to ERM training. The number in the bracket is the epoch number where we let $\alpha = 0$ (i.e. Mixup training becomes ERM training).

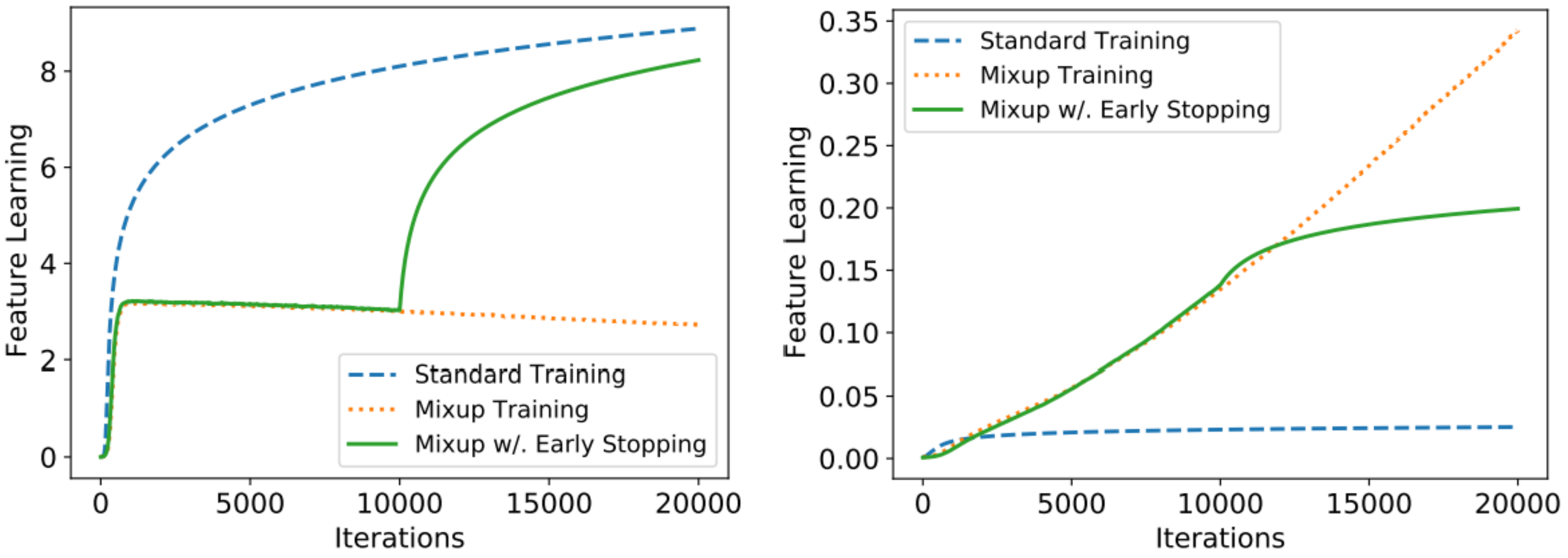
- Claim from papers: There exists an appropriate early stopping time of mixup to avoid generalization degradation.

  1. If the switch is too early → may not boost model performance (∵ small regularization),

  2. If the switch is too late → memorization of noisy data happen → degrade generalization.

Mixup data valuation – Q2

- Similarly, [D.Zou, 2023] proved that the early-stop of mix-up can be helpful for efficient learning of rare features in a given dataset.



(a) Common Feature Learning

(b) Rare Feature Learning

Figure 2: Common feature learning and rare feature learning on synthetic data, all experiments are conducted using full-batch gradient descent. Here we consider three training methods: standard training, Mixup training, and Mixup training with early stopping (at the 10000-th iteration).

Hypothesis to be verified

1. Our algorithm can benefit the model by boosting the rare feature learning (due to guidance of Teacher)

2. Hard→ Easy mix-up transition by easy α scheduler can potentially act as a smoothed version of Early stopping.

Mixup data valuation – Q2

Side note:
We may interpret the increase of noise term as the model's GraNd score differentiation ability

- But, why does the mix-up works well even if the model converges t
    - From [J. Zhang, 2021], it turns our that large-scale NN can generalize well without having the gradient norm vanish during training (implying no convergence to stationary points), which is a tremendous gap between theory and practice.

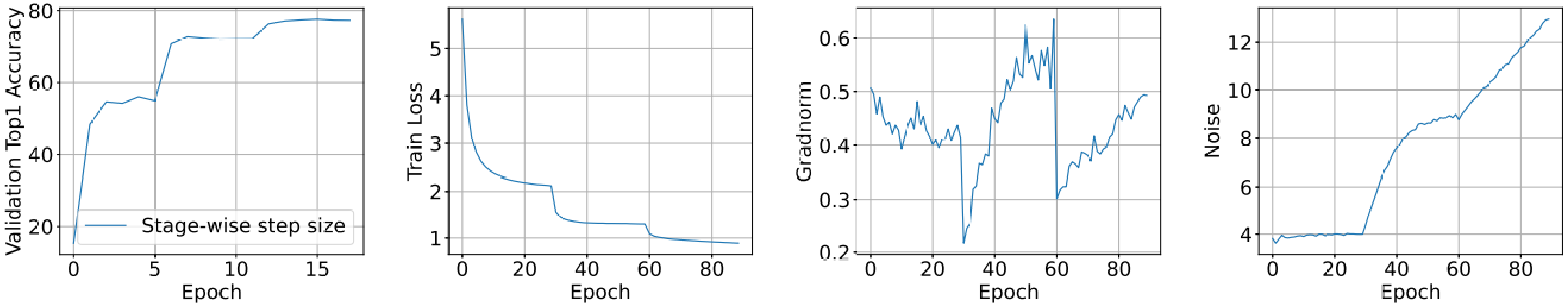


Figure 1. The validation accuracy and the quantities of interest (1) for the default training schedule of ImageNet + ResNet101 experiment.

where **GradNorm** : $\|\nabla_\theta L_S(\theta_k)\|_2 := \| \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta l\big(f\big(x^i, \theta_k\big), y_i\big)\|_2$, and

$$\textbf{Noise} : \sigma(\theta_k) := \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\nabla L_S(\theta_k) - \nabla_\theta l(f(x^i, \theta_k), y^i)\|_2^2}$$

Mixup data valuation - Summary

1. The mix-up method followed by guide of Teacher mix-up model can benefit the generalization performance of Student model for several datasets.

2. To resolve deteriorate guide from Teacher at the end tail of epoch, Easy α scheduler is adopted, which might not be optimal strategy for Student model.

3. While mix-up boosts the generalization performance model, overtraining can lead to wrong optimal solution.

4. For the solution, Two papers claim that Early-stopping of mix-up is beneficial to reduce the above effect, and the success of mix-up can be attributed to the non-vanishing Gradient Norm, which is a tremendous gap between theory and practice.

Mixup data valuation (Ablations)

- Questions to be resolved:

  1. Does the accuracy gain come from ‘filtering’ strategy?
  2. If yes, what is the role of filtering in mix-up?
  3. How the roles can contribute to the accuracy gain?

- Question 1 : *Does the accuracy gain come from ‘filtering’ strategy?*

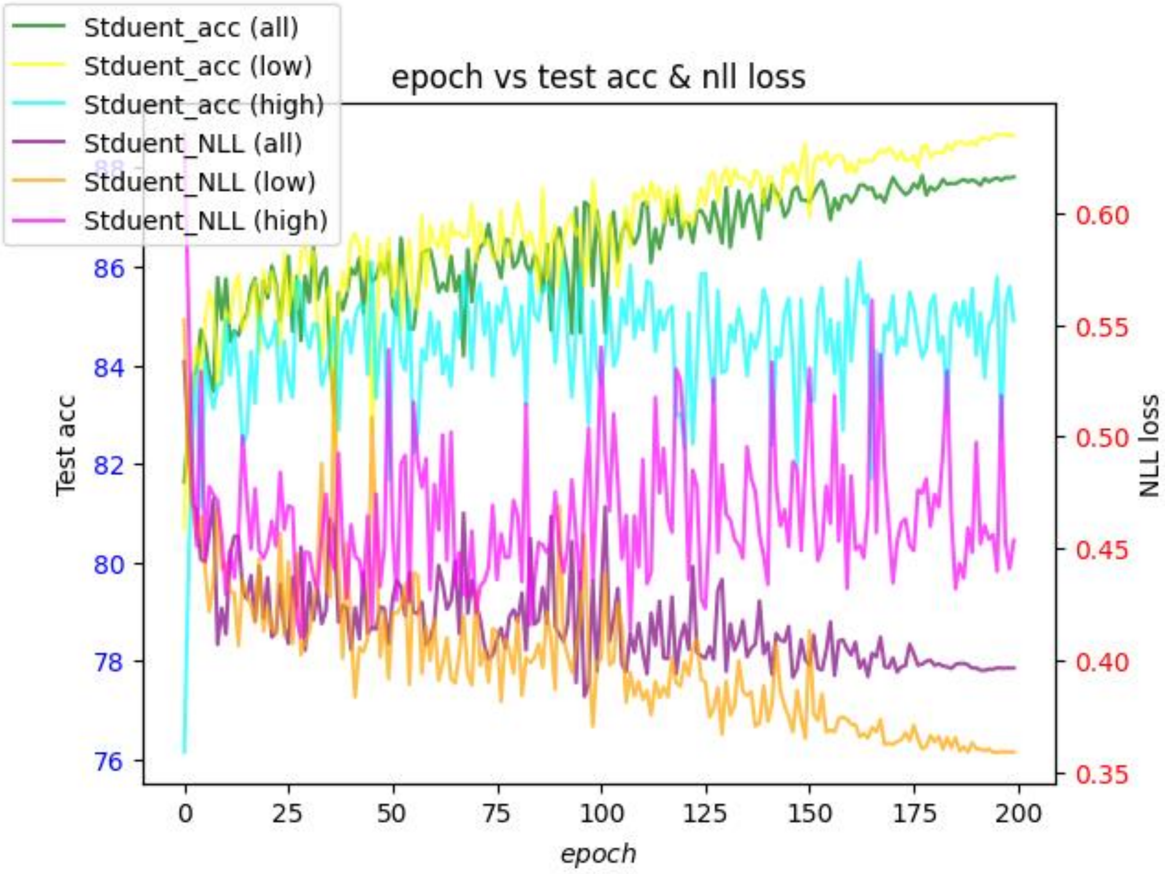
  - Method 1 (Ideal method) : when the Teacher is trained by (train set + test set)
  - Method 2 (Our method) : when the Teacher is only trained by (train set)

  → First, check whether the filtering effect exists or not in Method 1.

Mixup data valuation – Q1

- Experiment environment :

3-layer NN (786 – 300 – 100 – 10) w/ Cosine annealing lr scheduler | FashionMNIST (20%)

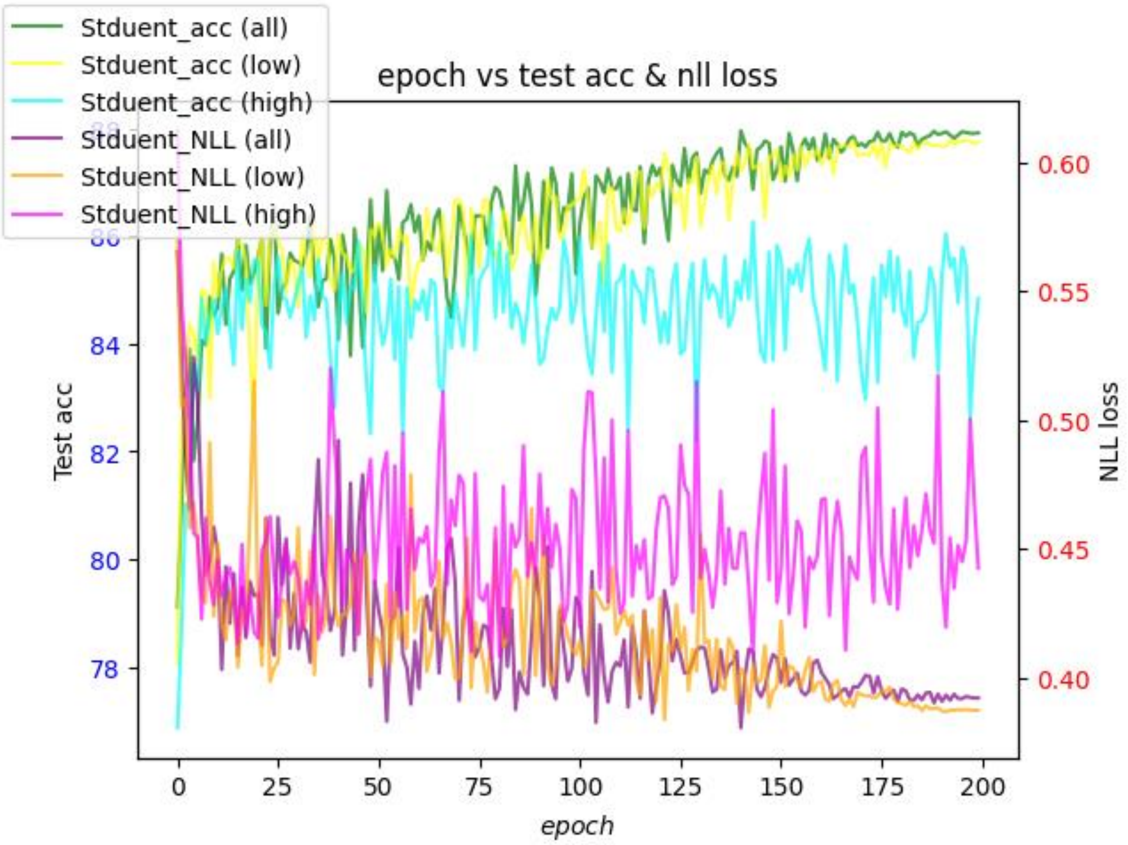


Results (Method 1):

- Teacher (100 ep) : 93.29% (acc) || 0.245 (NLL loss)
- Student w/ low filter outperforms student w/ high filter by 1% (acc), 0.04(NLL loss)
- Clearly, Student w/ high filter degrades even compared to the (original training) baseline (86.88 (acc), 0.51(NLL loss))
- This indicates when Method 1 (ideal Teacher) is adopted, the filter can guide mix-up strategy.

Mixup data valuation – Q1

- Method 2 : when the Teacher is only trained by (train set)



Results (Method 2):

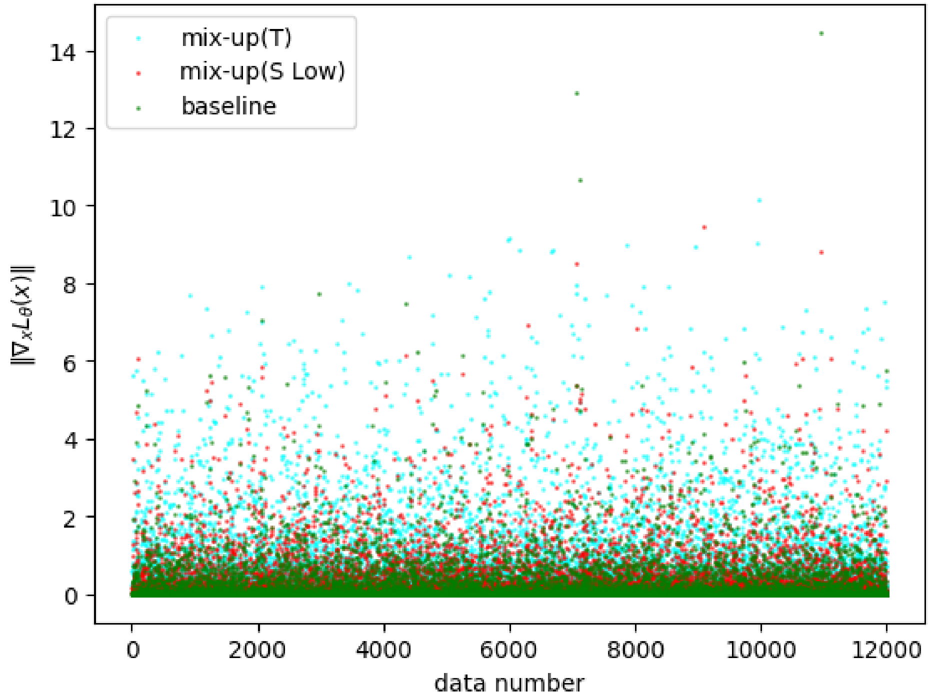
- Teacher (100 ep) : 87.27% (acc) || 0.397 (NLL loss)
- The low filtering begins to being not effective.
- But, clearly, Student w/ high filter degrades even compared to the (original training) baseline (86.88 (acc), 0.51(NLL loss))
- This indicates the Teacher model trained by train set only may not guide the mix-up strategy well.

Mixup data valuation - Q2

- Question 2 : *what is the role of filtering in mix-up?*
  - The main effect of mix-up is to regularize $\nabla_x f_\theta(x)$ and $\nabla_x^2 f_\theta$

Method 1	Mix-up (Teacher)	Guided (All)	Guided (Low)
Weight norm	18.7557	20.2712	19.9560
$\mathbb{E}_x[\lVert\nabla_x l(x,\theta)\rVert]$	0.7590	0.3177	0.3635

Method 2	Mix-up (Teacher)	Guided (All)	Guided (Low)
Weight norm	19.0546	20.2883	19.9803
$\mathbb{E}_x[\lVert\nabla_x l(x,\theta)\rVert]$	0.7472	0.3144	0.3542



- Does the mix-up indeed regularize input gradient in practice?
  - Maybe no; [MixupE, 2022] points out the regularization effect of mix-up can be wrong (+ suggest direct method to regularize 1st order regularization term, but the performance is worse than RegMixUp)

Mixup data valuation - Q2

- According to MixupE:

$$L_n^{mix}(\theta, S) = L_n^{std}(\theta, S) + \frac{\mathbb{E}_\lambda[a(\lambda)]}{n} \sum_{i=1}^{n} \left(g\left(f_\theta(x_i)\right) - y_i\right)^T \nabla f_\theta(x_i)(\bar{x} - x_i) + 2^{nd}\ term$$

- Problem:

Note $q(x_i) := \left(g\left(f_\theta(x_i)\right) - y_i\right)^T \nabla f_\theta(x_i)(\bar{x} - x_i)$

$= \sum_{k=1}^{d} \alpha_{k,i} \|\nabla f_k(x_i)\|_2 \|\bar{x} - x_i\|_2$

But, $\alpha_{k,i}$ can be negative in practice,

which leads to maximize $\|\nabla f_k(x_i)\|_2$.

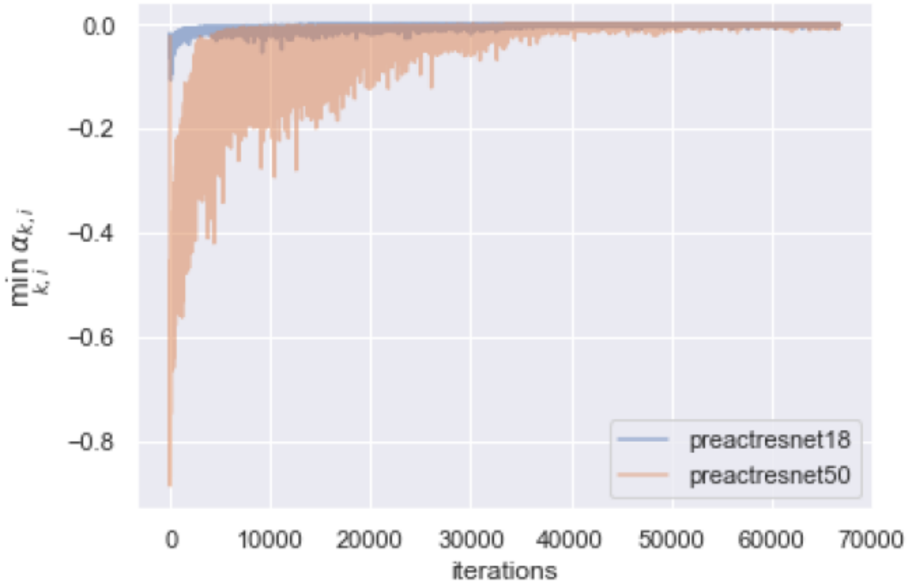


Figure 2: Minimum value of α over the coordinate k and sample i for different iterations during the training.

Mixup data valuation - Q2

- Then, how [Zhang, 2021] explains the generalization performance of Mixup?

Lemma 3.1. *Consider the loss function $l(\theta,(x,y)) = h(f_\theta(x)) - yf_\theta(x)$, where $h(\cdot)$ and $f_\theta(\cdot)$ for all $\theta \in \Theta$ are twice differentiable. We further denote $\tilde{\mathcal{D}}_\lambda$ as a uniform mixture of two Beta distributions, i.e., $\frac{\alpha}{\alpha+\beta}Beta(\alpha+1,\beta)+\frac{\beta}{\alpha+\beta}Beta(\beta+1,\alpha)$, and $\mathcal{D}_X$ as the empirical distribution of the training dataset $S=(x_1,\cdots,x_n)$, the corresponding Mixup loss $L_n^{mix}(\theta,S)$, as defined in Eq. (1) with $\lambda \sim D_\lambda = Beta(\alpha,\beta)$, can be rewritten as*

$$L_n^{mix}(\theta,S) = L_n^{std}(\theta,S) + \sum_{i=1}^{3}\mathcal{R}_i(\theta,S) + \mathbb{E}_{\lambda\sim\tilde{\mathcal{D}}_\lambda}[(1-\lambda)^2\varphi(1-\lambda)],$$

where $\lim_{a\to 0}\varphi(a)=0$ *and*

$$\mathcal{R}_1(\theta,S) = \frac{\mathbb{E}_{\lambda\sim\tilde{\mathcal{D}}_\lambda}[1-\lambda]}{n}\sum_{i=1}^{n}(h'(f_\theta(x_i))-y_i)\nabla f_\theta(x_i)^\top\mathbb{E}_{r_x\sim\mathcal{D}_X}[r_x-x_i],$$

$$\mathcal{R}_2(\theta,S) = \frac{\mathbb{E}_{\lambda\sim\tilde{\mathcal{D}}_\lambda}[(1-\lambda)^2]}{2n}\sum_{i=1}^{n}h''(f_\theta(x_i))\nabla f_\theta(x_i)^\top\mathbb{E}_{r_x\sim\mathcal{D}_X}[(r_x-x_i)(r_x-x_i)^\top]\nabla f_\theta(x_i),$$

$$\mathcal{R}_3(\theta,S) = \frac{\mathbb{E}_{\lambda\sim\tilde{\mathcal{D}}_\lambda}[(1-\lambda)^2]}{2n}\sum_{i=1}^{n}(h'(f_\theta(x_i))-y_i)\mathbb{E}_{r_x\sim\mathcal{D}_X}[(r_x-x_i)\nabla^2 f_\theta(x_i)(r_x-x_i)^\top].$$

Lemma 3.3. *Consider the centralized dataset S, that is, $1/n\sum_{i=1}^{n}x_i = 0$. and denote $\hat{\Sigma}_X = \frac{1}{n}x_ix_i^\top$. For a GLM, if $A(\cdot)$ is twice differentiable, then the regularization term obtained by the second-order approximation of $\tilde{L}_n^{mix}(\theta,S)$ is given by*

$$\frac{1}{2n}[\sum_{i=1}^{n}A''(\theta^\top x_i)]\cdot\mathbb{E}_{\lambda\sim\tilde{\mathcal{D}}_\lambda}[\frac{(1-\lambda)^2}{\lambda^2}]\theta^\top\hat{\Sigma}_X\theta, \tag{7}$$

where $\tilde{\mathcal{D}}_\lambda = \frac{\alpha}{\alpha+\beta}Beta(\alpha+1,\beta)+\frac{\alpha}{\alpha+\beta}Beta(\beta+1,\alpha)$.

Note:

- $A(\theta^T x) = \log(1+e^{\theta^T x})$ for logistic loss, which has $A''(\theta^T x) > 0$ always.
- By lemma, we can consider the following function class:

  $\mathcal{W}_\gamma \coloneqq \{x \to \theta^T x : \mathbb{E}_x[A''(\theta^T x)\cdot\theta^T\Sigma_X\theta] \le \gamma\}$

Mixup data valuation - Q2

- Then, how [Zhang, 2021] explains the generalization performance of Mixup?

Theorem 3.4. *Assume that the distribution of x_i is ρ-retentive, and let $\Sigma_X = \mathbb{E}[xx^\top]$. Then the empirical Rademacher complexity of $\mathcal{W}_\gamma$ satisfies*

$$Rad(\mathcal{W}_\gamma, S) \leq \max\{(\frac{\gamma}{\rho})^{1/4}, (\frac{\gamma}{\rho})^{1/2}\} \cdot \sqrt{\frac{rank(\Sigma_X)}{n}}.$$

Note:

- ρ-retentive : for any non-zero vector v, $[\mathbb{E}_x[A''(x^T v)]]^2 \geq \rho \cdot \min\{1, \mathbb{E}_x(v^T x)^2\}$ (achievable if weight is bounded)
- Compare to the baseline function class : $\mathcal{W}_\gamma^{ridge} := \{x \to \theta^T x : \|\theta\|^2 \leq \gamma\}$ (achieved by l2 regularization):

$$Rad\left(\mathcal{W}_r^{ridge}, S\right) \leq \max\left\{\left(\frac{\gamma}{\rho}\right)^{\frac{1}{4}}, \left(\frac{\gamma}{\rho}\right)^{\frac{1}{2}}\right\} \cdot \sqrt{\frac{p}{n}}$$

- If $rank(\Sigma_X) \leq p$ is much smaller than p, then the mix-up strategy can generalized better than l2 regularization.
- One explainable method via this theorem is to use contrastive learning and apply mix-up at linear evaluation.

Mixup data valuation – Q2

- Then, what is the good explanation for improved generalization performance in mix-up?

⇒ [D.Zou, 2023] proved that the early-stop of mix-up can be helpful for efficient learning of rare features in a given dataset.

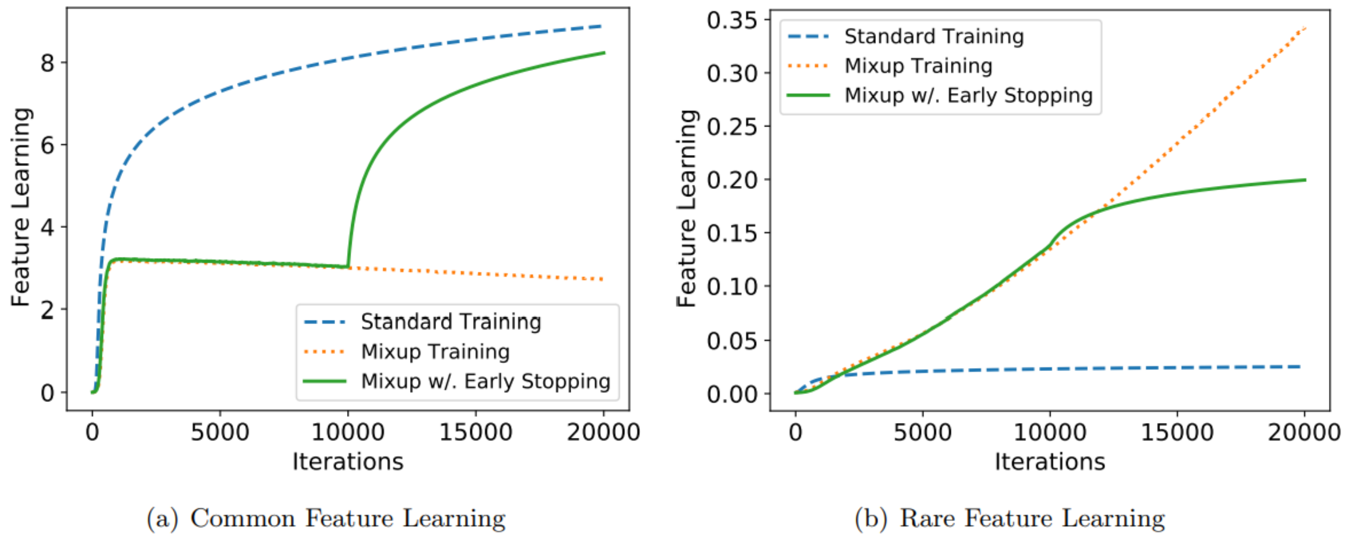


Figure 2: Common feature learning and rare feature learning on synthetic data, all experiments are conducted using full-batch gradient descent. Here we consider three training methods: standard training, Mixup training, and Mixup training with early stopping (at the 10000-th iteration).

Model : 2-layer CNN w/ logit:

$$F_k(W;x) = \sum_{p=1}^{P}\sum_{r=1}^{m}\left(< w_{k,r}, x^{(p)} >\right)^2$$

where $x = \left(x^{(1)}, \ldots, x^{(P)}\right) \in \mathbb{R}^{d\times P}$, and m = network width

Feature learning metric:

$\sum_{r=1}^{m}\left(< w_{1,r}, v >\right)^2$ (Common)

$\sum_{r=1}^{m}\left(< w_{1,r}, v' >\right)^2$ (Rare)

Mixup data valuation – Q2

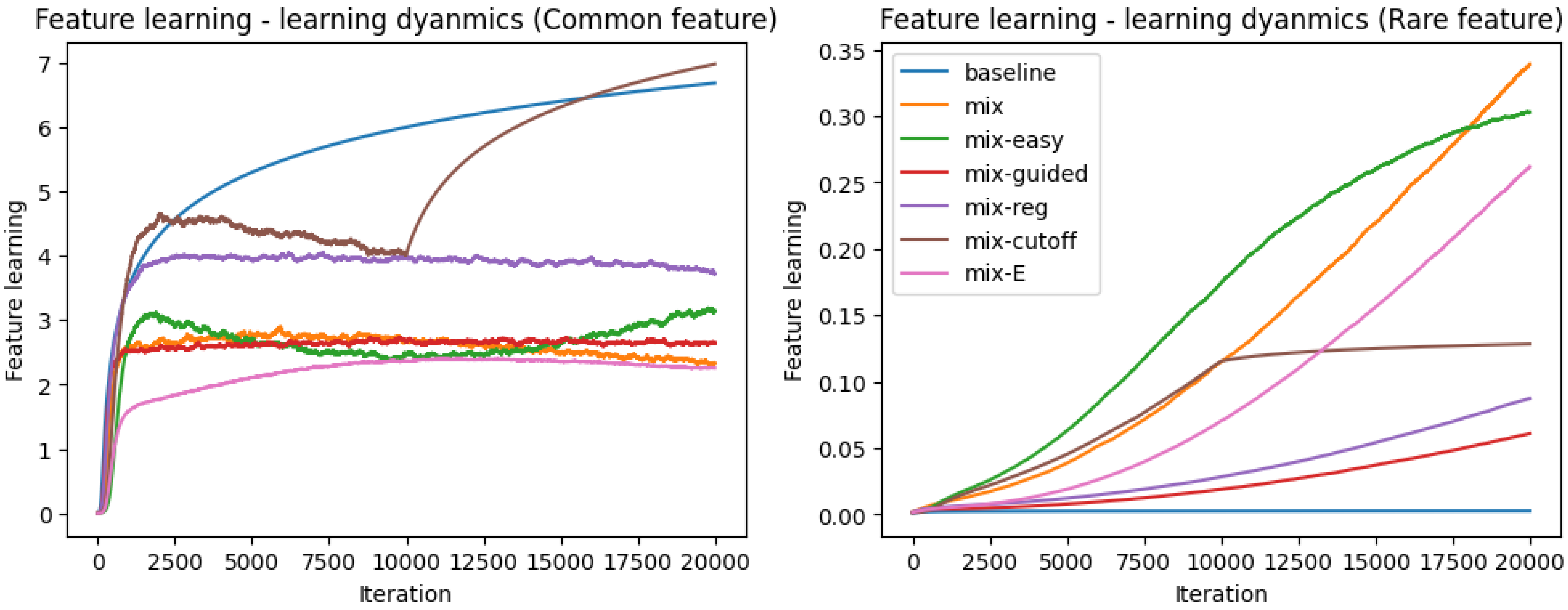
- (Side-Note) How to generate data?

Definition 3.1. Let $\mathcal{D}$ denote the data distribution, from which a data point $(\mathbf{x}, y) \in \mathbb{R}^{dP} \times \{1, 2\}$ is randomly generated as follows:

1. Generate $y \in \{1, 2\}$ uniformly.
2. Generate $\mathbf{x}$ as a vector with P patches $\mathbf{x} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(2)}) \in (\mathbb{R}^d)^P$, where
   - **Feature Patch.** One patch, among all P patches, will be randomly selected as the feature patch: with probability $1 - \rho$ for some $\rho \in (0, 1)$, this patch will contain a *common feature* ($\mathbf{v}$ for positive data, $\mathbf{u}$ for negative data); otherwise, this patch will contain a *rare feature* ($\mathbf{v}'$ for positive data, $\mathbf{u}'$ for negative data).
   - **Feature Noise.** For all data, a feature vector from $\alpha \cdot \{\mathbf{u}, \mathbf{v}\}$ is randomly sampled and assigned to up to b patches.
   - **Noise patch.** The remaining patches (those haven't been assigned with a feature or feature noise) are random Gaussian noise $\sim N(\mathbf{0}, \sigma_p^2 \cdot \mathbf{H})$, where $\mathbf{H} = \mathbf{I} - \frac{\mathbf{u}\mathbf{u}^\top}{\|\mathbf{u}\|_2^2} - \frac{\mathbf{v}\mathbf{v}^\top}{\|\mathbf{v}\|_2^2} - \frac{\mathbf{v}'\mathbf{v}'^\top}{\|\mathbf{v}'\|_2^2} - \frac{\mathbf{u}'\mathbf{u}'^\top}{\|\mathbf{u}'\|_2^2}$.

Mixup data valuation – Q2

- How about our current algorithms' feature learning performance?



- Crucial to find a method to improve both common feature / rare feature learning.

  (+ guided method fails in terms of feature learning)