# FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence

Kihyuk Sohn* David Berthelot* Chun-Liang Li Zizhao Zhang Nicholas Carlini
Ekin D. Cubuk Alex Kurakin Han Zhang Colin Raffel
Google Research
{kihyuks,dberth,chunliang,zizhaoz,ncarlini,cubuk,kurakin,zhanghan,craffel}@google.com

김 성 철

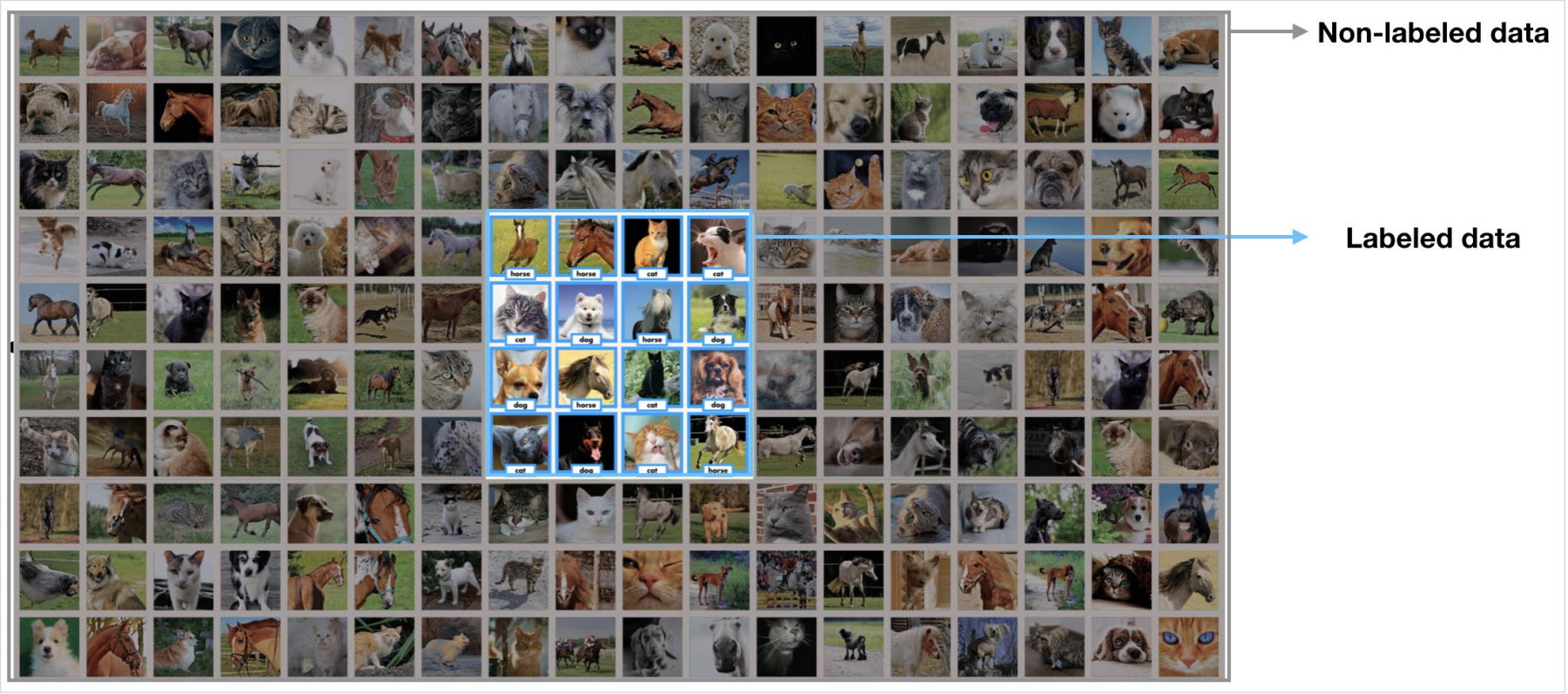
# Contents

# 용어 정리

- **Supervised Learning & Unsupervised Learning**
- **Weak-supervised Learning**
  - Weak label (e.g. dog, cat, …)을 가지고 strong label (e.g. segmentation mask)을 예측하자!
  - **FickleNet**, … (following을 안하고 있어서…)
- **Semi-supervised Learning**
  - Labeled data가 많지 않은 상황에서 unlabeled data의 도움을 받아 성능을 높이자!
  - **UDA**, **MixMatch**, **ReMixMatch**, **FixMatch**, …
- **Self-supervised Learning**
  - Unlabeled data를 이용해 데이터 고유의 특징을 뽑아내는 pretrained weight를 잘 만들어보자!
  - solving jigsaw puzzle, **CPC**, BERT, GPT, **MoCo**, **SimCLR**, **BYOL**, …

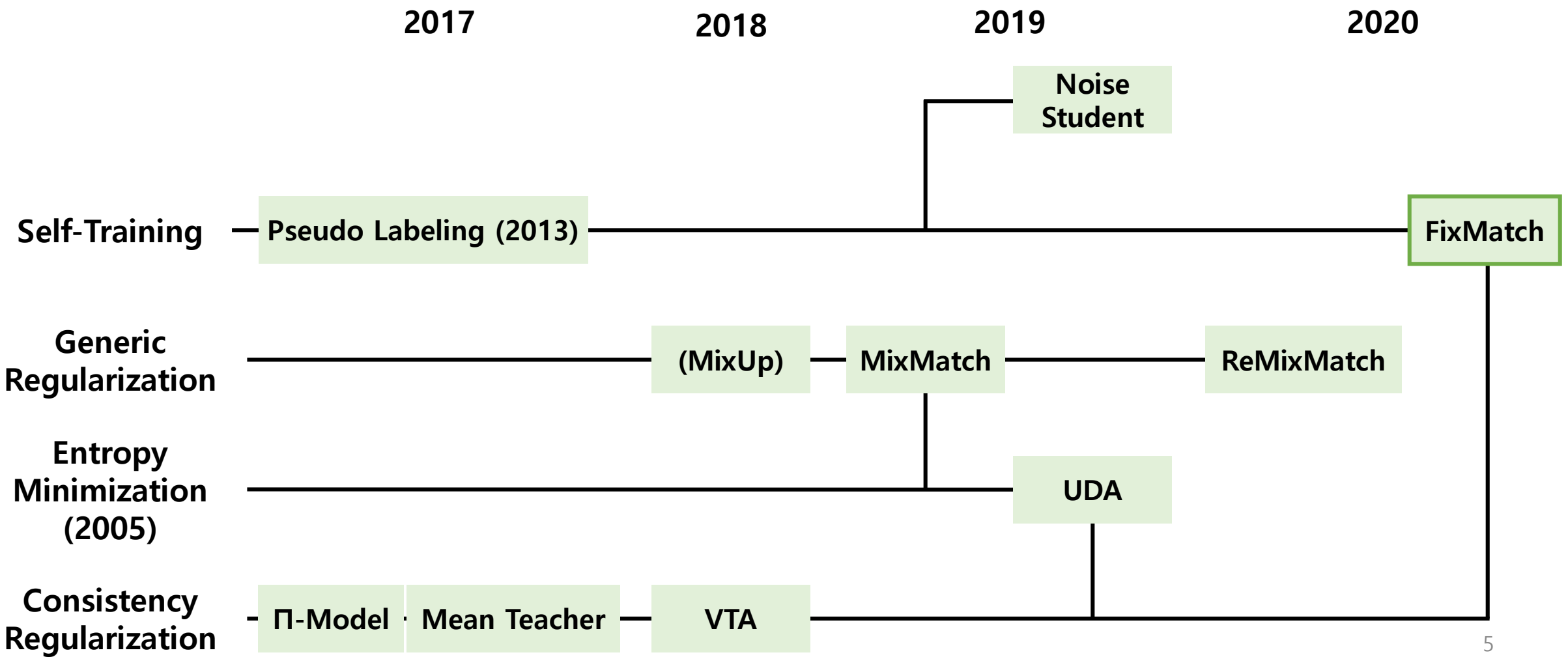
# Semi-supervised learning

- **목표**
  - Labeled data가 많지 않은 상황에서 unlabeled data의 도움을 받아 성능을 높이자!



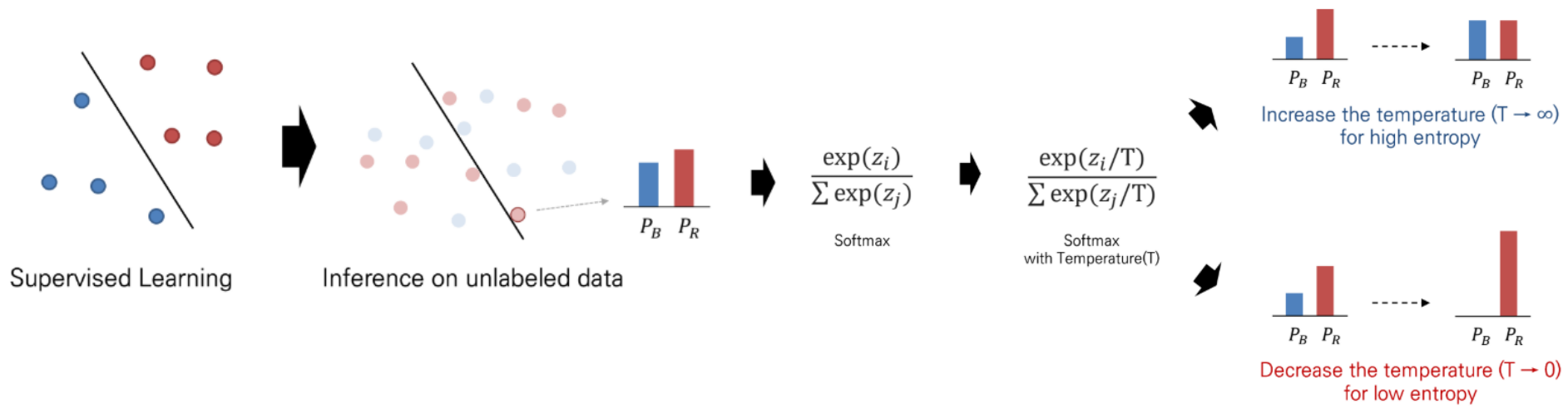

https://steemit.com/kr/@jiwoopapa/realistic-evaluation-of-semi-supervised-learning-algorithms

# Semi-supervised learning



# Semi-supervised learning

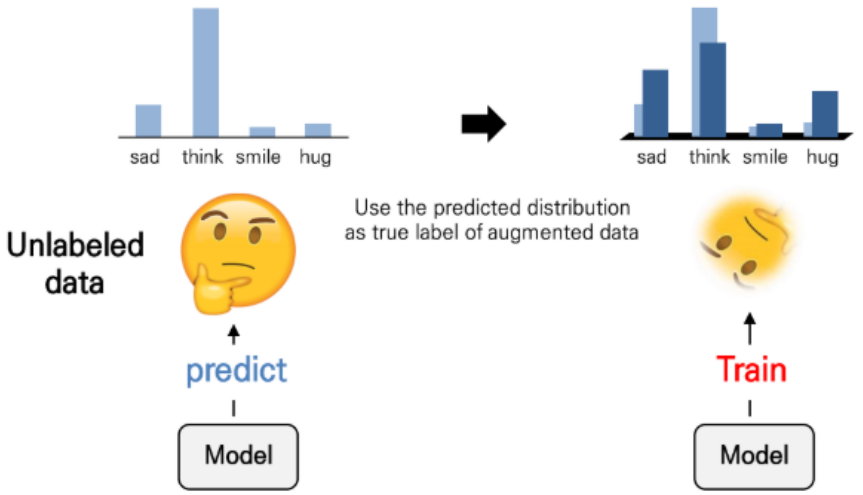
- **Entropy Minimization**
  - 예측값(softmax)의 confidenc를 높이기 위해 사용
  - 주로 softmax temperature를 이용
  - Temperature를 1보다 적게 설정할수록 entropy가 작아짐




https://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization.pdf
http://dsba.korea.ac.kr/seminar/?mod=document&uid=68
http://dsba.korea.ac.kr/seminar/?mod=document&uid=248

# Semi-supervised learning

- **Consistency Regularization**
  1. 모델을 이용해 unlabeled data의 분포 예측
  2. Unlabeled data에 noise 추가 (data augmentation)
  3. 예측한 분포를 augmented data의 정답 label로 사용해 모델로 학습






http://dsba.korea.ac.kr/seminar/?mod=document&uid=248

# Semi-supervised learning

- Unsupervised Data Augmentation (UDA, 2019)

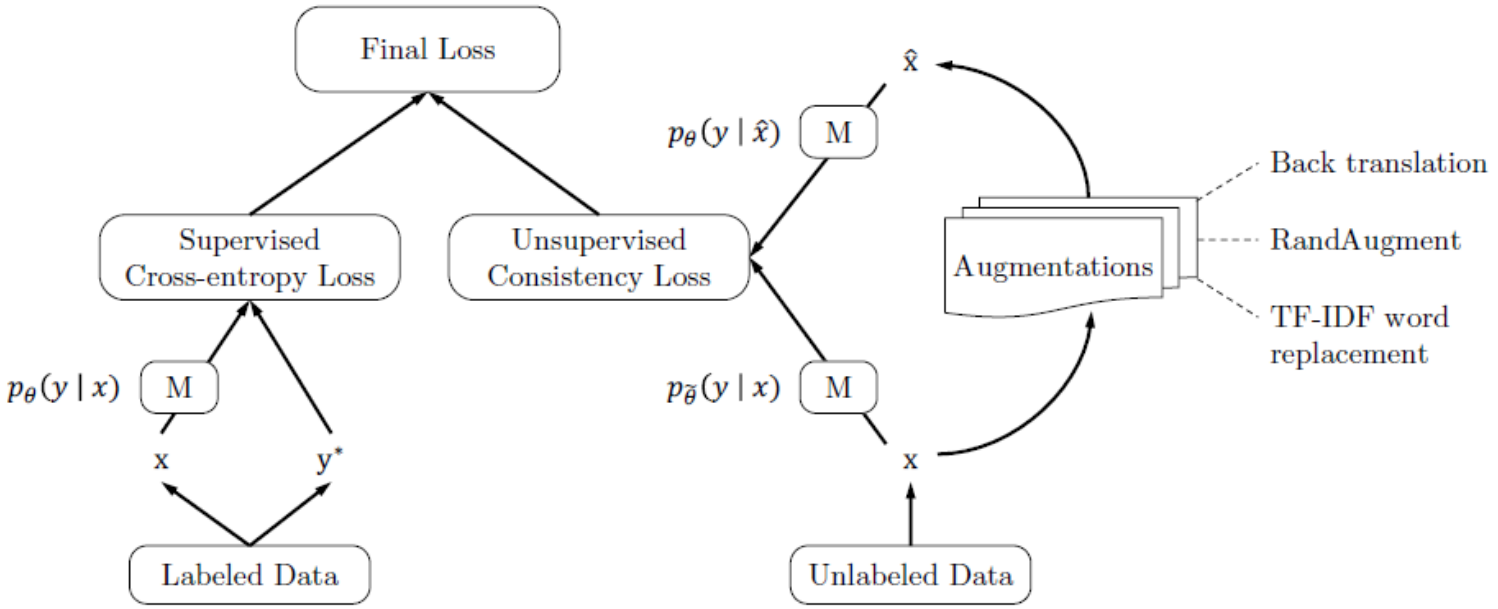


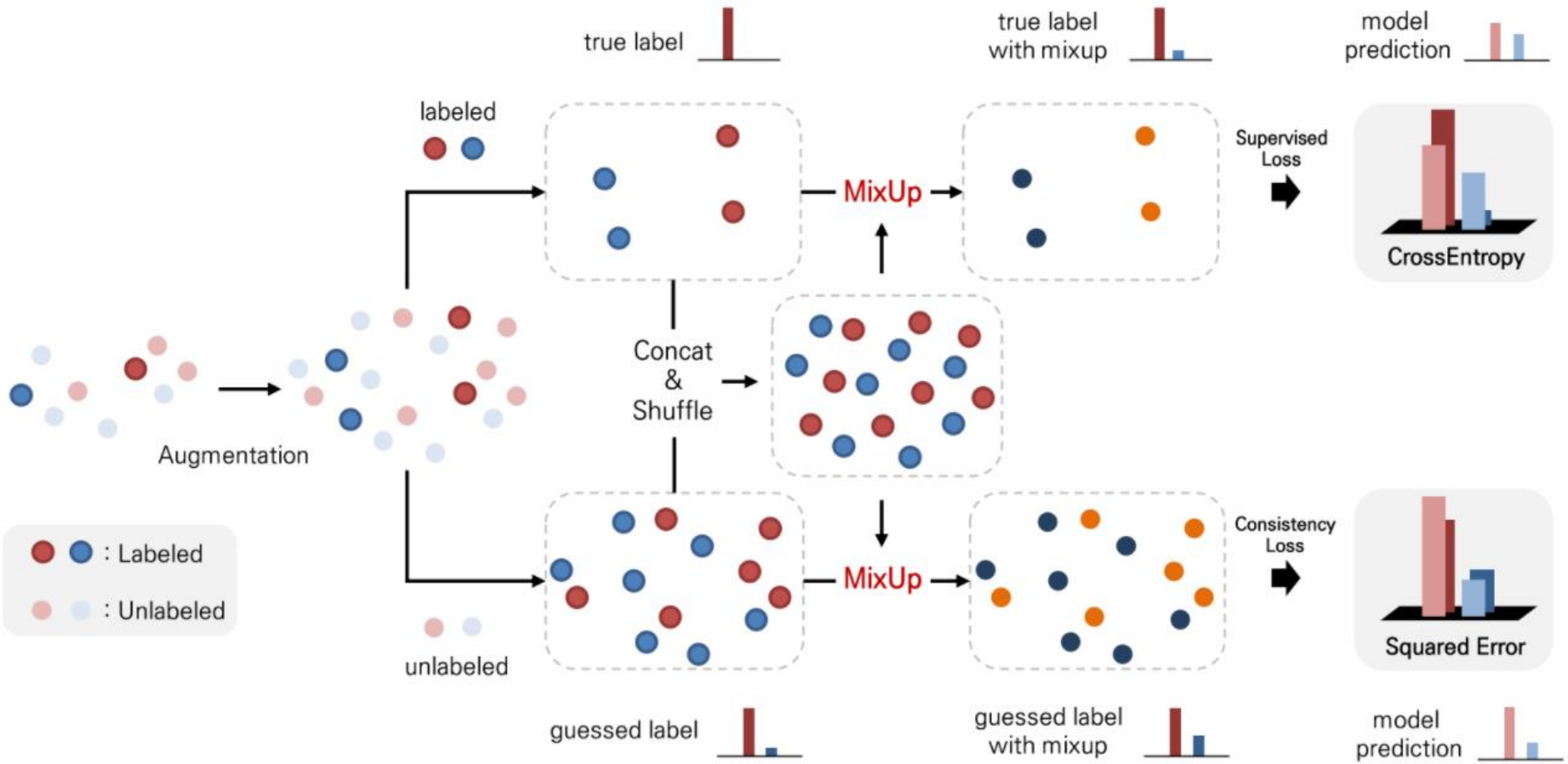

Figure 1: Training objective for UDA, where M is a model that predicts a distribution of $y$ given $x$.



https://arxiv.org/abs/1904.12848

# Semi-supervised learning

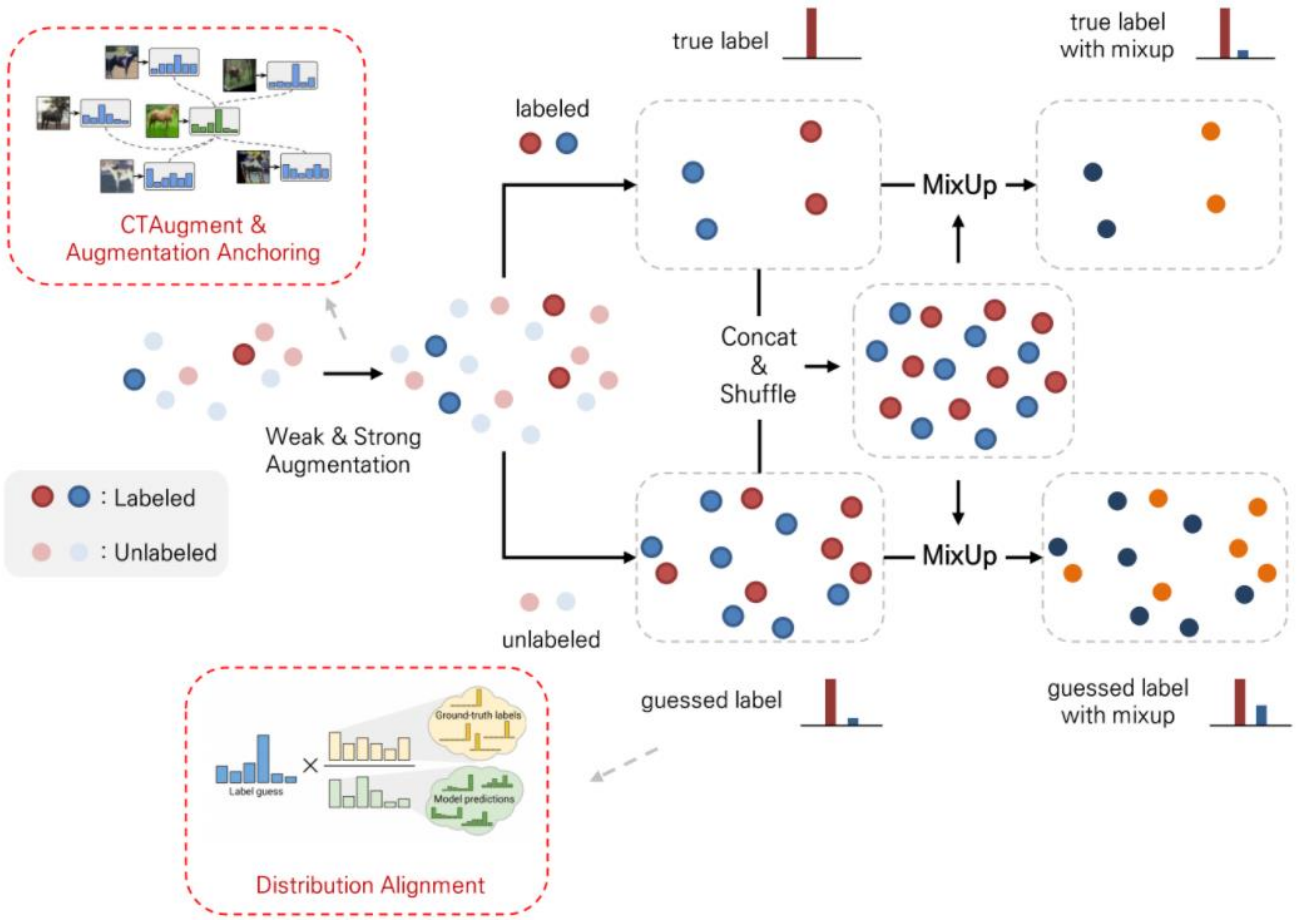
- MixMatch (NeurIPS 2019)




https://arxiv.org/abs/1905.02249
http://dsba.korea.ac.kr/seminar/?mod=document&uid=68
http://dsba.korea.ac.kr/seminar/?mod=document&uid=248

# Semi-supervised learning

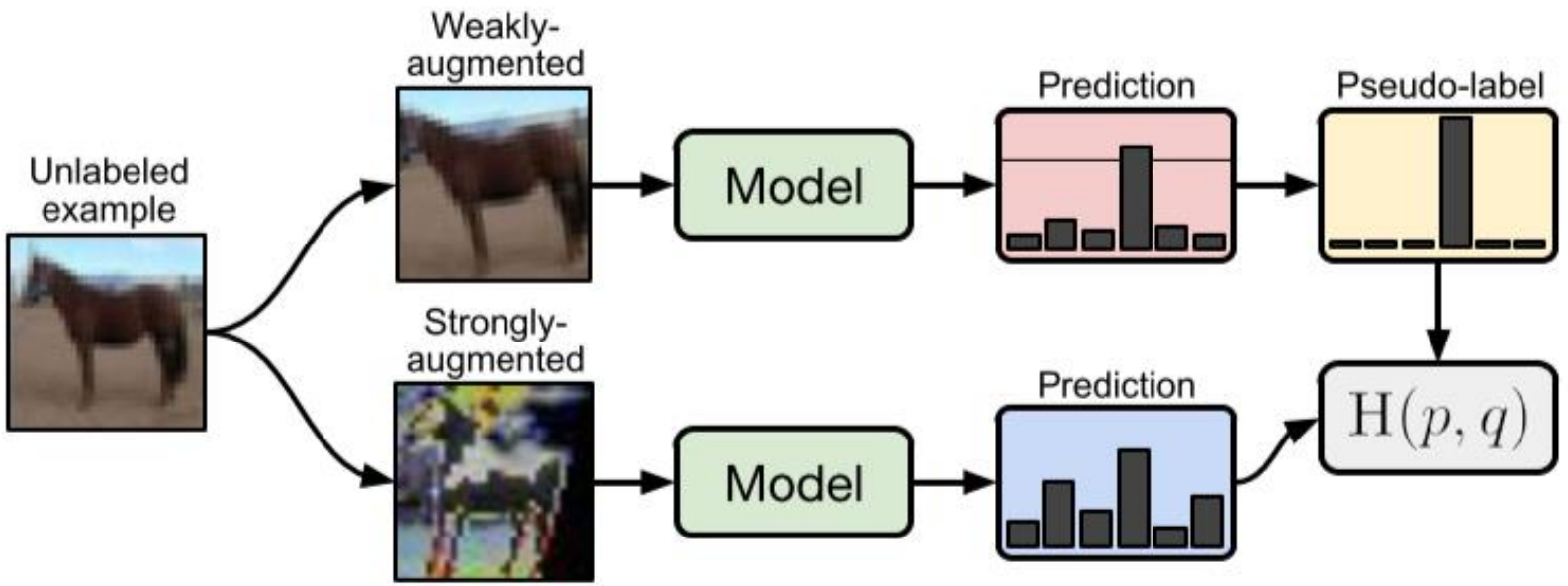
- ReMixMatch (ICLR 2020)




https://arxiv.org/abs/1911.09785
http://dsba.korea.ac.kr/seminar/?mod=document&uid=248

# Semi-supervised learning

- FixMatch (NeurIPS 2020)





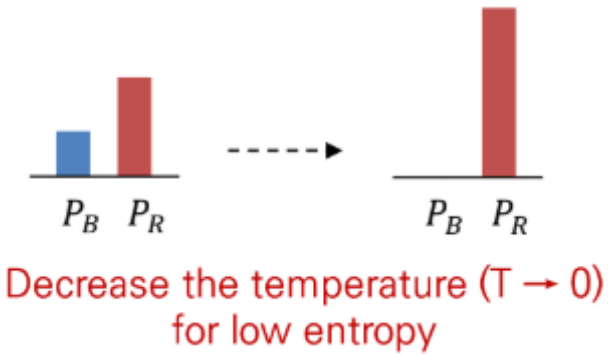

https://arxiv.org/abs/2001.07685

# FixMatch

- **Background**
  - Consistency regularization
  - Pseudo-labeling
    - Unlabeled data에 artificial label을 얻기 위해 모델 사용
    - Argmax로 hard label을 만들고 유지

$$\frac{1}{\mu B}\sum_{b=1}^{\mu B}\mathbf{1}(\max(q_b)\geq\tau)\mathrm{H}\left(\hat{q}_b,q_b\right)$$

      - $q_b = p_m(y|u_b)$, $\hat{q}_b = \mathrm{argmax}(q_b)$
      - $\tau$ : threshold hyperparameter
    - Hard label을 만드는 것은 entropy minimization과 깊은 관련이 있음



# FixMatch

- **FixMatch**
  - Notation
    - $\mathcal{X} = \{(x_b, p_b) : b \in (1, \dots, B)\}$ : a batch of $B$ labeled examples
      - $x_b, p_b$ : the training examples, one-hot labels
    - $\mathcal{U} = \{u_b : b \in (1, \dots, \mu B)\}$ : a batch of $\mu B$ unlabeled examples
      - $\mu$ : a hyperparameter that determines the relative sizes of $\mathcal{X}$ and $\mathcal{U}$
    - $p_m(y|x)$ : the predicted class distribution produced by the model for input $x$
    - $\mathcal{A}(\cdot), \alpha(\cdot)$ : strong and weak augmentation

# FixMatch

- **FixMatch**
  - Two cross-entropy loss term
    - Labeled data에 적용되는 supervised loss $l_s$

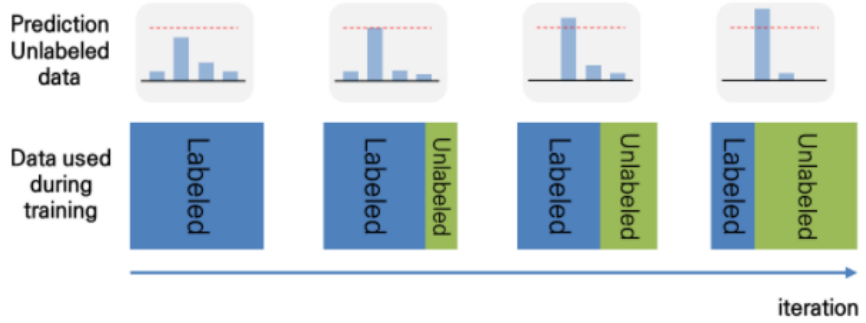
$$l_s = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}\left(p_b, p_m\left(y|\alpha(x_b)\right)\right)$$

    - Unsupervised loss $l_u$
      - Artificial label을 각 example에 대해 만들고 standard cross-entropy loss에 사용
        - $q_b = p_m\left(y|\alpha(u_b)\right)$ : weakly-augmented unlabeled image에 대한 model의 predicted class distribution
        - $\hat{q}_b = \mathrm{argmax}(q_b)$ : pseudo-label로 사용. Strongly-augmented image와 cross-entropy loss

$$l_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbf{1}\left(\max(q_b) \geq \tau\right)\mathrm{H}\left(\hat{q}_b, p_m\left(y|\mathcal{A}(u_b)\right)\right)$$

# FixMatch

- FixMatch
  - $l_s + \lambda_u l_u$를 최소화!
    - $\lambda_u$를 키우는 것이 최근 SSL algorithm의 경향이지만 FixMatch에서는 불필요
      - 학습 초기에는 $\max(q_b)$가 $\tau$보다 작고, 학습이 진행되면서 더 confident되고 $\max(q_b) > \tau$인 경우가 잦아짐




**Algorithm 1** FixMatch algorithm.

**Input:** Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \ldots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$, confidence threshold $\tau$, unlabeled data ratio $\mu$, unlabeled loss weight $\lambda_u$.
$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ *// Cross-entropy loss for labeled data*
**for** $b = 1$ **to** $\mu B$ **do**
  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to* $u_b$
  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ *// Compute prediction after applying weak data augmentation of* $u_b$
**end for**
$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), \tilde{u}_b)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*
**return** $\ell_s + \lambda_u \ell_u$

# FixMatch

- **Augmentation**
  - Two kinds of augmentation
    - Weak
      - Standard flip-and-shift augmentation
        - Randomly horizontally flipping with 50%
        - Randomly translating with up to 12.5% vertically and horizontally
    - Strong
      - AutoAugment
      - RandAugment
      - CTAugment (Control Theory Augment, in ReMixMatch)

      + Cutout

# FixMatch

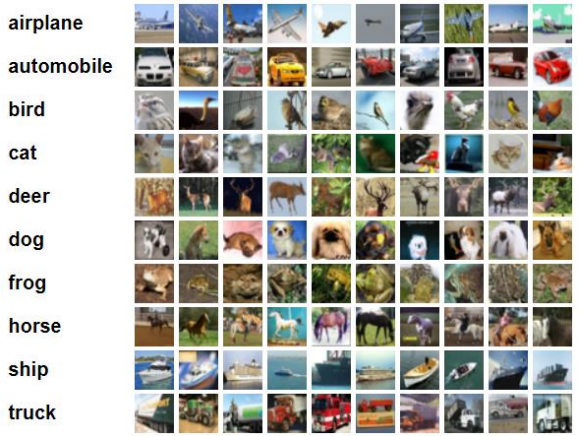
- **Additional important factors**
  - Semi-supervised learning의 성능은 알고리즘 이외에 regularization의 정도 등과 같은 다른 요소의 영향을 상당부분 받을 수 있음 (labeled data가 적은 경우에 특히!)
    - Regularization : simple weight decay
    - Optimizer : standard SGD with momentum
    - Learning rate scheduler : cosine learning rate decay ( $\eta \cos\left(\frac{7\pi k}{16K}\right)$ )
    - Exponential moving average of model parameters

# Related work

| Algorithm | Artificial label augmentation | Prediction augmentation | Artificial label post-processing | Notes |
|---|---|---|---|---|
| TS [39]/Π-Model [36] | Weak | Weak | None | |
| Temporal Ensembling [21] | Weak | Weak | None | Uses model from earlier in training |
| Mean Teacher [43] | Weak | Weak | None | Uses an EMA of parameters |
| Virtual Adversarial Training [28] | None | Adversarial | None | |
| UDA [45] | Weak | Strong | Sharpening | Ignores low-confidence artificial labels |
| MixMatch [3] | Weak | Weak | Sharpening | Averages multiple artificial labels |
| ReMixMatch [2] | Weak | Strong | Sharpening | Sums losses for multiple predictions |
| FixMatch | Weak | Strong | Pseudo-labeling | |

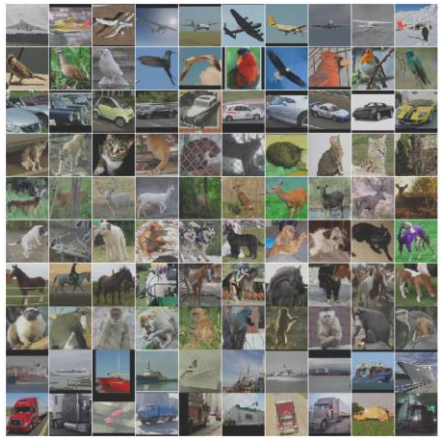
# Experiments

- Dataset
  - CIFAR-10
  - CIFAR-100
  - SVHN
  - STL-10
    - 96x96 pixels, 10 classes
    - 5,000 labeled images
    - 100,000 unlabeled images
  - ImageNet




**CIFAR-10**

**SVHN** (Street View House Number)



**STL-10**

https://www.cs.toronto.edu/~kriz/cifar.html
http://ufldl.stanford.edu/housenumbers/
https://cs.stanford.edu/~acoates/stl10/

# Experiments

- **CIFAR-10, CIFAR-100, SVHN**
  - Wide ResNet-28-2
  - 5 different "folds" of labeled data
  - CIFAR-100에서 ReMixMatch보다 낮은 성능
    - Distribution Alignment가 큰 영향을 미치는 것으로 보임

      → FixMatch + Distribution Alignment : 40.14% error rate with 400 labeled examples!

Table 2: Error rates for CIFAR-10, CIFAR-100 and SVHN on 5 different folds. FixMatch (RA) uses RandAugment [10] and FixMatch (CTA) uses CTAugment [2] for strong-augmentation. All baseline models (Π-Model [36], Pseudo-Labeling [22], Mean Teacher [43], MixMatch [3], UDA [45], and ReMixMatch [2]) are tested using the same codebase.

| | CIFAR-10 | | | CIFAR-100 | | | SVHN | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | 40 labels | 250 labels | 4000 labels | 400 labels | 2500 labels | 10000 labels | 40 labels | 250 labels | 1000 labels |
| Π-Model | - | 54.26±3.97 | 14.01±0.38 | - | 57.25±0.48 | 37.88±0.11 | - | 18.96±1.92 | 7.54±0.36 |
| Pseudo-Labeling | - | 49.78±0.43 | 16.09±0.28 | - | 57.38±0.46 | 36.21±0.19 | - | 20.21±1.09 | 9.94±0.61 |
| Mean Teacher | - | 32.32±2.30 | 9.19±0.19 | - | 53.91±0.57 | 35.83±0.24 | - | 3.57±0.11 | 3.42±0.07 |
| MixMatch | 47.54±11.50 | 11.05±0.86 | 6.42±0.10 | 67.61±1.32 | 39.94±0.37 | 28.31±0.33 | 42.55±14.53 | 3.98±0.23 | 3.50±0.28 |
| UDA | 29.05±5.93 | 8.82±1.08 | 4.88±0.18 | 59.28±0.88 | 33.13±0.22 | 24.50±0.25 | 52.63±20.51 | 5.69±2.76 | **2.46**±0.24 |
| ReMixMatch | **19.10**±9.64 | **5.44**±0.05 | 4.72±0.13 | **44.28**±2.06 | **27.43**±0.31 | **23.03**±0.56 | **3.34**±0.20 | **2.92**±0.48 | 2.65±0.08 |
| FixMatch (RA) | **13.81**±3.37 | **5.07**±0.65 | **4.26**±0.05 | 48.85±1.75 | 28.29±0.11 | **22.60**±0.12 | **3.96**±2.17 | **2.48**±0.38 | **2.28**±0.11 |
| FixMatch (CTA) | **11.39**±3.35 | **5.07**±0.33 | **4.31**±0.15 | 49.95±3.01 | 28.64±0.24 | 23.18±0.11 | 7.65±7.65 | **2.64**±0.64 | **2.36**±0.19 |

# Experiments

- STL-10
  - Wide ResNet-37-2
  - Five of the predefined folds of 1,000 labeled images each

Table 3: Error rates for STL-10 on 1000-label splits. All baseline models are tested using the same codebase.

| Method | Error rate | Method | Error rate |
|---|---|---|---|
| Π-Model | 26.23±0.82 | MixMatch | 10.41±0.61 |
| Pseudo-Labeling | 27.99±0.80 | UDA | 7.66±0.56 |
| Mean Teacher | 21.43±2.39 | ReMixMatch | **5.23**±0.45 |
| FixMatch (RA) | 7.98±1.50 | FixMatch (CTA) | **5.17**±0.63 |

# Experiments

- **ImageNet**
  - ResNet-50 & RandAugment
  - Like UDA, 10% labeled data + 나머지 unlabeled data
  - Top-1 error rate
    - FixMatch : 28.54 ± 0.52% (UDA보다 2.68% 낮음)
  - Top-5 error rate : 10.87 ± 0.28%
  - $S^4L$ : supervised → pseudo-label re-training (semi) → supervised fine-tuning (self)

    → pseudo-label re-training에서는 좋은 성능

    → 모든 과정을 통과했을 때 거의 유사한 성능을 얻을 수 있었음



https://arxiv.org/abs/1905.03670

# Experiments

- **Barely Supervised Learning**
  - CIFAR-10에서 class당 하나의 example만 사용해서 진행
    - 4개의 데이터셋 + 각 데이터셋에 대해 4번 학습
      - 48.58~85.32% (median of 64.28%) test accuracy
    - 첫 번째 데이터셋으로 학습한 4개의 모델은 모두 61~67%의 accuracy
    - 두 번째 데이터셋으로 학습한 4개의 모델은 모두 68~75%의 accuracy
  - Low-quality example을 선택한 경우 모델이 특정 class를 학습하는데 더 어렵지 않을까?
    - 8개의 training dataset을 구축 (prototypicality)
    - 여러 CIFAR-10 모델로 데이터를 representative한 순으로 sorting
    - 8개의 bucket으로 나누고 bucket별로 8개의 one labeled example per class dataset 구축
      → 가장 representative한 데이터는 첫 번째, outlier들은 마지막
    - 가장 prototypical한 example은 **78%**, middle of the distribution은 **65%**, outlier들은 **10%** accuracy

# Experiments

- Barely Supervised Learning

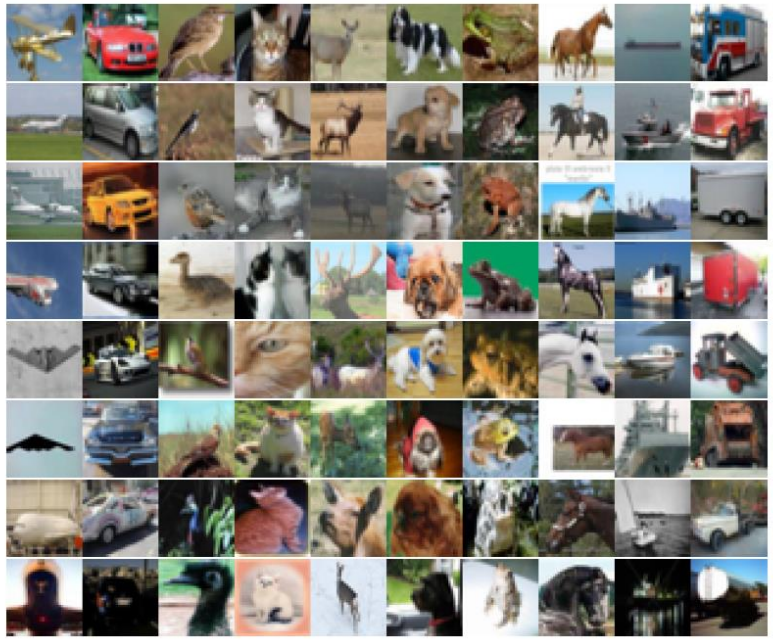


Figure 5: Labeled training data for the 1-label-per-class semi-supervised experiment. Each row corresponds to the complete labeled training set for one run of our algorithm, sorted from the most prototypical dataset (first row) to least prototypical dataset (last row).

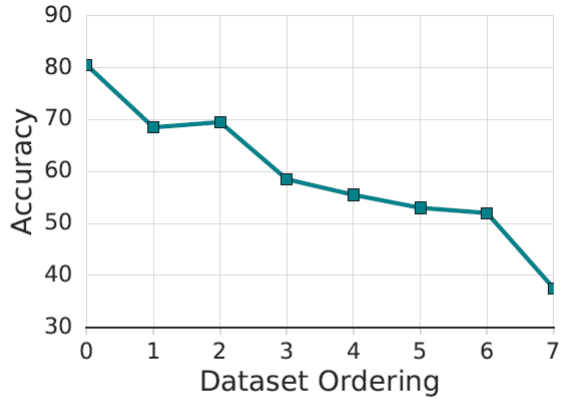
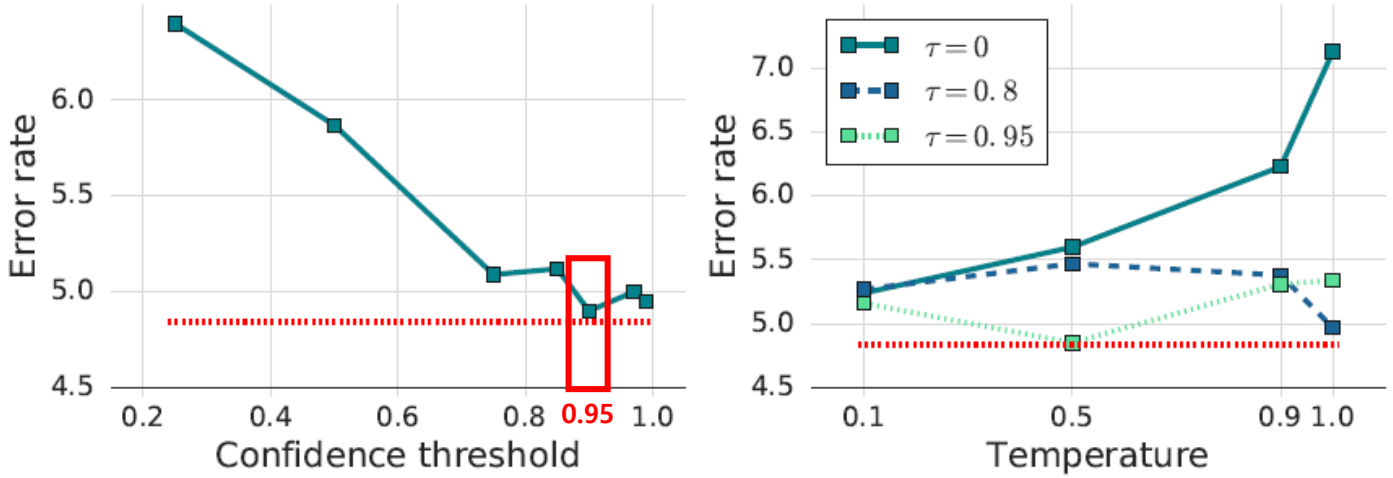



Figure 6: Accuracy of the model when trained on the 1-label-per-class datasets from Figure 5, ordered from most prototypical (top row) to least (bottom row).

# Experiments

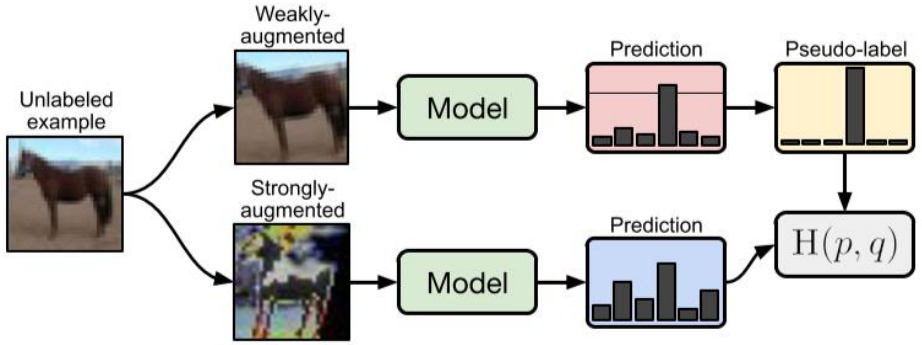
- **Ablation Study**
  - 250 label split from CIFAR-10 + CTAugment
  - **Sharpening and Thresholding**
    - Temperature $T$와 confidence threshold $\tau$에 대해 실험 진행
    - Confidence threshold $\tau = 0.95$
    - $\tau$가 적용되었다면 temperature $T$는 큰 영향을 미치지 않음



# Experiments

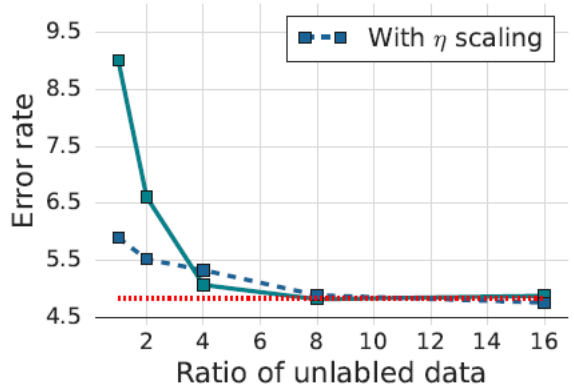
- **Ablation Study**
  - 250 label split from CIFAR-10 + CTAugment
  - **Augmentation Strategy**
    - Label guessing (pseudo-label)을 strong augmentation으로 교체
      - 학습 초기에 발산 → pseudo-label은 weakly augmented data로 생성되어야함
    - Model's prediction에 weak augmentation 적용
      - 45% accuracy에 도달했지만, 학습이 안정하지 않고 점차 12%로 떨어짐

        → model prediction에는 strong data augmentation이 필요
    - 이런 결과는 supervised learning에서도 확인됨



# Experiments

- **Ablation Study**
  - 250 label split from CIFAR-10 + CTAugment
  - **Ratio of Unlabeled Data**
    - Unlabeled data의 비율에 따른 성능 비교
      - Unlabeled data를 많이 사용할수록 성능이 좋아짐 → UDA와 동일
      - Learning rate $\eta$를 batch size에 따라 linear하게 변화를 주면 좋음 (특히 $\eta$가 작을 때!)




(a)

# Experiments

- **Ablation Study**
  - 250 label split from CIFAR-10 + CTAugment
  - **Optimizer and Learning Rate Schedule**
    - 이전 SSL 연구들에서는 실험이 거의 진행되지 않은 optimizers, hyperparameters
      - SGD with momentum of 0.9가 제일 좋음 (4.84%)
        - Momentum이 없을 때 5.19%
        - Nesterov variant of momentum은 5% 아래의 error를 얻는데 필요X
        - Adam은 별로
      - Cosine learning rate decay가 많이 사용되는데,
        FixMatch에서는 linear learning rate decay도 나쁘지 않음
        - Cosine learning rate decay는 적절한 decaying rate를 고르는 것이 중요
        - No decay는 0.86% accuracy가 감소

Table 5: Ablation study on optimizers. Error rates are reported on a single 250-label split from CIFAR-10.

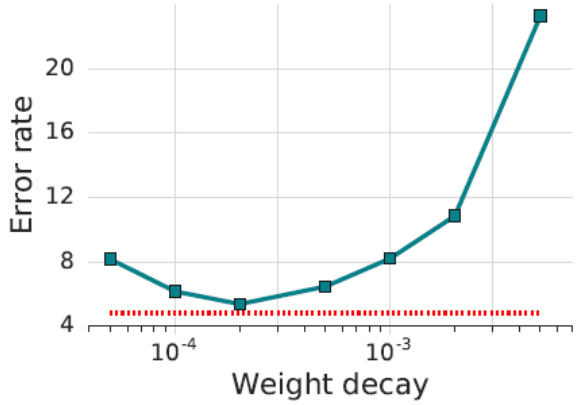
| Optimizer | Hyperparameters | | | Error |
|---|---|---|---|---|
| SGD | $\eta = 0.03$ | $\beta = 0.90$ | Nesterov | **4.84** |
| SGD | $\eta = 0.03$ | $\beta = 0.90$ | | **4.86** |
| SGD | $\eta = 0.20$ | $\beta = 0.0$ | Nesterov | 5.19 |
| Adam | $\eta = 0.0003$ | $\beta_1 = 0.9$ | $\beta_2 = 0.999$ | 5.37 |

Table 6: Ablation study on learning rate decay schedules. Error rates are reported on a single 250-label split from CIFAR-10.

| Decay Schedule | Error |
|---|---|
| Cosine (FixMatch) | 4.84 |
| Linear Decay (end 0.01) | 4.95 |
| Linear Decay (end 0.02) | 5.55 |
| No Decay | 5.70 |

# Experiments

- **Ablation Study**
  - 250 label split from CIFAR-10 + CTAugment
  - **Weight Decay**
    - Labeled data가 적은 경우에 weight decay가 굉장히 중요함
      - Optimal보다 크거나 작은 값을 선택하면 10%이상의 성능차가 나타날 수 있음




(d)

# 감 사 합 니 다