Learning with Noisy Labels by Adaptive GRAdient-Based Outlier Removal

Anastasiia Sedova*, Lena Zellinger*, Benjamin Roth

Data Science and Machine Learning Research Group, University of Vienna

ECML PKDD 2023

19.09.2023

* Equal contribution

Motivation

- To train a stable and well-performing model, we need some labeled data
  - … a *good amount* of labeled data …
  - … a *good amount* of *clean* labeled data.

- A way to obtain the labeled data more easily and cheaper is **automatic data labeling** (e.g. with weak supervision) - but on the cost of increased amount of noise.

  Ratner et al. 2016. Data programming: Creating large training sets, quickly.

- Even manually labeled data contains noise.

  Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.

Motivation

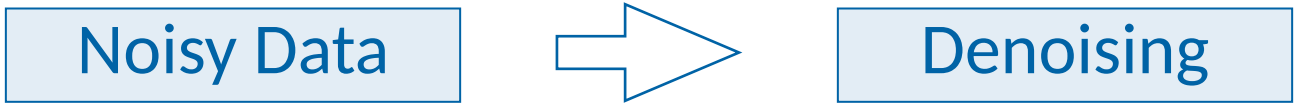
- To train a stable and well-performing model, we need some labeled data
  - … a *good amount* of labeled data …
  - … a *good amount* of *clean* labeled data.

- A way to obtain the labeled data more easily and cheaper is **automatic data labeling** (e.g. with weak supervision) - but on the cost of increased amount of noise.

Ratner et al. 2016. Data programming: Creating large training sets, quickly.

- Even manually labeled data contains noise.

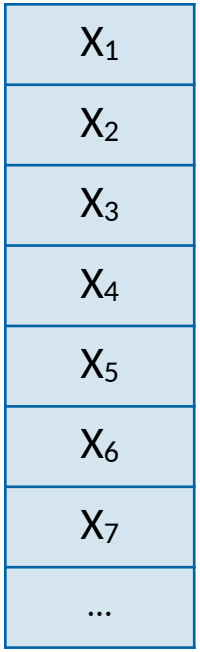
Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.



Background: Usual (Static) Denoising

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

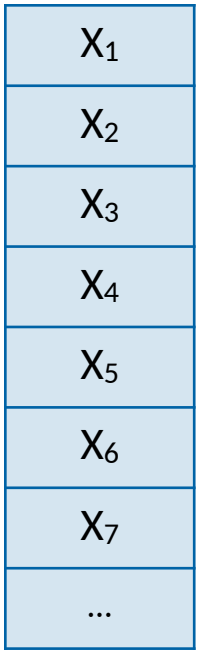
Background: Usual (Static) Denoising



Training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

Background: Usual (Static) Denoising

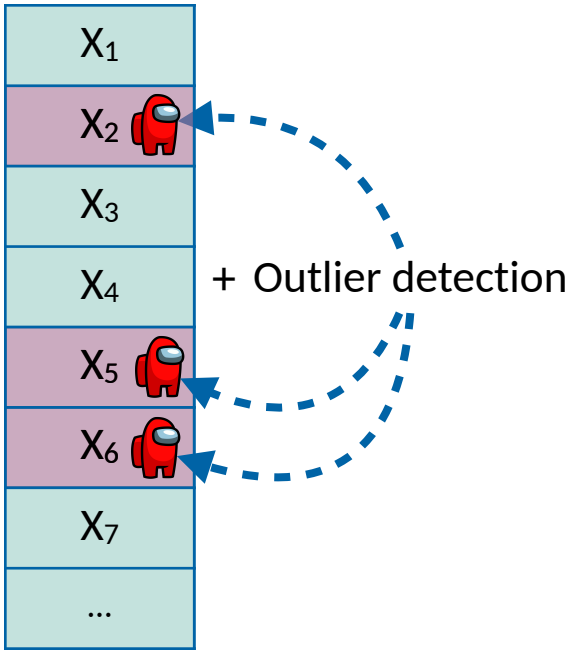


+ Outlier detection

Training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

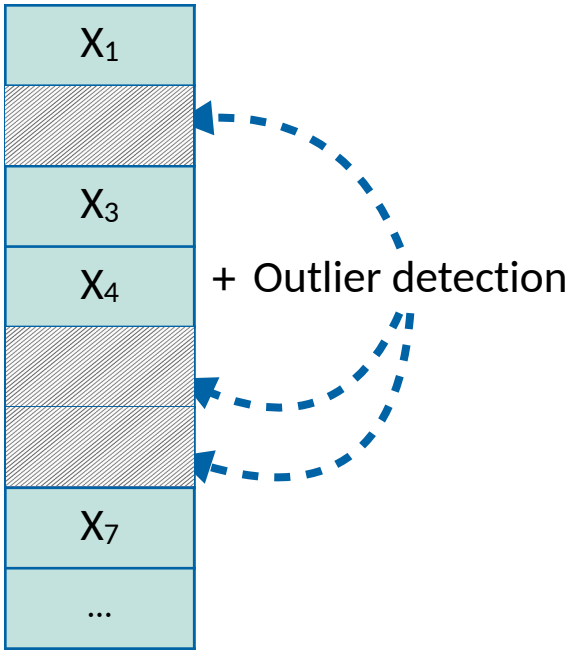
Background: Usual (Static) Denoising



Training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

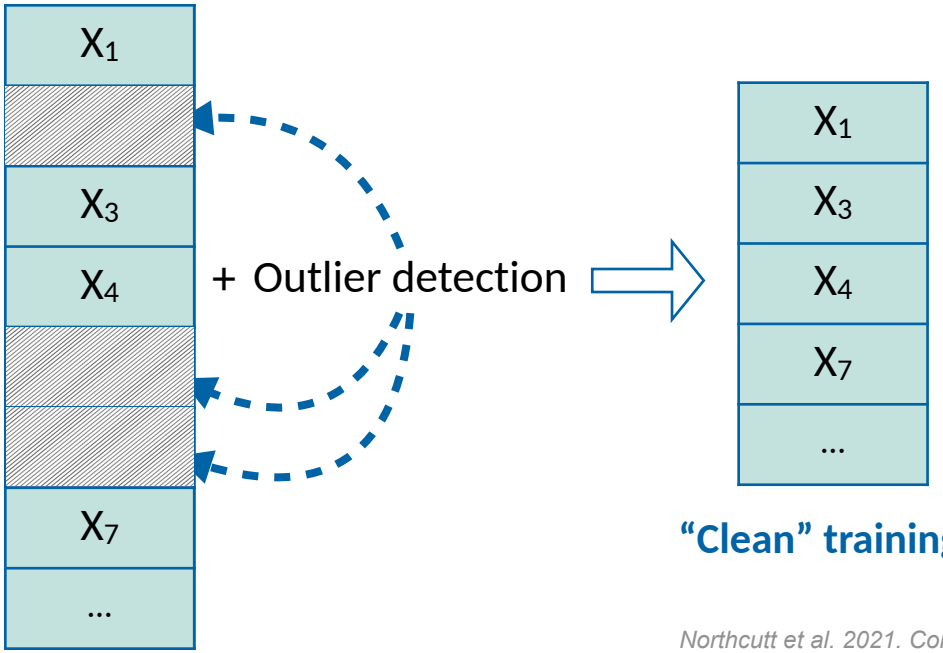
Background: Usual (Static) Denoising



Training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

Background: Usual (Static) Denoising

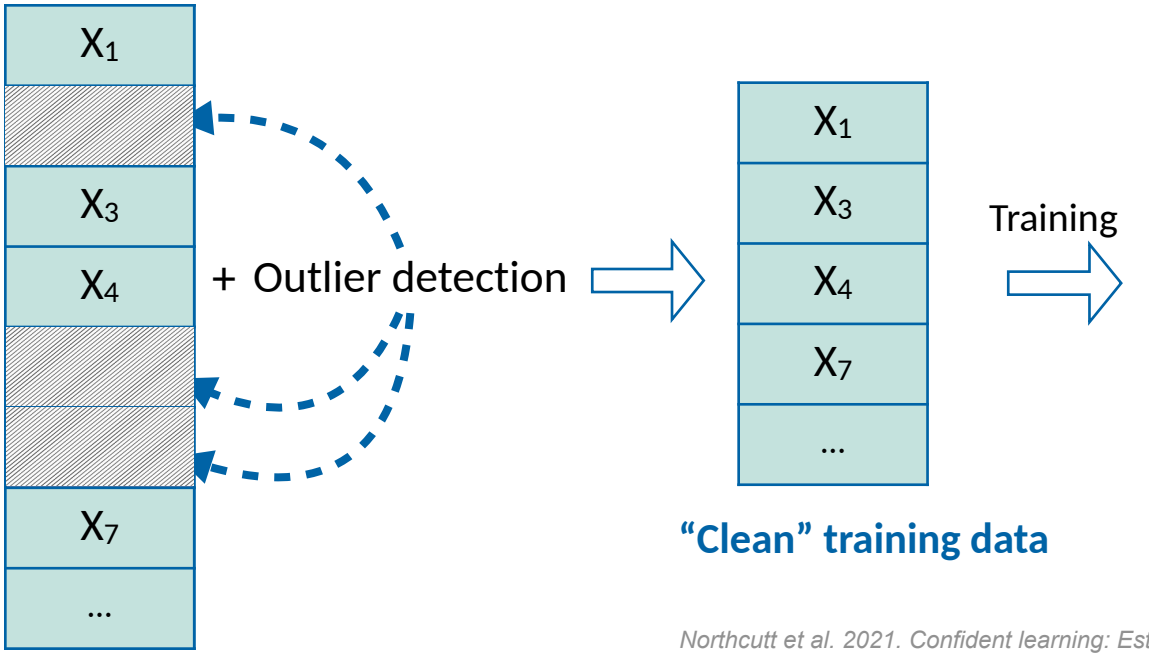


Training data

“Clean” training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

Background: Usual (Static) Denoising

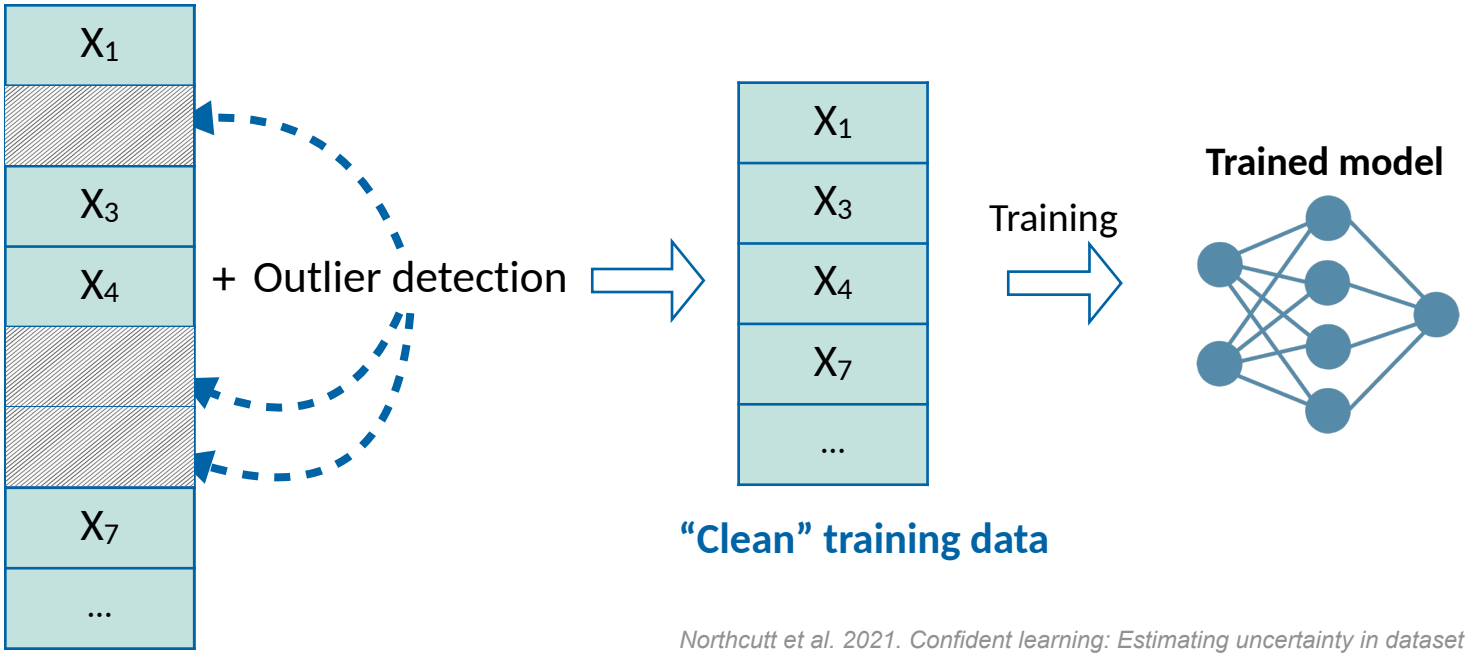


Training data

“Clean” training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

Background: Usual (Static) Denoising



Training data

"Clean" training data

Northcutt et al. 2021. Confident learning: Estimating uncertainty in dataset labels.
Huang et al. 2019. O2u-net: A simple noisy label detection approach for deep neural networks.
Chen et al. 2019. Understanding and utilizing deep neural networks trained with noisy labels.
Lipton et al. 2018. Detecting and correcting for label shift with black box predictors.

Our (Dynamic) Approach

Goal: To train a reliable model, not detect noisy samples

Our (Dynamic) Approach

Goal: To train a reliable model, not detect noisy samples

Our (Dynamic) Approach

- Correctness ≠ usefulness

Our (Dynamic) Approach

- Correctness ≠ usefulness
- *The movie was by no means great.*– POSITIVE
  A model that does not know anything about sentiment prediction might learn the useful association between the word great and the class POSITIVE.

Our (Dynamic) Approach

- Correctness ≠ usefulness
- *The movie was by no means great.*– POSITIVE
  A model that does not know anything about sentiment prediction might learn the useful association between the word great and the class POSITIVE.
- The same sample can be harmful when the model knows about negation.

Our (Dynamic) Approach

- Correctness ≠ usefulness
- *The movie was by no means great.*– POSITIVE
  A model that does not know anything about sentiment prediction might learn the useful association between the word great and the class POSITIVE.
- The same sample can be harmful when the model knows about negation.
- One sample can be **beneficial** for one model (or one stage of the model) but **harmful** for another.

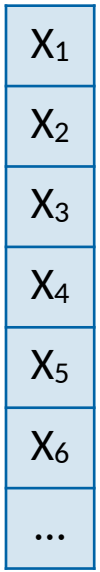
Our (Dynamic) Approach

- Correctness ≠ usefulness
- *The movie was by no means great.*– POSITIVE
  A model that does not know anything about sentiment prediction might learn the useful association between the word great and the class POSITIVE.
- The same sample can be harmful when the model knows about negation.
- One sample can be **beneficial** for one model (or one stage of the model) but **harmful** for another.

Instead of *static* removal of samples **before** training, we *dynamically* adjust the training set **during** training.

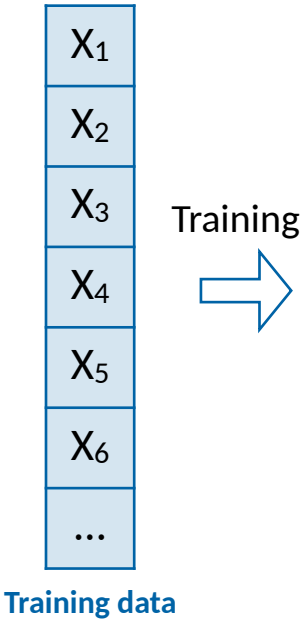
Dynamic Denoising

Dynamic Denoising

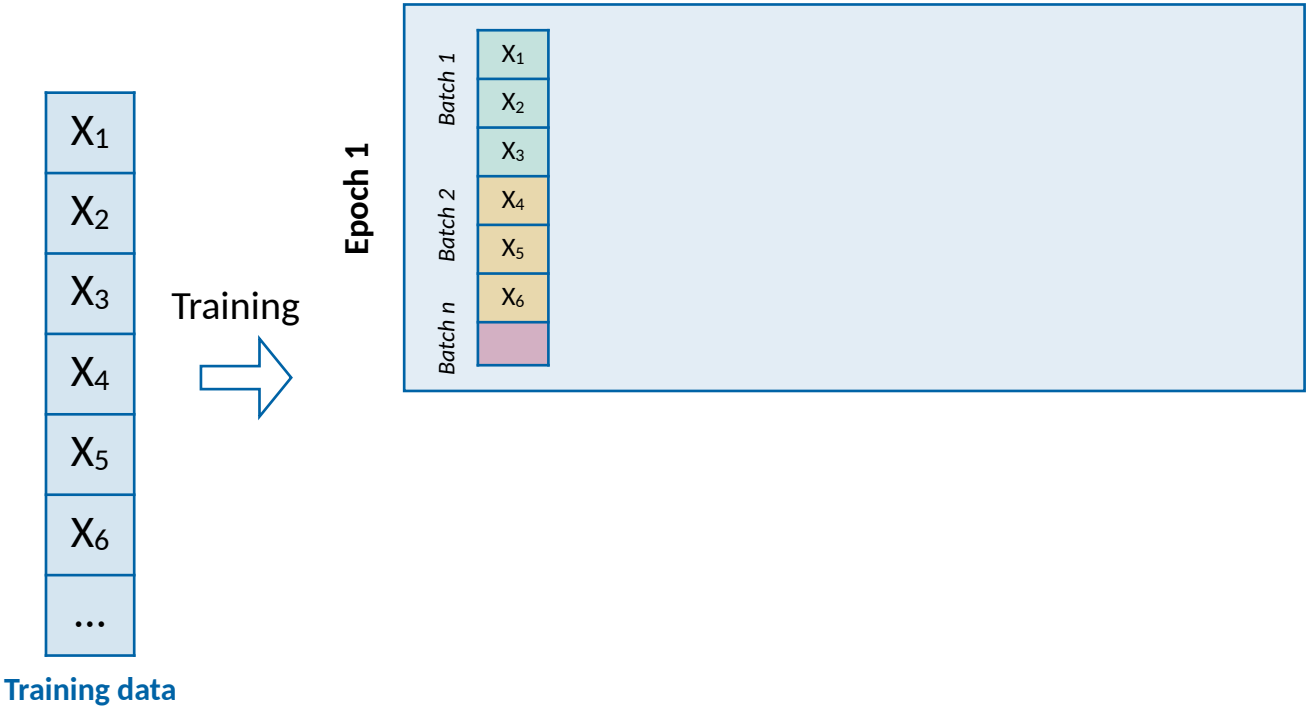


Training data

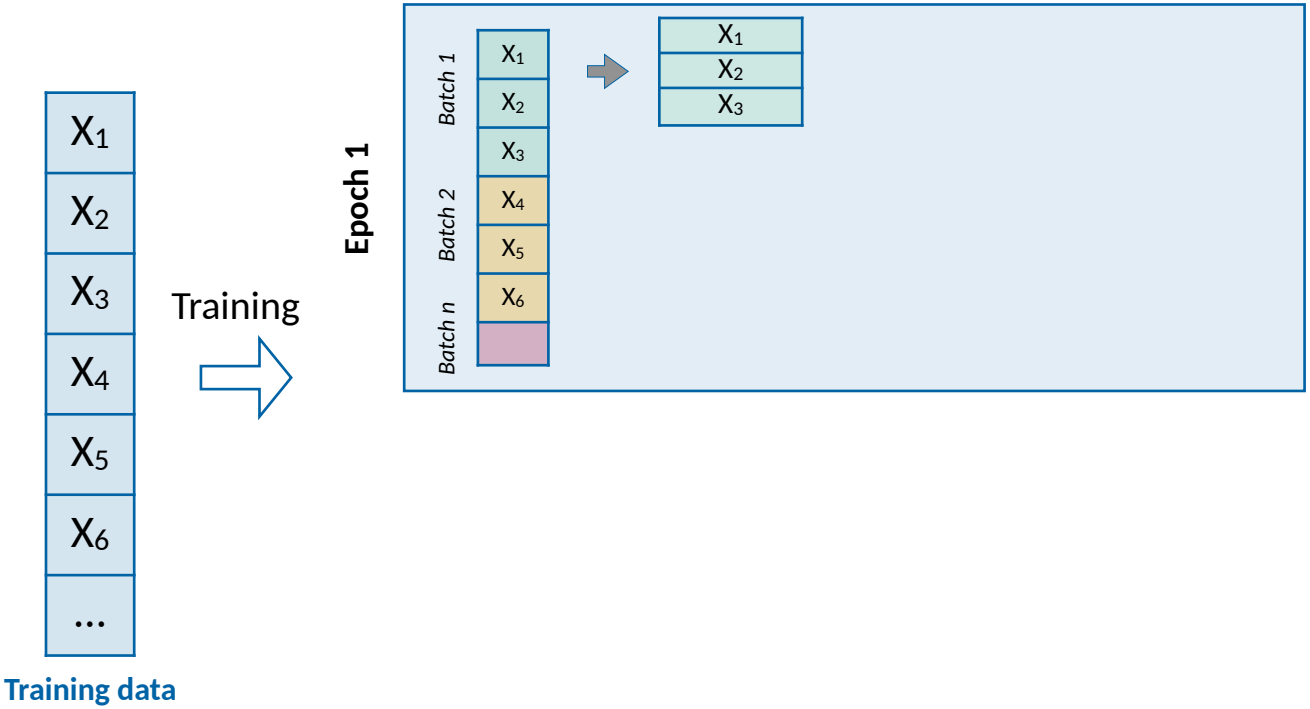
Dynamic Denoising



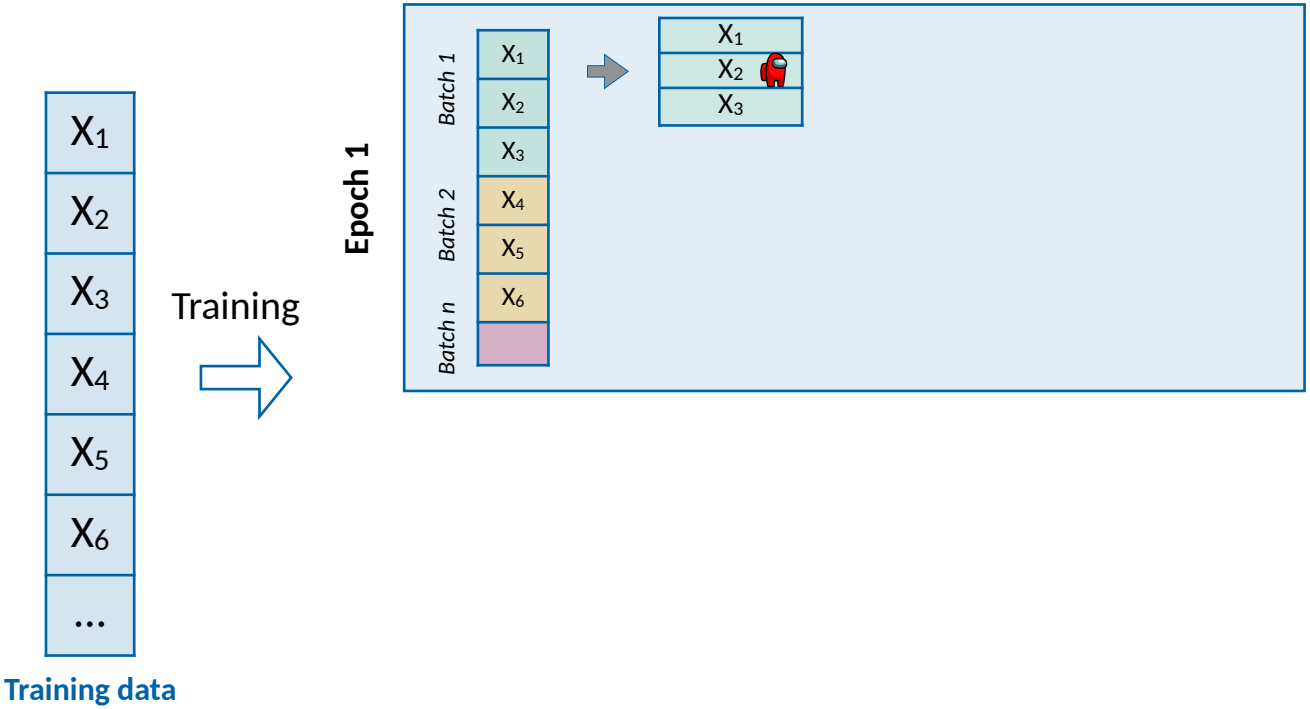
Dynamic Denoising



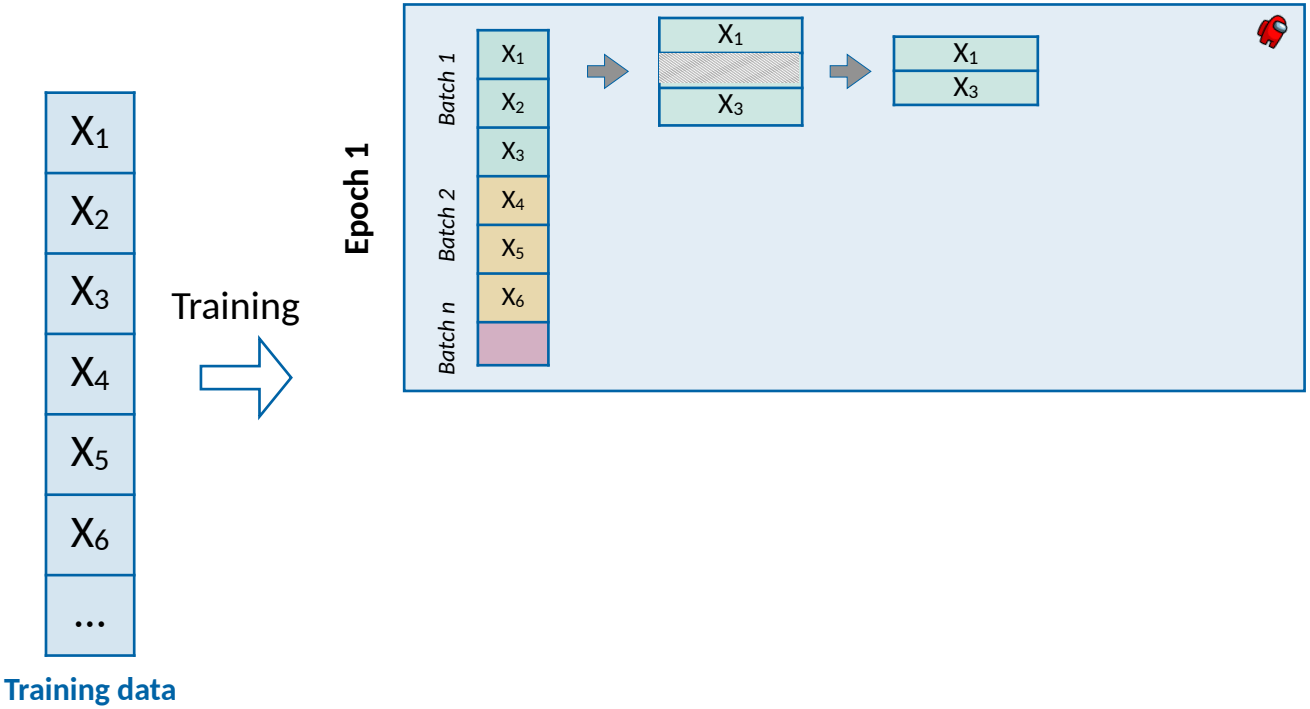
Dynamic Denoising



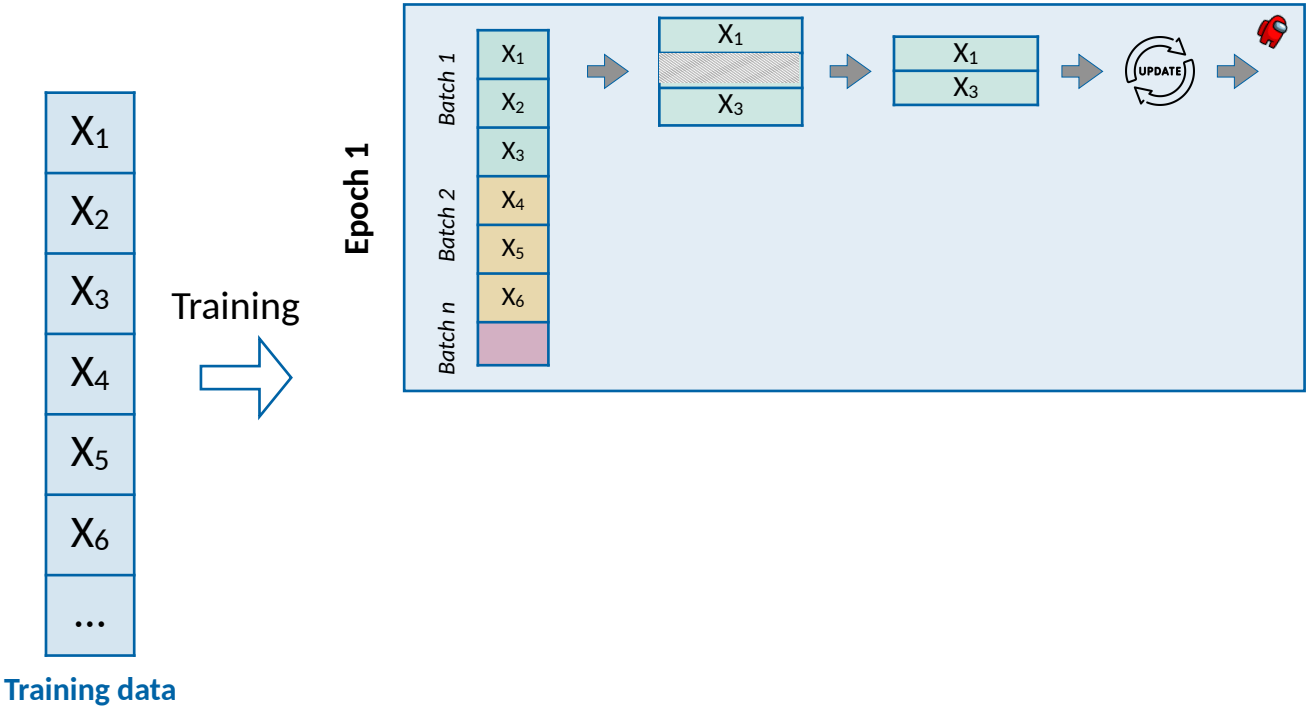
Dynamic Denoising



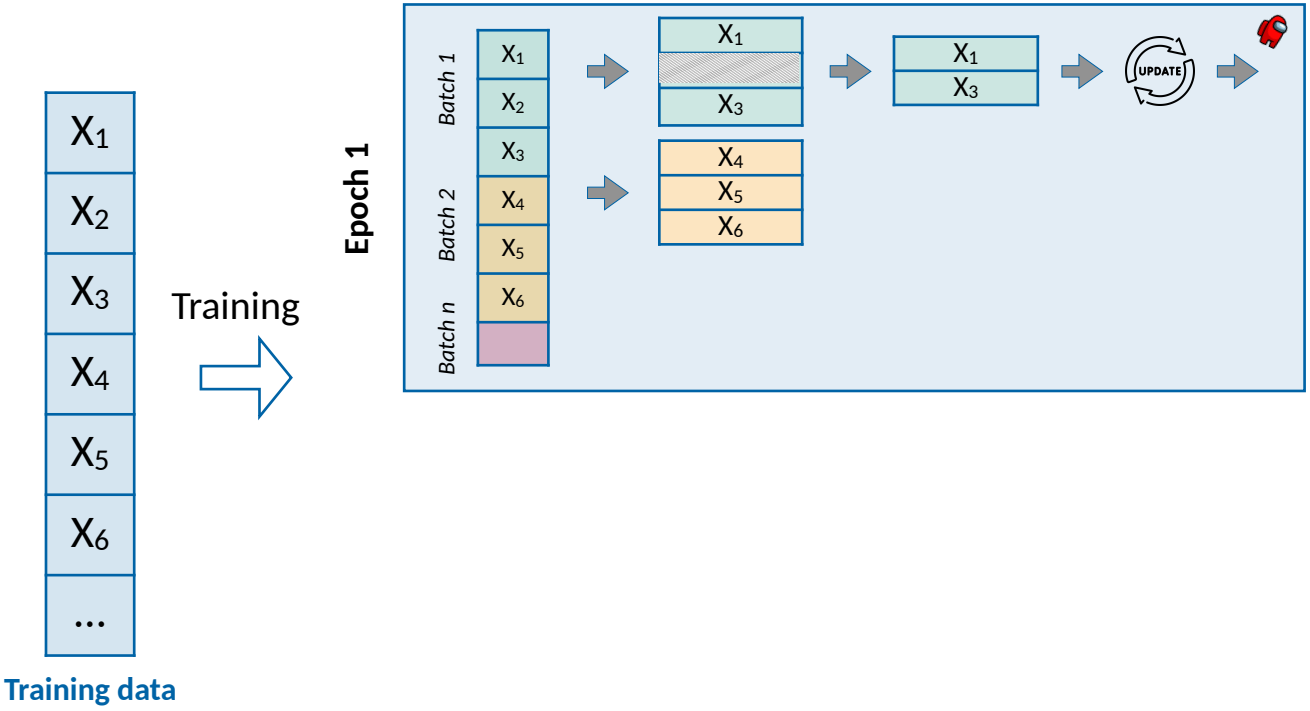
Dynamic Denoising



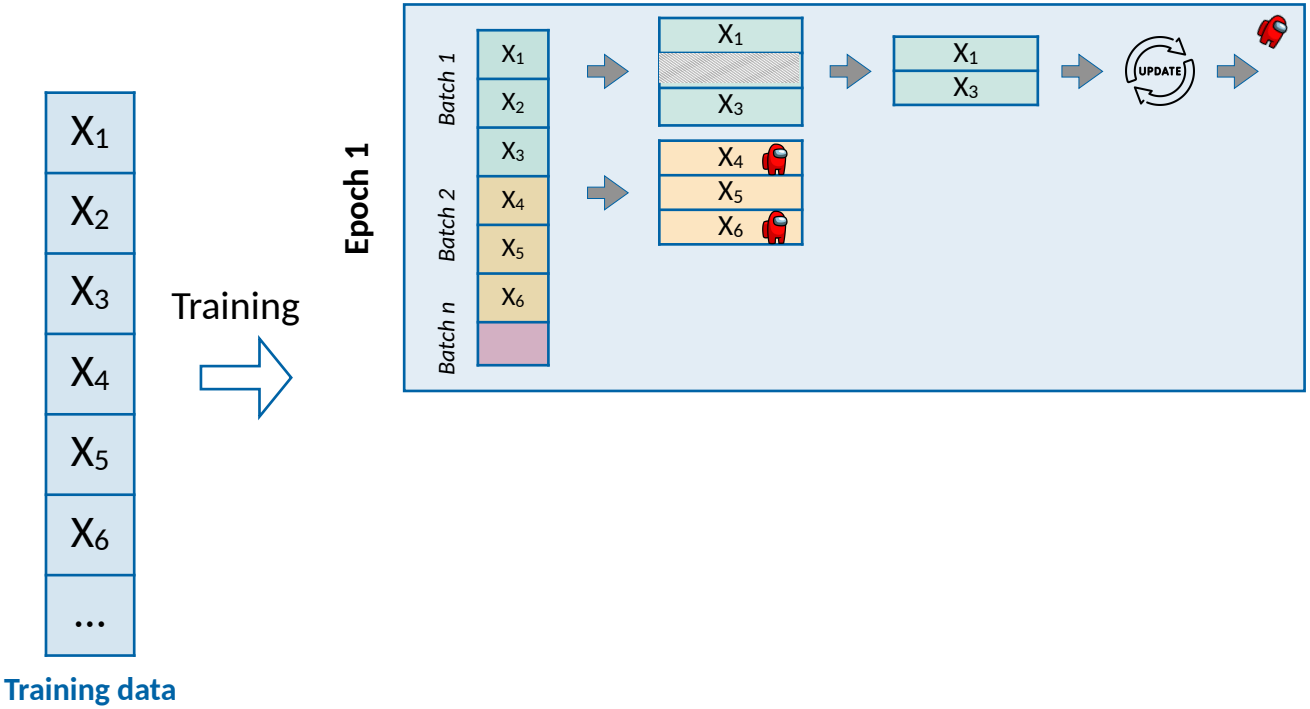
Dynamic Denoising



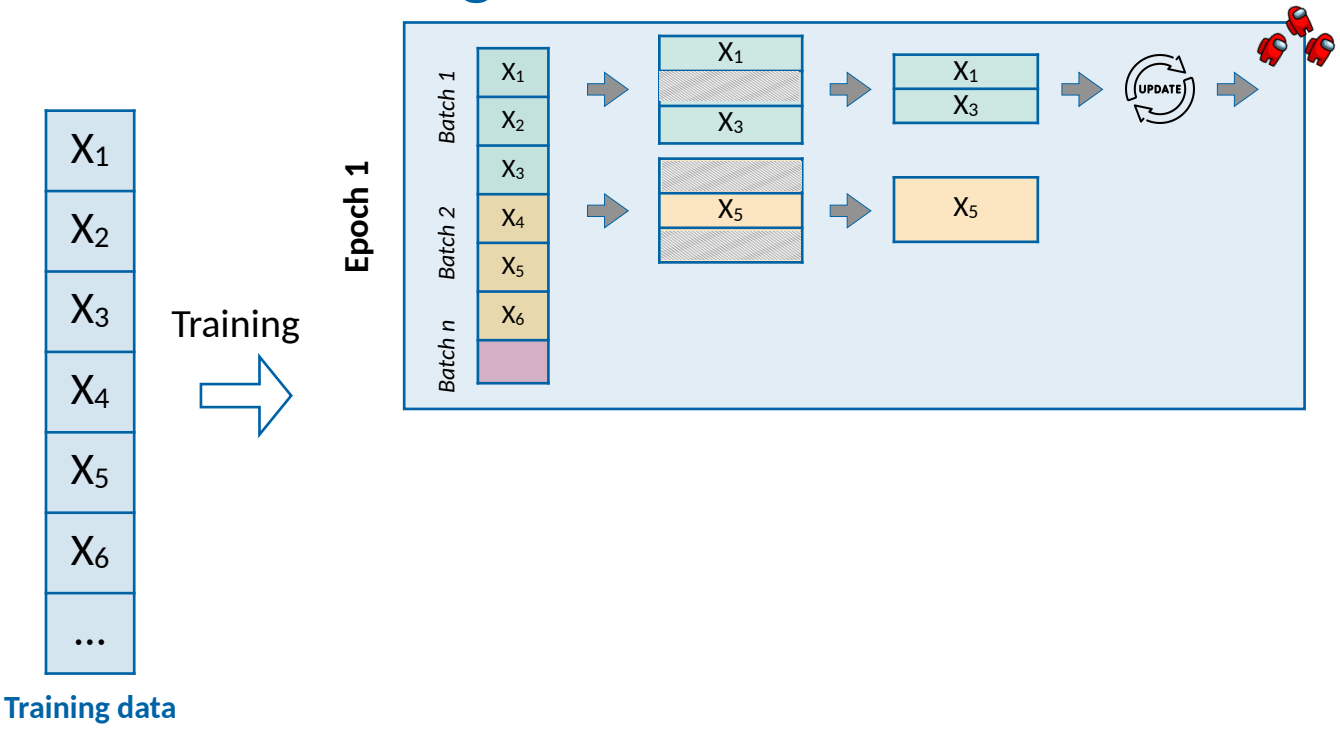
Dynamic Denoising



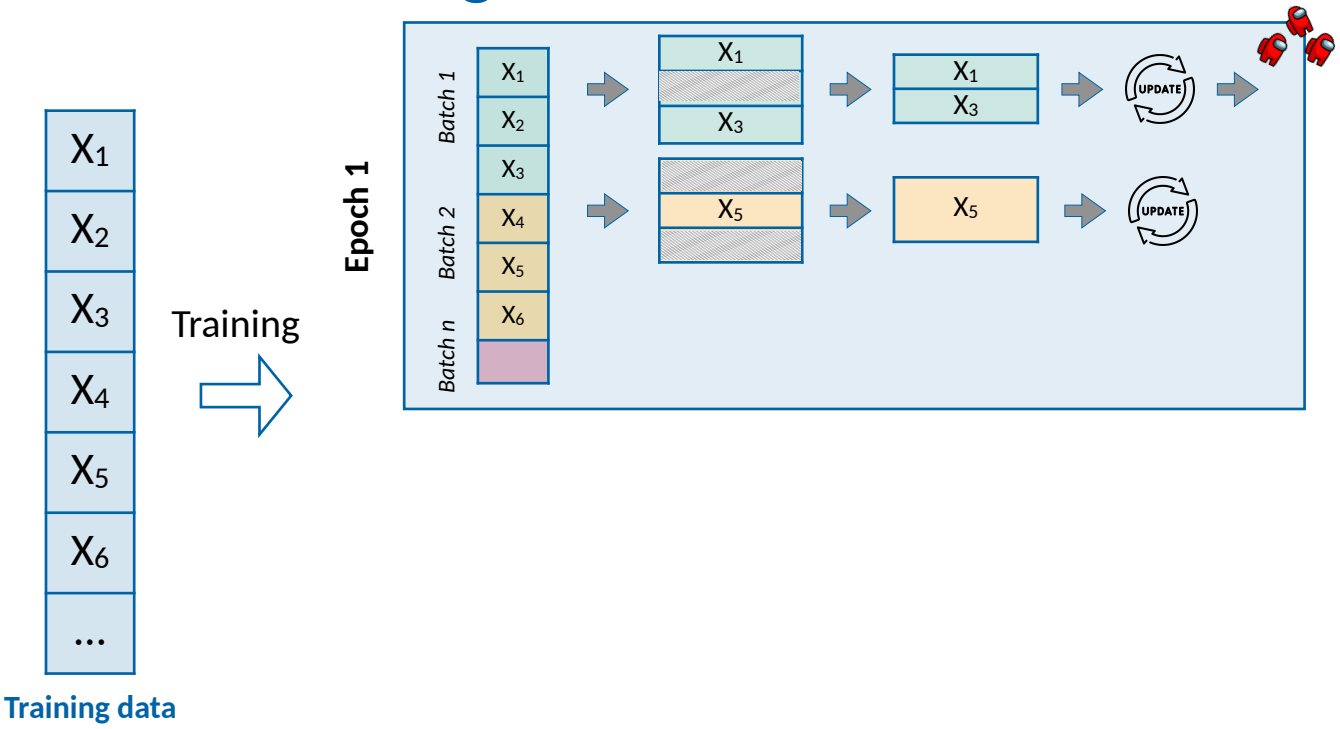
Dynamic Denoising



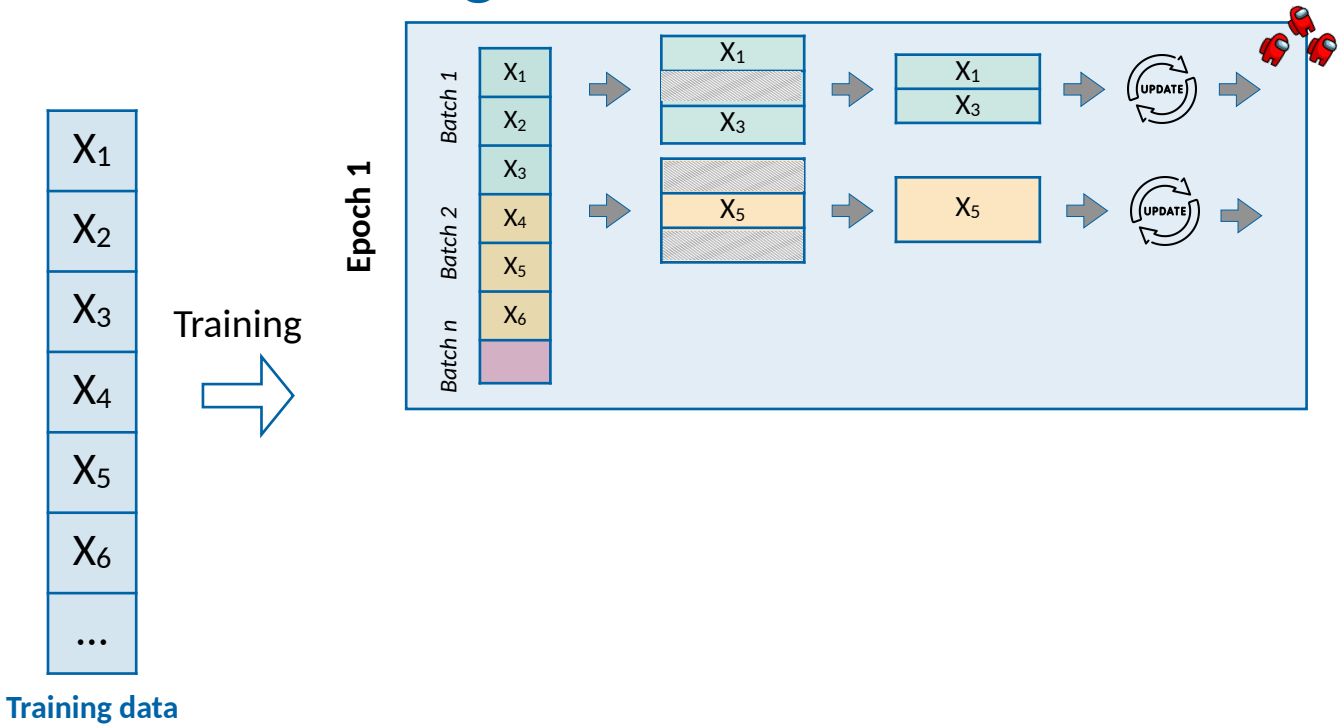
Dynamic Denoising



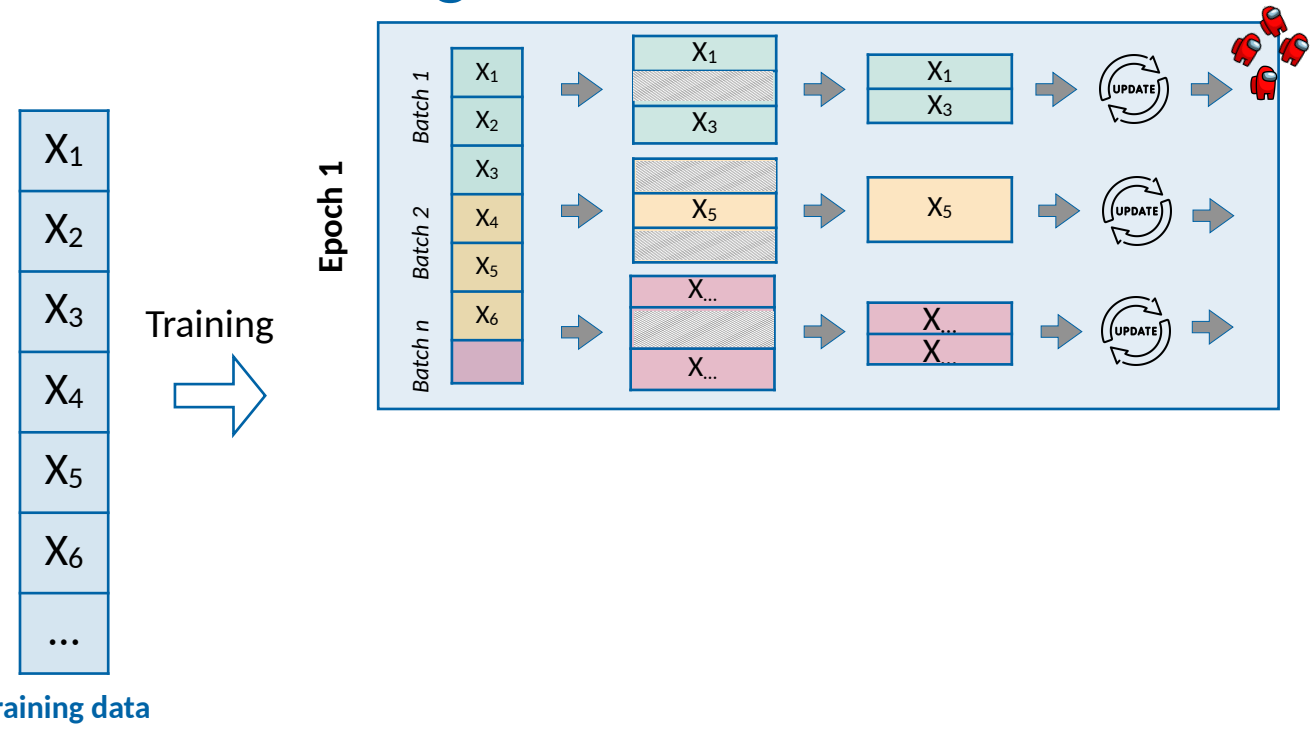
Dynamic Denoising



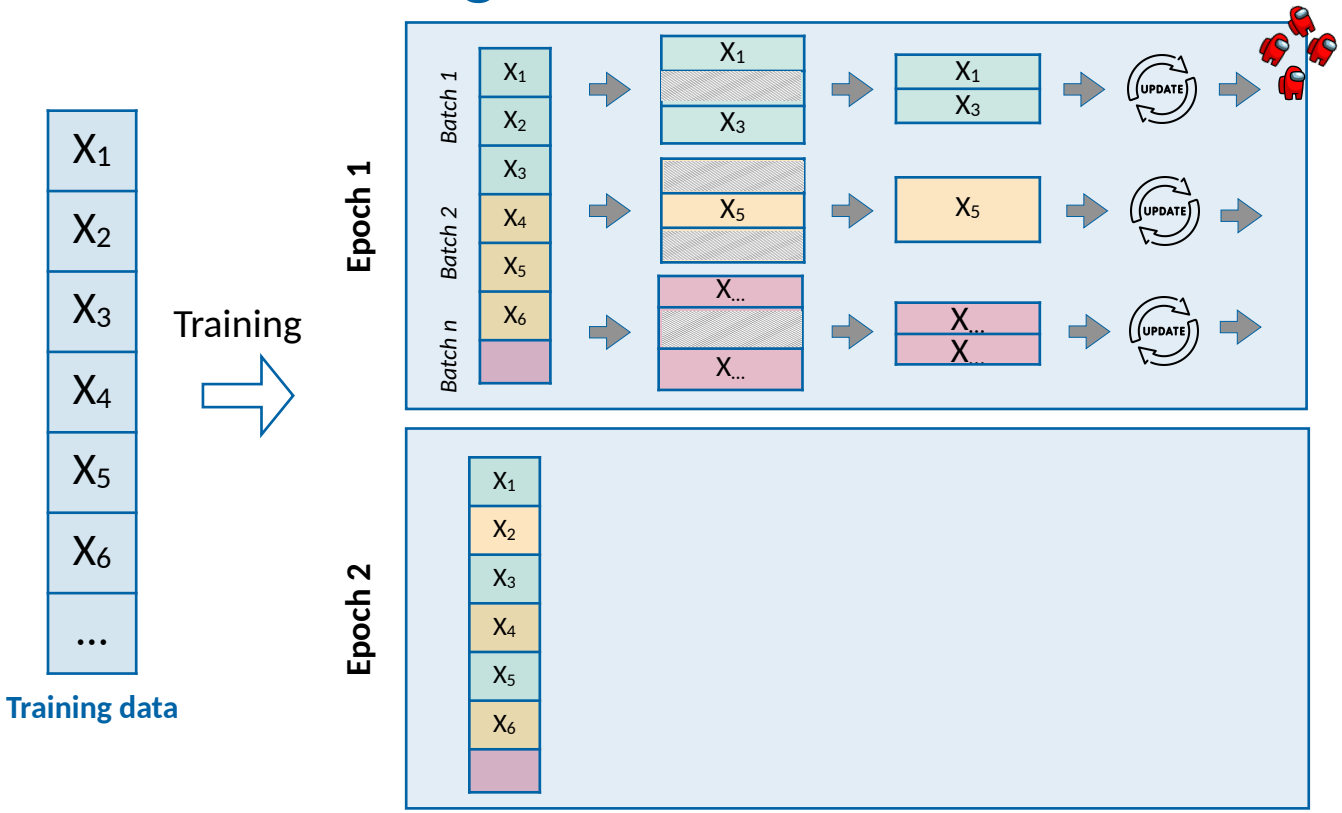
Dynamic Denoising



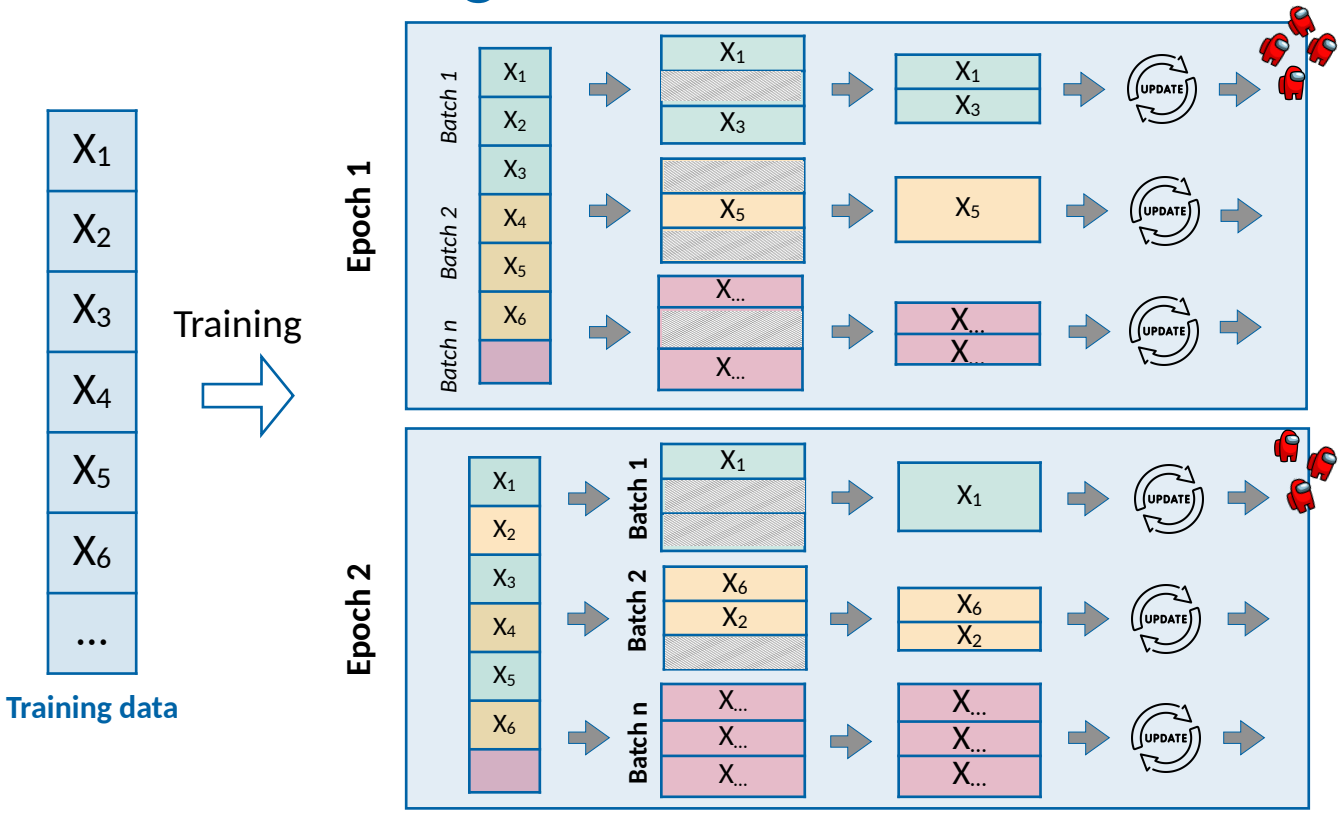
Dynamic Denoising



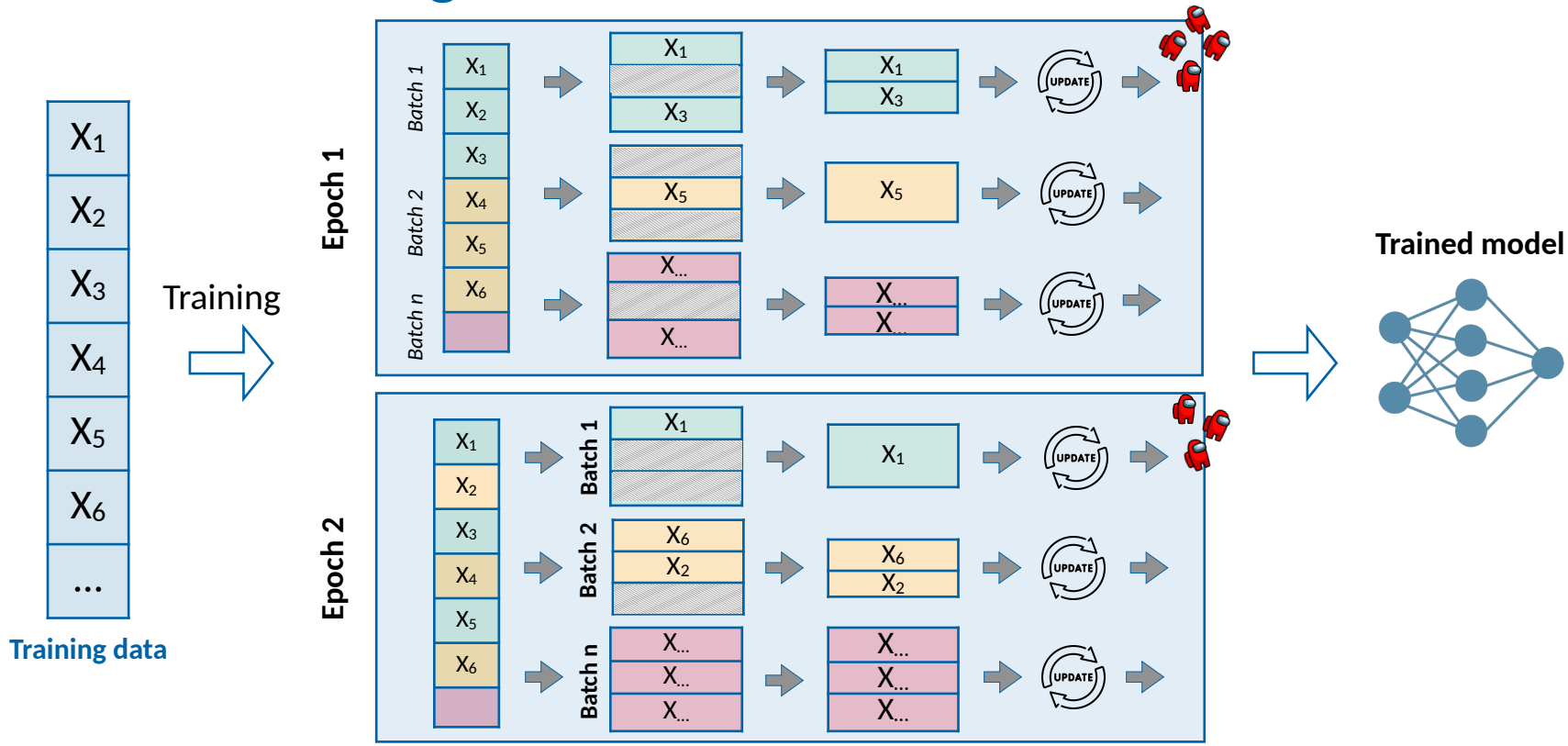
Dynamic Denoising



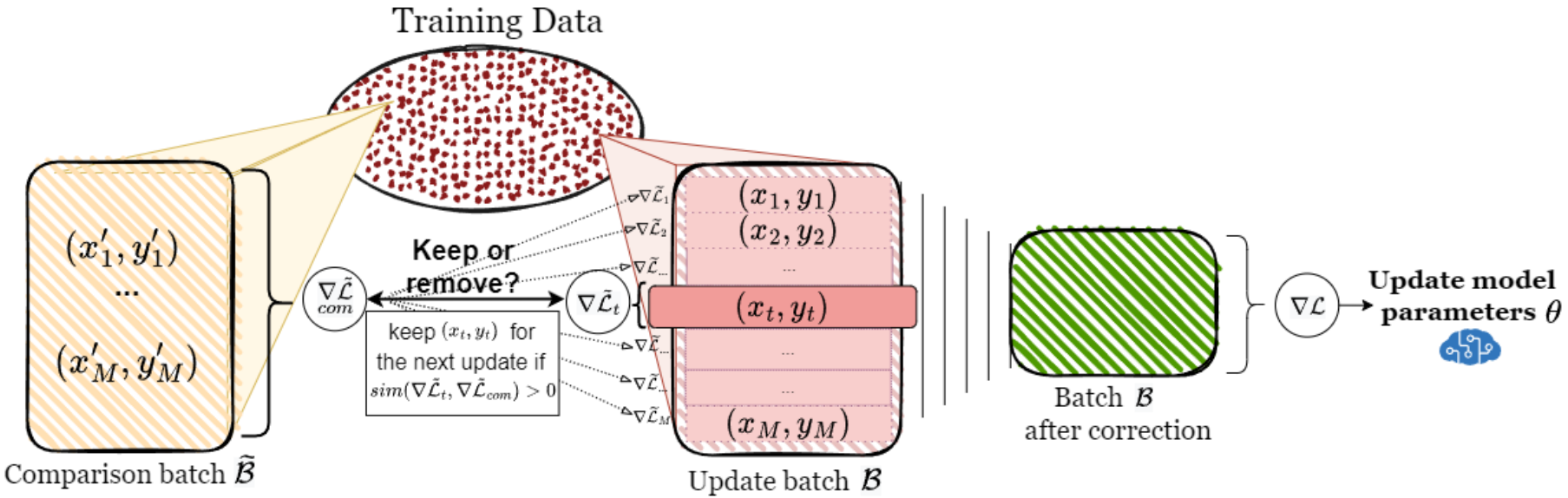
Dynamic Denoising



Dynamic Denoising

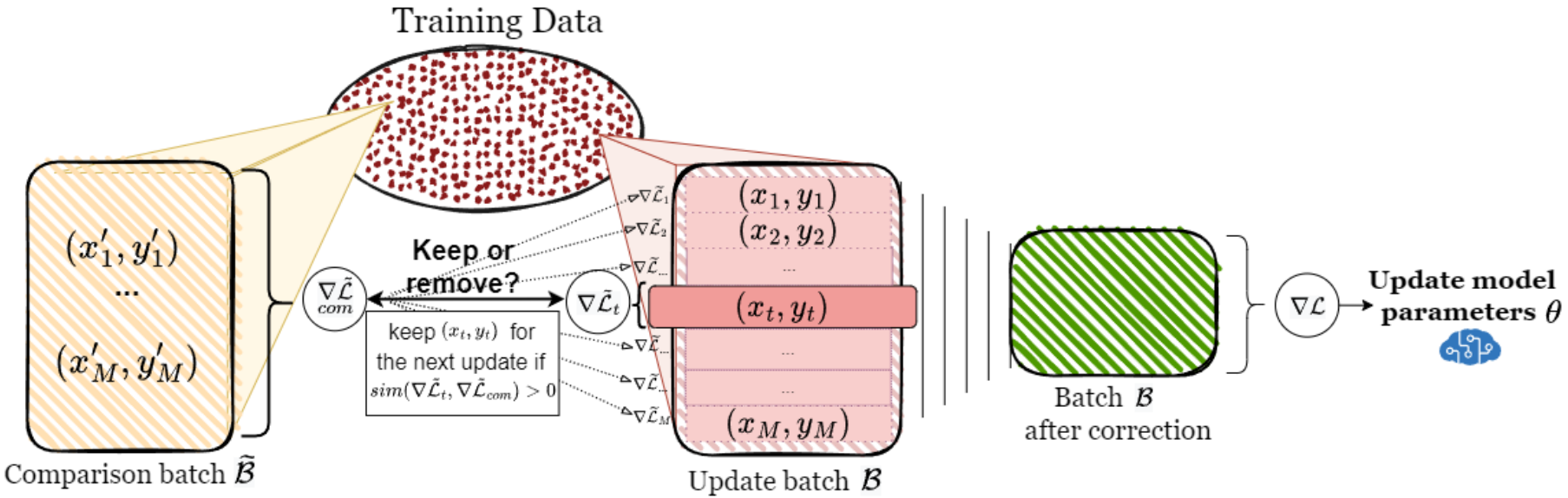


AGRA - Adaptive GRAdient-Based Outlier Removal



$$sim_{y_t} = sim\left(\nabla\widetilde{\mathscr{L}}(x_t, y_t), \nabla\widetilde{\mathscr{L}}_{com}\right) = \frac{\nabla\widetilde{\mathscr{L}}(x_t, y_t) \cdot \nabla\widetilde{\mathscr{L}}_{com}}{||\nabla\widetilde{\mathscr{L}}(x_t, y_t)||_2 \, ||\nabla\widetilde{\mathscr{L}}_{com}||_2}$$

AGRA - Adaptive GRAdient-Based Outlier Removal

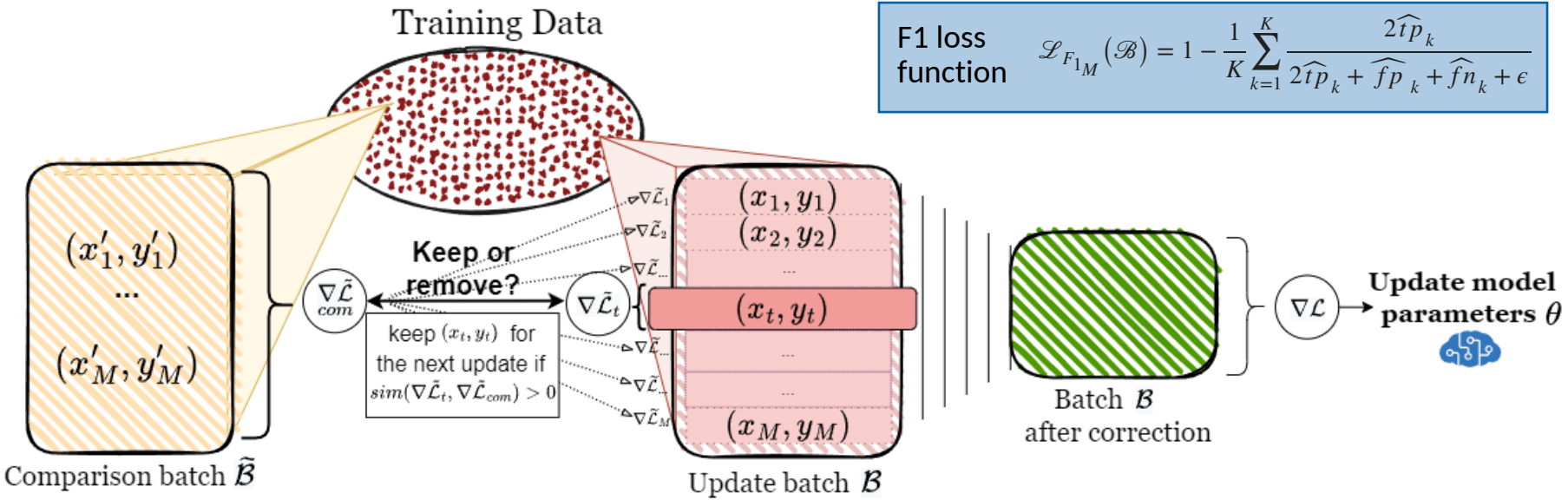


Class-weighted sampling:
Class weights shift the balance towards including more instances of minority classes to comparison batch

$$\frac{1}{\sum_{t=1}^{|\mathcal{X}|} \mathbb{1}(y_t = c_k)}$$

$$sim_{y_t} = sim\left(\nabla\widetilde{\mathscr{L}}(x_t, y_t), \nabla\widetilde{\mathscr{L}}_{com}\right) = \frac{\nabla\widetilde{\mathscr{L}}(x_t, y_t) \cdot \nabla\widetilde{\mathscr{L}}_{com}}{||\nabla\widetilde{\mathscr{L}}(x_t, y_t)||_2 \, ||\nabla\widetilde{\mathscr{L}}_{com}||_2}$$

AGRA - Adaptive GRAdient-Based Outlier Removal



F1 loss function

$$\mathscr{L}_{F1_M}(\mathscr{B}) = 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{2\widehat{tp}_k}{2\widehat{tp}_k + \widehat{fp}_k + \widehat{fn}_k + \epsilon}$$

Class-weighted sampling:
Class weights shift the balance towards including more instances of minority classes to comparison batch

$$\frac{1}{\sum_{t=1}^{|\mathcal{X}|}\mathbb{1}(y_t = c_k)}$$

$$sim_{y_t} = sim\left(\nabla\widetilde{\mathscr{L}}(x_t, y_t), \nabla\widetilde{\mathscr{L}}_{com}\right) = \frac{\nabla\widetilde{\mathscr{L}}(x_t, y_t) \cdot \nabla\widetilde{\mathscr{L}}_{com}}{||\nabla\widetilde{\mathscr{L}}(x_t, y_t)||_2 \, ||\nabla\widetilde{\mathscr{L}}_{com}||_2}$$

Results

	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

Weakly Supervised Text datasets

	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

	Weakly Supervised Text datasets						... with 20% noise added	
	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

	Weakly Supervised Text datasets						... with 20% noise added	Medical Image Data
	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

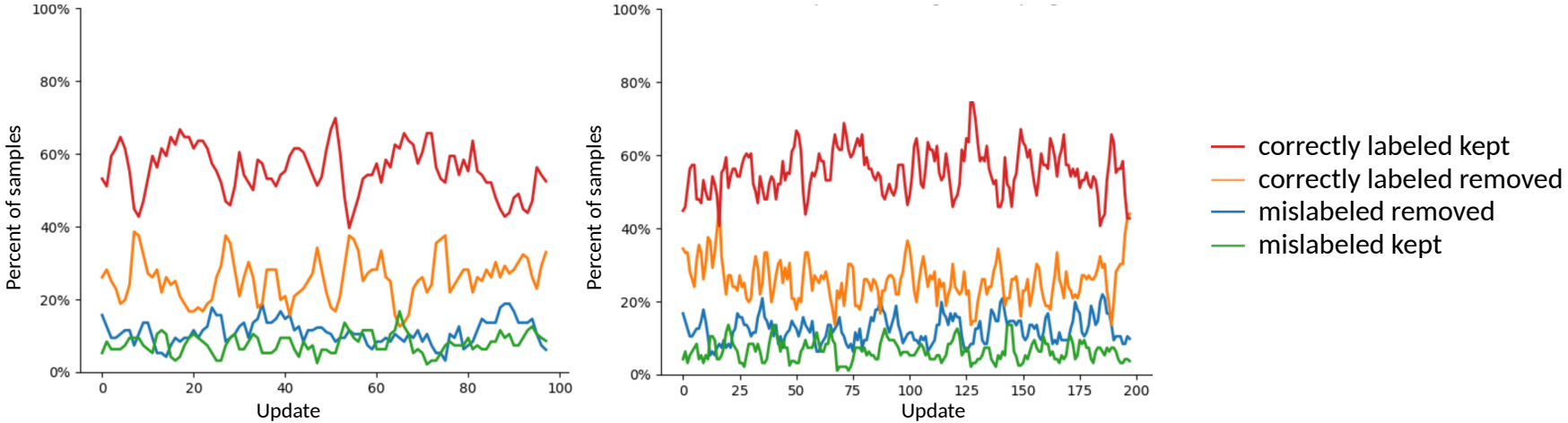
	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Results

	YouTube (Acc)	SMS (F1)	TREC (Acc)	Yorùbá (F1)	Hausa (F1)	Avg.	CIFAR (Acc)	CXT (AUR)
Gold	94.8±0.8	95.4±1.0	89.5±0.3	57.3±0.4	78.5±0.3	83.1	83.6±0.0	–
No Denoising	87.4±2.7	71.7±1.4	58.7±0.5	44.6±0.4	39.7±0.8	60.4	82.4±0.2	82.7±0.1
Weak Supervision								
DP [35]	90.8±1.0	44.1±6.7	54.3±0.5	**47.8**±1.7	40.9±0.6	55.6	–	–
MeTaL [34]	92.0±0.8	18.3±7.8	50.4±1.7	38.9±3.1	45.5±1.1	49.0	–	–
FS [14]	84.8±1.2	16.3±6.0	27.2±0.1	31.9±0.7	37.6±1.0	39.6	–	–
Noisy Learning								
CORES^2 [10]	88.8±3.6	85.8±1.8	61.8±0.5	43.0±0.7	**51.2**±0.5	66.1	83.4±0.1	–
Cleanlab [32]	91.3±1.2	80.6±0.3	60.9±0.4	43.8±1.3	40.3±0.3	63.4	83.3±0.0	81.5±0.4
AGRA	**93.9**±0.7	**87.7**±1.2	**63.6**±0.7	46.9±1.5	46.2±1.6	**67.7**	**83.6**±0.0	**83.9**±0.3

Result Analysis

- For all our datasets, we use the noisy heuristic labels, but we also know the gold labels
- -> We can measure the amount of **mislabeled kept, mislabeled removed, correcly labeled kept**, and **correctly labeled removed**.



Conclusion

- We experimented with dynamically adjusting the training set during the model training instead of hard outlier removal before the model training.
- Our method **AGRA** measures the sample-specific impact on the current model and removes the samples that negatively impact the model

Key Takeaways:

- Sample correctness ≠ Sample usefulness
- The model does not always benefit from hard outlier removal.

Learning with Noisy Labels by Adaptive GRAdient-Based Outlier Removal

Anastasiia Sedova*, Lena Zellinger*, Benjamin Roth

Data Science and Machine Learning Research Group, University of Vienna

ECML PKDD 2023

19.09.2023

* Equal contribution

AGRA Pseudocodes

Algorithm 1: AGRA Algorithm for Single-Label Datasets

Input: training set $\mathcal{X}$, initial model $f(\cdot;\theta)$, number of epochs E, batch size M, (optionally: alternative label y')
Output: trained model $f(\cdot;\theta^*)$
for *epoch = 1,..., E* **do**
  for *batch* $\mathcal{B}$ **do**
    Sample a comparison batch $\widetilde{\mathcal{B}}$, $\widetilde{\mathcal{B}} \subset \mathcal{X}$, $|\widetilde{\mathcal{B}}| = M$
    Compute $\nabla\widetilde{\mathcal{L}}_{com}$ on $\widetilde{\mathcal{B}}$
    for $(x_t, y_t) \in \mathcal{B}$ **do**
      Compute $\nabla\widetilde{\mathcal{L}}(x_t, y_t)$
      $sim_{y_t} = sim\left(\nabla\widetilde{\mathcal{L}}(x_t, y_t), \nabla\widetilde{\mathcal{L}}_{com}\right)$ (Eq. 1)
      if *an alternative label* y' *is specified* **then**
        Compute $\nabla\widetilde{\mathcal{L}}(x_t, y')$
        $sim_{y'} = sim\left(\nabla\widetilde{\mathcal{L}}(x_t, y'), \nabla\widetilde{\mathcal{L}}_{com}\right)$ (Eq. 1)
        if $sim_{y_t} \leq 0$ *and* $sim_{y'} \leq 0$ **then**
          $\mathcal{B} \leftarrow \mathcal{B} \setminus \{(x_t, y_t)\}$
        if $sim_{y'} > 0$ *and* $sim_{y'} > sim_{y_t}$ **then**
          $\mathcal{B} \leftarrow \mathcal{B} \setminus \{(x_t, y_t)\} \cup \{(x_t, y')\}$
      else
        if $sim_{y_t} \leq 0$ **then**
          $\mathcal{B} \leftarrow \mathcal{B} \setminus \{(x_t, y_t)\}$
    $\theta \leftarrow Optim(\theta, \mathcal{B}, \mathcal{L})$

Algorithm 2: AGRA Algorithm for Multi-Label Datasets

Input: training set $\mathcal{X}$, initial model $f(\cdot;\theta)$, removal threshold τ, number of epochs E, batch size M, number of classes K, label assigned to ignored samples i
Output: trained model $f(\cdot;\theta^*)$
for *epoch = 1,..., E* **do**
  for *batch* $\mathcal{B}$ **do**
    Sample a comparison batch $\widetilde{\mathcal{B}}$, $\widetilde{\mathcal{B}} \subset \mathcal{X}$, $|\widetilde{\mathcal{B}}| = M$
    Compute $\nabla\widetilde{\mathcal{L}}_{com}$ on $\widetilde{\mathcal{B}}$
    for $(x_t, y_t) \in \mathcal{B}$ **do**
      Compute $\nabla\widetilde{\mathcal{L}}(x_t, y_t)$
      Set up corrected label vector $\widetilde{y}_t \leftarrow y_t$
      for *k=1, ..., K* **do**
        $sim^k_{y_t} = sim\left(\left(\nabla\widetilde{\mathcal{L}}(x_t, y_t)\right)_k, \left(\nabla\widetilde{\mathcal{L}}_{com}\right)_k\right)$ (Eq.1)
        if $sim^k_{y_t} \leq 0$ **then**
          $\widetilde{y}_{t,k} \leftarrow i$
      $\mathcal{B} \leftarrow \mathcal{B} \setminus \{(x_t, y_t)\} \cup \{(x_t, \widetilde{y}_t)\}$
    $\theta \leftarrow Optim(\theta, \mathcal{B}, \mathcal{L})$

F1 Loss Function

- Directly represents the performance metric
- Maximizes the F1 score
- VS the standard F1 loss: the predicted labels are replaced by the model outputs transformed into predicted probabilities by a suitable activation function -> it is differentiable

$$\mathcal{L}_{F1_M}(\mathcal{B}) = 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{2\widehat{tp}_k}{2\widehat{tp}_k + \widehat{fp}_k + \widehat{fn}_k + \epsilon}$$

ε = 1e − 05

$$\widehat{tp}_k = \sum_{t=1}^{M} \hat{y}_{t,k} \times \mathbb{1}(y_t = c_k)$$

the predicted probability of class k for sample t after application of the softmax

$$\widehat{fp}_k = \sum_{t=1}^{M} \hat{y}_{t,k} \times (1 - \mathbb{1}(y_t = c_k))$$

$$\widehat{fn}_k = \sum_{t=1}^{M} (1 - \hat{y}_{t,k}) \times \mathbb{1}(y_t = c_k)$$

Binary-F1 Loss Function for Single-Label Settings

- Based on the F1 score of the positive class
- Aims to maximize the F1 score of the positive class

$$\mathcal{L}_{F_1}(\mathcal{B}) = 1 - \frac{2\widehat{tp}}{2\widehat{tp} + \widehat{fp} + \widehat{fn} + \epsilon}$$

the predicted probability for the positive class for sample t after application of the softmax

$$\widehat{tp} = \sum_{t=1}^{M} \hat{y}_{t,1} \times y_t$$

$$\widehat{fp} = \sum_{t=1}^{M} \hat{y}_{t,1} \times (1 - y_t)$$

$$\widehat{fn} = \sum_{t=1}^{M} (1 - \hat{y}_{t,1}) \times y_t$$

$$y_t \in \{0, 1\}$$

Macro-F1 Loss Function for Single-Label Settings

- Averages the differentiable F1 scores of all K classes

the predicted probability of class k for sample t after application of the sigmoid function

$$\mathcal{L}_{F1_M}(\mathcal{B}) = 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{2\widehat{tp}_k}{2\widehat{tp}_k + \widehat{fp}_k + \widehat{fn}_k + \epsilon}$$

$$\widehat{tp}_k = \sum_{t=1}^{M} \hat{y}_{t,k} \times y_{t,k}$$

$$\widehat{fp}_k = \sum_{t=1}^{M} \hat{y}_{t,k} \times (1 - y_{t,k})$$

$$\widehat{fn}_k = \sum_{t=1}^{M} (1 - \hat{y}_{t,k}) \times y_{t,k}$$

$$y_{t,k} \in \{0, 1\}$$

Datasets

Dataset	#Class	#Train	#Dev	#Test	%Noise
YouTube	2	1586	120	250	18.8
SMS	2	4571	500	500	31.9
TREC	6	4965	500	500	48.2
Yorùbá	7	1340	189	379	42.3
Hausa	5	2045	290	582	50.6
CheXpert	12	200599	22815	234	-
CIFAR-10	10	50000	5000	5000	20

Ablation Study

	No Weighted Sampling		Weighted Sampling	
	CE/CE	CE/F_1	CE/CE	CE/F_1
YouTube	92.0 ± 1.0	$\mathbf{93.9 \pm 0.7}$	91.9 ± 0.5	93.4 ± 0.8
YouTube†	90.5 ± 1.0	–	92.0 ± 0.7	–
SMS	79.0 ± 3.2	61.1 ± 5.2	$\mathbf{87.7 \pm 1.2}$	49.1 ± 3.0
SMS†	71.1 ± 3.1	–	86.3 ± 1.2	–
TREC	61.6 ± 0.6	62.1 ± 0.4	62.8 ± 1.1	$\mathbf{63.6 \pm 0.7}$
Yorùbá	44.3 ± 2.5	44.2 ± 1.4	43.5 ± 1.0	$\mathbf{46.9 \pm 1.5}$
Hausa	41.2 ± 0.4	40.9 ± 0.6	43.8 ± 2.8	$\mathbf{46.2 \pm 1.6}$
CheXpert	82.6 ± 0.6	$\mathbf{83.9 \pm 0.3}$	–	–
CIFAR	82.2 ± 0.2	83.5 ± 0.0	83.1 ± 0.0	$\mathbf{83.6 \pm 0.0}$