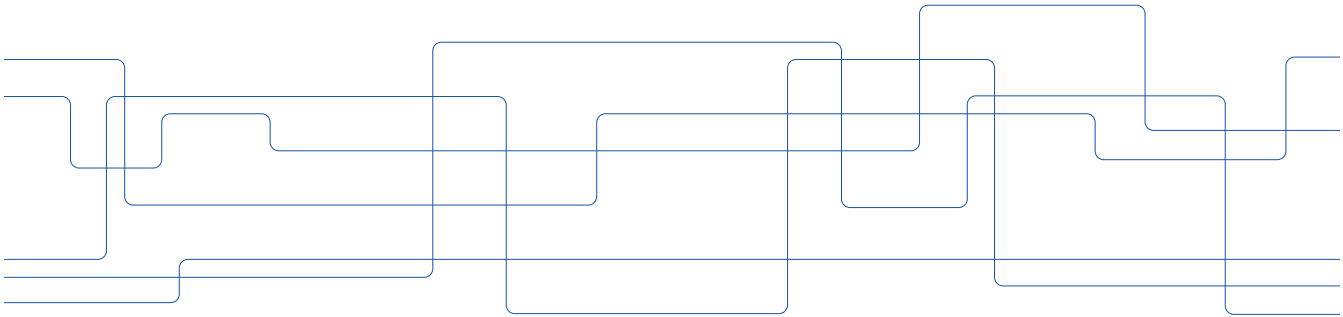
Computer Vision Reading Group

ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring – *Berthelot et al.*

7th of April 2020 – Johan Fredin Haslum



Papers

- **ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring -** *Berthelot et al.*

- **FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence –** *Sohn et al.*

Semi-Supervised Learning

- Why use it?
  - Improve predictive performance on supervised prediction tasks by leveraging unlabeled data
- When can we use it?
  - When abundant amount of unlabeled data is available
  - When gathering labeled data is hard
    - > *Expert labels might be expensive to gather*
    - > *Trade of between cost of gathering more data vs more labels*
  - Rule of thumb: More than x10 more unlabeled data than labeled
- How?
  - Pseudo-labeling, self-training, consistency, adverserial training, etc.

Papers

- **ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring -** *Berthelot et al.* – ICLR 2020

- **FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence –** *Sohn et al. – Arxiv 2020*

- **MixMatch: A Holistic Approach to Semi-Supervised Learning -** *Berthelot et al.* – NeurIPS 2019

Three core ideas

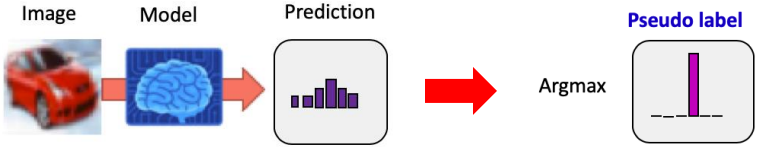
- Entropy Minimization
- Consistency Regularization
- Generic Regularization

Contributions - ReMixMatch

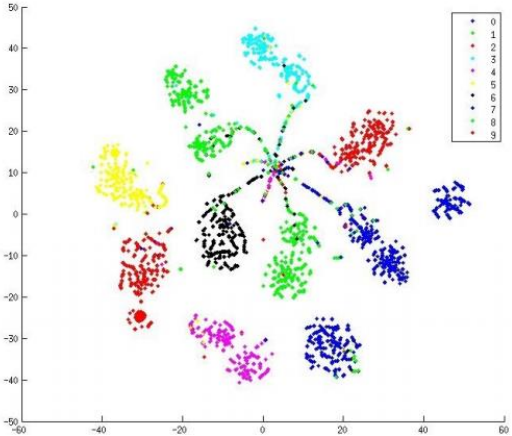
- Entropy Minimization
  - **Distribution Alignment**
- Consistency Regularization
  - **Augmentation Anchoring**
- Generic Regularization

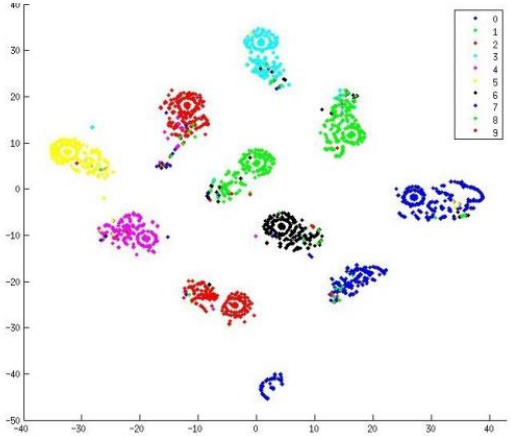
Entropy Minimization - Artificial Labeling

- Pseudo-labeling



  - Use models prediction on unlabeled example as training label
  - $loss = CE_{supervised} + \lambda * CE_{unsupervised}$ (Gradually increasing $\lambda \in [0, 3]$)

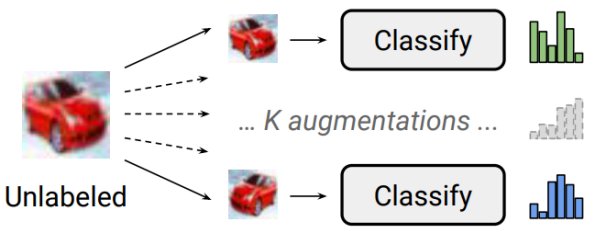


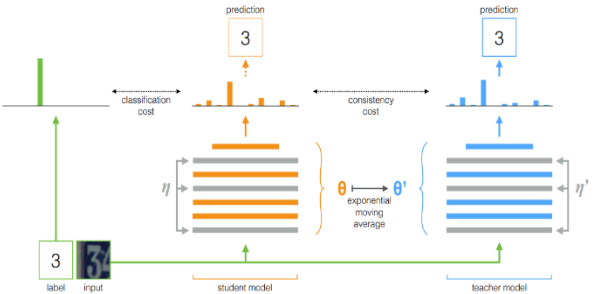


Lee, Dong-Hyun. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." *Workshop on challenges in representation learning, ICML*. Vol. 3. 2013.

Consistency Regularization

- Enforcing robustness to input perturbations
  - Mainly using augmentations, network stochasticity (dropout), temporal ensambling, etc.
  - Loss often calculated between augmented and non-augmented input
  - $loss_{consistensy} = \|pred(x) - pred(aug(x)\|_2^2$
- Basic idea: Similar input should yield similar output





Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

Verma, Vikas, et al. "Interpolation consistency training for semi-supervised learning." *arXiv preprint arXiv:1903.03825* (2019).

Tarvainen, Antti, and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results." *Advances in neural information processing systems*. 2017.

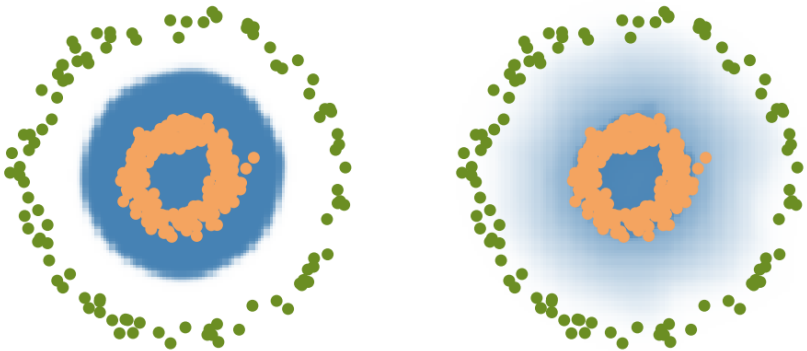
Laine, Samuli, and Timo Aila. "Temporal ensembling for semi-supervised learning." *arXiv preprint arXiv:1610.02242* (2016).

Xie, Qizhe, et al. "Unsupervised data augmentation." *arXiv preprint arXiv:1904.12848* (2019).

Generic Regularization

- Training with few labeled examples can be prone to overfitting
- Strong regularization might be necessary for good generalization for many Semi-Supervised Learning problems
- mixup: Beyond empirical risk minimization

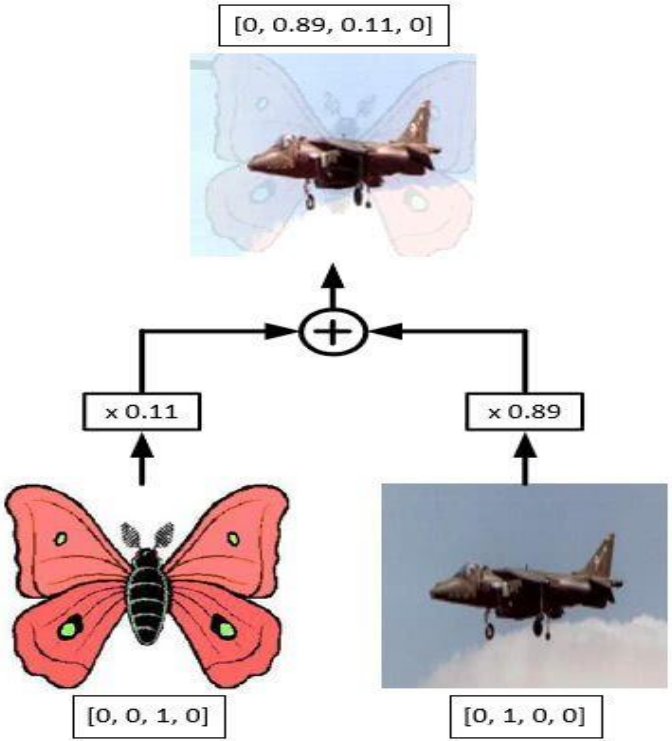
ERM *mixup*



$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412* (2017).

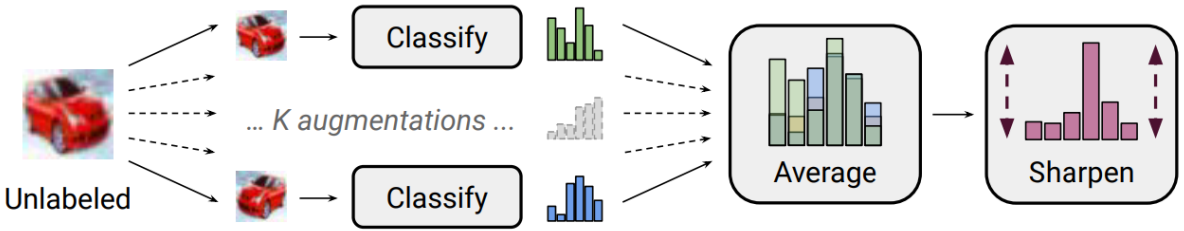
Generic Regularization



Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412* (2017).

MixMatch

- Pushing the State-of-the-art by employing these three core ideas
  - Entropy Minimization
  - Consistency Regularization
  - Generic Regularization



Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

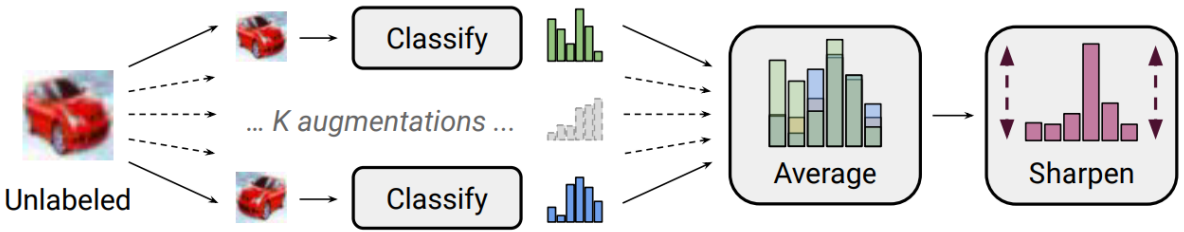
MixMatch – cont.

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 $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.
for $b = 1$ **to** B **do**
  $\hat{x}_b = \text{Augment}(x_b)$ *// Apply data augmentation to* x_b
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \text{Augment}(u_b)$ *// Apply* k^{th} *round of data augmentation to* u_b
  end for
  $\bar{q}_b = \frac{1}{K} \sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of* u_b
  $q_b = \text{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*
end for
$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ *// Augmented labeled examples and their labels*
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ *// Augmented unlabeled examples, guessed labels*
$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ *// Apply* MixUp *to labeled data and entries from* $\mathcal{W}$
$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ *// Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$
return $\mathcal{X}', \mathcal{U}'$

MixMatch – cont.

for $b = 1$ **to** B **do**
  $\hat{x}_b = \text{Augment}(x_b)$ // *Apply data augmentation to* x_b
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \text{Augment}(u_b)$ // *Apply* k^{th} *round of data augmentation to* u_b
  end for
  $\bar{q}_b = \frac{1}{K} \sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ // *Compute average predictions across all augmentations of* u_b
  $q_b = \text{Sharpen}(\bar{q}_b, T)$ // *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*



Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

MixMatch – cont.

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

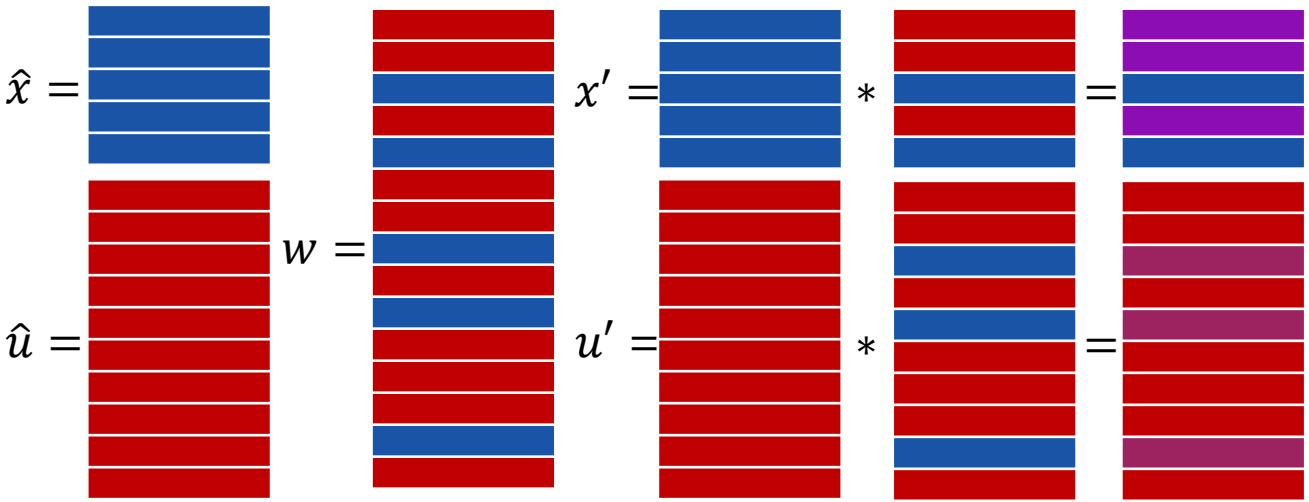
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ // *Augmented unlabeled examples, guessed labels*

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

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

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

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



Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

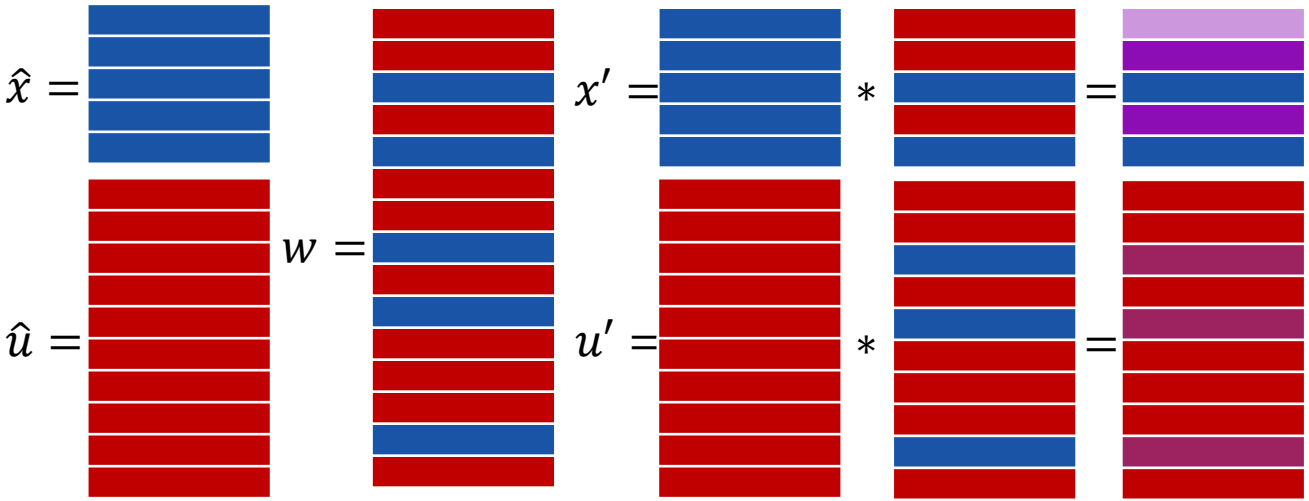
MixMatch – cont.

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha) \tag{2}$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta)) \tag{3}$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2 \tag{4}$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}} \tag{5}$$



Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

MixMatch – cont.

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha) \tag{2}$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta)) \tag{3}$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2 \tag{4}$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}} \tag{5}$$

Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

MixMatch - Results

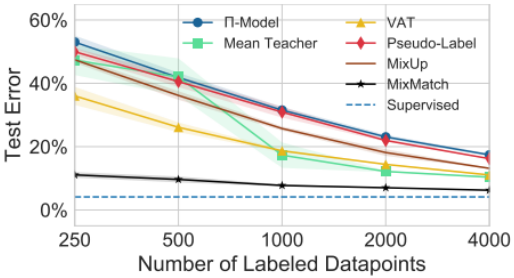


Figure 2: Error rate comparison of MixMatch to baseline methods on CIFAR-10 for a varying number of labels. Exact numbers are provided in table 5 (appendix). "Supervised" refers to training with all 50000 training examples and no unlabeled data. With 250 labels MixMatch reaches an error rate comparable to next-best method's performance with 4000 labels.

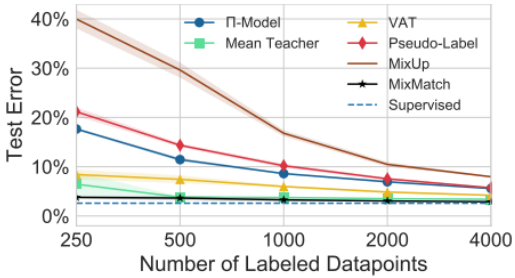


Figure 3: Error rate comparison of MixMatch to baseline methods on SVHN for a varying number of labels. Exact numbers are provided in table 6 (appendix). "Supervised" refers to training with all 73257 training examples and no unlabeled data. With 250 examples MixMatch nearly reaches the accuracy of supervised training for this model.

Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

MixMatch – Results –Cont.

Ablation	250 labels	4000 labels
MixMatch	11.80	6.00
MixMatch without distribution averaging ($K = 1$)	17.09	8.06
MixMatch with $K = 3$	11.55	6.23
MixMatch with $K = 4$	12.45	5.88
MixMatch without temperature sharpening ($T = 1$)	27.83	10.59
MixMatch with parameter EMA	11.86	6.47
MixMatch without MixUp	39.11	10.97
MixMatch with MixUp on labeled only	32.16	9.22
MixMatch with MixUp on unlabeled only	12.35	6.83
MixMatch with MixUp on separate labeled and unlabeled	12.26	6.50
Interpolation Consistency Training [45]	38.60	6.81

Table 4: Ablation study results. All values are error rates on CIFAR-10 with 250 or 4000 labels.

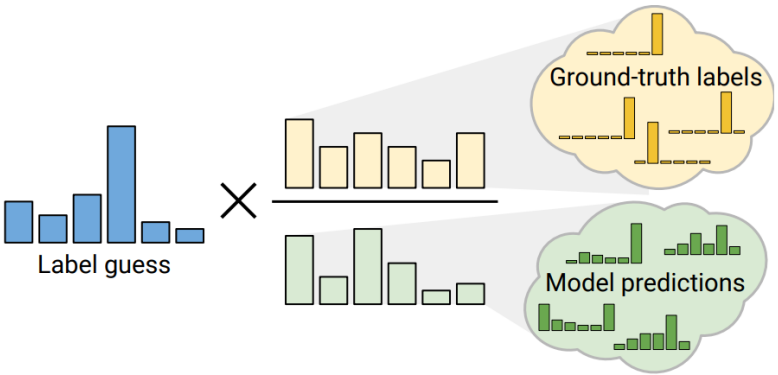
Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

ReMixMatch (Finally)

- Entropy Minimization
  - **Distribution Alignment**
- Consistency Regularization
  - **Augmentation Anchoring**
- Generic Regularization

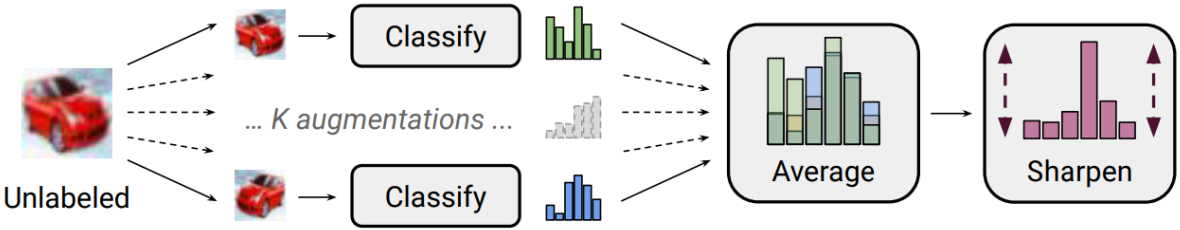
Distributional Alignment

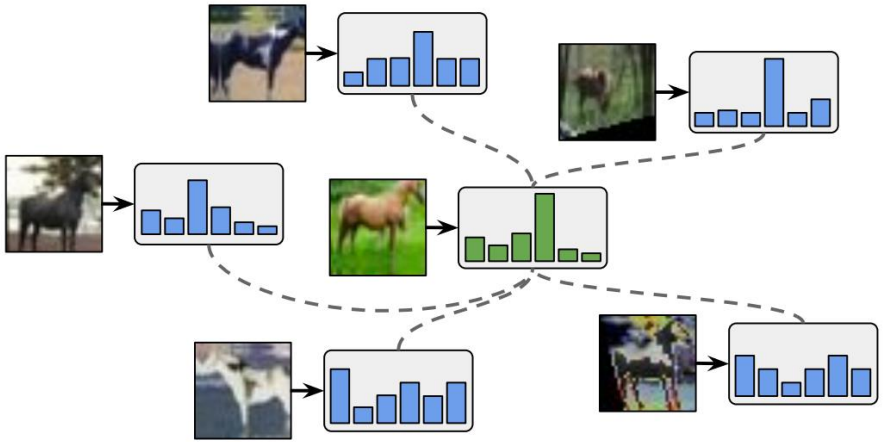
- Enforce ”Fairness”
  - Push the network class output frequency to be aligned with class frequency for the labeled portion of the data
  - Basically: Proportionally increase/decrease the label guess value for those classes that are disproportionally under/over represented in the models predicted outputs



Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

Augmentation Anchoring





Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

ReMixMatch - Algorithm

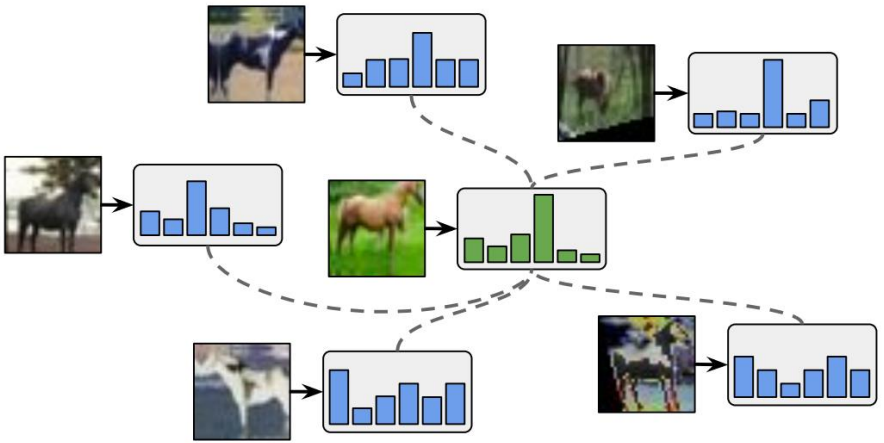
Algorithm 1 ReMixMatch algorithm for producing a collection of processed labeled examples and processed unlabeled examples with label guesses (cf. Berthelot et al. (2019) Algorithm 1.)

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b) : b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b : b \in (1, \dots, B)\big)$, 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
  $\hat{u}_{b,k} = \text{StrongAugment}(u_b); k \in \{1, \dots, K\}$ // *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*
  $q_b = \text{Normalize}\big(q_b^{1/T}\big)$ // *Apply temperature sharpening to label guess*
end for
$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ // *Augmented labeled examples and their labels*
$\hat{\mathcal{U}}_1 = \big((\hat{u}_{b,1}, q_b); b \in (1, \dots, B)\big)$ // *First strongly augmented unlabeled example and guessed label*
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ // *All strongly augmented unlabeled examples*
$\hat{\mathcal{U}} = \hat{\mathcal{U}} \cup \big((\tilde{u}_b, q_b); b \in (1, \dots, B)\big)$ // *Add weakly augmented unlabeled examples*
$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ // *Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ // *Apply* MixUp *to labeled data and entries from* $\mathcal{W}$
$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ // *Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$
return $\mathcal{X}', \mathcal{U}', \hat{\mathcal{U}}_1$

Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

ReMixMatch – Algorithm – Cont.

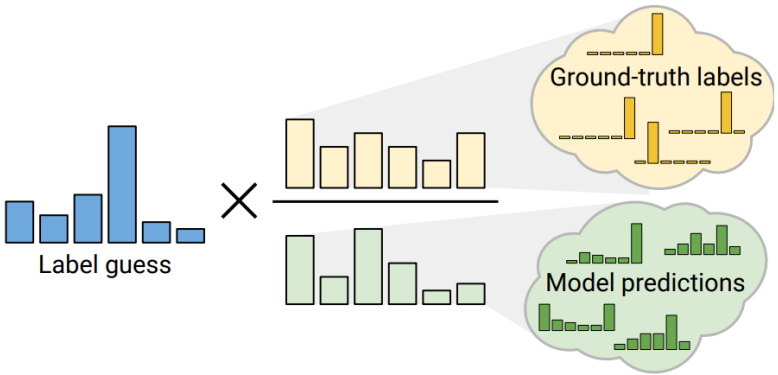
2: **for** $b = 1$ **to** B **do**
3: $\hat{x}_b = \text{StrongAugment}(x_b)$ // *Apply strong data augmentation to* x_b
4: $\hat{u}_{b,k} = \text{StrongAugment}(u_b); k \in \{1, \dots, K\}$ // *Apply strong data augmentation* K *times to* u_b
5: $\tilde{u}_b = \text{WeakAugment}(u_b)$ // *Apply weak data augmentation to* u_b
6: $q_b = p_{\text{model}}(y \mid \tilde{u}_b; \theta)$ // *Compute prediction for weak augmentation of* u_b
7: $q_b = \text{Normalize}(q_b \times p(y)/\tilde{p}(y))$ // *Apply distribution alignment*
8: $q_b = \text{Normalize}(q_b^{1/T})$ // *Apply temperature sharpening to label guess*
9: **end for**



Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

ReMixMatch – Algorithm – Cont.

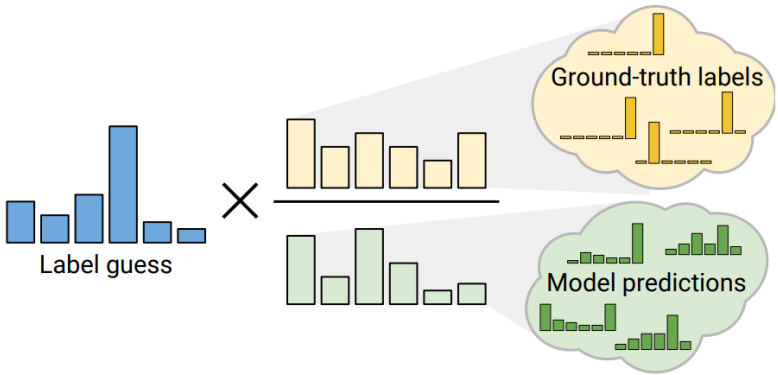
2: **for** $b = 1$ **to** B **do**
3: $\hat{x}_b = \text{StrongAugment}(x_b)$ // *Apply strong data augmentation to* x_b
4: $\hat{u}_{b,k} = \text{StrongAugment}(u_b); k \in \{1, \dots, K\}$ // *Apply strong data augmentation* K *times to* u_b
5: $\tilde{u}_b = \text{WeakAugment}(u_b)$ // *Apply weak data augmentation to* u_b
6: $q_b = p_{\text{model}}(y \mid \tilde{u}_b; \theta)$ // *Compute prediction for weak augmentation of* u_b
7: $q_b = \text{Normalize}(q_b \times p(y)/\tilde{p}(y))$ // *Apply distribution alignment*
8: $q_b = \text{Normalize}(q_b^{1/T})$ // *Apply temperature sharpening to label guess*
9: **end for**



Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

ReMixMatch – Algorithm – Cont.

2: **for** $b = 1$ **to** B **do**
3: $\hat{x}_b = \text{StrongAugment}(x_b)$ // *Apply strong data augmentation to* x_b
4: $\hat{u}_{b,k} = \text{StrongAugment}(u_b); k \in \{1, \dots, K\}$ // *Apply strong data augmentation* K *times to* u_b
5: $\tilde{u}_b = \text{WeakAugment}(u_b)$ // *Apply weak data augmentation to* u_b
6: $q_b = p_{\text{model}}(y \mid \tilde{u}_b; \theta)$ // *Compute prediction for weak augmentation of* u_b
7: $q_b = \text{Normalize}(q_b \times p(y)/\tilde{p}(y))$ // *Apply distribution alignment*
8: $q_b = \text{Normalize}(q_b^{1/T})$ // *Apply temperature sharpening to label guess*
9: **end for**



Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

ReMixMatch – Algorithm – Cont.

(Labeled+Mixed) MixUp, same as in MixMatch

(Unlabeled+Mixed) MixUp, same as in MixMatch but with Cross Entropy instead

$$\sum_{x,p\in\mathcal{X}'} \mathrm{H}(p, p_{\mathrm{model}}(y|x;\theta)) + \lambda_{\mathcal{U}} \sum_{u,q\in\mathcal{U}'} \mathrm{H}(q, p_{\mathrm{model}}(y|u;\theta)) \tag{3}$$

$$+\lambda_{\hat{\mathcal{U}}_1} \sum_{u,q\in\hat{\mathcal{U}}_1} \mathrm{H}(q, p_{\mathrm{model}}(y|u;\theta)) + \lambda_r \sum_{u\in\hat{\mathcal{U}}_1} \mathrm{H}(r, p_{\mathrm{model}}(r|\,\mathrm{Rotate}(u,r);\theta)) \tag{4}$$

Unlabeled images with standard Cross Entropy

Unlabeled images, Self-Supervised rotational prediction

Berthelot, David, et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring." *arXiv preprint arXiv:1911.09785* (2019).

Experiments

- Datasets:
  - CIFAR-10
  - SVHN
  - STL-10
- Implementation
  - Same codebase for all experiments and different methods
  - Wide ResNet-28-2
  - Same training algorithm
  - Five random splits per dataset and training examples
- Comparison with:
  - *VAT – Virtual Adversarial Training*
  - *Mean Teacher*
  - *MixMatch*
  - *UDA – Unsupervised Data Augmentation*

ReMixMatch Results CIFAR-10 and SVHN

	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 – Ablation Study

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.

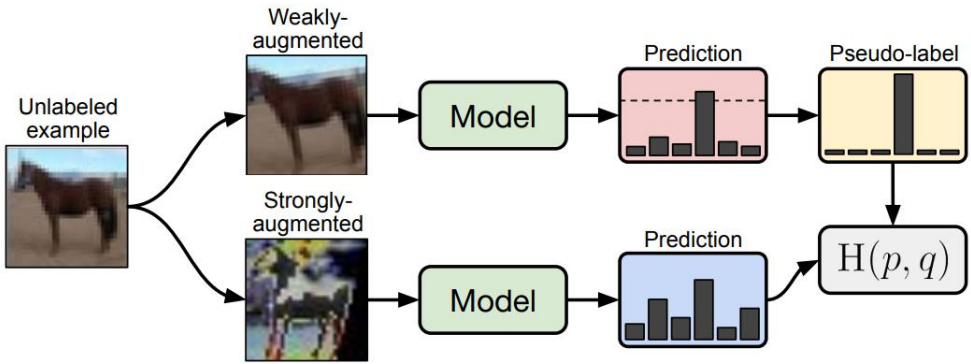
ReMixMatch - Conclusions

- Adding Distributional Alignment and Augmentation Anchoring to MixMatch reduces the need for labeled data even further

- My take aways:
  - The strong augmentation is what really pushes performance
  - SOTA results but only on small datasets
  - A lot of moving parts, a lot of hyper parameter optimization

FixMatch

- Back to basic
  - Simplified version of ReMixMatch and MixMatch
- Core ideas
  - Consistency Regularization
  - Pseudo-labeling



FixMatch - Algorithm

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \dots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \dots, \mu B)\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_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** μB **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 $\ell_s + \lambda_u \ell_u$

$$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, p_\mathrm{m}(y \mid \alpha(x_b))) \qquad (3)$$

$$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, p_\mathrm{m}(y \mid \mathcal{A}(u_b))) \qquad (4)$$

Loss = $\ell_s + \lambda_u \ell_u$

FixMatch - Results

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

FixMatch - Results

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

Table 4: Error rates of FixMatch (CTA) on a single 40-label split of CIFAR-10 and SVHN with different random seeds.

Dataset	Runs (ordered by accuracy)				
	1	2	3	4	5
CIFAR-10	5.46	6.17	9.37	10.85	13.32
SVHN	2.40	2.47	6.24	6.32	6.38

FixMatch – Ablation Study

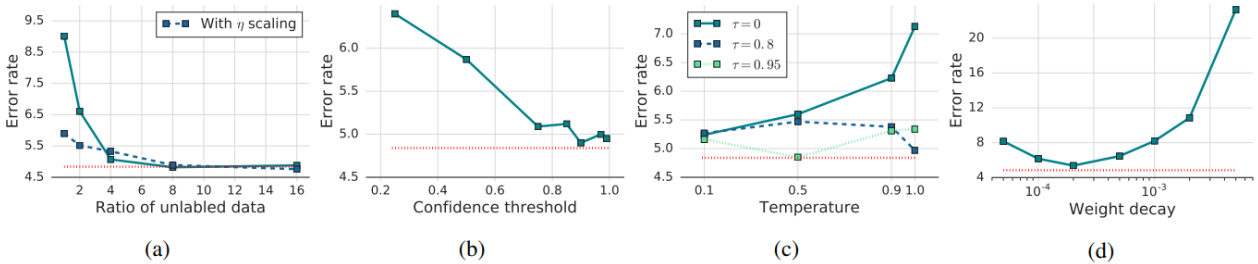


Figure 3: Plots of various ablation studies on FixMatch. (a) Varying the ratio of unlabeled data (μ) with different learning rate (η) scaling strategies. (b) Varying the confidence threshold for pseudo-labels. (c) Measuring the effect of "sharpening" the predicted label distribution while varying the confidence threshold (τ). (d) Varying the loss coefficient for weight decay. We include the error rate of FixMatch with the default hyperparameter setting in red dotted line for each plot.

FixMatch – CIFAR-10 Single Image per Class

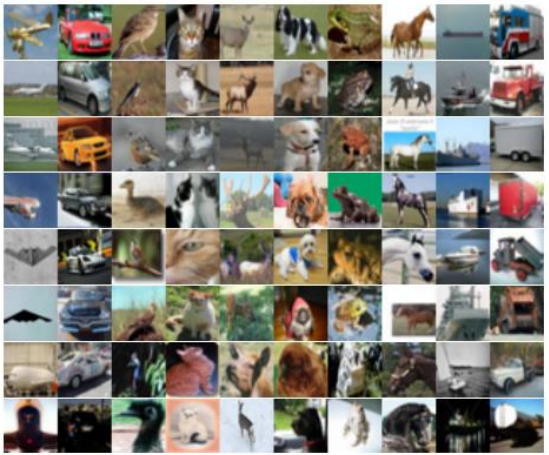


Figure 5: Labeled training data for the 1-label-per-class semi-supervised experiment. Each row corresponds to the complete labeled training set for one run of our algorithm, sorted from the most prototypical dataset (first row) to least prototypical dataset (last row).

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).

Conclusions

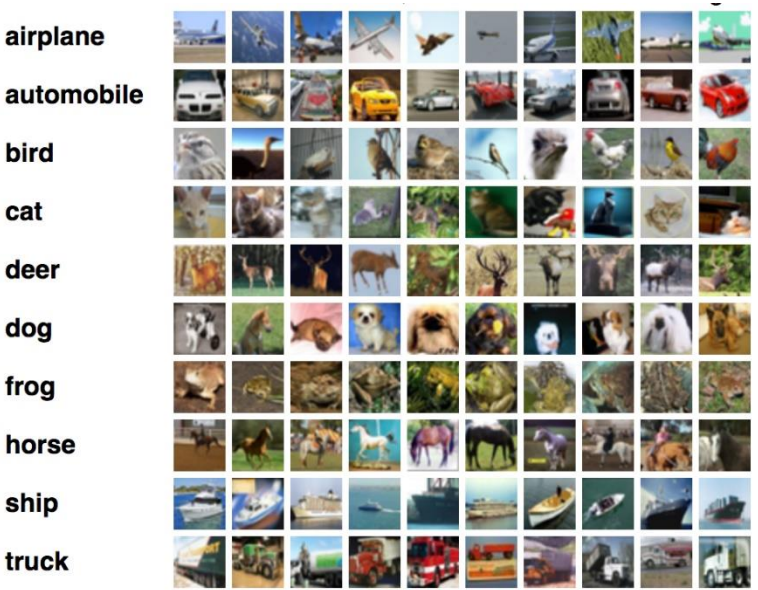
- FixMatch is able to reach SOTA performance on SSL datasets by simply enforcing consistency between augmented samples and creating pseudo-labels for unlabeled examples when sufficiently confident
- Simple yet effective
- Highlight how delicate SSL settings can be to small deviations to optimal parameter selections

- Drawbacks:
  - Limited novelty
  - Non-significant improvement compared to previous work

Thank you for listening!

ReMixMatch – CIFAR-10

	% of data labeled	Error rate
Fully supervised	100.0	3.62
ReMixMatch	8.0	5.14
ReMixMatch	2.0	5.73
ReMixMatch	0.50	6.27



Common Experimental setup

- Standard image classification datasets with varying number of labels removed
  - CIFAR-10, CIFAR-100, SVHN, ImageNet
  - Keep x % of labels for each class, regard the rest as unlabeled
- STL-10
  - 10 classes
  - 5,000 labeled training images
  - 8,000 labeled test images
  - 100,000 unlabeled images, contains other classes, class frequency not uniform