

- ❖ FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling (NeurIPS, 2021)
- Tokyo Institute of Technology에서 연구하였으며, 2022년 12월 20일 기준으로 135회 인용됨
- 모든 클래스에 동일한 Confidence 기준을 적용하지 않고 각 클래스의 난이도에 따른 다른 기준을 적용하자는 것이 핵심!

FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling

Bowen Zhang* Tokyo Institute of Technology bowen.z.ab@m.titech.ac.jp	Yidong Wang* Tokyo Institute of Technology wang.y.ca@m.titech.ac.jp	
Wenxin Hou Microsoft wenxinhou@microsoft.com	Hao Wu Tokyo Institute of Technology wu.h.aj@m.titech.ac.jp	Jindong Wang† Microsoft Research Asia jindongwang@microsoft.com
Manabu Okumura† Tokyo Institute of Technology oku@pi.titech.ac.jp	Takahiro Shinozaki† Tokyo Institute of Technology shinot@ict.e.titech.ac.jp	

Abstract

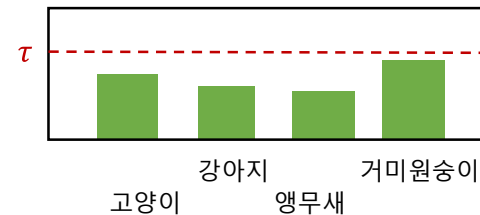
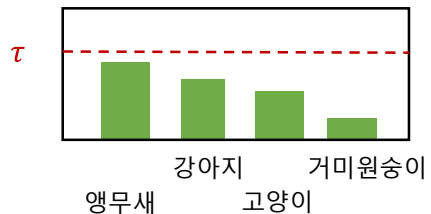
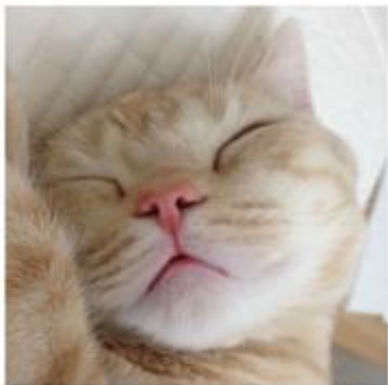
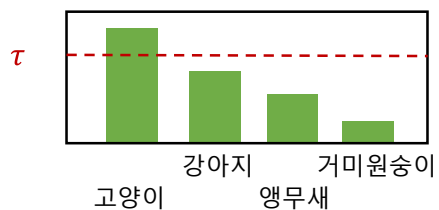
The recently proposed FixMatch achieved state-of-the-art results on most semi-supervised learning (SSL) benchmarks. However, like other modern SSL algorithms, FixMatch uses a pre-defined constant threshold for all classes to select unlabeled data that contribute to the training, thus failing to consider different learning status and learning difficulties of different classes. To address this issue, we propose Curriculum Pseudo Labeling (CPL), a curriculum learning approach to leverage unlabeled data according to the model's learning status. The core of CPL is to flexibly adjust thresholds for different classes at each time step to let pass informative unlabeled data and their pseudo labels. CPL does not introduce additional parameters or computations (forward or backward propagation). We apply CPL to FixMatch and call our improved algorithm *FlexMatch*. FlexMatch achieves state-of-the-art performance on a variety of SSL benchmarks, with especially strong performances when the labeled data are extremely limited or when the task is challenging. For example, FlexMatch achieves **13.96%** and **18.96%** error rate reduction over FixMatch on CIFAR-100 and STL-10 datasets respectively, when there are only 4 labels per class. CPL also significantly boosts the convergence speed, e.g., FlexMatch can use only 1/5 training time of FixMatch to achieve even better performance. Furthermore, we show that CPL can be easily adapted to other SSL algorithms and remarkably improve their performances. We open-source our code at <https://github.com/TorchSSL/TorchSSL>.

FlexMatch

❖ FlexMatch

- FixMatch의 문제점을 개선한 방법론

- ✓ 분류가 쉬운 범주의 경우 처음부터 Confidence가 높은 데이터가 다수 Pseudo Labeling이 되어 계속 더 잘 학습할 수 있게 유도됨
- ✓ 하지만, 분류가 어려운 범주는 Confidence가 높은 레이블이 없는 데이터가 많지 않기 때문에 비지도 학습의 본 의도인 레이블이 없는 데이터의 정보 활용이 어려움

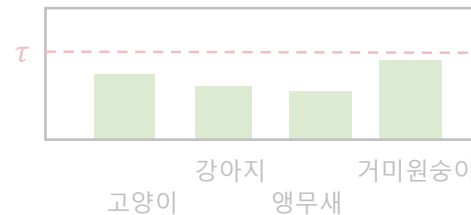
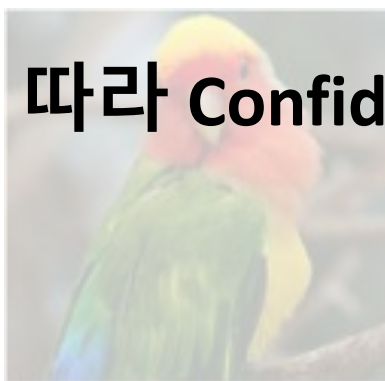
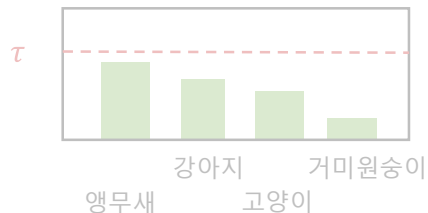
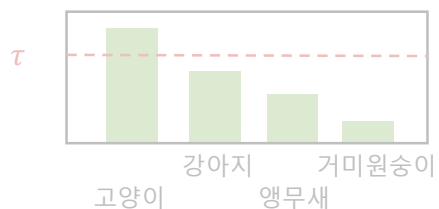


FlexMatch

❖ FlexMatch

- FixMatch의 문제점을 개선한 방법론

- ✓ 분류가 쉬운 범주의 경우 처음부터 Confidence가 높은 데이터가 다수 Pseudo Labeling이 되어 계속 더 잘 학습할 수 있게 유도됨
- ✓ 하지만, 분류가 어려운 범주는 Confidence가 높은 레이블이 없는 데이터가 많지 않기 때문에 비지도 학습의 본 의도인 레이블이 없는 데이터의 정보 활용이 어려움

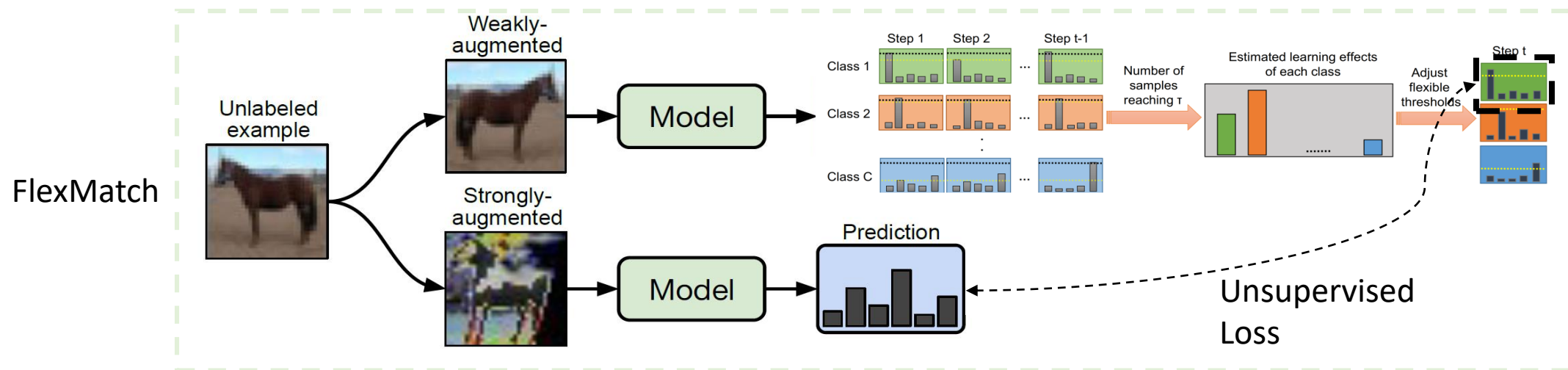
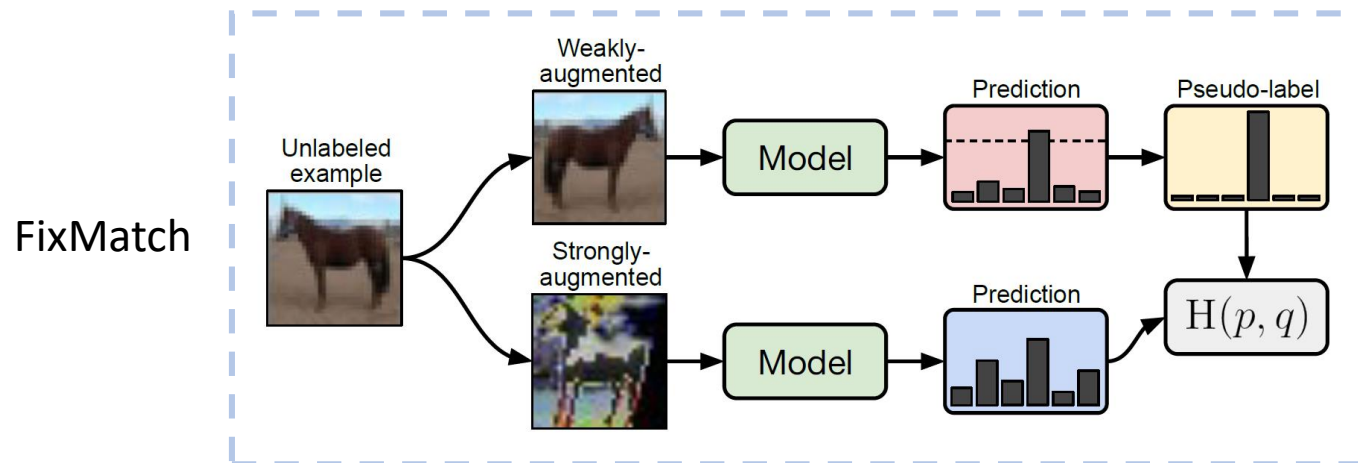


범주의 난이도에 따라 Confidence를 적절하게 조절!

FlexMatch

❖ FlexMatch

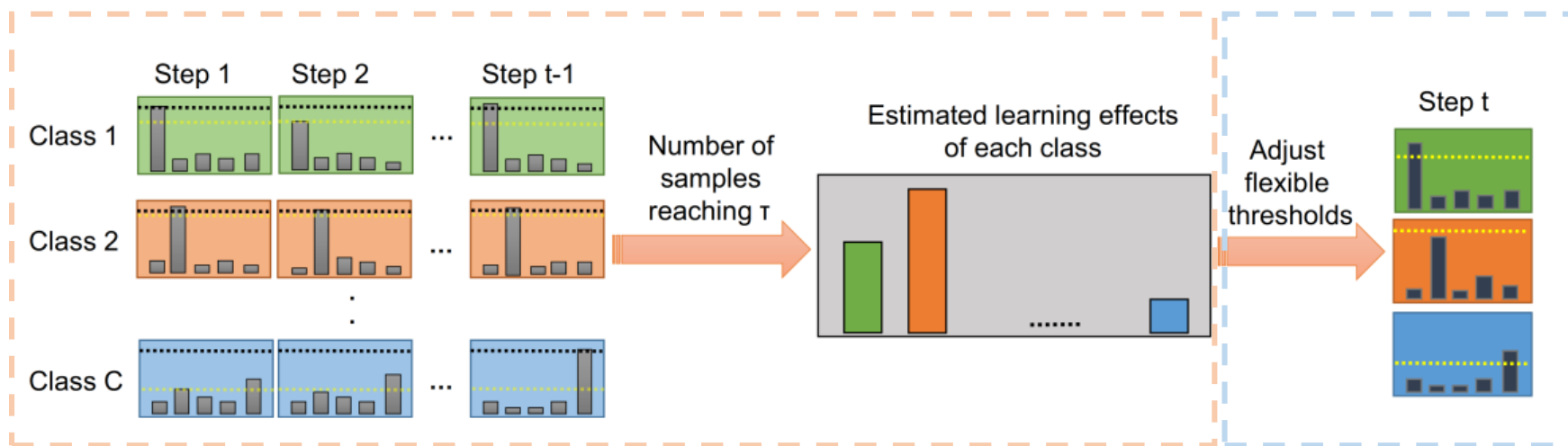
- 범주의 난이도에 따른 Confidence를 적절하게 조절하기 위해 CPL(Curriculum Pseudo Labeling)을 기존 FixMatch에 결합



FlexMatch

❖ FlexMatch

- 범주의 난이도에 따른 Confidence를 적절하게 조절하기 위해 CPL(Curriculum Pseudo Labeling)을 기존 FixMatch에 결합



1. 클래스 별 학습 난이도 설정 및 정규화

2. 유연한 임계값 설정 및

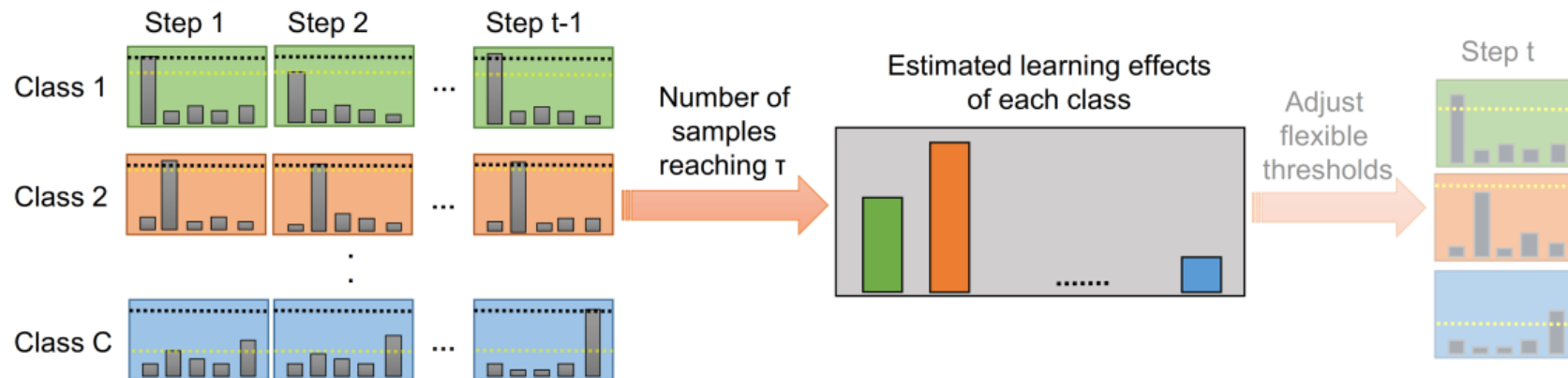
FlexMatch

❖ FlexMatch

- Learning effect 측정

- ✓ 매 time step마다 학습에 사용되는 레이블이 없는 데이터의 양을 클래스 별 난이도를 사용해 조절
- ✓ Learning effect → 클래스 별 학습 난이도
- ✓ Confidence가 미리 지정한 임계값을 넘기면서, 해당 클래스로 예측된 레이블이 없는 샘플의 개수를 통해 학습 난이도 결정

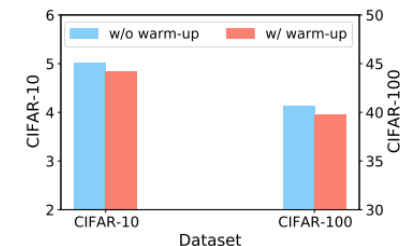
$$\sigma_t(c) = \sum_{n=1}^N \underbrace{\mathbb{1}(\max(p_{m,t}(y|u_n)) > \tau)}_{\text{Number of samples reaching } \tau} \cdot \underbrace{\mathbb{1}(\arg \max(p_{m,t}(y|u_n)) = c)}_{\text{Estimated learning effects of each class}}$$



FlexMatch

❖ FlexMatch

- Learning effect 정규화 → 0~1 사이의 값으로 조절
- 추가로 전체 데이터의 수에서 학습 난이도 설정에 사용된 레이블이 없는 데이터의 수를 뺀 값을 분모에 넣어 Max 연산에 사용함
 - ✓ Heavy fluctuation 방지하거나, 모든 클래스의 threshold를 0에서부터 점진적으로 증가시키기 위해 적용

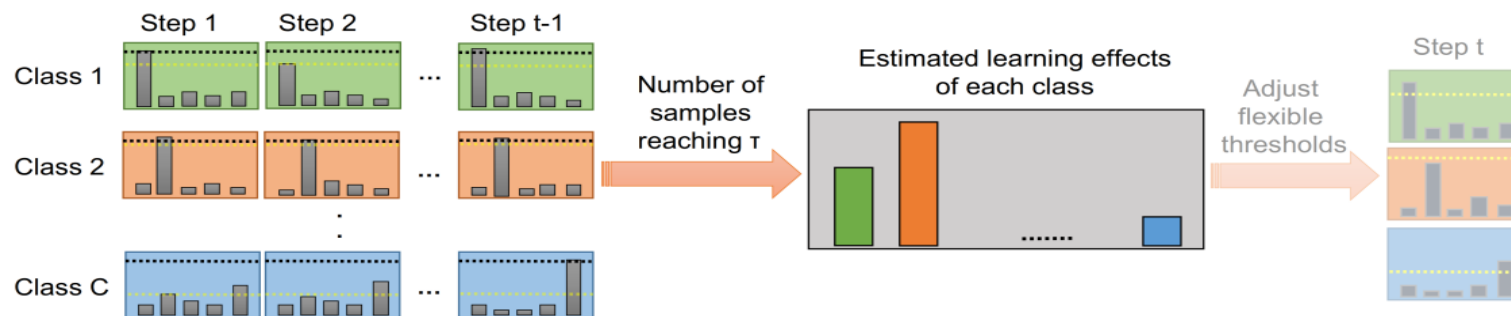


(c) Threshold warm-up

$$\sigma_t(c) = \sum_{n=1}^N \mathbb{1}(\max(p_{m,t}(y|u_n)) > \tau) \cdot \mathbb{1}(\arg \max(p_{m,t}(y|u_n)) = c)$$

Threshold Warm-up

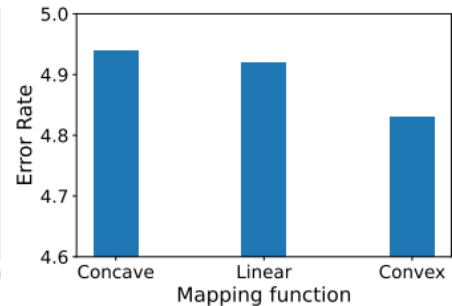
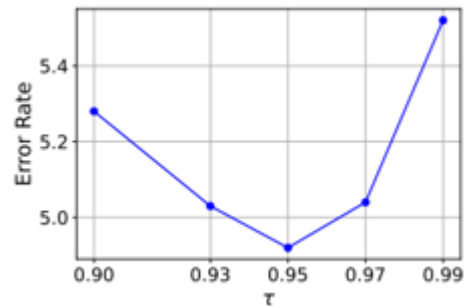
$$\beta_t(c) = \frac{\sigma_t(c)}{\max_c \sigma_t} \xrightarrow{\text{추가 진행}} \beta_t(c) = \frac{\sigma_t(c)}{\max \left\{ \max_c \sigma_t, N - \sum_c \sigma_t \right\}}$$



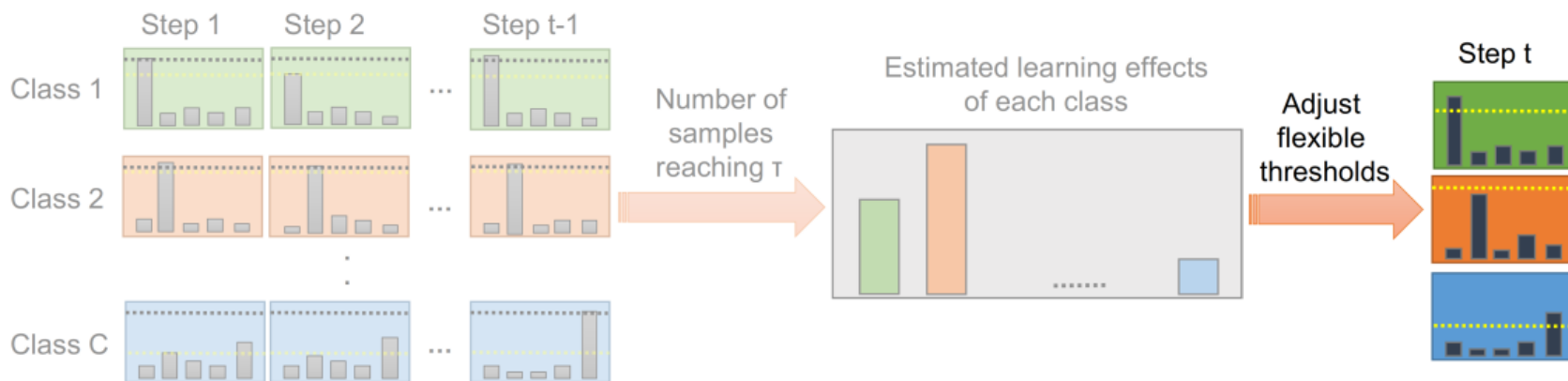
FlexMatch

❖ FlexMatch

- Flexible threshold 설정 / 매 time step마다 클래스 별 임계 값 업데이트
- 학습 초기에 베타(학습 난이도)의 변동을 완화하기 위해서 Mapping function을 적용함
 - ✓ 이는 베타의 변동에 따른 임계 값의 변화를 완화시켜주는 장치 → Convex function 사용



$$\mathcal{T}_t(c) = \beta_t(c) \cdot \tau \xrightarrow[\text{threshold}]{\text{Mapping Function 적용}} \mathcal{T}_t(c) = \mathcal{M}(\beta_t(c)) \cdot \tau$$



FlexMatch

❖ FlexMatch – Overall Framework

