南京航空航天大學
Nanjing University of Aeronautics and Astronautics

模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

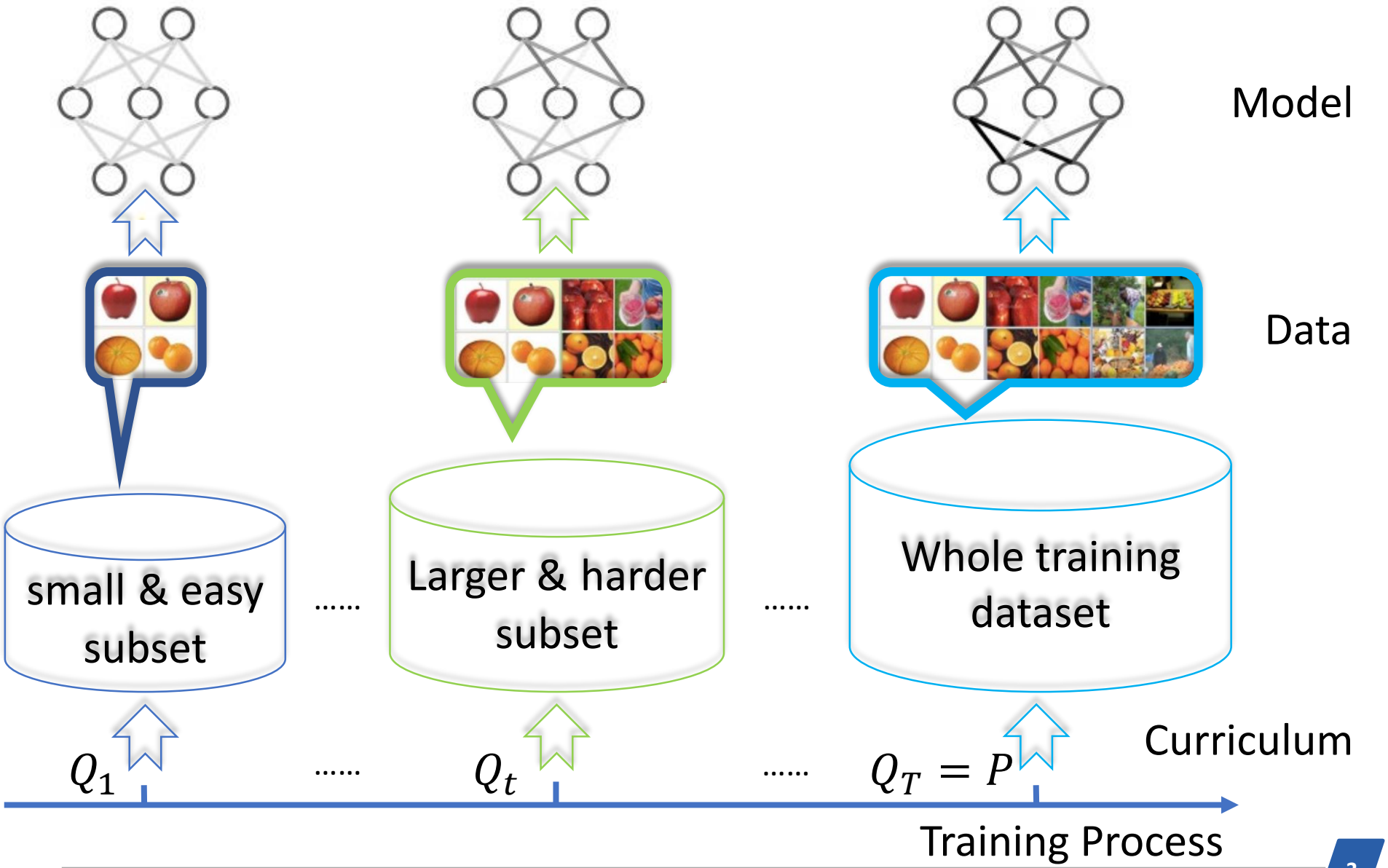
# MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels

Lu Jiang [1] Zhengyuan Zhou [2] Thomas Leung [1] Li-Jia Li [1] Li Fei-Fei [1 2]

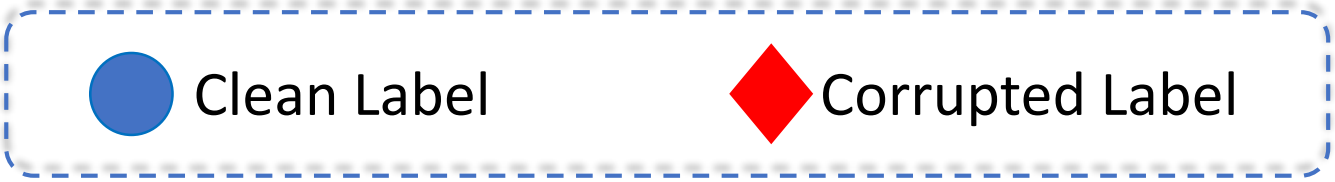
*ICML 2018*

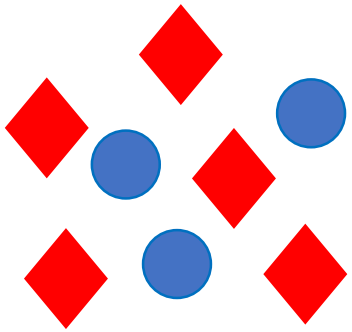
# Background

# Curriculum Learning



# Traditional Learning

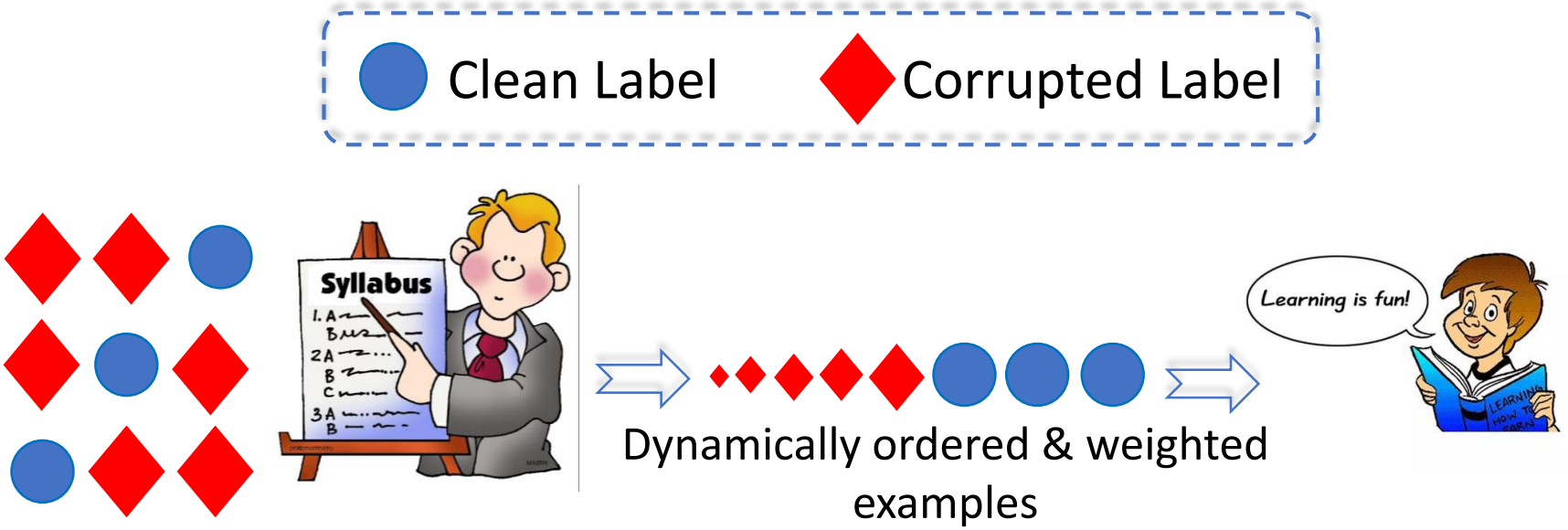


Random shuffled examples

# Curriculum Learning

- **Curriculum learning:** learning examples with focus
  - Introduce a teacher to determine the weight and timing to learn every example.




*Bengio, Y et al. Curriculum learning. ICML, 2009*
*Kumar, M et al. Self-paced learning for latent variable models. NIPS 2010*

# Curriculum Learning

Latent weight variable

Regularizer G

$$\min_{\mathbf{w}\in\mathbb{R}^d,\mathbf{v}\in[0,1]^{n\times m}} \mathbb{F}(\mathbf{w},\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^T \mathbf{L}(\mathbf{y}_i, g_s(\mathbf{x}_i,\mathbf{w}_i)) + G(\mathbf{v};\lambda) + \theta\|\mathbf{w}\|_2$$

- An example: self-paced (*Kumar, M et al. 2010*)

Favor example with smaller loss

$$G(\mathbf{v}) = -\|\mathbf{v}\|_1 \quad \Rightarrow \quad v_i^* = \begin{cases} 1 & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda \end{cases} \qquad \begin{array}{l} \ell_i = L(\mathbf{y}_i, g_s(\mathbf{x}_i;\mathbf{w})) \\ \mathbf{v}_i \Rightarrow v_i \end{array}$$

Regularizer

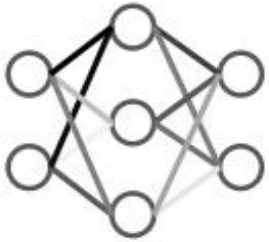
Weighting scheme

*Kumar, M et al. Self-paced learning for latent variable models. NIPS 2010*

# Motivation

- Existing studies define a curriculum as a function:
  - Self-paced (*Kumar, M et al. 2010*)
  - Regularizer $G$: $G(\mathbf{v}) = -\|\mathbf{v}\|_1$
  - Weight $v^*$: $v_i^* = \mathbb{I}(\ell_i < \lambda)$
  - Linear weighting (*Jiang et al. 2015*)
  - Regularizer $G$: $G(\mathbf{v}) = \frac{1}{2}\sum_{i=1}^{n}(v_i^2 - 2v_i)$
  - Weight $v^*$: $v_i^* = \max(0, 1 - \frac{1}{\lambda}\ell_i)$

Learning a curriculum by a neural network from data?

Mini-batch ⇒  ⇒ Weights

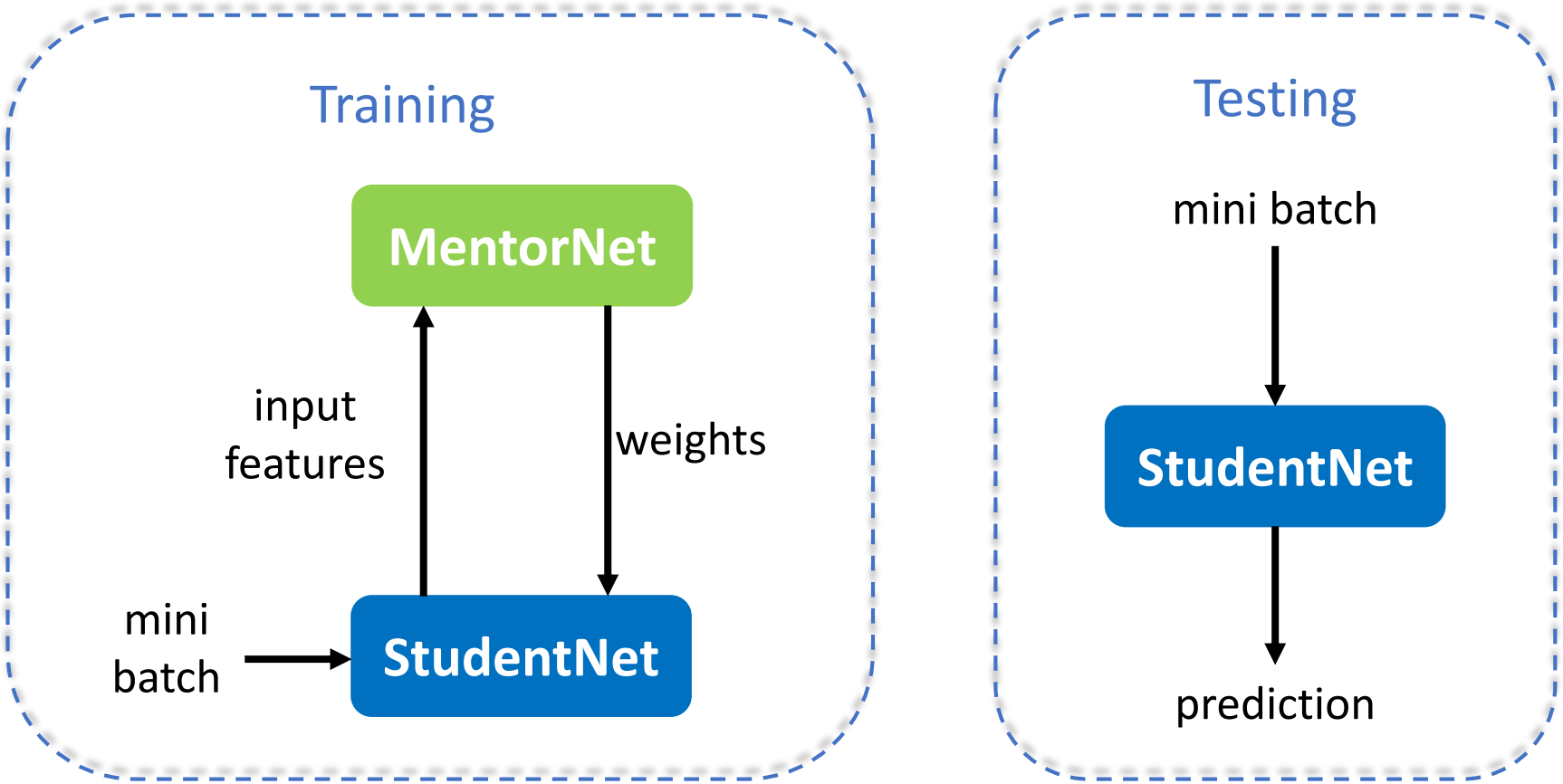
Network

*Kumar, M et al. Self-paced learning for latent variable models. NIPS 2010*
*Jiang, L et al. Self-paced curriculum learning. AAAI 2015*

# Method

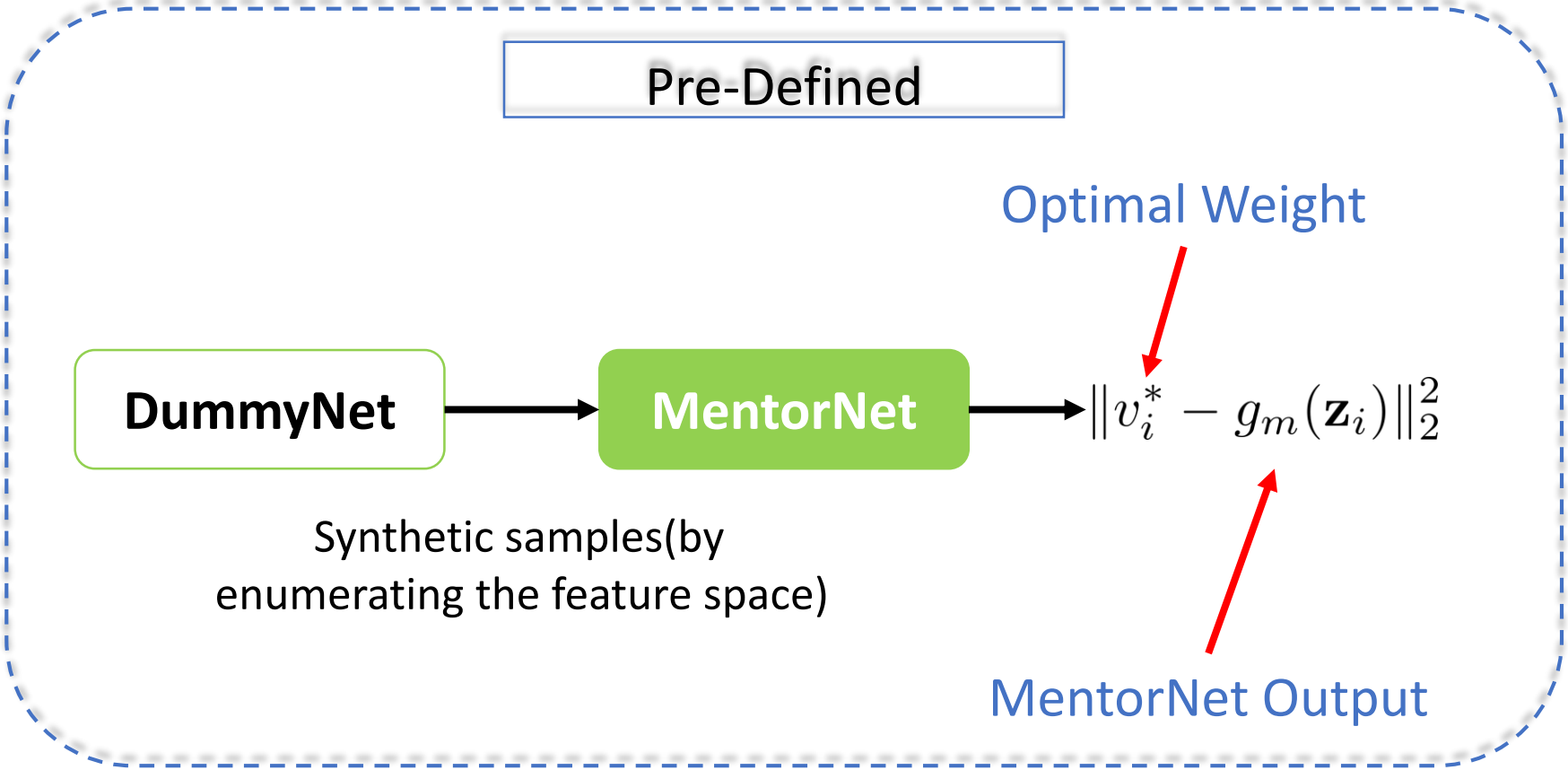
# MentorNet

- Each curriculum is implemented as a network (called MentorNet)



# Learning the MentorNet

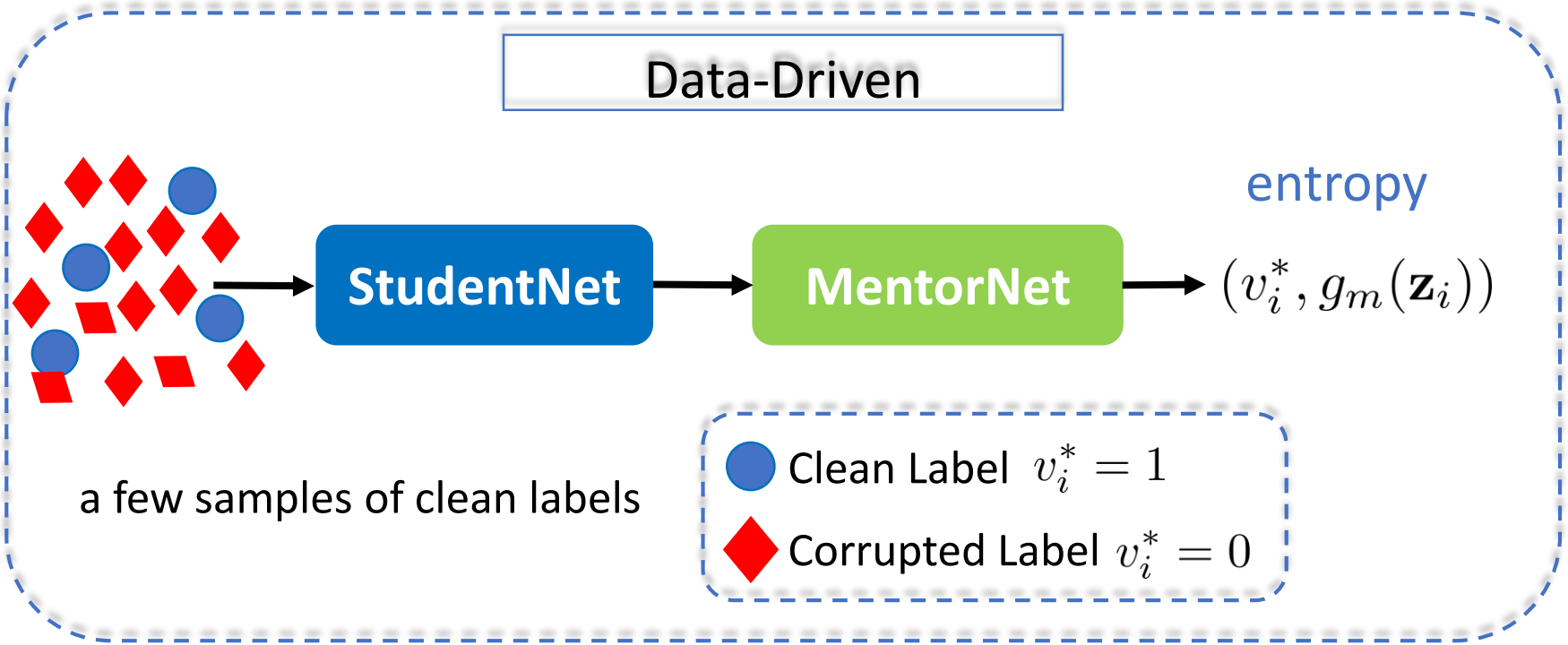
- Two ways to learn a MentorNet:
  - Pre-Defined: approximating existing curriculum



# Learning the MentorNet

- Two ways to learn a MentorNet:
  - Data-Driven: learn new curriculum from a small set (~10%) with clean labels.

Learn the MentorNet of CIFAR-10 and use it on CIFAR-100



# Algorithm

## Training MentorNet with StudentNet

Existing curriculum/ self-paced learning uses alternative optimization.

High computational complexity

This paper proposes a mini-batch SGD and shows good property on model convergence under standard assumptions.

**Algorithm 1** SPADE for minimizing Eq. (1)

**Input** : Dataset $\mathcal{D}$, a predefined $G$ or a learned $g_m(\cdot;\Theta)$

**Output** : The model parameter $\mathbf{w}$ of StudentNet.

Initialize $\mathbf{w}^0$, $\mathbf{v}^0$, $t = 0$

**while** *Not Converged* **do**

  Fetch a mini-batch $\Xi_t$ uniformly at random

  For every $(\mathbf{x}_i, y_i)$ in $\Xi_t$ compute $\phi(\mathbf{x}_i, y_i, \mathbf{w}^t)$

  **if** *update curriculum* **then**

    $\Theta \leftarrow \Theta^*$, where $\Theta^*$ is learned in Sec. 3.1

  **end**

  **if** *$G$ is used* **then**

    $\mathbf{v}_\Xi^t \leftarrow \mathbf{v}_\Xi^{t-1} - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^{t-1})|_{\Xi_t}$

  **end**

  **else** $\mathbf{v}_\Xi^t \leftarrow g_m(\phi(\Xi_t, \mathbf{w}^{t-1}); \Theta)$ ;

  $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^t)|_{\Xi_t}$

  $t \leftarrow t + 1$

**end**

**return** $\mathbf{w}^t$

# Experiment

# Setups

- **Experiments on controlled corrupted labels**
  - Nosie type: uniform noise
  - Nosie fractions: $p \in \{0.2,\ 0.4,\ 0.8\}$
  - Dataset and StudentNet:

*Table 1.* StudentNet and their accuracies on the clean training data.

| Dataset | Model | #para | train acc | val acc |
|---|---|---|---|---|
| CIFAR10 | inception | 1.7M | 0.83 | 0.81 |
| | resnet101 | 84M | 1.00 | 0.96 |
| CIFAR100 | inception | 1.7M | 0.64 | 0.49 |
| | resnet101 | 84M | 1.00 | 0.79 |
| ImageNet | inception_resnet | 59M | 0.88 | 0.77 |

# Results on CIFAR

Baseline comparisons on **CIFAR-10 & CIFAR-100**
(under 20%, 40% and 80% noise fractions)

| | Resnet-101 StudentNet | | | | | | Inception StudentNet | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CIFAR-100 | | | CIFAR-10 | | | CIFAR-100 | | | CIFAR-10 | | |
| Method | 0.2 | 0.4 | 0.8 | 0.2 | 0.4 | 0.8 | 0.2 | 0.4 | 0.8 | 0.2 | 0.4 | 0.8 |
| FullModel | 0.60 | 0.45 | 0.08 | 0.82 | 0.69 | 0.18 | 0.43 | 0.38 | 0.15 | 0.76 | 0.73 | 0.42 |
| Forgetting | 0.61 | 0.44 | 0.16 | 0.78 | 0.63 | 0.35 | 0.42 | 0.37 | 0.17 | 0.76 | 0.71 | 0.44 |
| Self-paced | 0.70 | 0.55 | 0.13 | 0.89 | 0.85 | 0.28 | 0.44 | 0.38 | 0.14 | **0.80** | 0.74 | 0.33 |
| Focal Loss | 0.59 | 0.44 | 0.09 | 0.79 | 0.65 | 0.28 | 0.43 | 0.38 | 0.15 | 0.77 | 0.74 | 0.40 |
| Reed Soft | 0.62 | 0.46 | 0.08 | 0.81 | 0.63 | 0.18 | 0.42 | 0.39 | 0.12 | 0.78 | 0.73 | 0.39 |
| MentorNet PD | 0.72 | 0.56 | 0.14 | 0.91 | 0.77 | 0.33 | 0.44 | 0.39 | 0.16 | 0.79 | 0.74 | 0.44 |
| MentorNet DD | **0.73** | **0.68** | **0.35** | **0.92** | **0.89** | **0.49** | **0.46** | **0.41** | **0.20** | 0.79 | **0.76** | **0.46** |

Significant improvement over baselines.

**D**ata-**D**irven performs better than **P**re-**D**efined MentorNet.

# Results on ImageNet

Baseline comparisons on **ImageNet**
(under 40% noise fractions)

| Method | P@1 | P@5 |
|---|---|---|
| NoReg | 0.538 | 0.770 |
| NoReg+WeDecay | 0.560 | 0.809 |
| NoReg+Dropout | 0.575 | 0.807 |
| NoReg+DataAug | 0.522 | 0.772 |
| NoReg+MentorNet | 0.590 | 0.814 |
| FullModel | 0.612 | 0.844 |
| Forgetting(FullModel) | 0.628 | 0.845 |
| **MentorNet(FullModel)** | **0.651** | **0.859** |

**NoReg:** Vanilla model with no regularization

**FullModel:** Added weight decay, dropout, and data augmentation.

# Results on WebVision

**2.4 million** images of noisy labels crawled by Flickr/Google Search.

**1000 classes** defined in ImageNet ILSVRC 2012.

**Real-World noisy labels.**

| Dataset | Method | ILSVRC12 | WebVision |
|---|---|---|---|
| Entire | Li *et al.* (2017a) | 0.476 (0.704) | 0.570 (0.779) |
| Entire | Forgetting | 0.590 (0.808) | 0.666 (0.856) |
| Entire | Lee *et al.* (2017)* | 0.602 (0.811) | 0.685 (0.865) |
| Entire | MentorNet | 0.625 (0.830) | 0.708 (0.880) |
| Entire | **MentorNet*** | **0.642 (0.848)** | **0.726 (0.889)** |

**The best-published** result on the WebVision benchmark!

Substantiate that MentorNet is **beneficial** for training very deep networks on noisy data.

# My Work

# Partial Multi-Label Learning


building tree
bicycle cloud

window tree
bicycle light

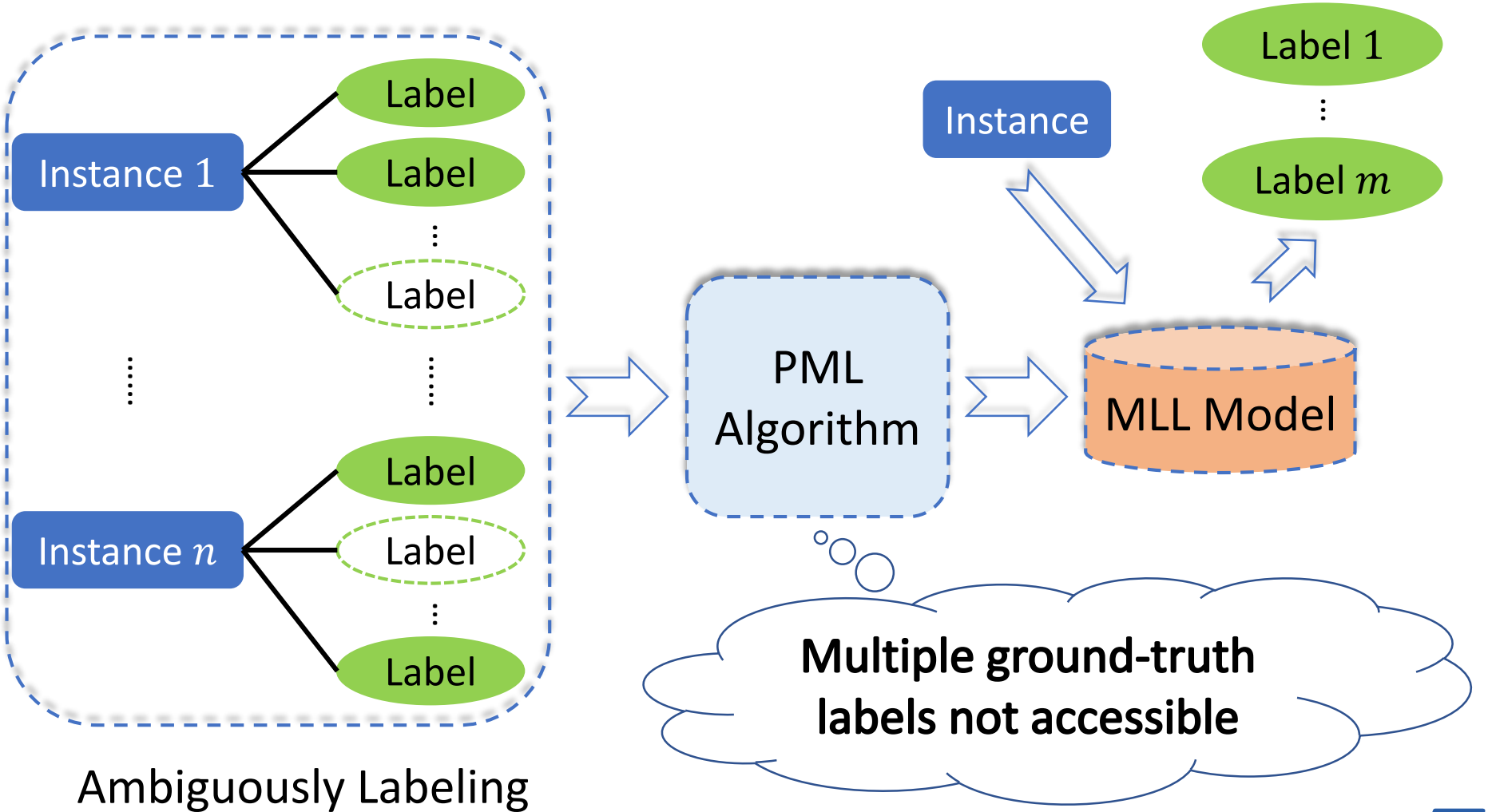
building people
flower light

Crowdsourcing Platform


Candidate Label Set = { window bicycle light tree building people flower cloud }

# The PML Framework

- Learning a multi-label classifier from partial-labeled examples



# Motivation & Thought

- Considering the commonly used hinge loss

$$\ell_{ijk} = \max(0, 1 - (f_{ij} - f_{ik}))$$

$$\mathcal{L}(D, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in \hat{\boldsymbol{y}}_i} \sum_{k \notin \hat{\boldsymbol{y}}_i} \ell \left( f_{ij}(\boldsymbol{x}_i, \theta) - f_{ik}(\boldsymbol{x}_i, \theta) \right)$$


**Candidate label set $\hat{y}$**

building tree
street window
…
people flower
…

**Non-Candidate label set $\overline{y}$**

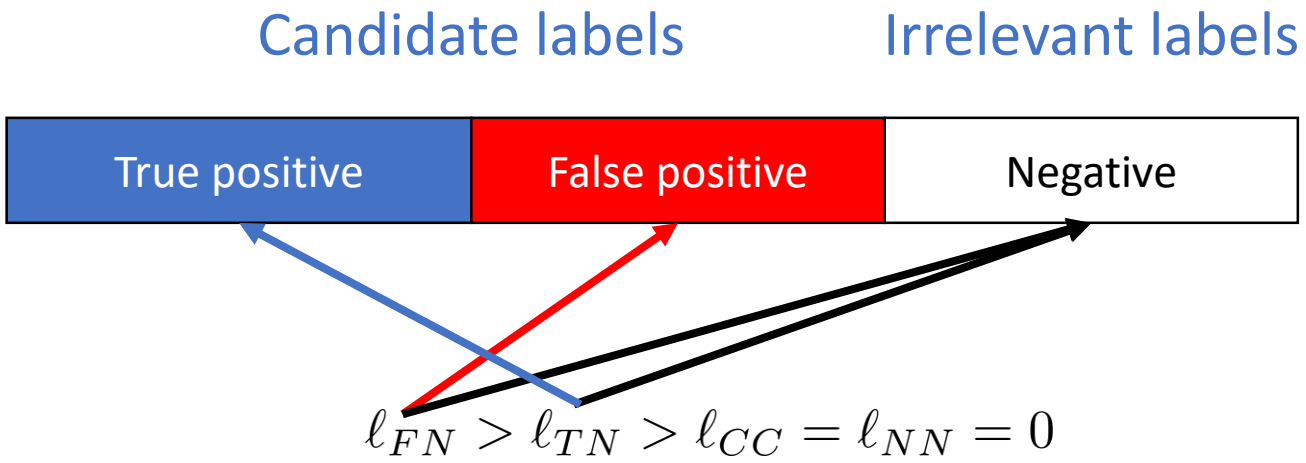
dog car
cat
….

Ranking the candidate label **people** before **dog** (as done in hinge loss)

Misleading the classifier by the **false positive labels** in the candidate set

# Motivation & Thought

- Intuitively, the loss value of pairs between **false positive labels** and negative labels is **larger** than that of pairs between **true positive labels** and negative labels.



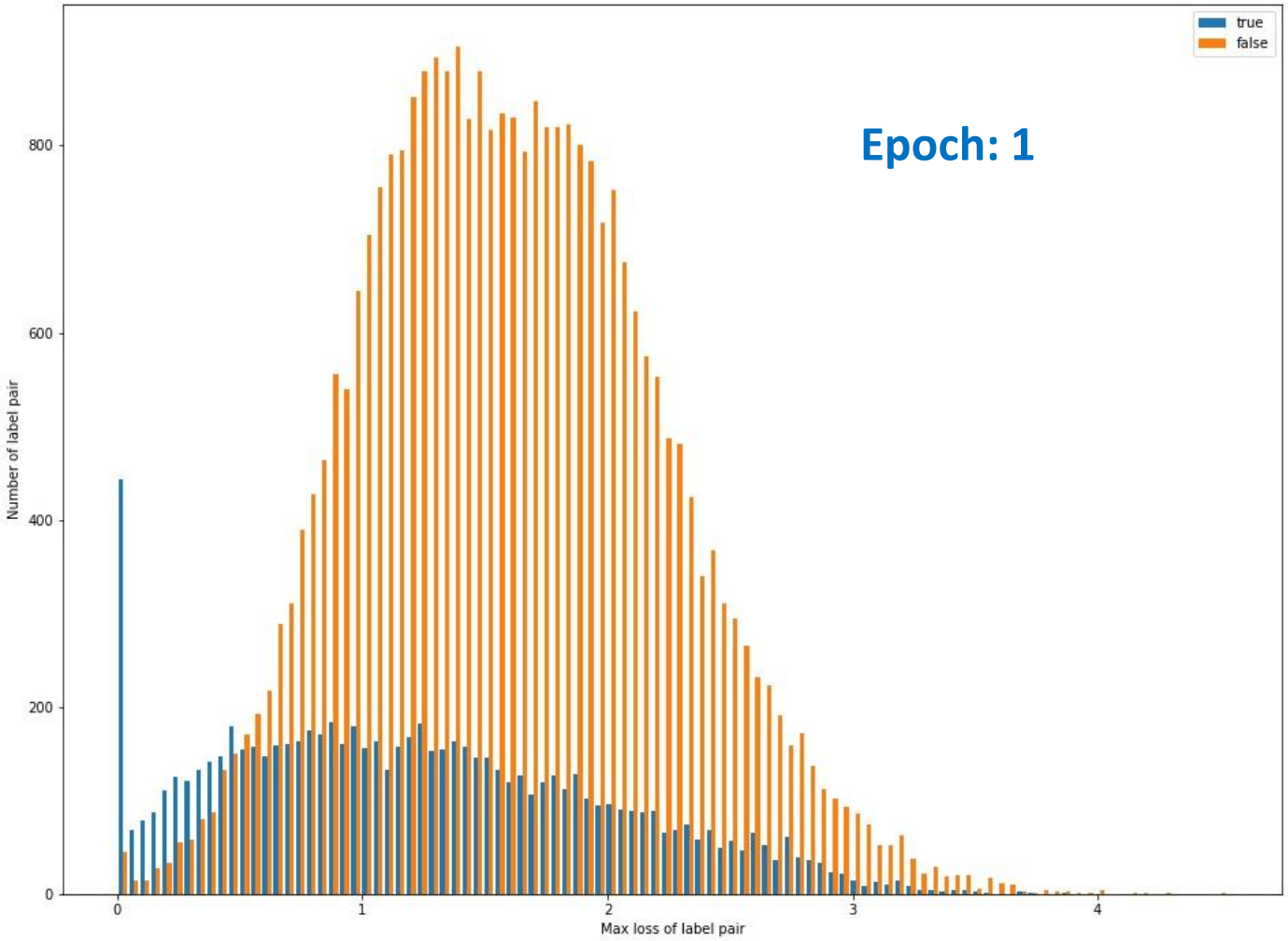

Can we implement the disambiguation in a self-paced way?

- We firstly select the labels with small loss, and gradually add the labels with larger loss.

# Preliminary Verification

Dataset: **VOC**

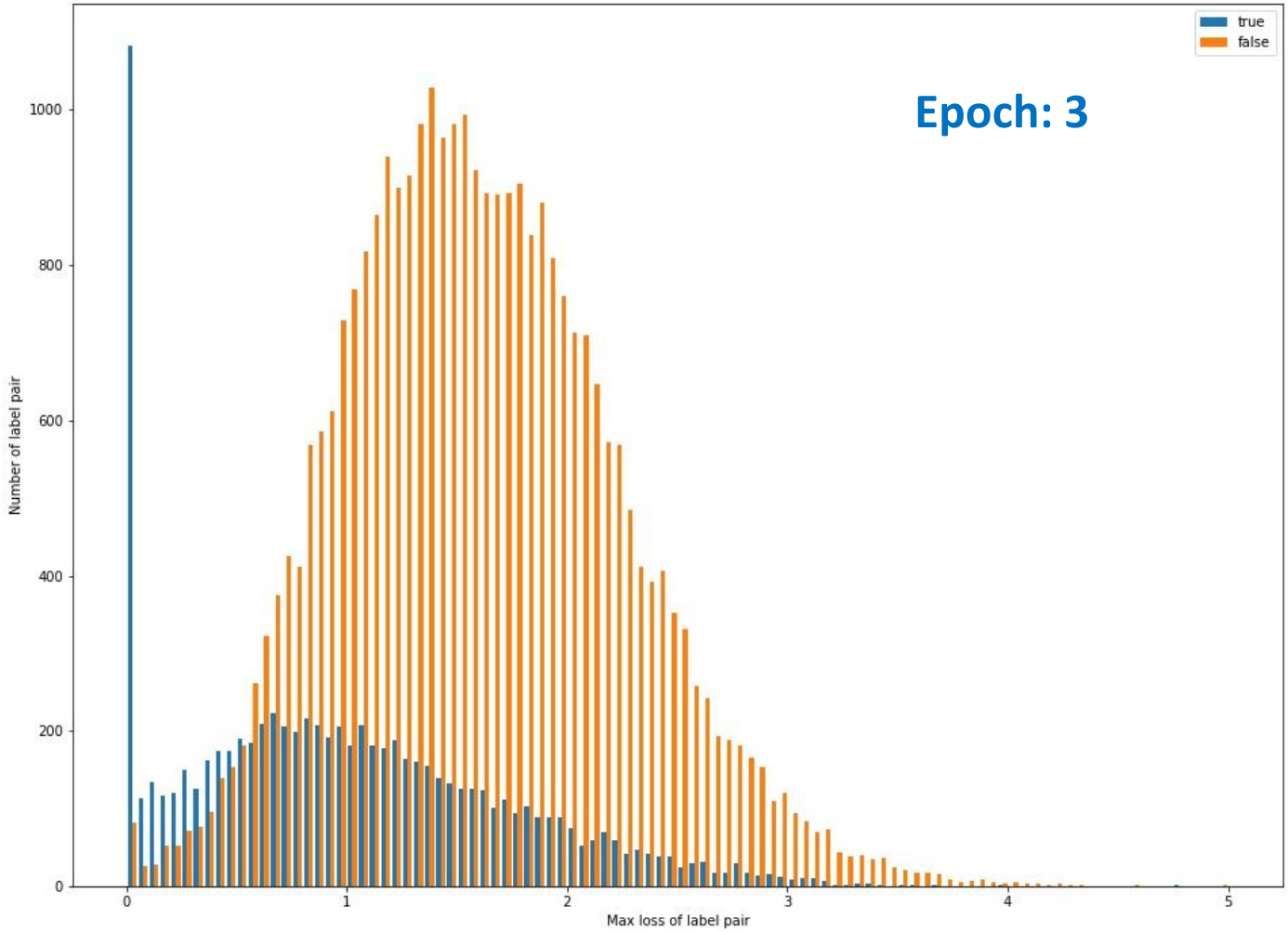
Noise rate: $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$



# Preliminary Verification

Dataset: **VOC**

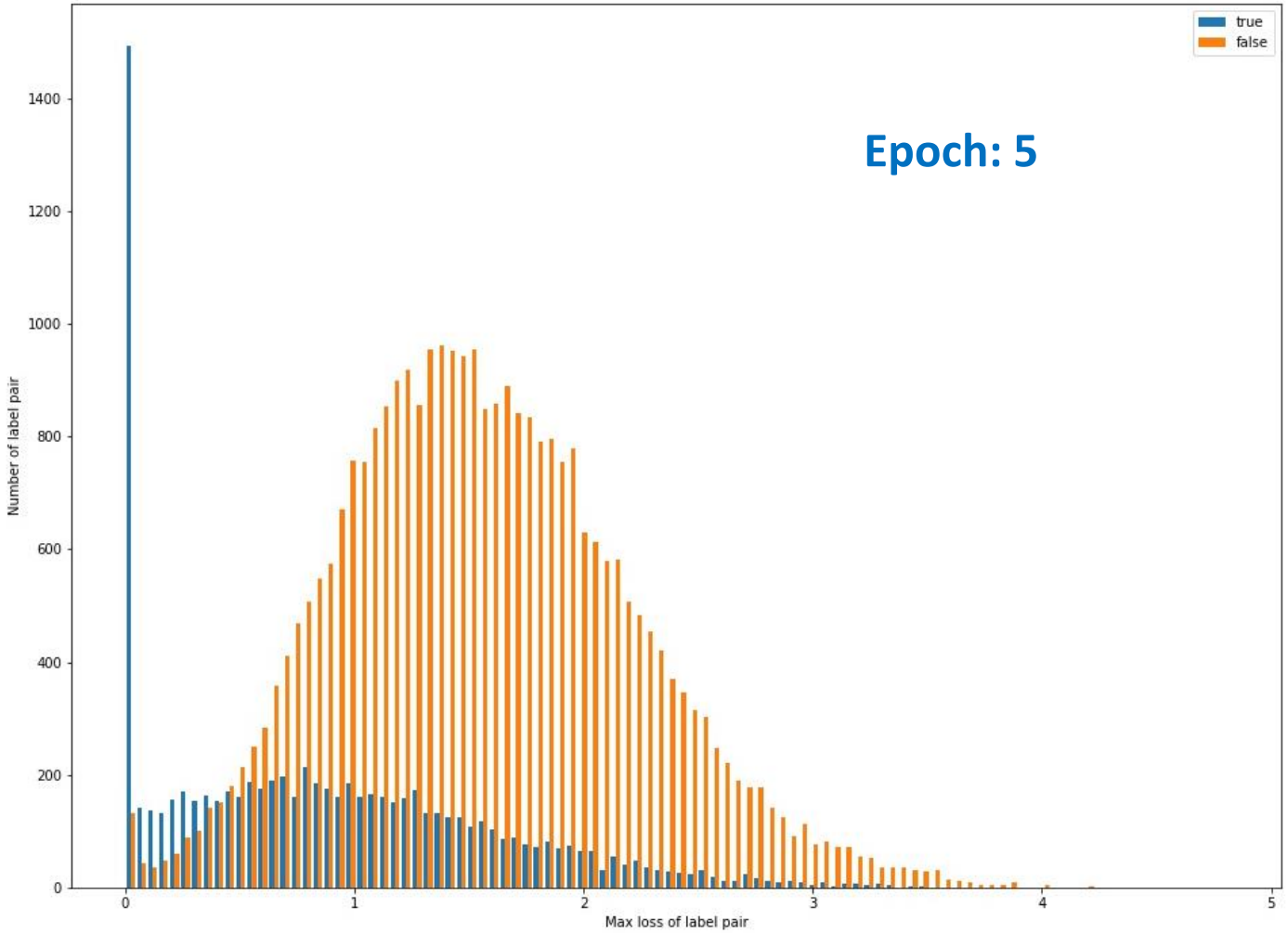
Noise rate: $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$



# Preliminary Verification

Dataset: **VOC**

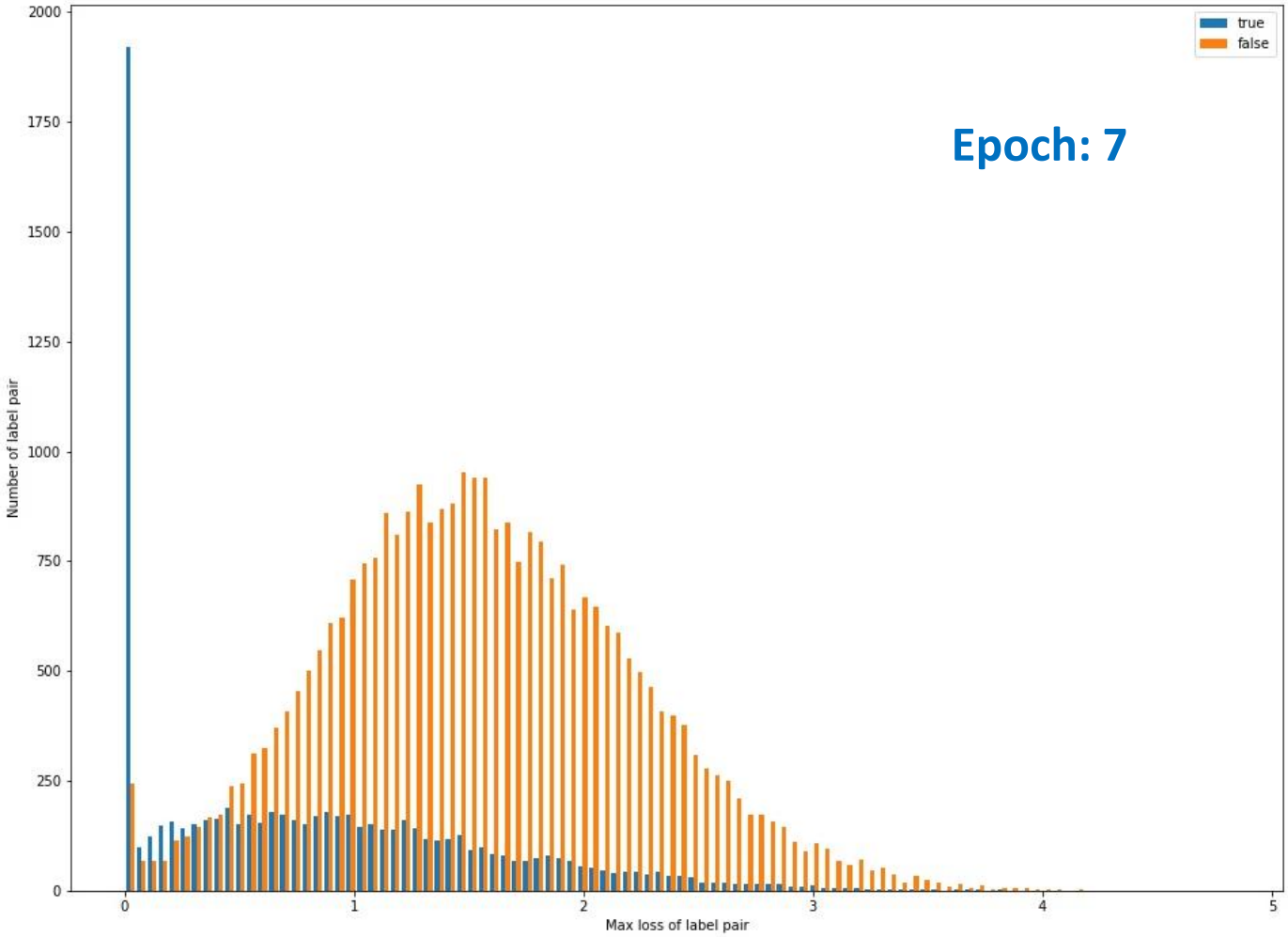
Noise rate: $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$



# Preliminary Verification

Dataset: **VOC**

Noise rate: $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$




Thanks!