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

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

How to Reweight Examples in SSL Using Meta-Learning

Background

What's semi-supervised learning?

Few labeled data

+

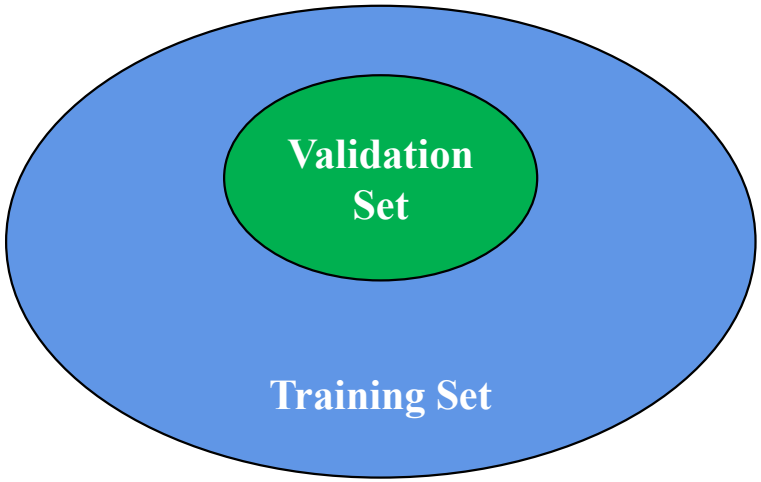
Numerous unlabeled data

What's Meta-learning?

Meta-learning refers to **learning about learning**.
If machine learning learns how to best use information in data to make predictions, then meta-learning or meta machine learning learns how to best use the predictions from machine learning algorithms to make predictions.

We only consider **Model Selection and Tuning** in Meta-Learning here.

- Validation Set(**clean**, **unbiased**, **full-labeled** …)
- Training Set(**noisy**, **biased**, **unlabeled** or **partial labeled** …)



How to use a small validation set to guide training?

Meta Pseudo Labels

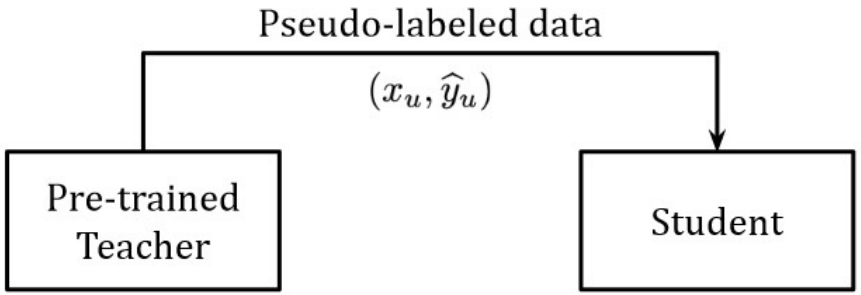
Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, Quoc V. Le
Google AI, Brain Team, Mountain View, CA 94043
{hyhieu,zihangd,qizhex,thangluong,qvl}@google.com

CVPR 2021

Motivation

Pseudo-Label

$$y_i' = \begin{cases} 1 & \text{if } i = \text{argmax}_{i'} f_{i'}(x) \\ 0 & \text{otherwise} \end{cases}$$

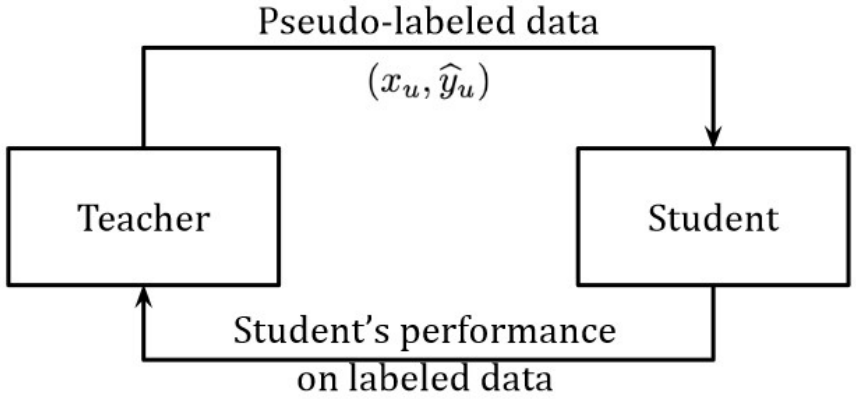


Confirmation Bias

If the pseudo labels are inaccurate, the student will learn from **inaccurate** data. As a result, the student may **not** get **significantly better** than the teacher.

Meta Pseudo Labels

Utilize **feedback** from the student to help the teacher update its parameters to generate more accurate pseudo-labels.



Method

Pseudo Labels (PL) trains the student model to minimize the cross-entropy loss on unlabeled data:

$$\theta_S^{\mathrm{PL}} = \underset{\theta_S}{\operatorname{argmin}} \underbrace{\mathbb{E}_{x_u}\Big[\mathrm{CE}\big(T(x_u;\theta_T), S(x_u;\theta_S)\big)\Big]}_{:=\mathcal{L}_u\left(\theta_T,\theta_S\right)}$$

Given a good teacher, the hope of Pseudo Labels is that the obtained θ_S^{PL} would ultimately achieve a low loss on labeled data:

$$\mathbb{E}_{x_l,y_l}\Big[\mathrm{CE}\big(y_l, S(x_l;\theta_S^{\mathrm{PL}})\big)\Big] := \mathcal{L}_l\left(\theta_S^{\mathrm{PL}}\right) \Rightarrow$$

Depends on θ_T via $T(x_u;\theta_T)$

We can explicitly express the dependency as $\theta_S^{\mathrm{PL}}(\theta_T)$, Therefore, we could further optimize $\mathcal{L}_l$ with respect to θ_T

$$\begin{aligned} &\min_{\theta_T} \quad \mathcal{L}_l\left(\theta_S^{\mathrm{PL}}(\theta_T)\right), \\ &\text{where} \quad \theta_S^{\mathrm{PL}}(\theta_T) = \underset{\theta_S}{\operatorname{argmin}}\, \mathcal{L}_u\big(\theta_T, \theta_S\big). \end{aligned}$$

Method

Practical approximation

Approximate the multi-step argmin_{θ_S} with the one-step gradient update of θ_S.

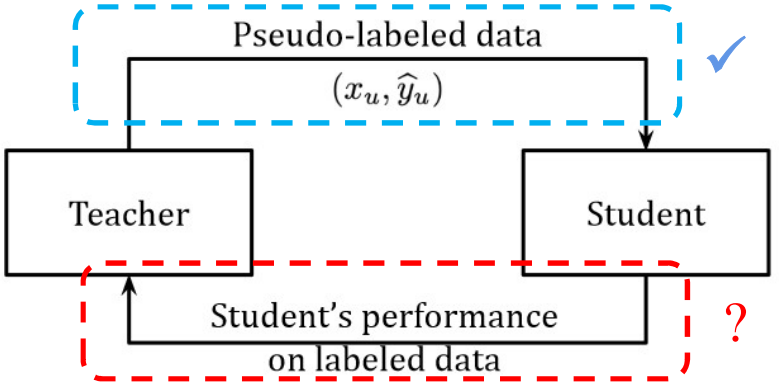
$$\theta_S^{\text{PL}}(\theta_T) \approx \theta_S - \eta_S \cdot \nabla_{\theta_S} \mathcal{L}_u\left(\theta_T, \theta_S\right)$$

The final practical teacher objective is :

$$\min_{\theta_T} \quad \mathcal{L}_l\Big(\theta_S - \eta_S \cdot \nabla_{\theta_S} \mathcal{L}_u\left(\theta_T, \theta_S\right)\Big).$$

So the teacher parameter is **not fixed** anymore. The learning objectives of teacher and student give rise to an **alternating optimization** procedure.

- Student: draw a batch of unlabeled data x_u, then sample $T(x_u; \theta_T)$ from teacher's prediction, and optimize objective 1 with SGD: $\theta'_S = \theta_S - \eta_S \nabla_{\theta_S} \mathcal{L}_u(\theta_T, \theta_S)$, ✓
- Teacher: draw a batch of labeled data (x_l, y_l), and "reuse" the student's update to optimize objective 3 with SGD: $\theta'_T = \theta_T - \eta_T \nabla_{\theta_T} \mathcal{L}_l\big(\underbrace{\theta_S - \nabla_{\theta_S} \mathcal{L}_u(\theta_T, \theta_S)}_{=\theta'_S \text{ reused from student's update}}\big)$. ?



Method

Derivation of the Teacher's Update Rule

$$\underbrace{\frac{\partial R}{\partial \theta_T}}_{1\times|T|} = \frac{\partial}{\partial \theta_T}\mathrm{CE}\left(y_l, S\left(x_l; \mathbb{E}_{\widehat{y}_u\sim T(x_u;\theta_T)}\left[\theta_S - \eta_S \nabla_{\eta_S}\mathrm{CE}(\widehat{y}_u, S(x_u;\theta_S))\right]\right)\right)$$

$$= \frac{\partial}{\partial \theta_T}\mathrm{CE}\left(y_l, S\left(x_l; \bar{\theta}'_S\right)\right) \longrightarrow \mathbb{E}_{\widehat{y}_u\sim T(x_u;\theta_T)}\left[\theta_S - \eta_S \nabla_{\eta_S}\mathrm{CE}(\widehat{y}_u, S(x_u;\theta_S))\right]$$

$$= \underbrace{\frac{\partial \mathrm{CE}\left(y_l, S\left(x_l;\theta'_S\right)\right)}{\partial \theta_S}\bigg|_{\theta_S=\bar{\theta}'_S}}_{1\times|S|} \cdot \underbrace{\frac{\partial \bar{\theta}'_S}{\partial \theta_T}}_{|S|\times|T|} \quad (1)$$

Can be computed via back-propagation

$$\underbrace{\frac{\partial \bar{\theta}'_S}{\partial \theta_T}}_{|S|\times|T|} = \frac{\partial}{\partial \theta_T}\mathbb{E}_{\widehat{y}_u\sim T(x_u;\theta_T)}\left[\theta_S - \eta_S \nabla_{\eta_S}\mathrm{CE}(\widehat{y}_u, S(x_u;\theta_S))\right]$$

$$= \frac{\partial}{\partial \theta_T}\mathbb{E}_{\widehat{y}_u\sim T(x_u;\theta_T)}\left[\theta_S - \eta_S \cdot \left(\frac{\partial \mathrm{CE}\left(\widehat{y}_u, S(x_u;\theta_S)\right)}{\partial \theta_S}\bigg|_{\theta_S=\theta_S}\right)^\top\right]$$

$$= -\eta_S \cdot \frac{\partial}{\partial \theta_T}\mathbb{E}_{\widehat{y}_u\sim T(x_u;\theta_T)}\left[\underbrace{g_S(\widehat{y}_u)}_{|S|\times 1}\right] \quad (2)$$

We always treat θ_S as **fixed** parameters and ignore its **higher-order** dependency on θ_T .

Method

Derivation of the Teacher's Update Rule

$$\underbrace{\frac{\partial \bar{\theta}_S^{(t+1)}}{\partial \theta_T}}_{|S|\times|T|} = -\eta_S \cdot \frac{\partial}{\partial \theta_T} \mathbb{E}_{\widehat{y}_u \sim T(x_u;\theta_T)} \left[g_S(\widehat{y}_u) \right]$$

$$= -\eta_S \cdot \mathbb{E}_{\widehat{y}_u \sim T(x_u;\theta_T)} \left[\underbrace{g_S(\widehat{y}_u)}_{|S|\times 1} \cdot \underbrace{\frac{\partial \log P\left(\widehat{y}_u | x_u; \theta_T\right)}{\partial \theta_T}}_{1\times|T|} \right]$$

$$= \eta_S \cdot \mathbb{E}_{\widehat{y}_u \sim T(x_u;\theta_T)} \left[\underbrace{g_S(\widehat{y}_u)}_{|S|\times 1} \cdot \underbrace{\frac{\partial \mathrm{CE}\left(\widehat{y}_u, T(x_u;\theta_T)\right)}{\partial \theta_T}}_{1\times|T|} \right] \quad (3)$$

Substitute Equation (3) into Equation (1)

$$\underbrace{\frac{\partial R}{\partial \theta_T}}_{1\times|T|} = \underbrace{\left. \frac{\partial \mathrm{CE}\left(y_l, S\left(x_l; \bar{\theta}'_S\right)\right)}{\partial \theta_S} \right|_{\theta_S = \bar{\theta}'_S}}_{1\times|S|} \cdot \underbrace{\frac{\partial \bar{\theta}'_S}{\partial \theta_T}}_{|S|\times|T|}$$

$$= \eta_S \cdot \underbrace{\left. \frac{\partial \mathrm{CE}\left(y_l, S\left(x_l; \bar{\theta}'_S\right)\right)}{\partial \theta_S} \right|_{\theta_S = \bar{\theta}'_S}}_{1\times|S|} \cdot \mathbb{E}_{\widehat{y}_u \sim T(x_u;\theta_T)} \left[\underbrace{g_S(\widehat{y}_u)}_{|S|\times 1} \cdot \underbrace{\frac{\partial \mathrm{CE}\left(\widehat{y}_u, T(x_u;\theta_T)\right)}{\partial \theta_T}}_{1\times|T|} \right] \quad (4)$$

Method

Derivation of the Teacher's Update Rule

$$\nabla_{\theta_T}\mathcal{L}_l = \eta_S \cdot \underbrace{\frac{\partial \mathrm{CE}\left(y_l, S(x_l;\theta'_S)\right)}{\partial \theta_S}}_{1\times|S|} \cdot \underbrace{\left(\frac{\partial \mathrm{CE}\left(\widehat{y}_u, S(x_u;\theta_S)\right)}{\partial \theta_S}\bigg|_{\theta_S=\theta_S}\right)^{\top}}_{|S|\times 1} \cdot \underbrace{\frac{\partial \mathrm{CE}\left(\widehat{y}_u, T(x_u;\theta_T)\right)}{\partial \theta_T}}_{1\times|T|}$$

$$= \eta_S \cdot \underbrace{\left(\left(\nabla_{\theta'_S}\mathrm{CE}\left(y_l, S(x_l;\theta'_S)\right)\right)^{\top} \cdot \nabla_{\theta_S}\mathrm{CE}\left(\widehat{y}_u, S(x_u;\theta_S)\right)\right)}_{\text{A scalar} := h} \cdot \nabla_{\theta_T}\mathrm{CE}\left(\widehat{y}_u, T(x_u;\theta_T)\right)$$

Similarity of Student gradients between labeled and unlabeled samples

→ **How to approximate this?** → **First Order Taylor Expansion**

$$f(x) - f(a) = (x-a)f'(a) \qquad \theta'_S = \theta_S - \eta_S \nabla_{\theta_S}\mathrm{CE}(\hat{y}_u, S(x_u;\theta_S))$$

$$\mathrm{CE}(y_l, S(x_l;\theta_S)) - \mathrm{CE}\left(y_l, S(x_l;\theta'_S)\right) = \eta_S \cdot \left(\nabla_{\theta'_S}\mathrm{CE}\left(y_l, S(x_l;\theta'_S)\right)\right)^{\top} \cdot \nabla_{\theta_S}\mathrm{CE}(\hat{y}_u, S(x_u;\theta_S))$$

A scalar, a weight

$$\nabla_{\theta_T}\mathcal{L}_l = \eta_S h \cdot \nabla_{\theta_T}\mathrm{CE}(\hat{y}_u, T(x_u;\theta_T))$$

Method

Algorithm 1 The Meta Pseudo Labels method, applied to a teacher trained with UDA [76].

Input: Labeled data x_l, y_l and unlabeled data x_u.
Initialize $\theta_T^{(0)}$ and $\theta_S^{(0)}$
for $t = 0$ **to** $N - 1$ **do**
Sample an unlabeled example x_u and a labeled example x_l, y_l
Sample a pseudo label $\widehat{y}_u \sim P(\cdot|x_u; \theta_T)$
Update the student using the pseudo label $\widehat{y}_u$:

$$\theta_S^{(t+1)} = \theta_S^{(t)} - \eta_S \, \nabla_{\theta_S} \mathrm{CE}(\widehat{y}_u, S(x_u; \theta_S))|_{\theta_S = \theta_S^{(t)}}$$

Compute the teacher's feedback coefficient as in Equation 12:

$$h = \eta_S \cdot \left(\left(\nabla_{\theta'_S} \mathrm{CE}\left(y_l, S(x_l; \theta_S^{(t+1)})\right) \right)^\top \cdot \nabla_{\theta_S} \mathrm{CE}\left(\widehat{y}_u, S(x_u; \theta_S^{(t)})\right) \right) \Rightarrow \mathrm{CE}(y_l, S(x_l; \theta_S)) - \mathrm{CE}(y_l, S(x_l; \theta'_S))$$

Compute the teacher's gradient from the student's feedback:

$$g_T^{(t)} = h \cdot \nabla_{\theta_T} \mathrm{CE}(\widehat{y}_u, T(x_u; \theta_T))|_{\theta_T = \theta_T^{(t)}}$$

Compute the teacher's gradient on labeled data:

$$g_{T,\text{supervised}}^{(t)} = \nabla_{\theta_T} \, \mathrm{CE}(y_l, T(x_l; \theta_T))|_{\theta_T = \theta_T^{(t)}}$$

Compute the teacher's gradient on the UDA loss with unlabeled data:

$$g_{T,\text{UDA}}^{(t)} = \nabla_{\theta_T} \mathrm{CE}\Big((T(x_u^w); \theta_T), T((x_u^s); \theta_T) \Big)|_{\theta_T = \theta_T^{(t)}}$$

Consistency Loss

Update the teacher:

$$\theta_T^{(t+1)} = \theta_T^{(t)} - \eta_T \cdot \left(g_T^{(t)} + g_{T,\text{supervised}}^{(t)} + g_{T,\text{UDA}}^{(t)} \right)$$

end
return $\theta_S^{(N)}$ ▷ *Only the student model is returned for predictions and evaluations*

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

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Meta-Semi: A Meta-learning Approach for Semi-supervised Learning

Yulin Wang, Jiayi Guo, Shiji Song, Gao Huang*
Department of Automation, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology (BNRist)
wang-yl19@mails.tsinghua.edu.cn, guojy821@gmail.com
{shijis, gaohuang}@tsinghua.edu.cn

Motivation

Insufficient usage of labeled data

Most of SSL algorithms model the labeled and unlabeled data in separate terms in loss function

$$\mathcal{L}_{all} = \mathcal{L}_l + \lambda \mathcal{L}_u$$

The unlabeled data receives no supervision.

Intuition: *If the network is trained with correctly pseudo-labeled unannotated samples, the final loss on labeled data should be minimized.*

Meta Reweighting Objective

Find the optimal **weights** for different pseudo-labeled samples to train a network, such that the final loss on labeled data is minimized.

Meta-Semi

$$\mathcal{L}_{meta} = \frac{1}{\sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j^*} \sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j^* L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}))$$

Use **weight** to find those most beneficial pseudo-labeled samples

Meta Pseudo Label

$$\mathbb{E}_{x_l, y_l}\left[\mathrm{CE}\left(y_l, S(x_l; \theta_S^{\mathrm{PL}})\right)\right] := \mathcal{L}_l\left(\theta_S^{\mathrm{PL}}\right)$$

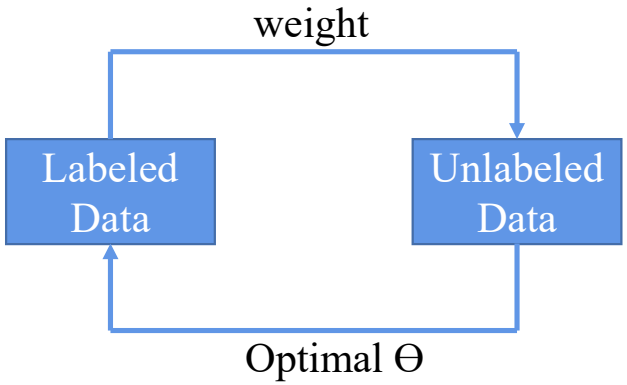
Use **feedback** to increase accuracy of pseudo labels

Method

Find Optimal Weight on Labeled Data

Labeled samples $\mathcal{X}=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}$, and unlabeled samples $\mathcal{U}=\{(\boldsymbol{u}_j,\hat{\boldsymbol{y}}_j)\}$

$$\boldsymbol{w}^* = \underset{w_j\in[0,1],j=1,\ldots,|\tilde{\mathcal{U}}|}{\arg\min} \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\boldsymbol{\theta}^*(\boldsymbol{w})))$$



Optimize Θ on Unlabeled Data

$$\boldsymbol{\theta}^*(\boldsymbol{w}) = \arg\min_{\boldsymbol{\theta}} \sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}))$$

Approximating the Meta Solution

At t^{th} step in the training process, consider estimating $\theta^*(w)$by performing M times of gradient descents starting from current values of network parameters θ^t :

$$\overline{\boldsymbol{\theta}}_M^t \approx \boldsymbol{\theta}^*(\boldsymbol{w}),\ \ \overline{\boldsymbol{\theta}}_0^t = \boldsymbol{\theta}^t,$$

$$\overline{\boldsymbol{\theta}}_{m+1}^t = \overline{\boldsymbol{\theta}}_m^t - \alpha^t \left[\frac{\partial \sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\overline{\boldsymbol{\theta}}_m^t))}{\partial \overline{\boldsymbol{\theta}}_m^t} \right], m = 0, 1, \ldots, M-1,$$

It's computationally intensive to do so!

$\overline{\boldsymbol{\theta}}_M^t$ is a reliable alternate of $\boldsymbol{\theta}^*(\boldsymbol{w})$ as long as M is sufficiently large. Then, estimate the gradient $\nabla_{\boldsymbol{w}} \sum_{i=1}^{|\mathcal{X}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\boldsymbol{\theta}^*(\boldsymbol{w})))$ to determine w.

Method

Approximating the Meta Solution

$$\boldsymbol{w}^* = \underset{w_j \in [0,1], j=1,\ldots,|\tilde{\mathcal{U}}|}{\arg\min} \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i | \boldsymbol{\theta}^*(\boldsymbol{w})))$$

First Order Taylor Expansion

$$\boldsymbol{w}^* \approx \underset{w_j \in [0,1], j=1,\ldots,|\tilde{\mathcal{U}}|}{\arg\min} \boldsymbol{w}^T \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i | \overline{\boldsymbol{\theta}}_M^t))}{\partial \boldsymbol{w}} \Big|_{\boldsymbol{w}=0} \right].$$

As the optimization objective is linear. The solution is:

$$w_j^* \approx w_j^t = \begin{cases} 1 & \frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i | \overline{\boldsymbol{\theta}}_M^t))}{\partial w_j} \Big|_{\boldsymbol{w}=0} \leq 0 \\ 0 & \frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i | \overline{\boldsymbol{\theta}}_M^t))}{\partial w_j} \Big|_{\boldsymbol{w}=0} > 0 \end{cases},$$

But how to understand this? What's the complete form of gradient on w?
Why does its sign tell us whether a pseudo labeled sample is important or not?

Method

Derivation of gradient on w

$$
\begin{aligned}
&\left.\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_M^t))}{\partial w_j}\right|_{\boldsymbol{w}=0} \\
&= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_M^t))}{\partial \overline{\boldsymbol{\theta}}_M^t}\right]^T \left[\frac{\partial \overline{\boldsymbol{\theta}}_M^t}{\partial w_j}\right]\Bigg|_{\boldsymbol{w}=0} \\
&= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_M^t))}{\partial \overline{\boldsymbol{\theta}}_M^t}\right]^T \left[\frac{\partial (\overline{\boldsymbol{\theta}}_{M-1}^t - \alpha^t \nabla_{\overline{\boldsymbol{\theta}}_{M-1}^t} \sum_{k=1}^{|\tilde{\mathcal{U}}|} w_k L(\hat{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k|\overline{\boldsymbol{\theta}}_{M-1}^t)))}{\partial w_j}\right]\Bigg|_{\boldsymbol{w}=0} \\
&= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_M^t))}{\partial \overline{\boldsymbol{\theta}}_M^t}\right]^T \left[\frac{\partial \overline{\boldsymbol{\theta}}_{M-1}^t}{\partial w_j} - \alpha^t \sum_{k=1}^{|\tilde{\mathcal{U}}|} \left[\frac{\partial w_k \nabla_{\overline{\boldsymbol{\theta}}_{M-1}^t} L(\hat{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k|\overline{\boldsymbol{\theta}}_{M-1}^t))}{\partial w_k} \boxed{\frac{\partial w_k}{\partial w_j}}\right.\right. \\
&\qquad \left.\left. + \frac{\partial w_k \nabla_{\overline{\boldsymbol{\theta}}_{M-1}^t} L(\hat{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k|\overline{\boldsymbol{\theta}}_{M-1}^t))}{\partial \nabla_{\overline{\boldsymbol{\theta}}_{M-1}^t} L(\hat{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k|\overline{\boldsymbol{\theta}}_{M-1}^t))} \frac{\partial \nabla_{\overline{\boldsymbol{\theta}}_{M-1}^t} L(\hat{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k|\overline{\boldsymbol{\theta}}_{M-1}^t))}{\partial w_j}\right]\right]\Bigg|_{\boldsymbol{w}=0}
\end{aligned}
$$

Non-zero only when k=j

Method

Derivation of gradient on w

$$= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_M^t))}{\partial \overline{\boldsymbol{\theta}}_M^t}\right]^T \left[\frac{\partial \overline{\boldsymbol{\theta}}_{M-1}^t}{\partial w_j} - \alpha^t \left[\frac{\partial L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\overline{\boldsymbol{\theta}}_{M-1}^t))}{\partial \overline{\boldsymbol{\theta}}_{M-1}^t} + \sum_{k=1}^{|\tilde{\mathcal{U}}|} w_k \frac{\partial \nabla_{\overline{\boldsymbol{\theta}}_{M-1}^t} L(\hat{\boldsymbol{y}}_k, p(\tilde{\boldsymbol{u}}_k|\overline{\boldsymbol{\theta}}_{M-1}^t))}{\partial w_j}\right]\right]\Bigg|_{\boldsymbol{w}=0}$$

$$= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]^T \left[\frac{\partial \overline{\boldsymbol{\theta}}_{M-1}^t}{\partial w_j}\Bigg|_{\boldsymbol{w}=0} - \alpha^t \frac{\partial L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]$$

$$\overline{\boldsymbol{\theta}}_M^t = \overline{\boldsymbol{\theta}}_{M-1}^t = \ldots = \overline{\boldsymbol{\theta}}_0^t \text{ when } \boldsymbol{w} = 0$$

$$= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]^T \left[\frac{\partial \overline{\boldsymbol{\theta}}_{M-2}^t}{\partial w_j}\Bigg|_{\boldsymbol{w}=0} - 2\alpha^t \frac{\partial L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]$$

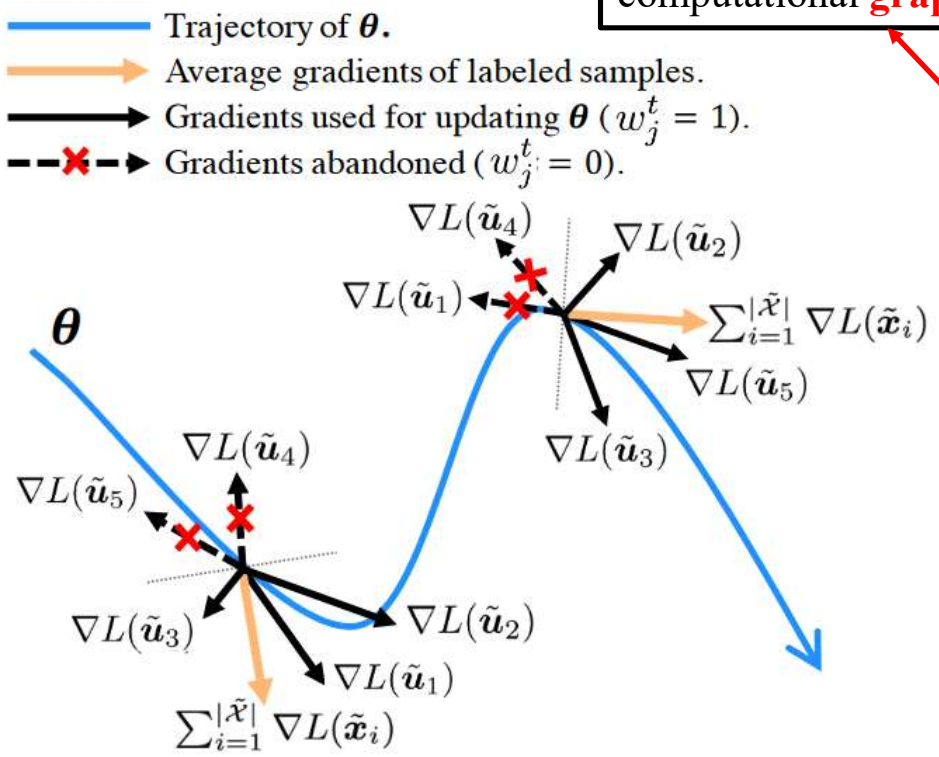
$$= \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]^T \left[\frac{\partial \overline{\boldsymbol{\theta}}_0^t}{\partial w_j}\Bigg|_{\boldsymbol{w}=0} - M\alpha^t \frac{\partial L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]$$

$$= -M\alpha^t \left[\frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]^T \left[\frac{\partial L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}\right]$$

Similarity of gradients between labeled and unlabeled samples

Method

Interpretation of meta gradients



Meta-Semi trains deep networks using pseudo-labeled samples whose gradient directions are similar to labeled samples.

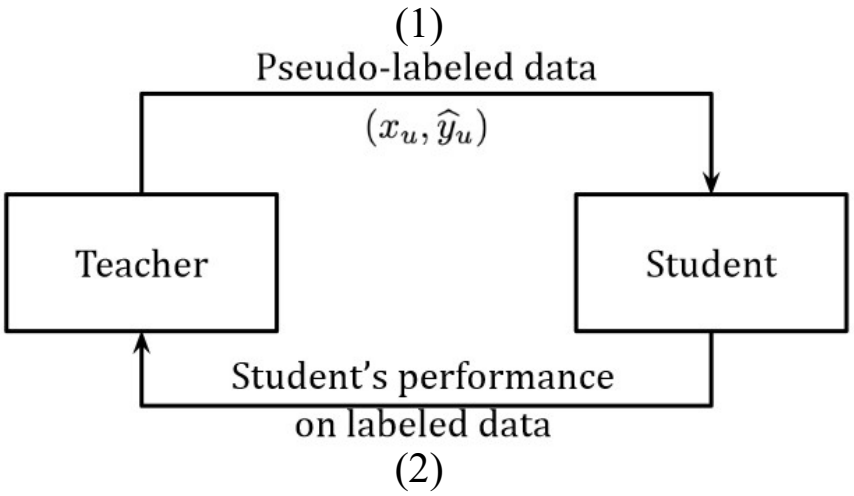
Such a meta updating step does **not change** the values of **parameters**, but **construct** a differentiable computational **graph**.

Algorithm 1 The Meta-Semi Algorithm.

Initialize: $\boldsymbol{\theta}^0$
for $t = 1$ **to** T **do**
  Randomly sample $\mathcal{X}, \mathcal{U}$
  Generate $\tilde{\mathcal{X}}, \tilde{\mathcal{U}}$
  Compute $p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t)$, $\tilde{\boldsymbol{u}}_j \in \tilde{\mathcal{U}}$
  $\boldsymbol{w} \leftarrow 0, \overline{\boldsymbol{\theta}}_0^t \leftarrow \boldsymbol{\theta}^t$
  $\nabla_{\overline{\boldsymbol{\theta}}_0^t} \leftarrow \frac{\partial \sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t))}{\partial \boldsymbol{\theta}^t}$
  $\overline{\boldsymbol{\theta}}_1^t \leftarrow \overline{\boldsymbol{\theta}}_0^t - \alpha^t \nabla_{\overline{\boldsymbol{\theta}}_0^t}$
  Compute $p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_1^t)$, $\tilde{\boldsymbol{x}}_i \in \tilde{\mathcal{X}}$
  Meta Gradient: $\nabla_{\boldsymbol{w}}^t \leftarrow \frac{\partial \sum_{i=1}^{|\tilde{\mathcal{X}}|} L(\tilde{\boldsymbol{y}}_i, p(\tilde{\boldsymbol{x}}_i|\overline{\boldsymbol{\theta}}_1^t))}{\partial \boldsymbol{w}}$
  $\boldsymbol{w}^t \leftarrow sign(max(-\nabla_{\boldsymbol{w}}^t, 0))$ (Eq. (9))
  $\mathcal{L}_{meta} \leftarrow \frac{1}{\sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j^t} \sum_{j=1}^{|\tilde{\mathcal{U}}|} w_j^t L(\hat{\boldsymbol{y}}_j, p(\tilde{\boldsymbol{u}}_j|\boldsymbol{\theta}^t))$
  $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^t - \alpha^t \frac{\partial \mathcal{L}_{meta}}{\partial \boldsymbol{\theta}^t}$
end for

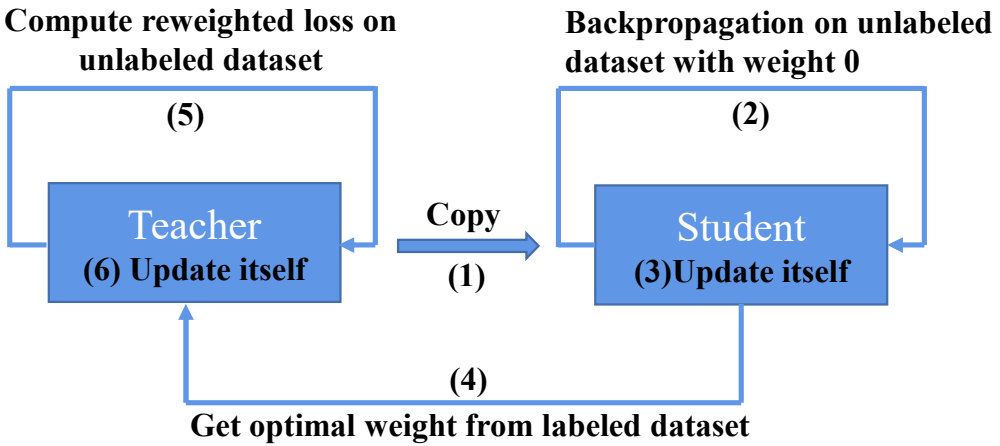
Comparison

Meta Pseudo Label



Calculate feedback to generate more accurate pseudo labels in every iteration.

Meta Semi



Find optimal weight from labeled data to reweight pseudo labeled data.

Both of them use inner product of gradients on labeled and unlabeled data. In MPL, we use it as a coefficient to build **loss term**. While in MS, we use it as a **weight** of loss function.

Experiment (Meta Pseudo Label)

	Method	CIFAR-10-4K (mean ± std)	SVHN-1K (mean ± std)	ImageNet-10% Top-1	ImageNet-10% Top-5
Label Propagation Methods	Temporal Ensemble [35]	83.63 ± 0.63	92.81 ± 0.27	–	–
	Mean Teacher [64]	84.13 ± 0.28	94.35 ± 0.47	–	–
	VAT + EntMin [44]	86.87 ± 0.39	94.65 ± 0.19	–	83.39
	LGA + VAT [30]	87.94 ± 0.19	93.42 ± 0.36	–	–
	ICT [71]	92.71 ± 0.02	96.11 ± 0.04	–	–
	MixMatch [5]	93.76 ± 0.06	96.73 ± 0.31	–	–
	ReMixMatch [4]	94.86 ± 0.04	97.17 ± 0.30	–	–
	EnAET [72]	94.65	97.08	–	–
	FixMatch [58]	95.74 ± 0.05	97.72 ± 0.38	71.5	89.1
	UDA* [76]	94.53 ± 0.18	97.11 ± 0.17	68.07	88.19
Self-Supervised Methods	SimCLR [8, 9]	–	–	71.7	90.4
	MOCOv2 [10]	–	–	71.1	–
	PCL [38]	–	–	–	85.6
	PIRL [43]	–	–	–	84.9
	BYOL [21]	–	–	68.8	89.0
	Meta Pseudo Labels	**96.11 ± 0.07**	**98.01 ± 0.07**	**73.89**	**91.38**
	Supervised Learning with full dataset*	94.92 ± 0.17	97.41 ± 0.16	76.89	93.27

Table 2: Image classification accuracy on CIFAR-10-4K, SVHN-1K, and ImageNet-10%. Higher is better. For CIFAR-10-4K and SVHN-1K, we report mean ± std over 10 runs, while for ImageNet-10%, we report Top-1/Top-5 accuracy of a single run. For fair comparison, we only include results that share the same model architecture: WideResNet-28-2 for CIFAR-10-4K and SVHN-1K, and ResNet-50 for ImageNet-10%. * indicates our implementation which uses the same experimental protocols. Except for UDA, results in the first two blocks are from representative important papers, and hence do not share the same controlled environment with ours.

Experiment (Meta Pseudo Label)

Method	Unlabeled Images	Accuracy (top-1/top-5)
Supervised [24]	None	76.9/93.3
AutoAugment [12]	None	77.6/93.8
DropBlock [18]	None	78.4/94.2
FixRes [68]	None	79.1/94.6
FixRes+CutMix [83]	None	79.8/94.9
NoisyStudent [77]	JFT	78.9/94.3
UDA [76]	JFT	79.0/94.5
Billion-scale SSL [68, 79]	YFCC	82.5/**96.6**
Meta Pseudo Labels	JFT	**83.2**/96.5

Table 3: Top-1 and Top-5 accuracy of Meta Pseudo Labels and other representative supervised and semi-supervised methods on ImageNet with ResNet-50.

Supervised learning methods with data augmentation or regularization methods

Semi-supervised learning methods that leverage the labeled training images from ImageNet and unlabeled images elsewhere.

Benchmark Meta Pseudo Labels on the entire ImageNet dataset plus unlabeled images from the JFT dataset to verify MPL works well on ResNet-50.

Experiment (Meta Pseudo Label)

Method	# Params	Extra Data	ImageNet		ImageNet-ReaL [6]
			Top-1	Top-5	Precision@1
ResNet-50 [24]	26M	–	76.0	93.0	82.94
ResNet-152 [24]	60M	–	77.8	93.8	84.79
DenseNet-264 [28]	34M	–	77.9	93.9	–
Inception-v3 [62]	24M	–	78.8	94.4	83.58
Xception [11]	23M	–	79.0	94.5	–
Inception-v4 [61]	48M	–	80.0	95.0	–
Inception-resnet-v2 [61]	56M	–	80.1	95.1	–
ResNeXt-101 [78]	84M	–	80.9	95.6	85.18
PolyNet [87]	92M	–	81.3	95.8	–
SENet [27]	146M	–	82.7	96.2	–
NASNet-A [90]	89M	–	82.7	96.2	82.56
AmoebaNet-A [52]	87M	–	82.8	96.1	–
PNASNet [39]	86M	–	82.9	96.2	–
AmoebaNet-C + AutoAugment [12]	155M	–	83.5	96.5	–
GPipe [29]	557M	–	84.3	97.0	–
EfficientNet-B7 [63]	66M	–	85.0	97.2	–
EfficientNet-B7 + FixRes [70]	66M	–	85.3	97.4	–
EfficientNet-L2 [63]	480M	–	85.5	97.5	–
ResNet-50 Billion-scale SSL [79]	26M	3.5B labeled Instagram	81.2	96.0	–
ResNeXt-101 Billion-scale SSL [79]	193M	3.5B labeled Instagram	84.8	–	–
ResNeXt-101 WSL [42]	829M	3.5B labeled Instagram	85.4	97.6	88.19
FixRes ResNeXt-101 WSL [69]	829M	3.5B labeled Instagram	86.4	98.0	89.73
Big Transfer (BiT-L) [33]	928M	300M labeled JFT	87.5	98.5	90.54
Noisy Student (EfficientNet-L2) [77]	480M	300M unlabeled JFT	88.4	98.7	90.55
Noisy Student + FixRes [70]	480M	300M unlabeled JFT	88.5	98.7	–
Vision Transformer (ViT-H) [14]	632M	300M labeled JFT	88.55	–	90.72
EfficientNet-L2-NoisyStudent + SAM [16]	480M	300M unlabeled JFT	88.6	98.6	–
Meta Pseudo Labels (EfficientNet-B6-Wide)	390M	300M unlabeled JFT	90.0	98.7	**91.12**
Meta Pseudo Labels (EfficientNet-L2)	480M	300M unlabeled JFT	**90.2**	**98.8**	91.02

Table 4: Top-1 and Top-5 accuracy of Meta Pseudo Labels and previous state-of-the-art methods on ImageNet. With EfficientNet-L2 and EfficientNet-B6-Wide, Meta Pseudo Labels achieves an improvement of 1.6% on top of the state-of-the-art [16], despite the fact that the latter uses 300 million *labeled* training examples from JFT.

Experiment (Meta Pseudo Label)

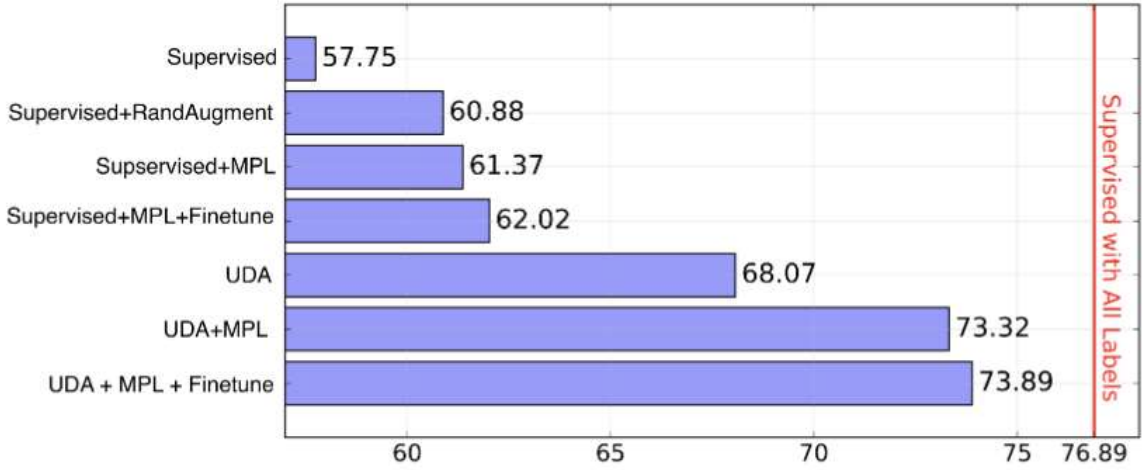


Figure 3: Breakdown of the gains of different components in Meta Pseudo Labels (MPL). The gain of Meta Pseudo Labels over UDA, albeit smaller than the gain of UDA over RandAugment, is significant as UDA is already very strong.

Experiment (Meta Pseudo Label)

MPL as A Regularization Strategy

	CIFAR-10-4K	SVHN-1K
Supervised	82.14 ± 0.25	88.17 ± 0.47
Label Smoothing	82.21 ± 0.18	89.39 ± 0.25
Meta Pseudo Labels	$\mathbf{83.71 \pm 0.21}$	$\mathbf{91.89 \pm 0.14}$

Table 9: Meta Pseudo Labels can be used as a regularization method for supervised learning.

Meta Pseudo Labels can be seen as an **adaptive** form of Label Smoothing: the teacher generates soft labels on labeled data for the student, just like the way **Label Smoothing** smooths the hard labels to regularize the model.

Experiment (Meta Pseudo Label)

MPL Is a Mechanism to Addresses the Confirmation Bias of Pseudo Labels

MPL converges much slower

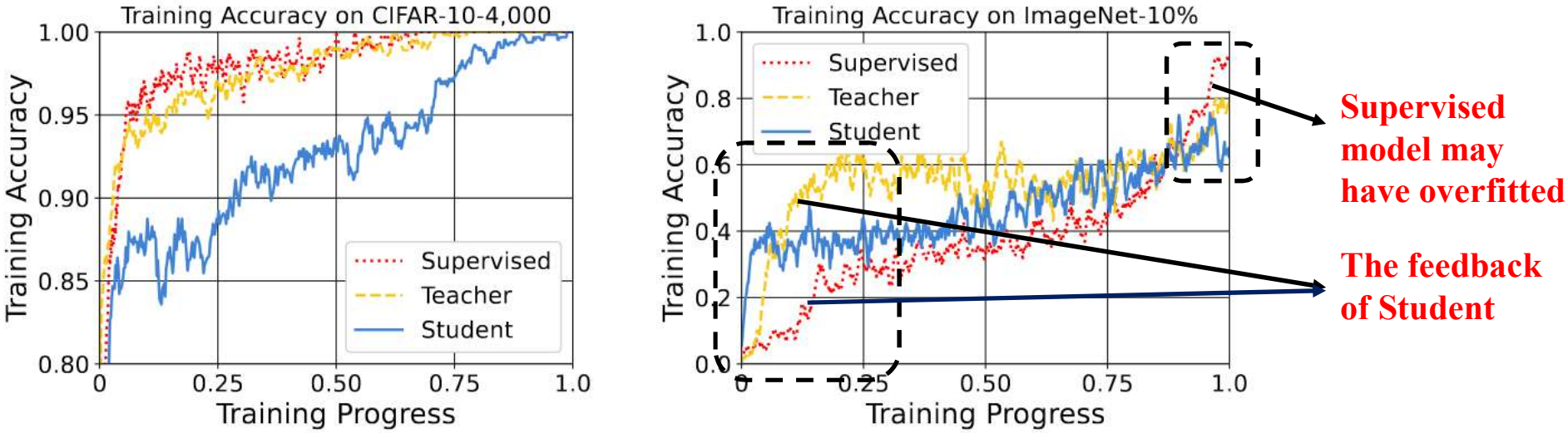


Figure 4: Training accuracy of Meta Pseudo Labels and of supervised learning on CIFAR-10-4,000 and ImageNet-10%. Both the teacher and the student in Meta Pseudo Labels have lower training accuracy, effectively avoiding overfitting.

Experiment (Meta Pseudo Label)

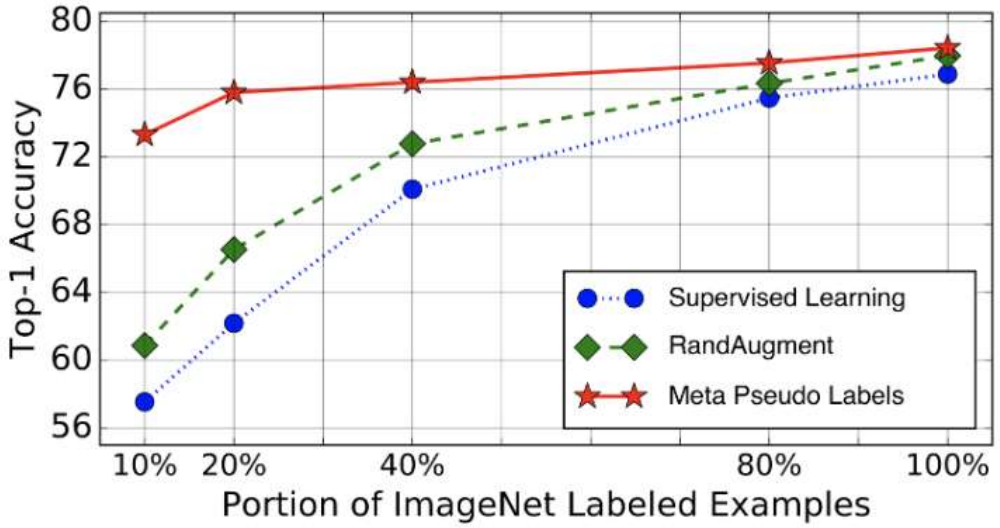


Figure 5: Performance of Supervised Learning, RandAugment, and Meta Pseudo Labels at different amounts of labeled examples.

Experiment (Meta-Semi)

Table 1: Performance of *Meta-Semi* and state-of-the-art SSL algorithms on CIFAR with varying amount of labeled data. We report the average test errors and the standard deviations of 5 trials. † refers to the experiments using the WRN-28 network, while all others use the CNN-13 network. In each setting, the best two results with CNN-13 and the best result with WRN-28 are **bold-faced**.

Dataset	CIFAR-10			CIFAR-100	
Number of Labeled Samples	1000	2000	4000	4000	10000
Supervised	39.95 ± 0.75%	27.67 ± 0.12%	20.42 ± 0.21%	58.31 ± 0.89%	44.56 ± 0.30%
Supervised + MixUp [43]	31.83 ± 0.65%	24.22 ± 0.15%	17.37 ± 0.35%	54.87 ± 0.07%	40.97 ± 0.47%
Π-model [22]	28.74 ± 0.48%	17.57 ± 0.44%	12.36 ± 0.17%	55.39 ± 0.55%	38.06 ± 0.37%
Temp-ensemble [22]	25.15 ± 1.46%	15.78 ± 0.44%	11.90 ± 0.25%	-	38.65 ± 0.51%
Mean Teacher [40]	18.27 ± 0.53%	13.45 ± 0.30%	10.73 ± 0.14%	45.36 ± 0.49%	35.96 ± 0.77%
VAT [28]	18.12 ± 0.82%	13.93 ± 0.33%	11.10 ± 0.24%	-	-
SNTG [27]	18.41 ± 0.52%	13.64 ± 0.32%	10.93 ± 0.14%	-	37.97 ± 0.29%
Learning to Reweight [36]	11.74 ± 0.12%	-	9.44 ± 0.17%	46.62 ± 0.29%	37.31 ± 0.47%
MT + Fast SWA [2]	15.58%	11.02%	9.05%	-	33.62 ± 0.54%
ICT [42]	12.44 ± 0.57%	8.69 ± 0.15%	7.18 ± 0.24%	40.07 ± 0.38%	32.24 ± 0.16%
Meta-Semi	**10.27 ± 0.66%**	**8.42 ± 0.30%**	**7.05 ± 0.27%**	**37.61 ± 0.56%**	**30.51 ± 0.32%**
Meta-Semi + ICT	**9.29 ± 0.62%**	**7.05 ± 0.12%**	**6.42 ± 0.18%**	**37.12 ± 0.59%**	**29.68 ± 0.05%**
Mean Teacher † [40]	17.32 ± 4.00%	12.17 ± 0.22%	10.36 ± 0.25%	-	-
MixMatch † [7]	7.75 ± 0.32%	7.03 ± 0.15%	6.24 ± 0.06%	-	30.84 ± 0.29%
Meta-Semi †	**7.34 ± 0.22%**	**6.58 ± 0.07%**	**6.10 ± 0.10%**	-	**29.69 ± 0.18%**

Table 2: Test errors on STL-10. We adopt the same experimental setups as [7]. The best result is **bold-faced**.

Method	STL-10, 1000 labels
SWWAE [44]	25.70%
CC-GAN [41]	22.20%
MixMatch [7]	10.18 ± 1.46%
Meta-Semi	**8.03 ± 0.24%**

Experiment (Meta-Semi)

Table 3: Test errors on SVHN with varying amount of labeled data. We report the average results and the standard deviations of 5 independent experiments. All results are based on CNN-13. The best results are **bold-faced**.

Methods	SVHN 500 labels	SVHN 1000 labels
VAT [28]	-	5.42%
Π-model [22]	6.65 ± 0.53%	4.82 ± 0.17%
Temp-ensemble [22]	5.12 ± 0.13%	4.42 ± 0.16%
Mean Teacher [40]	4.18 ± 0.27%	3.95 ± 0.19%
ICT [42]	4.23 ± 0.15%	3.89 ± 0.04%
SNTG [27]	3.99 ± 0.24%	3.86 ± 0.27%
Meta-Semi	4.12 ± 0.21%	3.92 ± 0.11%
Meta-Semi + ICT	**3.98 ± 0.09%**	**3.77 ± 0.05%**

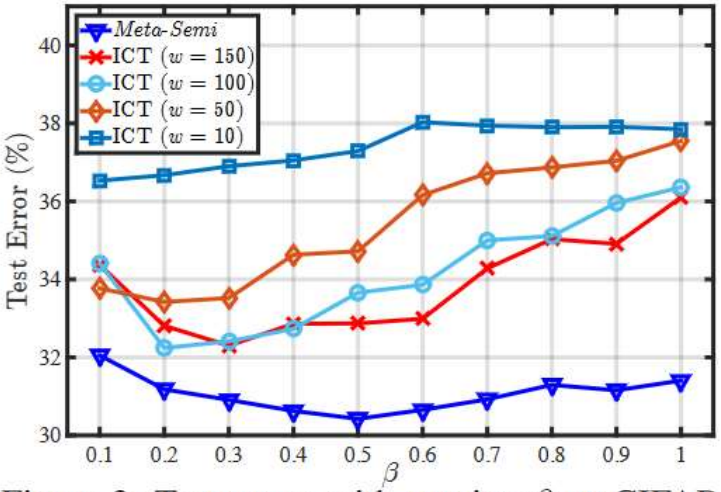


Figure 3: Test errors with varying β on CIFAR-100 using 10,000 labels. The CNN-13 network is used. We also report the results of ICT [42] when the unsupervised consistency coefficient w changes among the recommended range.

Table 4: Performance of *Meta-Semi* v.s. baselines with fixed amount of training time. We report the mean test errors of both networks on CIFAR-100 with 10,000 labels. The best results are **bold-faced**

(a) CNN-13

Training Time	5.0h	7.5h	10.0h	12.6h
ICT [42]	33.43%	32.84%	32.61%	32.24%
Meta-Semi	**32.73%**	**31.81%**	**31.06%**	**30.84%**

(b) WRN-28

Training Time	13.7h	18.3h	22.8h	29.2h
MixMatch [7]	32.94%	31.91%	31.26%	30.84%
Meta-Semi	**31.74%**	**30.85%**	**30.50%**	**30.13%**

Experiment (Meta-Semi)

Ablation Study

Table 5: Ablation study results. We report the test errors on CIFAR-100 with 4,000 and 10,000 labels. The CNN-13 network is used.

Ablation	CIFAR-100 4000 labels	CIFAR-100 10000 labels
Without parameter EMA	$47.68 \pm 0.27\%$	$37.15 \pm 1.02\%$
One-hot pseudo labels	$41.52 \pm 0.51\%$	$32.78 \pm 0.41\%$
MixUp on unlabeled data only	$37.69 \pm 0.50\%$	$30.56 \pm 0.39\%$
MixUp on labeled data only	$45.90 \pm 0.15\%$	$36.11 \pm 0.21\%$
Without MixUp	$46.71 \pm 0.05\%$	$35.98 \pm 0.69\%$
Reweighting with the constant 1	$40.26 \pm 0.64\%$	$32.17 \pm 0.14\%$
Reweighting with -1 and 1	$45.41 \pm 0.38\%$	$36.39 \pm 0.44\%$
Meta-Semi	$37.61 \pm 0.56\%$	$30.51 \pm 0.32\%$
Meta-Semi + ICT	$37.12 \pm 0.59\%$	$29.68 \pm 0.05\%$

Thanks