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

NeurIPS 2020

FixMatch-Experiment

Wide ResNet-28-2 for CIFAR-10 and SVHN

WRN-28-8 for CIFAR-100 WRN-37-2 for STL-10

$\lambda_u = 1,\ \eta = 0.03,\ \beta = 0.9,\ \tau = 0.95,\ \mu = 7,\ B = 64,\ K = 2^{20}$

	CIFAR-10			CIFAR-100			SVHN			STL-10
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 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	26.23±0.82
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	27.99±0.83
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	21.43±2.39
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	10.41±0.61
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	7.66±0.56
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	**5.23**±0.45
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	7.98±1.50
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	**5.17**±0.63

Table 2: Error rates for CIFAR-10, CIFAR-100, SVHN and STL-10 on 5 different folds. FixMatch (RA) uses RandAugment [11] and FixMatch (CTA) uses CTAugment [3] for strong-augmentation. All baseline models (Π-Model [43], Pseudo-Labeling [25], Mean Teacher [51], MixMatch [4], UDA [54], and ReMixMatch [3]) are tested using the same codebase.

Figure 2: FixMatch reaches 78% CIFAR-10 accuracy using only above 10 labeled images.

FixMatch-Experiment

- Adversarial Robustness

Examples that well represent the dataset should be more adversarially robust.

- Holdout Retraining

A model should treat a well-represented example the same regardless of whether or not it is used in the training process.

- Ensemble Agreement $\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\text{JS-Divergence}(f_{\theta_i}(x), f_{\theta_j}(x))$

- Model Confidence $\frac{1}{N}\sum_{i=1}^{N}\max f_{\theta_i}(x)$

- Privacy-preserving Training

MixMatch: A Holistic Approach to Semi-Supervised Learning

David Berthelot
Google Research
dberth@google.com

Nicholas Carlini
Google Research
ncarlini@google.com

Ian Goodfellow
Work done at Google
ian-academic@mailfence.com

Avital Oliver
Google Research
avitalo@google.com

Nicolas Papernot
Google Research
papernot@google.com

Colin Raffel
Google Research
craffel@google.com

NeurIPS 2019

Related Work for SSL

□ Consistency Regularization

A classifier should output the same class distribution for an unlabeled example even after it has been augmented.

$$\|\mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) - \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta)\|_2^2$$

□ Entropy Minimization

Require that the classifier output low-entropy predictions on unlabeled data.

□ Traditional Regularization

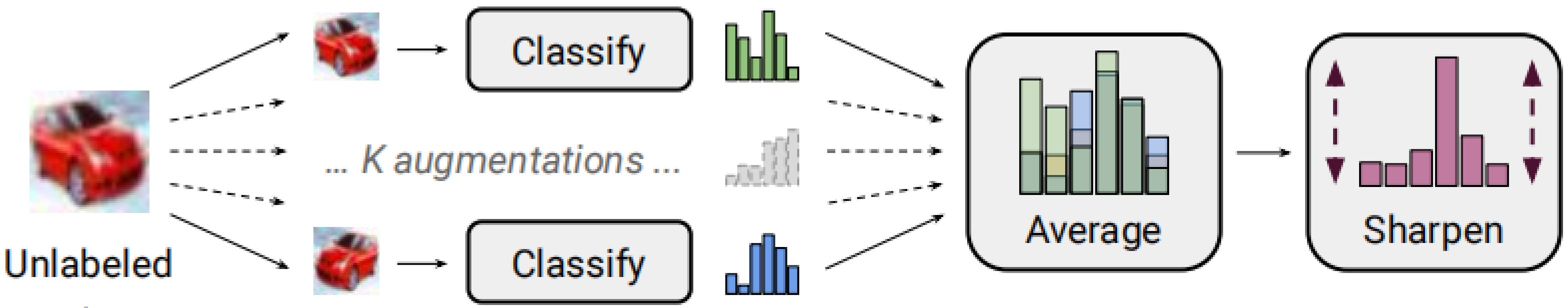
1. Penalize the L2 norm of the model parameters.
2. MixUp

MixMatch

☐ Data Augmentation

$$\hat{x}_b = \text{Augment}(x_b)$$

$$\hat{u}_{b,k} = \text{Augment}(u_b), k \in (1, \ldots, K)$$



☐ Label Guessing

$$\bar{q}_b = \frac{1}{K} \sum_{k=1}^{K} \mathrm{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$$

$$\text{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \bigg/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$

☐ MixUp

Mix both labeled examples and unlabeled examples with label guesses.

$$\lambda \sim \text{Beta}(\alpha, \alpha)$$

$$\lambda' = \max(\lambda, 1 - \lambda)$$

$$x' = \lambda' x_1 + (1 - \lambda') x_2$$

$$p' = \lambda' p_1 + (1 - \lambda') p_2$$

MixMatch

Algorithm 1 MixMatch takes a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ (resp. $\mathcal{U}'$) of processed labeled examples (resp. unlabeled with guessed labels).

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \ldots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \ldots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.

for $b = 1$ **to** B **do**

    $\hat{x}_b = \mathrm{Augment}(x_b)$ *// Apply data augmentation to x_b*

    for $k = 1$ **to** K **do**

        $\hat{u}_{b,k} = \mathrm{Augment}(u_b)$ *// Apply k^{th} round of data augmentation to u_b*

    end for

    $\bar{q}_b = \frac{1}{K} \sum_k \mathrm{p_{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of u_b*

    $q_b = \mathrm{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*

end for

$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \ldots, B)\big)$ *// Augmented labeled examples and their labels*

$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \ldots, B), k \in (1, \ldots, K)\big)$ *// Augmented unlabeled examples, guessed labels*

$\mathcal{W} = \mathrm{Shuffle}\big(\mathrm{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*

$\mathcal{X}' = \big(\mathrm{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \ldots, |\hat{\mathcal{X}}|)\big)$ *// Apply MixUp to labeled data and entries from $\mathcal{W}$*

$\mathcal{U}' = \big(\mathrm{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \ldots, |\hat{\mathcal{U}}|)\big)$ *// Apply MixUp to unlabeled data and the rest of $\mathcal{W}$*

return $\mathcal{X}', \mathcal{U}'$

MixMatch

Loss Function

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \mathrm{H}(p, \mathrm{p}_{\text{model}}(y \mid x; \theta))$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \mathrm{p}_{\text{model}}(y \mid u; \theta)\|_2^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$

Hyperparameters

$T = 0.5$ $\quad$ $K = 2$ $\quad$ $\alpha = 0.75$ $\quad$ $\lambda_{\mathcal{U}} = 100$

ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, Colin Raffel
Google Research
{dberth,ncarlini,cubuk,kurakin,zhanghan,craffel}@google.com

Kihyuk Sohn
Google Cloud AI
kihyuks@google.com

ArXiv 2020

ReMixMatch-Distribution Alignment

☐ Input-Output Mutual Information

A good classifier's prediction should depend as much as possible on the input.

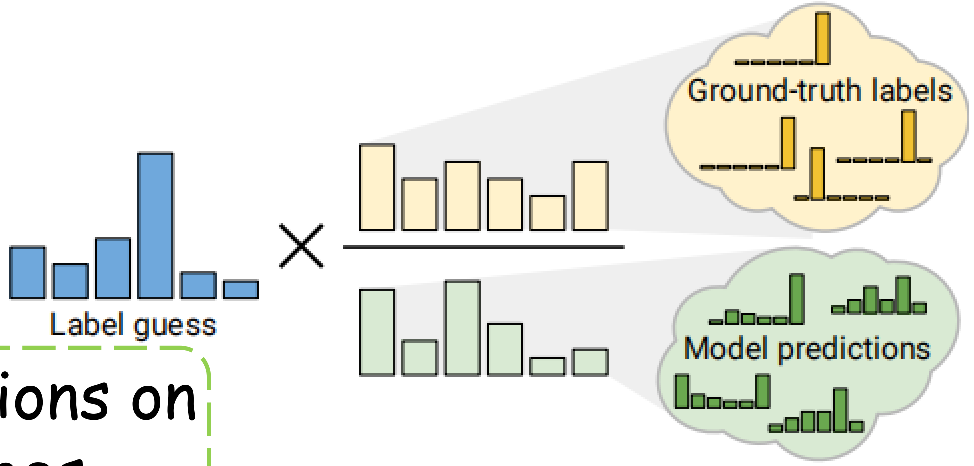
$$\begin{aligned}\mathcal{I}(y;x) &= \iint p(y,x)\log\frac{p(y,x)}{p(y)p(x)}\,\mathrm{d}y\,\mathrm{d}x \\ &= \mathcal{H}(\mathbb{E}_x[p_{\text{model}}(y|x;\theta)]) - \mathbb{E}_x[\mathcal{H}(p_{\text{model}}(y|x;\theta))]\end{aligned}$$

☐ Distribution Alignment

Enforces the aggregate of predictions on unlabeled data matches the distribution of the provided labeled data.

$$\tilde{q} = \text{Normalize}(q \times p(y)/\tilde{p}(y))$$

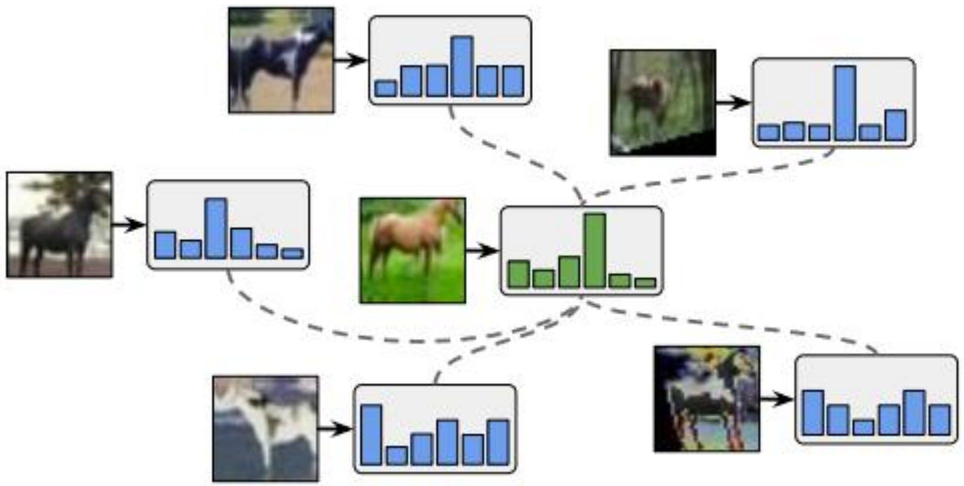
The moving average of the model's predictions on unlabeled examples over the last 128 batches.



ReMixMatch- Improved Consistency Regularization

- Augmentation Anchoring

Enforces the aggregate of predictions on unlabeled data matches the distribution of the provided labeled data.



ReMixMatch

Algorithm 1 ReMixMatch algorithm for producing a collection of processed labeled examples and processed unlabeled examples with label guesses (cf. Berthelot et al. (2019) Algorithm 1.)

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \{(x_b, p_b) : b \in (1, \ldots, B)\}$, batch of unlabeled examples $\mathcal{U} = \{u_b : b \in (1, \ldots, B)\}$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.

for $b = 1$ **to** B **do**

- $\hat{x}_b = \text{StrongAugment}(x_b)$ *// Apply strong data augmentation to x_b*
- $\hat{u}_{b,k} = \text{StrongAugment}(u_b); k \in \{1, \ldots, K\}$ *// Apply strong data augmentation K times to u_b*
- $\tilde{u}_b = \text{WeakAugment}(u_b)$ *// Apply weak data augmentation to u_b*
- $q_b = p_{\text{model}}(y \mid \tilde{u}_b; \theta)$ *// Compute prediction for weak augmentation of u_b*
- $q_b = \text{Normalize}(q_b \times p(y)/\tilde{p}(y))$ *// Apply distribution alignment*
- $q_b = \text{Normalize}(q_b^{1/T})$ *// Apply temperature sharpening to label guess*

end for

$\hat{\mathcal{X}} = ((\hat{x}_b, p_b); b \in (1, \ldots, B))$ *// Augmented labeled examples and their labels*

$\hat{\mathcal{U}}_1 = ((\hat{u}_{b,1}, q_b); b \in (1, \ldots, B))$ *// First strongly augmented unlabeled example and guessed label*

$\hat{\mathcal{U}} = ((\hat{u}_{b,k}, q_b); b \in (1, \ldots, B), k \in (1, \ldots, K))$ *// All strongly augmented unlabeled examples*

$\hat{\mathcal{U}} = \hat{\mathcal{U}} \cup ((\tilde{u}_b, q_b); b \in (1, \ldots, B))$ *// Add weakly augmented unlabeled examples*

$\mathcal{W} = \text{Shuffle}(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}}))$ *// Combine and shuffle labeled and unlabeled data*

$\mathcal{X}' = (\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \ldots, |\hat{\mathcal{X}}|))$ *// Apply MixUp to labeled data and entries from $\mathcal{W}$*

$\mathcal{U}' = (\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \ldots, |\hat{\mathcal{U}}|))$ *// Apply MixUp to unlabeled data and the rest of $\mathcal{W}$*

return $\mathcal{X}', \mathcal{U}', \hat{\mathcal{U}}_1$

ReMixMatch

Loss Function

$$\sum_{x,p\in\mathcal{X}'} \mathrm{H}(p, p_{\text{model}}(y|x;\theta)) + \lambda_{\mathcal{U}} \sum_{u,q\in\mathcal{U}'} \mathrm{H}(q, p_{\text{model}}(y|u;\theta))$$

$$+\lambda_{\hat{\mathcal{U}}_1} \sum_{u,q\in\hat{\mathcal{U}}_1} \mathrm{H}(q, p_{\text{model}}(y|u;\theta)) + \lambda_r \sum_{u\in\hat{\mathcal{U}}_1} \mathrm{H}(r, p_{\text{model}}(r|\,\mathrm{Rotate}(u,r);\theta))$$

$$r \sim \{0, 90, 180, 270\}$$

Hyperparameters

$\lambda_r = \lambda_{\hat{\mathcal{U}}_1} = 0.5$ $\quad T = 0.5$ $\quad \alpha = 0.75$ $\quad \lambda_{\mathcal{U}} = 1.5$

ReMixMatch-Experiment

Model: Wide ResNet-28-2

	CIFAR-10			SVHN		
Method	250 labels	1000 labels	4000 labels	250 labels	1000 labels	4000 labels
VAT	36.03±2.82	18.64±0.40	11.05±0.31	8.41±1.01	5.98±0.21	4.20±0.15
Mean Teacher	47.32±4.71	17.32±4.00	10.36±0.25	6.45±2.43	3.75±0.10	3.39±0.11
MixMatch	11.08±0.87	7.75±0.32	6.24±0.06	3.78±0.26	3.27±0.31	2.89±0.06
ReMixMatch	6.27±0.34	5.73±0.16	5.14±0.04	3.10±0.50	2.83±0.30	2.42±0.09
UDA, reported*	8.76±0.90	5.87±0.13	5.29±0.25	2.76±0.17	2.55±0.09	2.47±0.15

Table 1: Results on CIFAR-10 and SVHN. * For UDA, due to adaptation difficulties, we report the results from Xie et al. (2019) which are not comparable to our results due to a different network implementation, training procedure, etc. For VAT, Mean Teacher, and MixMatch, we report results using our reimplementation, which makes them directly comparable to ReMixMatch's scores.

ReMixMatch-Experiment

Method	Error Rate
SWWAE	25.70
CC-GAN	22.20
MixMatch	10.18 ± 1.46
ReMixMatch (K=1)	6.77 ± 1.66
ReMixMatch (K=4)	6.18 ± 1.24

Table 2: STL-10 error rate using 1000-label splits. SWWAE and CC-GAN results are from (Zhao et al., 2015) and (Denton et al., 2016).

Ablation	Error Rate	Ablation	Error Rate
ReMixMatch	5.94	No rotation loss	6.08
With K=1	7.32	No pre-mixup loss	6.66
With K=2	6.74	No dist. alignment	7.28
With K=4	6.21	L2 unlabeled loss	17.28
With K=16	5.93	No strong aug.	12.51
MixMatch	11.08	No weak aug.	29.36

Table 3: Ablation study. Error rates are reported on a single 250-label split from CIFAR-10.

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

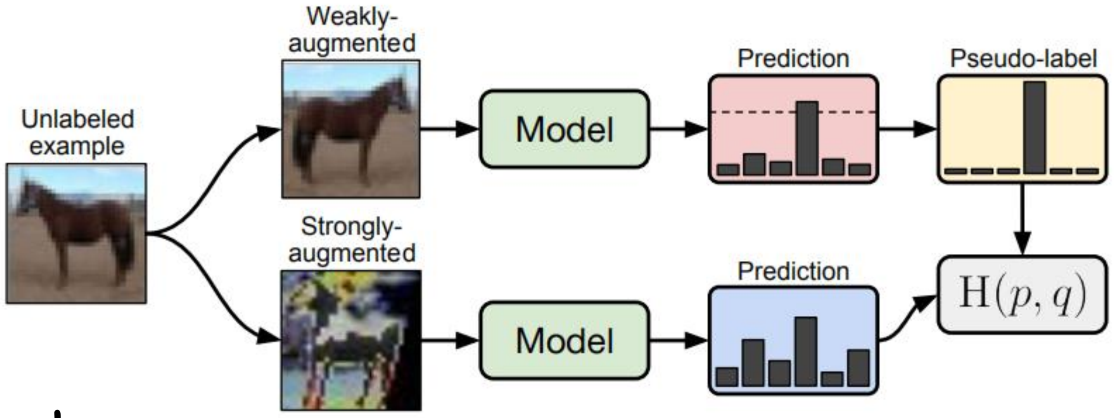
NeurIPS 2020

FixMatch

☐ Pseudo-labeling

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

$q_b = p_{\mathrm{m}}(y|u_b)$ $\hat{q}_b = \arg\max(q_b)$ one-hot



☐ Loss Function

$$\ell_s = \frac{1}{B}\sum_{b=1}^{B}\mathrm{H}(p_b,p_{\mathrm{m}}(y\mid\alpha(x_b)))$$

weakly augment

$$\ell_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B}\mathbb{1}(\max(q_b)\geq\tau)\,\mathrm{H}(\hat{q}_b,p_{\mathrm{m}}(y\mid\mathcal{A}(u_b)))$$

$q_b = p_{\mathrm{m}}(y\mid\alpha(u_b))$

$\hat{q}_b = \arg\max(q_b)$

$\ell_s + \lambda_u\ell_u$

strongly augment $\begin{cases}\text{RandAugment}\\ \text{CTAugment}\end{cases}$

FixMatch

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

Table 1: Comparison of SSL algorithms which include a form of consistency regularization and which (optionally) apply some form of post-processing to the artificial labels. We only mention those components of the SSL algorithm relevant to producing the artificial labels (for example, Virtual Adversarial Training additionally uses entropy minimization [17], MixMatch and ReMixMatch also use MixUp [59], UDA includes additional techniques like training signal annealing, etc.).

FixMatch-Experiment

Wide ResNet-28-2 for CIFAR-10 and SVHN

WRN-28-8 for CIFAR-100 WRN-37-2 for STL-10

$$\lambda_u = 1,\ \eta = 0.03,\ \beta = 0.9,\ \tau = 0.95,\ \mu = 7,\ B = 64,\ K = 2^{20}$$

	CIFAR-10			CIFAR-100			SVHN			STL-10
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 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	26.23±0.82
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	27.99±0.83
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	21.43±2.39
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	10.41±0.61
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	7.66±0.56
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	**5.23**±0.45
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	7.98±1.50
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	**5.17**±0.63

Table 2: Error rates for CIFAR-10, CIFAR-100, SVHN and STL-10 on 5 different folds. FixMatch (RA) uses RandAugment [11] and FixMatch (CTA) uses CTAugment [3] for strong-augmentation. All baseline models (Π-Model [43], Pseudo-Labeling [25], Mean Teacher [51], MixMatch [4], UDA [54], and ReMixMatch [3]) are tested using the same codebase.

Figure 2: FixMatch reaches 78% CIFAR-10 accuracy using only above 10 labeled images.

FixMatch-Experiment

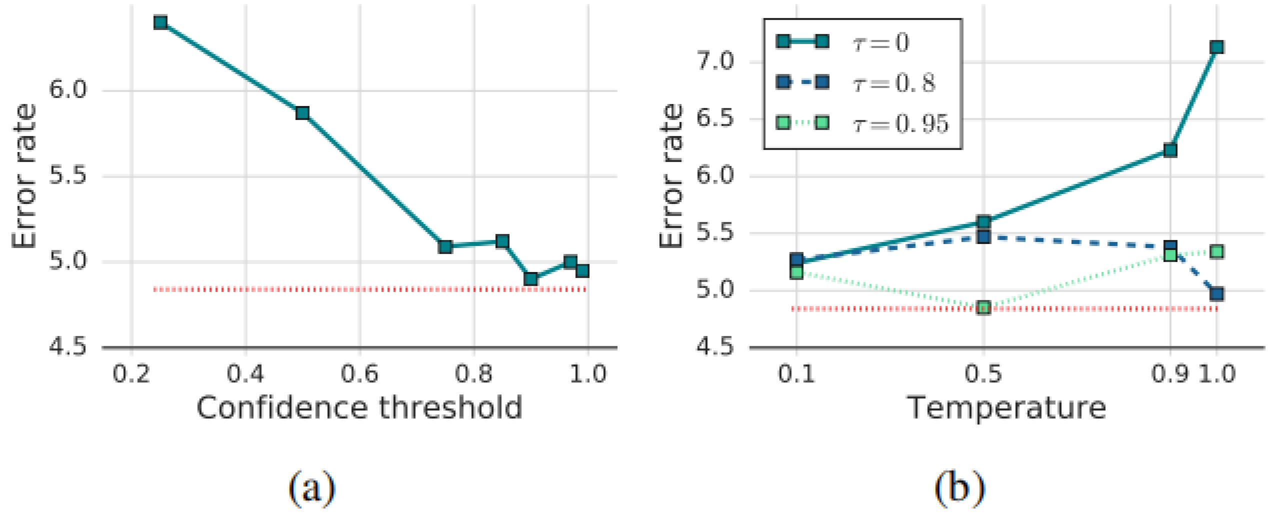


Figure 3: Plots of ablation studies on FixMatch. (a) Varying the confidence threshold for pseudo-labels. (b) Measuring the effect of "sharpening" the predicted label distribution while varying the confidence threshold (τ). Error rate of FixMatch with default hyperparameters is in red dotted line.

Ablation	Error
FixMatch	**4.84**
Only Cutout	6.15
No Cutout	6.15

Table 3: Ablation study with different strong data augmentation of FixMatch. Error rates are reported on a single 250-label split from CIFAR-10.

Thanks