

# MixMatch: A Holistic Approach to SSL

Yuan Feng, Yaan Zhang, Patrick Wisniewski, Kaveh Safavigerdini

11-18-2022



University of Missouri

# Introduction

## ❖ Labeled datasets

- Expert knowledge → expensive
- Private information → difficult to get

## ❖ Unlabeled datasets

- Much easier or cheaper

## ★ Semi-supervised learning (SSL)

- Labeled datasets + unlabeled datasets

## 📦 MixMatch

- 📦 Guesses low-entropy labels for data-augmented unlabeled examples.
- 📦 Mixes labeled and unlabeled data using MixUp.
- 📦 Obtains state-of-the-art results by a large margin across many datasets and labeled data amounts.

# Related works

## ❖ **Loss term** $p_{\text{model}}(y \mid x; \theta)$

➤ **Consistency Regularization:** encourages the model to produce the same output distribution when it's inputs are perturbed.

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

■  $\|p_{\text{model}}(y \mid \text{Augment}(x); \theta) - p_{\text{model}}(y \mid \text{Augment}(x); \theta)\|_2^2$

■  $x$  is unlabeled example,  $\theta$  is parameters,  $y$  is prediction results.

■  $\text{Augment}(x)$  is a stochastic transformation, so two  $\text{Augment}(x)$  are different.

# Related works

## ❖ **Loss term** $p_{\text{model}}(y \mid x; \theta)$

- Consistency Regularization
- **Entropy Minimization**: encourages the model to output confident predictions on unlabeled data
  - Assumption: the classifier's decision boundary should not pass through high-density regions of the marginal data distribution.
  - ★ Require the classifier output low-entropy predictions on unlabeled data.

# Related works

## ❖ **Loss term** $p_{\text{model}}(y | x; \theta)$

- Consistency Regularization
- Entropy Minimization
- **Traditional Regularization:** encourages the model to generalize well and avoid overfitting the training data.
  - ★ General approach of imposing a constraint on a model to make it harder to memorize the training data and therefore generalize better to unseen data\*.
- ❑ Loss term used mainly based on one of three classes

\* Geoffrey Hinton and Drew van Camp. Keeping neural networks simple by minimizing the description length of the weights. In proceedings of the 6th Annual ACM Conference on Computational Learning Theory, 1993.

# Related works

## ❖ MixMatch

- Consistency Regularization
  - Utilizes a form of consistency regularization through the use of standard data augmentation for images (random horizontal flips and crops).
- Entropy Minimization
  - Uses a “sharpening” function on the target distribution for unlabeled data to minimize entropy.
- Traditional Regularization
  - Uses weight decay and **MixUp** as regularizer.

# Related works

## ❖ MixMatch

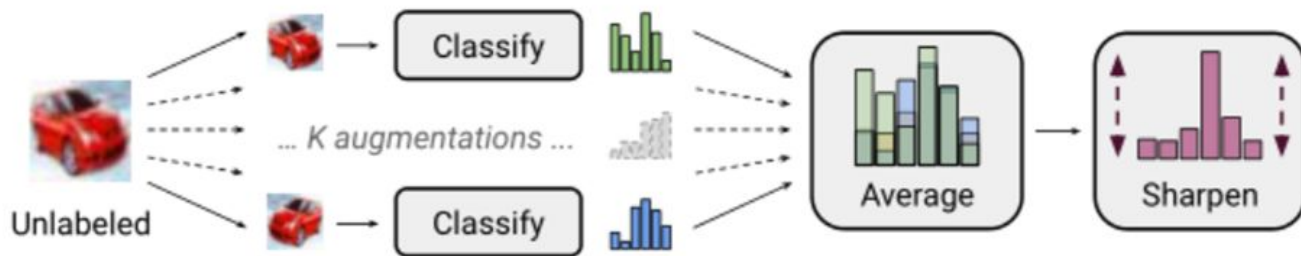


Diagram of the label guessing process

### ➤ Advantages:

- Obtains state-of-the-art results on all standard image benchmarks, reduce the error rate on CIFAR-10 by a factor of 4;
- MixMatch is greater than the sum of its parts;
- Simultaneously strength both privacy guarantees and accuracy.

# MixMatch: quick introduction

The high-level idea of MixMatch is to guess low-entropy labels for the augmented unlabeled data and apply further regularization by using MixUp in both labeled and unlabeled data.

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



# MixMatch: data augmentation

We're going to do data augmentation. This is to alleviate insufficient labeled data. This paper does 1 enhancement for marked labels and k enhancements for unmarked labels.

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

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

# MixMatch: Label Guessing & sharpening

The classifier averages the classification results of K times augmented unlabeled data and guesses the label.

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

Sharpening is a very important process. This idea is equivalent to the relu process in deep learning. Not sharpening after averaging will have a great impact on the result.

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

# MixMatch: MixUp

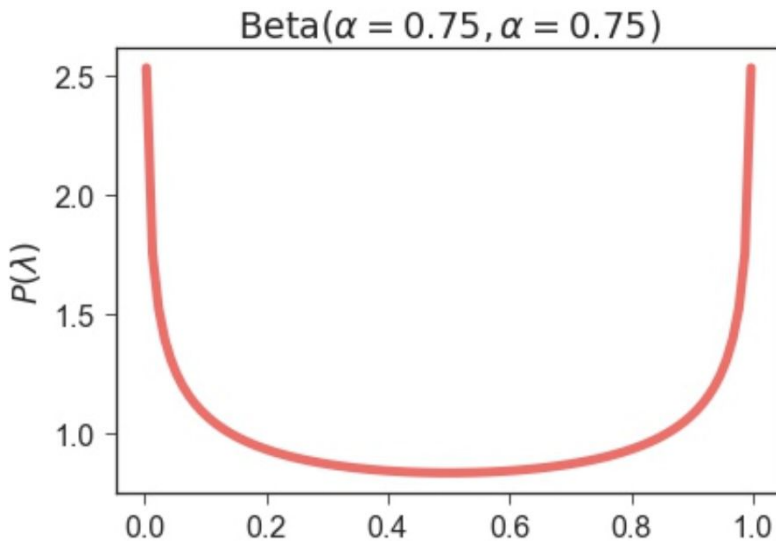
Different from the previous Mixup method, the MixMatch method mixes labeled data and unlabeled data for Mixup.

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

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

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

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



# MixMatch: Loss Function

The loss term is calculated separately for the augmented labeled data  $\mathcal{x}'$ , and the unlabeled augmented data  $\mathcal{u}'$ .

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

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

Cross Entropy calculation needs to use the Softmax function first.

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

$$\text{softmax}(z_i + c) = \frac{\exp(z_i + c)}{\sum_j \exp(z_j + c)} = \frac{\exp(z_i)}{\sum_j \exp(z_j)} = \text{softmax}(z_i)$$

The final overall loss function is a weighted combination of the two.

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

# Introduction to the experiment

Datasets used:

CIFAR-10

CIFAR-100

STL-10

SVHN

Model compared against:

Pseudo label

VAT

MeanTeacher

Pi Model

Baseline supervised learning model

# Baseline Methods

For a standard comparison, the authors provide 4 baseline methods from a reference paper. [1]

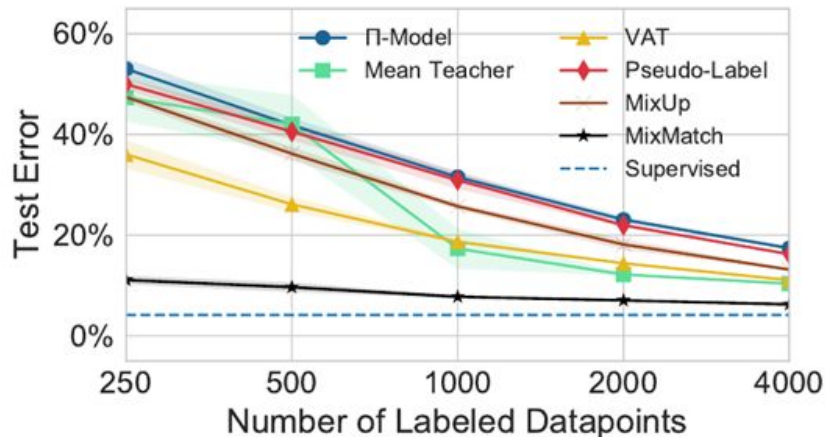
$\Pi$ -Model, Mean Teacher, Virtual Adversarial Training, Pseudo-Label

Also they compare the model with MixUp by some modification to be used in SSL

# Comparing MixMatch performance with other SSL methods on:

## 1. Cifar10

- Comparing MixMatch Error on Cifar10 with other base methods
- They re-implement the baseline methods and refine the hyperparameters
- Supervised works on labeled data  
MixUp is used for augmenting
- For 250 labels, the error rate of MixMatch is 11% but VAT has 36% error rate.



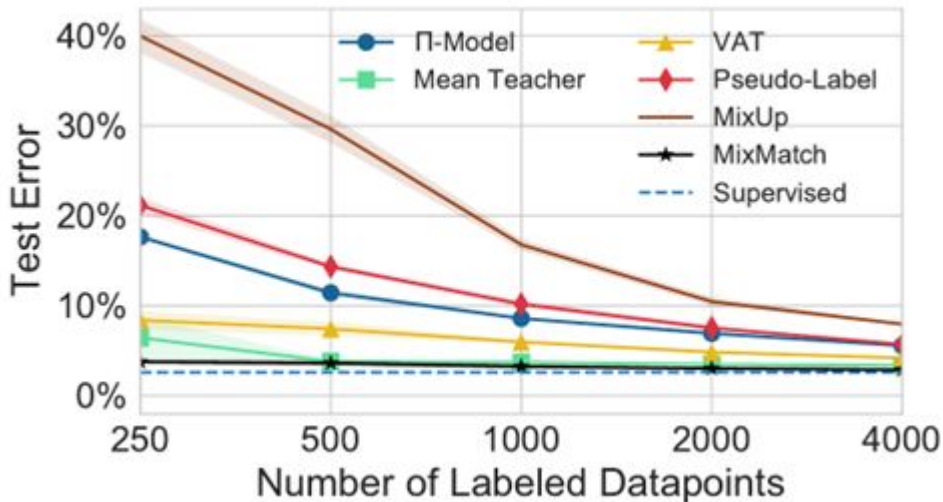
# Comparing MixMatch performance with other SSL methods on:

## 2. SVHN and SVHN+Extra Dataset

- 5 different datasets containing 250 to 4000 data points
- MixMatch's performance is relatively constant
- Two parts: Train (73257-example) + Extra 604388 samples full data -> give

higher ratio

- $\alpha$  = Beta distribution used in MixUp
- $\alpha = 0.25$ ,  $\lambda_u = 250$ ,





# Comparing MixMatch performance with other SSL methods on:

## 3. STL-10

- Comparing with state-of-the-art methods
- Comparing with 1000 and 5000 training test
- The experiment setup is not similar but because the error is lower by factor of 2, Therefore, the MixMatch can be regarded as confidence.

Method	1000 labels	5000 labels
CutOut [12]	-	12.74
IIC [20]	-	11.20
SWWAE [48]	25.70	-
CC-GAN <sup>2</sup> [11]	22.20	-
MixMatch	10.18 $\pm$ 1.46	5.59

# Ablation Study

Since MixMatch consists of several Semi-supervised learning mechanisms  
It's a good idea to investigate the effect of removing each components

As you can see, the effect of  
Temperature sharpening and  
Removing the MixUp is much bigger  
Than others.

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.

# Strengths of Mix Match

Achieved lowest error rates of all models in these experiments

Performed best with SVHN datasets

Achieved lowest error rates with less labels

## B.2 SVHN

Training the same model with supervised learning on the entire 73257-example training set achieved an error rate of 2.59%.

Methods/Labels	250	500	1000	2000	4000
PiModel	$17.65 \pm 0.27$	$11.44 \pm 0.39$	$8.60 \pm 0.18$	$6.94 \pm 0.27$	$5.57 \pm 0.14$
PseudoLabel	$21.16 \pm 0.88$	$14.35 \pm 0.37$	$10.19 \pm 0.41$	$7.54 \pm 0.27$	$5.71 \pm 0.07$
Mixup	$39.97 \pm 1.89$	$29.62 \pm 1.54$	$16.79 \pm 0.63$	$10.47 \pm 0.48$	$7.96 \pm 0.14$
VAT	$8.41 \pm 1.01$	$7.44 \pm 0.79$	$5.98 \pm 0.21$	$4.85 \pm 0.23$	$4.20 \pm 0.15$
MeanTeacher	$6.45 \pm 2.43$	$3.82 \pm 0.17$	$3.75 \pm 0.10$	$3.51 \pm 0.09$	$3.39 \pm 0.11$
MixMatch	$3.78 \pm 0.26$	$3.64 \pm 0.46$	$3.27 \pm 0.31$	$3.04 \pm 0.13$	$2.89 \pm 0.06$

Table 6: Error rate (%) for SVHN.



# Weaknesses of MixMatch

Did not perform substantially better than MeanTeacher with SVHN datasets

More entropy = higher error rates

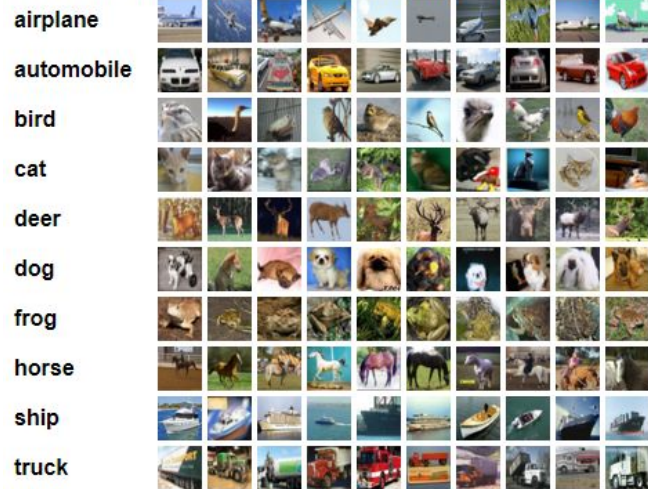
## B.1 CIFAR-10

Training the same model with supervised learning on the entire 50000-example training set achieved an error rate of 4.13%.

Methods/Labels	250	500	1000	2000	4000
PiModel	$53.02 \pm 2.05$	$41.82 \pm 1.52$	$31.53 \pm 0.98$	$23.07 \pm 0.66$	$17.41 \pm 0.37$
PseudoLabel	$49.98 \pm 1.17$	$40.55 \pm 1.70$	$30.91 \pm 1.73$	$21.96 \pm 0.42$	$16.21 \pm 0.11$
Mixup	$47.43 \pm 0.92$	$36.17 \pm 1.36$	$25.72 \pm 0.66$	$18.14 \pm 1.06$	$13.15 \pm 0.20$
VAT	$36.03 \pm 2.82$	$26.11 \pm 1.52$	$18.68 \pm 0.40$	$14.40 \pm 0.15$	$11.05 \pm 0.31$
MeanTeacher	$47.32 \pm 4.71$	$42.01 \pm 5.86$	$17.32 \pm 4.00$	$12.17 \pm 0.22$	$10.36 \pm 0.25$
MixMatch	$11.08 \pm 0.87$	$9.65 \pm 0.94$	$7.75 \pm 0.32$	$7.03 \pm 0.15$	$6.24 \pm 0.06$

Table 5: Error rate (%) for CIFAR10.

Here are the classes in the dataset, as well as 10 random images from each:



Example of cifar10 dataset

# Limitations of MixMatch

The paper did not list the amount of time it took each model to perform

Seems to perform better with datasets that have lower entropy

## B.1 CIFAR-10

Training the same model with supervised learning on the entire 50000-example training set achieved an error rate of 4.13%.

Methods/Labels	250	500	1000	2000	4000
PiModel	53.02 $\pm$ 2.05	41.82 $\pm$ 1.52	31.53 $\pm$ 0.98	23.07 $\pm$ 0.66	17.41 $\pm$ 0.37
PseudoLabel	49.98 $\pm$ 1.17	40.55 $\pm$ 1.70	30.91 $\pm$ 1.73	21.96 $\pm$ 0.42	16.21 $\pm$ 0.11
Mixup	47.43 $\pm$ 0.92	36.17 $\pm$ 1.36	25.72 $\pm$ 0.66	18.14 $\pm$ 1.06	13.15 $\pm$ 0.20
VAT	36.03 $\pm$ 2.82	26.11 $\pm$ 1.52	18.68 $\pm$ 0.40	14.40 $\pm$ 0.15	11.05 $\pm$ 0.31
MeanTeacher	47.32 $\pm$ 4.71	42.01 $\pm$ 5.86	17.32 $\pm$ 4.00	12.17 $\pm$ 0.22	10.36 $\pm$ 0.25
MixMatch	11.08 $\pm$ 0.87	9.65 $\pm$ 0.94	7.75 $\pm$ 0.32	7.03 $\pm$ 0.15	6.24 $\pm$ 0.06

Table 5: Error rate (%) for CIFAR10.

## B.2 SVHN

Training the same model with supervised learning on the entire 73257-example training set achieved an error rate of 2.59%.

Methods/Labels	250	500	1000	2000	4000
PiModel	17.65 $\pm$ 0.27	11.44 $\pm$ 0.39	8.60 $\pm$ 0.18	6.94 $\pm$ 0.27	5.57 $\pm$ 0.14
PseudoLabel	21.16 $\pm$ 0.88	14.35 $\pm$ 0.37	10.19 $\pm$ 0.41	7.54 $\pm$ 0.27	5.71 $\pm$ 0.07
Mixup	39.97 $\pm$ 1.89	29.62 $\pm$ 1.54	16.79 $\pm$ 0.63	10.47 $\pm$ 0.48	7.96 $\pm$ 0.14
VAT	8.41 $\pm$ 1.01	7.44 $\pm$ 0.79	5.98 $\pm$ 0.21	4.85 $\pm$ 0.23	4.20 $\pm$ 0.15
MeanTeacher	6.45 $\pm$ 2.43	3.82 $\pm$ 0.17	3.75 $\pm$ 0.10	3.51 $\pm$ 0.09	3.39 $\pm$ 0.11
MixMatch	3.78 $\pm$ 0.26	3.64 $\pm$ 0.46	3.27 $\pm$ 0.31	3.04 $\pm$ 0.13	2.89 $\pm$ 0.06

Table 6: Error rate (%) for SVHN.

# Possible future works and improvements

Explore scenarios where it is best to use MixMatch

Compare time to complete for MixMatch and other models

Compare compute resources for MixMatch and other models

# Conclusion


Has lowest error rates of all models

Performs well with low and high number of classes

Performs best with low entropy datasets

Time to compute is unknown





Thanks