Modern Machine Learning Paradigms

- Semi-Supervised Learning
- Self-Supervised Learning
- Self / Semi-Supervised Learning

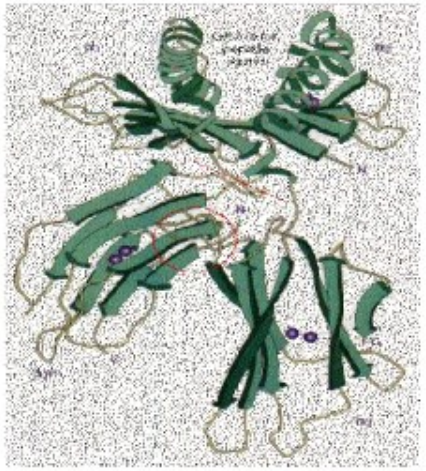
S. A. Seyedi
Seyedi.ml

Semi-Supervised Learning

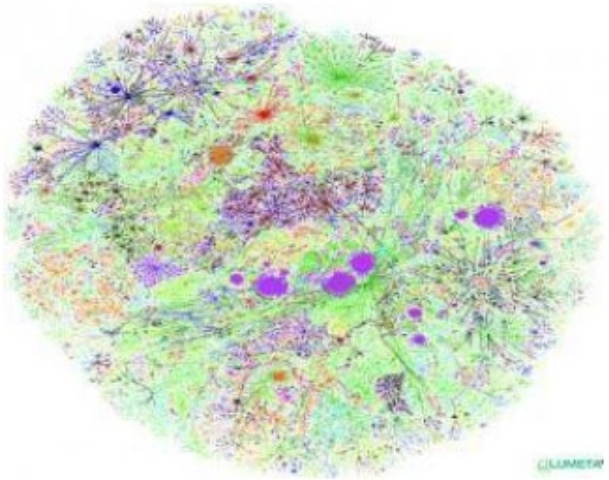
Classic Paradigm Insufficient Nowadays

Modern applications: **massive amounts** of raw data.

Only a tiny fraction can be annotated by human experts.



Protein sequences



Billions of webpages

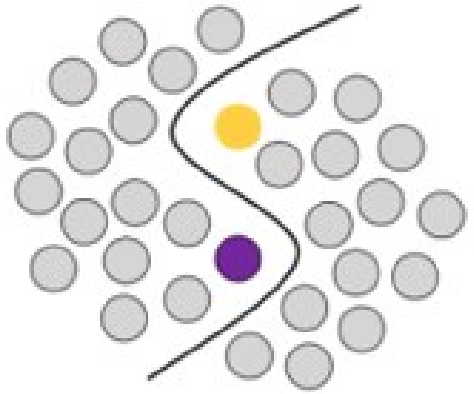
Images

Modern ML: New Learning Approaches

Modern applications: **massive amounts** of raw data.

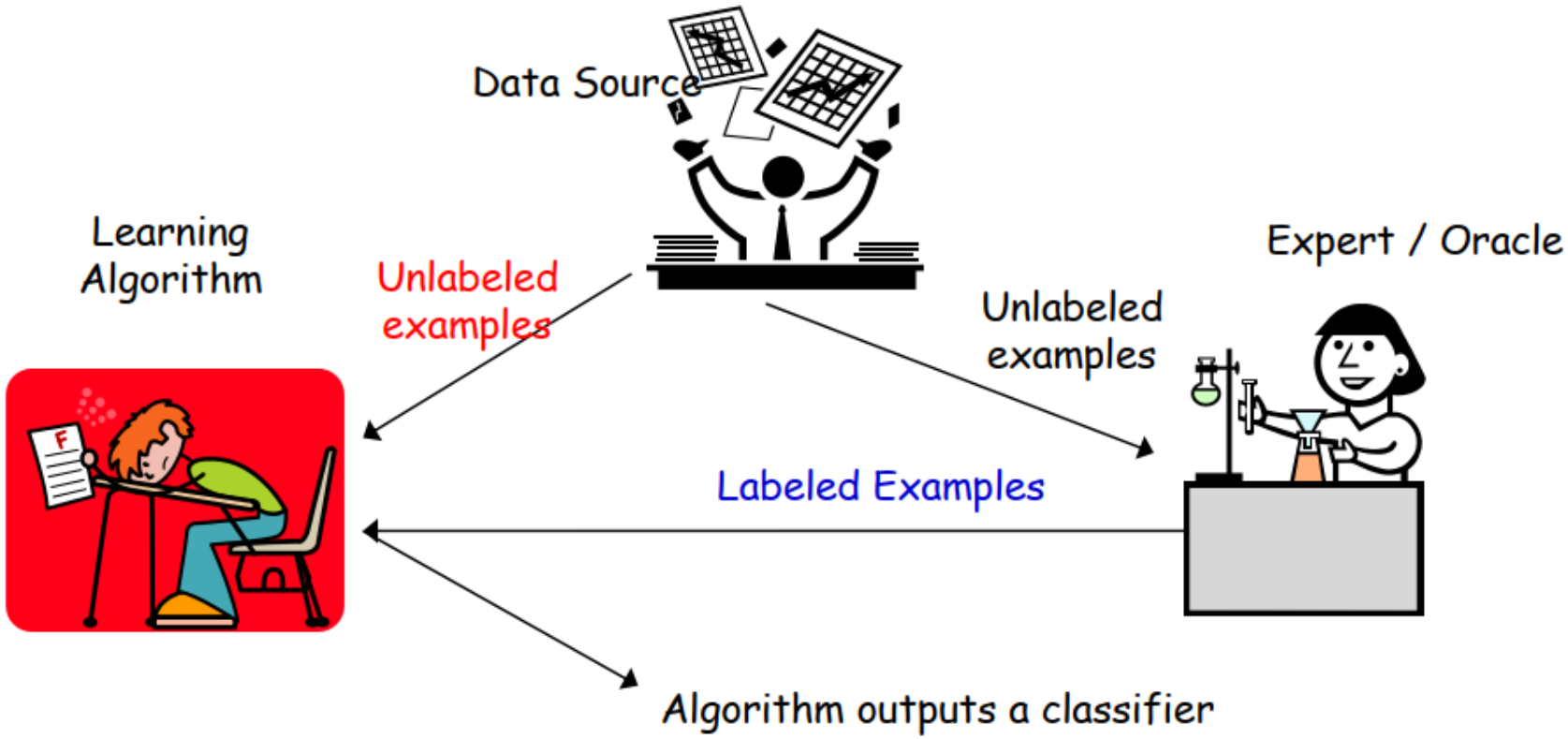
Techniques that best utilize data, minimizing need for expert/human intervention.

Supervised Learning



Semi-Supervised Learning

Semi-Supervised Learning



$S_l=\{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

x_i drawn i.i.d from D, $y_i = c^*(x_i)$

$S_u=\{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from D

Goal: h has small error over D.

$$err_D(h) = \Pr_{x\sim D}(h(x) \neq c^*(x))$$

Semi-Supervised Learning

Major topic of research in ML.

• Several methods have been developed to try to use unlabeled data to improve performance, e.g.:

– Transductive SVM [Joachims '99]

– Co-training [Blum & Mitchell '98]

– Graph-based methods [B&C01], [ZGL03]

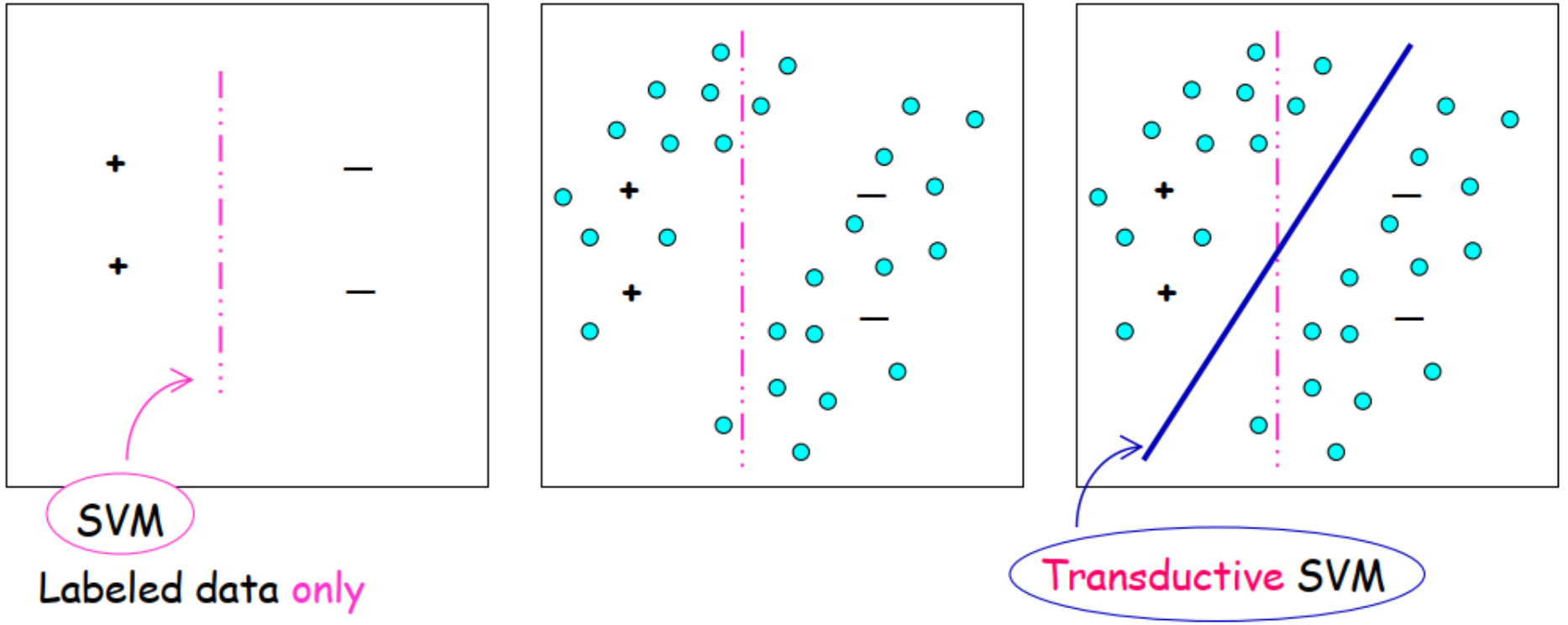
Today: discuss these methods.

Very interesting, they all exploit unlabeled data in different, very interesting and creative ways.

Margins based regularity

Target goes through low density regions (large margin).

- assume we are looking for linear separator
- belief: should exist one with large separation



33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

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

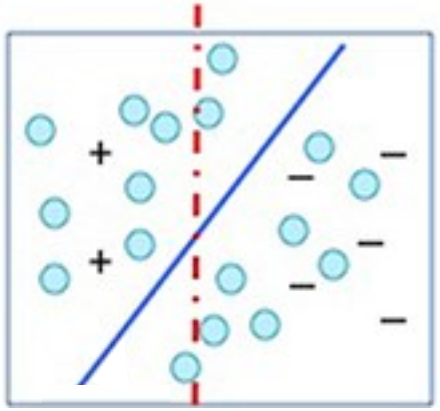
Nicolas Papernot
Google Research
papernot@google.com

Colin Raffel
Google Research
craffel@google.com

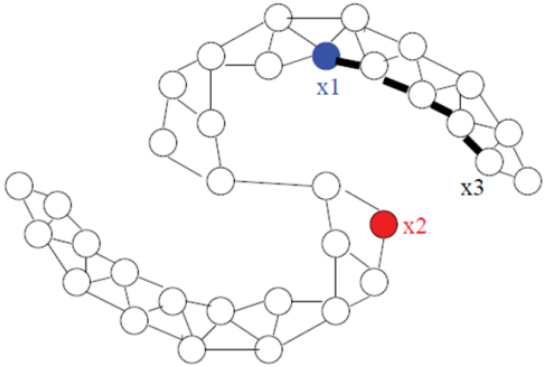
Introduction

Semi-supervised learning (SSL) seeks to largely alleviate the need for labeled data by allowing a model to leverage unlabeled data.

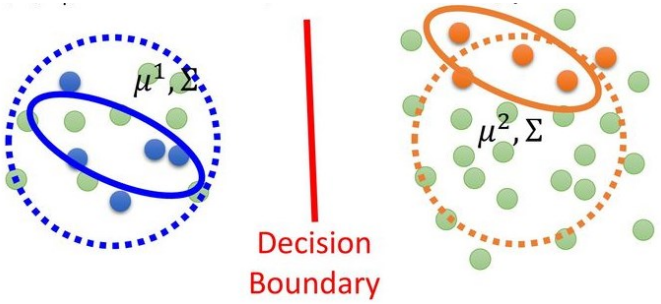
Semi-supervised techniques:



Transductive models



Graph-based methods



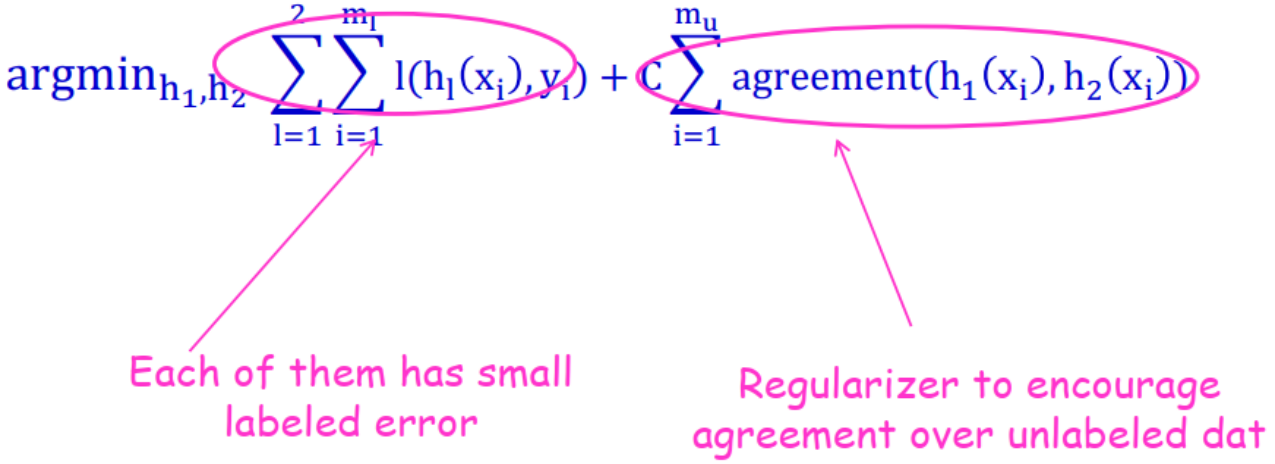
Generative modeling

- Generic Model: A generic model $p_{model}(y \mid x; \theta)$ produces a distribution over class labels y for an input x with parameters θ.

Background

Many recent approaches for semi-supervised learning add a loss term which is computed on unlabeled data and encourages the model to generalize better to unseen data.

Co-Training

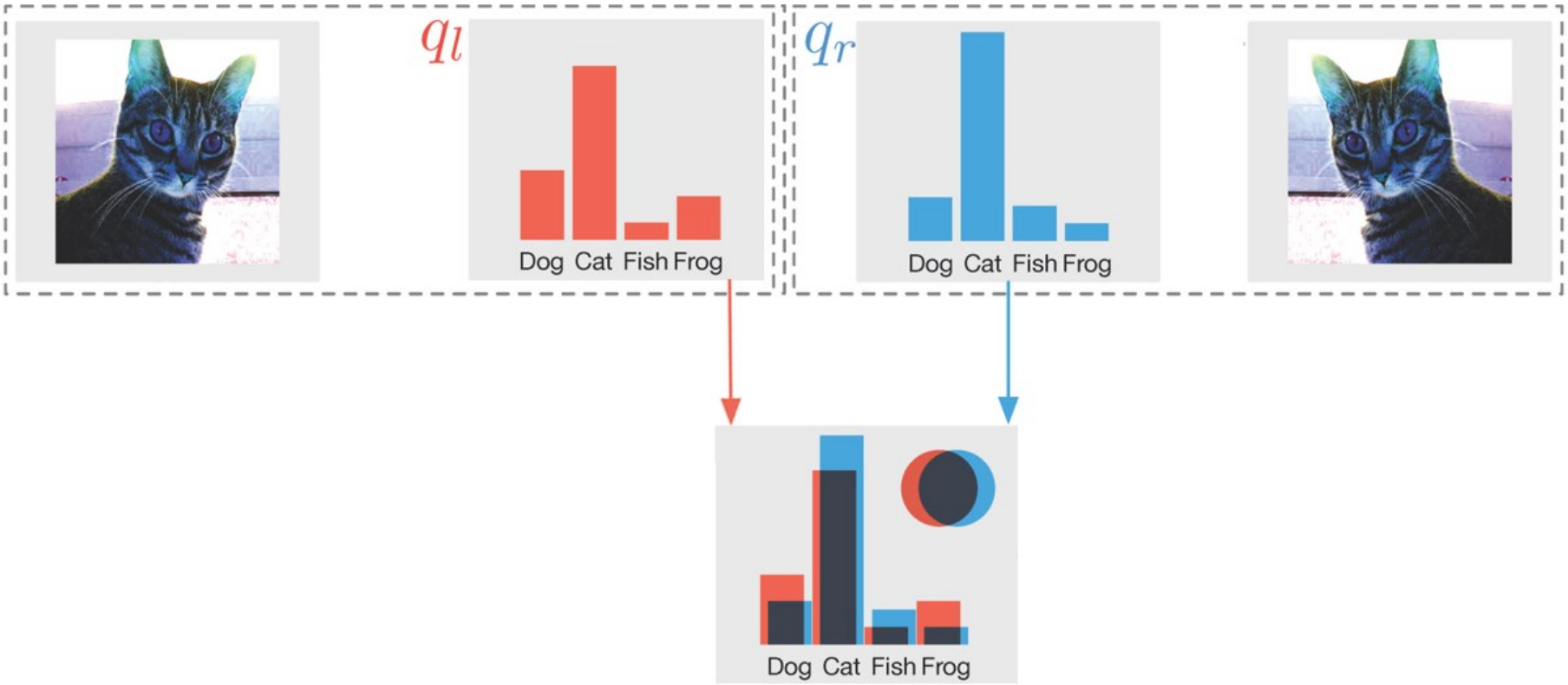


In much recent work, the loss term falls into one of three classes:

- **Entropy minimization** encourages the model to output confident predictions on unlabeled data;
- **Consistency regularization** encourages the model to produce the same output distribution when its inputs are perturbed;
- **Generic regularization** encourages the model to generalize well and avoid overfitting the training data.

Background

1. Consistency Regularization



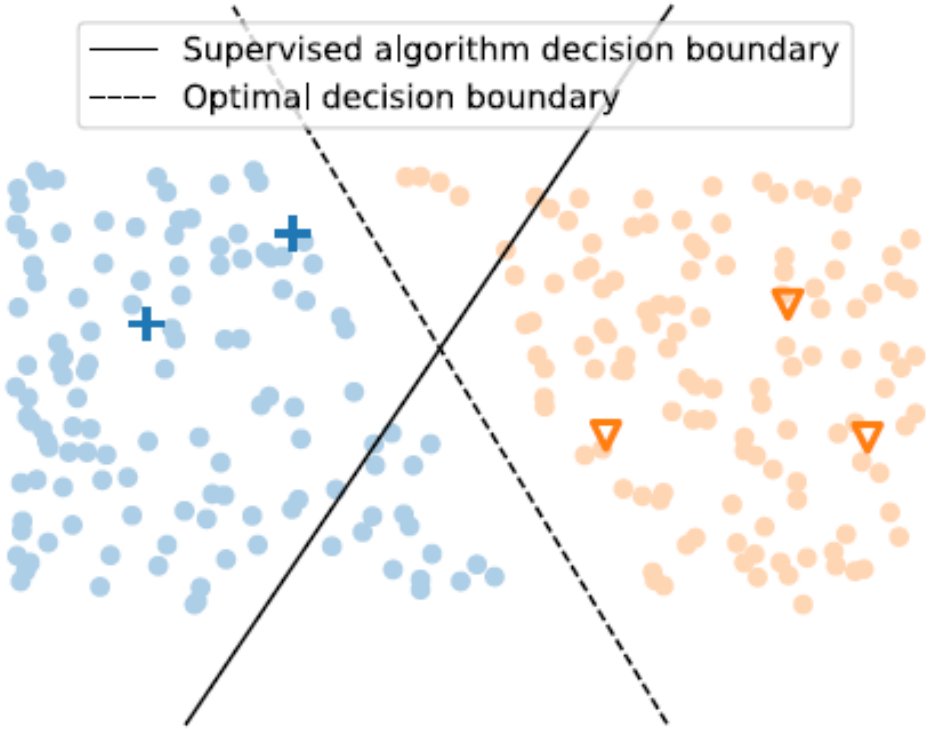
$$\|\mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) - \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta)\|_2^2.$$

Augment(x) is a stochastic transformation, so the two terms are not identical.

Background

2. Entropy Minimization

Density assumption: classifier's decision boundary should not pass through high-density regions.



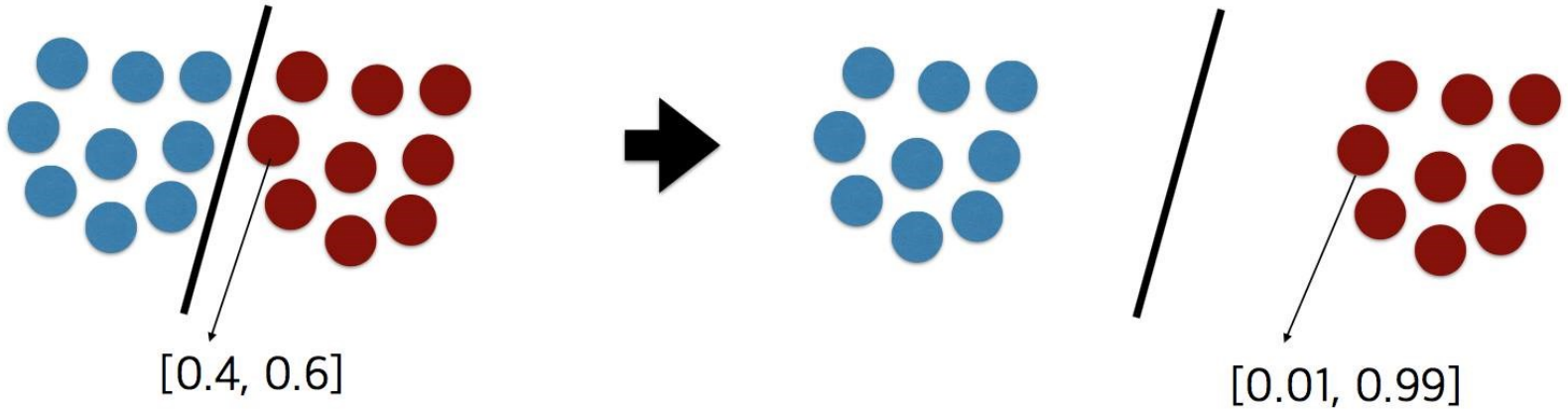
Background

2. Entropy Minimization

- One way to enforce this is to require that the classifier output low-entropy predictions on unlabeled data.

This is done explicitly with a loss term which minimizes the entropy of $p_{model}(y \mid x; \theta)$ for unlabeled data x.

- **Minimize the entropy of unlabeled data.**

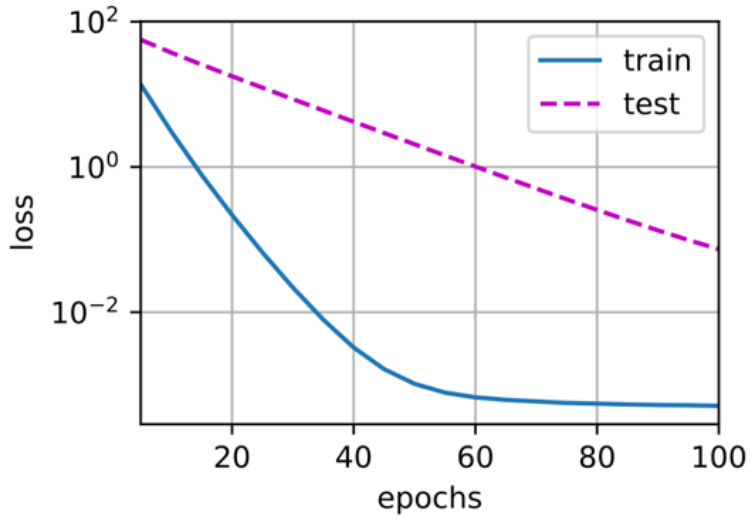


Background

3. Generic Regularization

Regularization refers to the general approach of imposing a constraint on a model to make it harder to memorize the training data and therefore hopefully make it generalize better to unseen data.

We use weight decay which penalizes the L_2 norm of the model parameters.



$$\min_{\theta} \sum_{x,p \in \mathcal{X}} \ell(p, \mathrm{p}_{model}(y|x;\theta)) + \lambda \|\theta\|_2^2$$

MixMatch

MixMatch introduces a unified loss term for unlabeled data that seamlessly reduces entropy while maintaining consistency and remaining compatible with traditional regularization techniques.

MixMatch

MixMatch introduces a unified loss term for unlabeled data that seamlessly reduces entropy while maintaining consistency and remaining compatible with traditional regularization techniques.

Given a batch X of labeled examples with one-hot targets (representing one of L possible labels) and an equally-sized batch U of unlabeled examples.

MixMatch

MixMatch introduces a unified loss term for unlabeled data that seamlessly reduces entropy while maintaining consistency and remaining compatible with traditional regularization techniques.

Given a batch *X* of labeled examples with one-hot targets (representing one of *L* possible labels) and an equally-sized batch *U* of unlabeled examples.

MixMatch produces a processed batch of augmented labeled examples *X'* and a batch of augmented unlabeled examples with "guessed" labels *U'*.

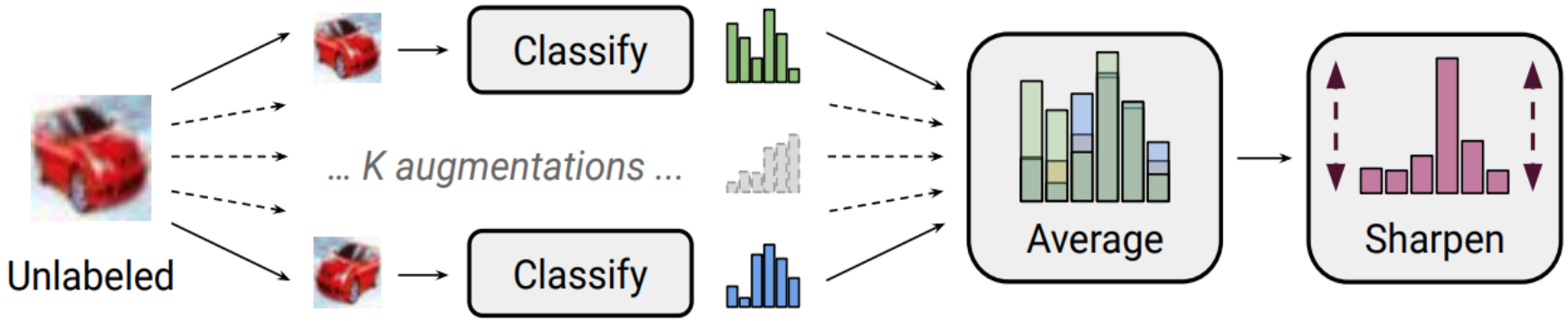
U' and *X'* are then used in computing separate labeled and unlabeled loss terms

MixMatch

1. Data Augmentation

For each unlabeled example in *U*, *MixMatch* produces a “guess” for the example’s label using the model’s predictions.

This guess is later used in the unsupervised loss term.



To do so, we compute the average of the model’s predicted class distributions across all the *K* augmentations of u_b by

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

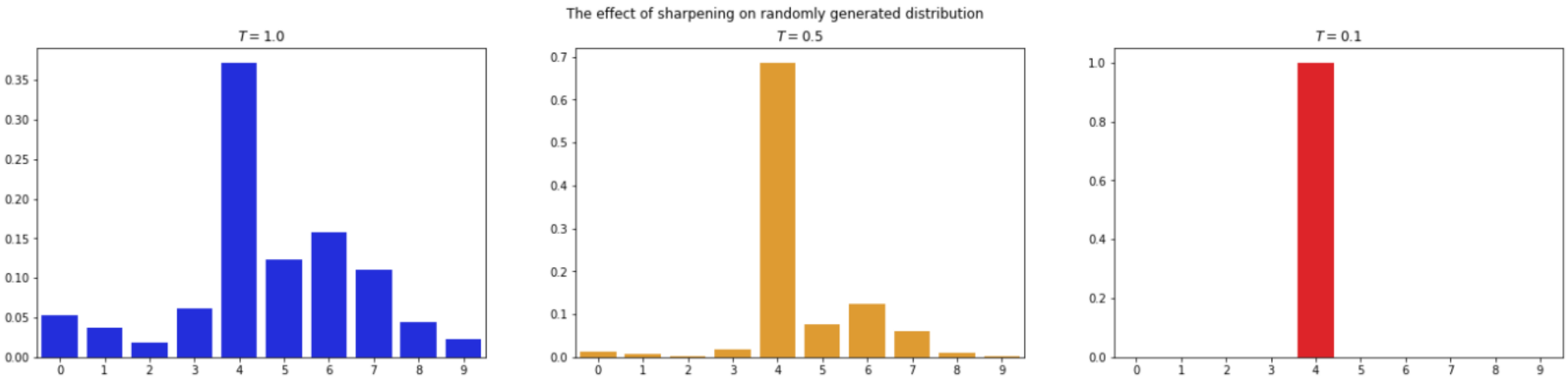
Using data augmentation to obtain an artificial target for an unlabeled example is common in consistency regularization methods.

MixMatch

2. Label Guessing and Sharpening

In generating a label guess, we perform one additional step inspired by the success of entropy minimization in semi-supervised learning.

Given the average prediction over augmentations $\bar{q}_b$, we apply a sharpening function to reduce the entropy of the label distribution.



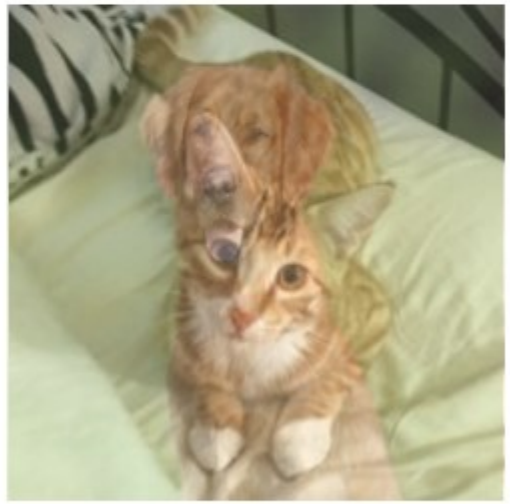
$$\text{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$

MixUp

$$\begin{aligned} \hat{x} &= \lambda x_i + (1-\lambda)x_j, \\ \hat{y} &= \lambda y_i + (1-\lambda)y_j, \end{aligned}$$

where $\lambda \in [0, 1]$ is a random number

Image



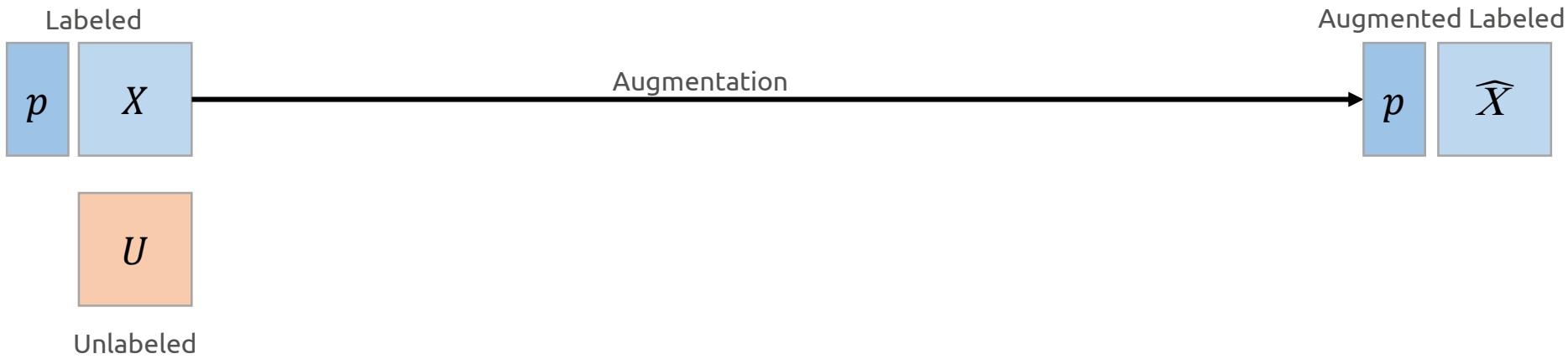
Label

[1.0, 0.0]
cat dog

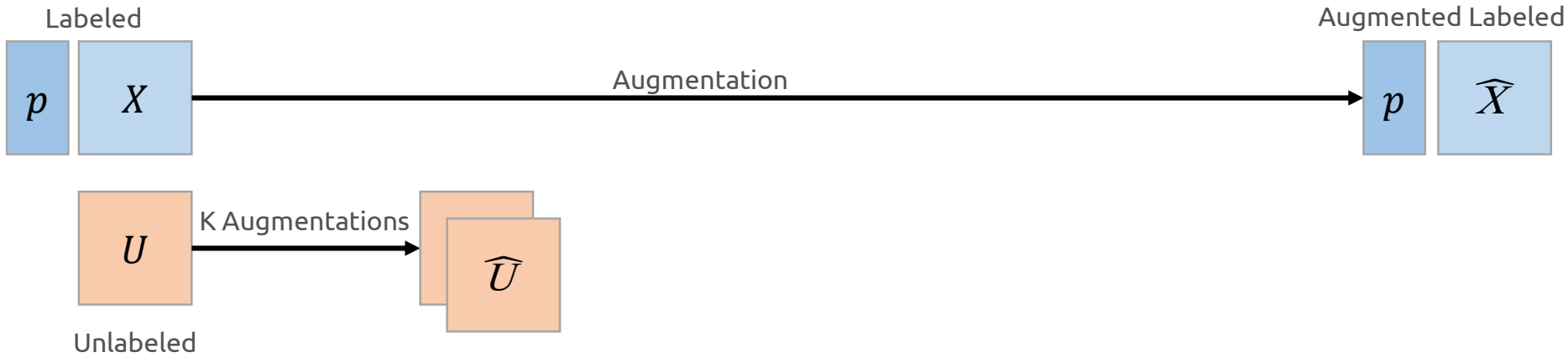
[0.0, 1.0]
cat dog

[0.7, 0.3]
cat dog

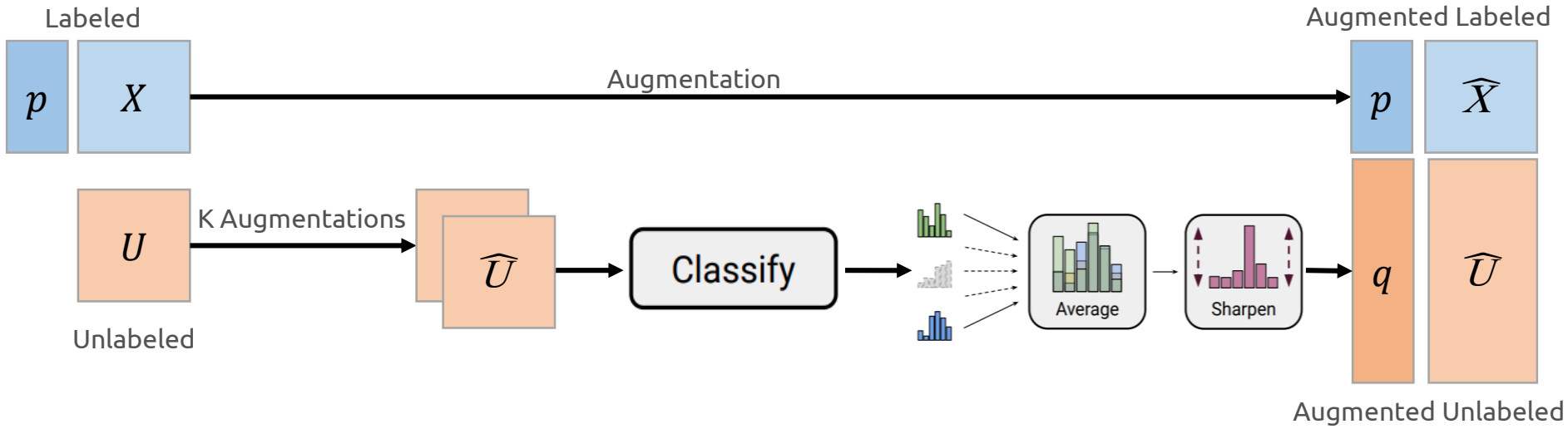
MixMatch Diagram



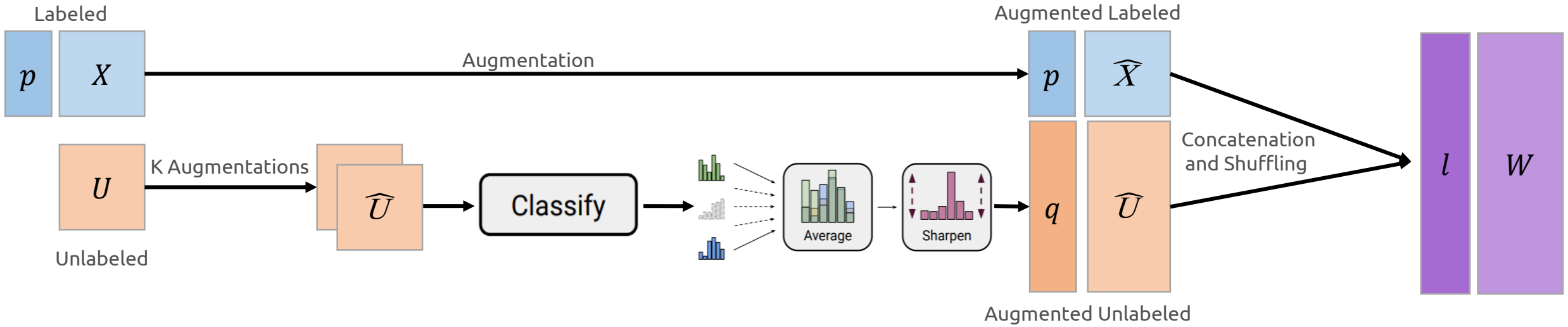
MixMatch Diagram



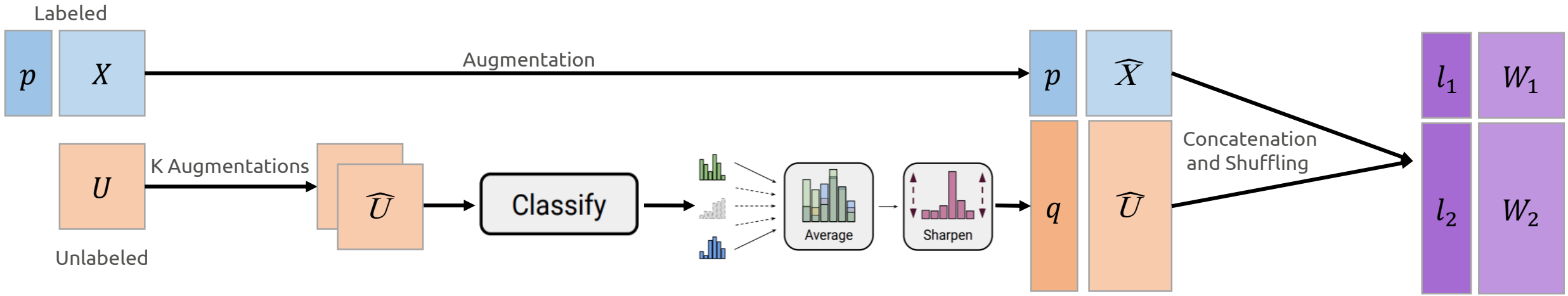
MixMatch Diagram



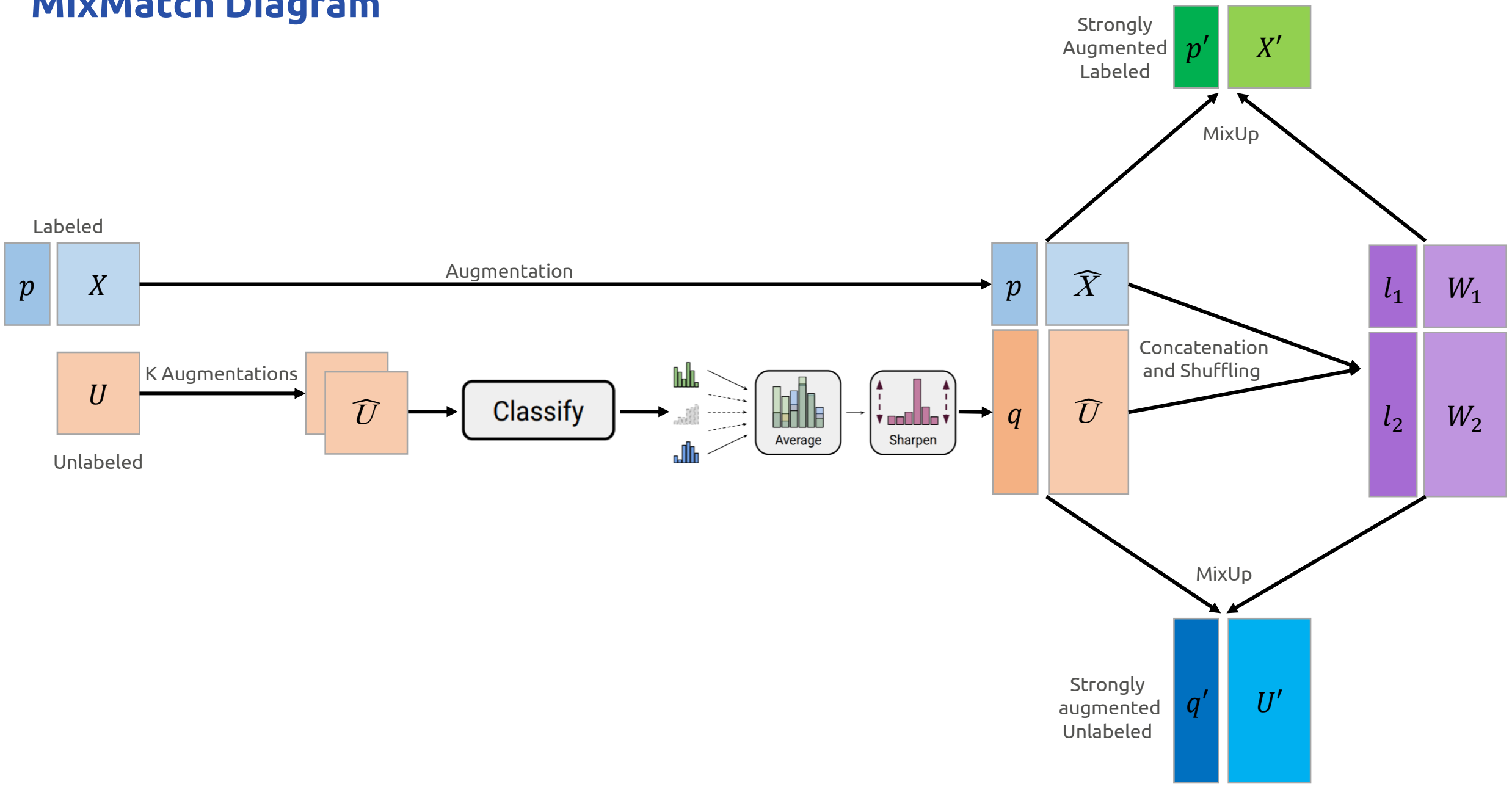
MixMatch Diagram



MixMatch Diagram



MixMatch Diagram



MixMatch

U' and *X'* are then used in computing separate labeled and unlabeled loss terms.

More formally, the combined loss *L* for semi-supervised learning is defined as

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

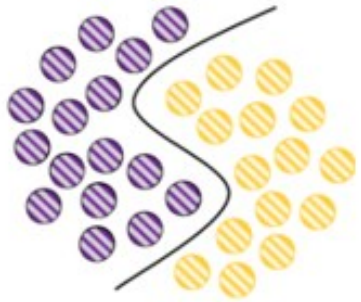
$$\mathcal{L} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta)) + \lambda_{\mathcal{U}} \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2$$

where *H(p, q)* is the cross-entropy between distributions *p* and *q*, and *T*, *K*, α, and *U* are hyperparameters.

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

Self-Supervised Learning

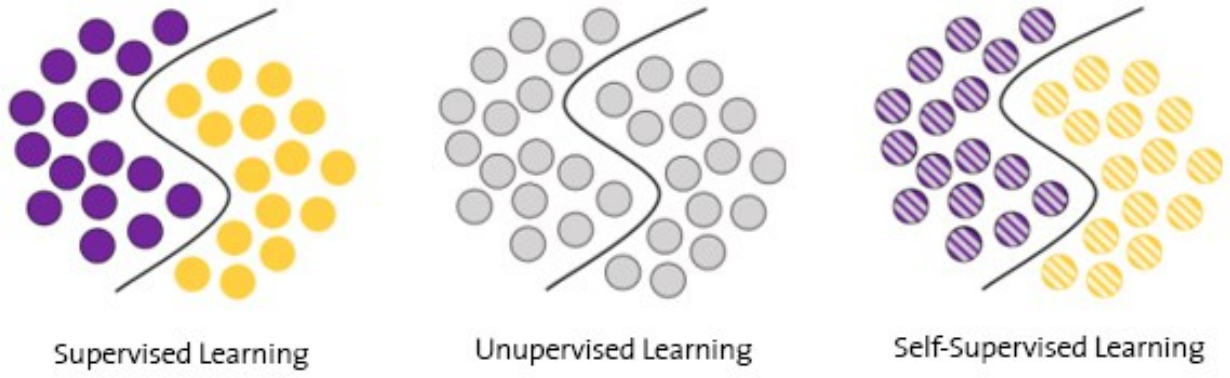


Self-Supervised Learning

- Self-supervised learning learns from unlabeled sample data.
- Self-supervised learning obtains supervisory signals from the data itself.
- It can be regarded as an intermediate form between supervised and unsupervised learning.

The model learns in two steps.

1) The task is solved based on pseudo-labels which help to initialize the network weights.
2) The actual task is performed with supervised or unsupervised learning.



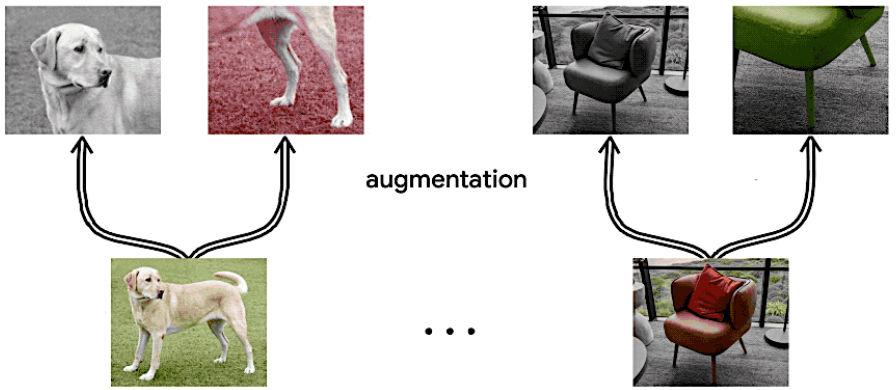
Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart.

. . .

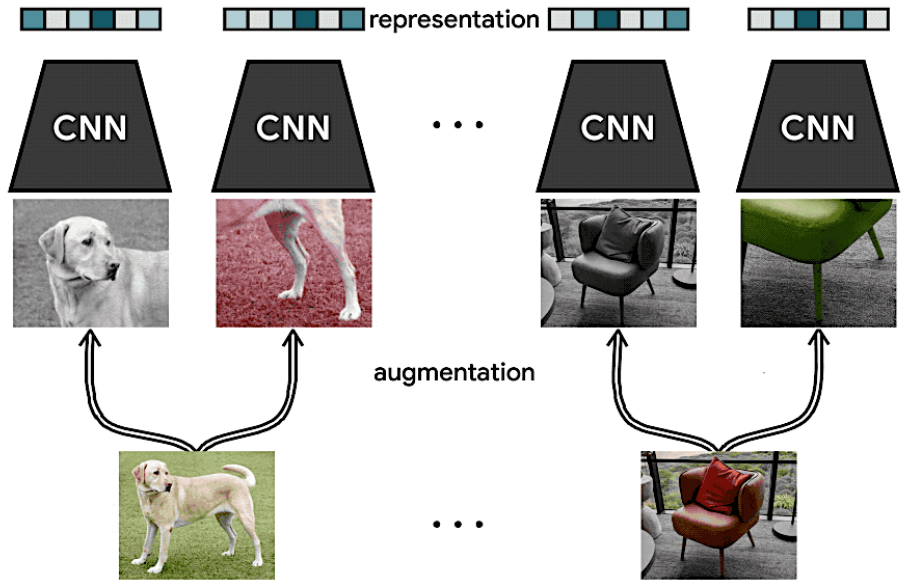
Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart.



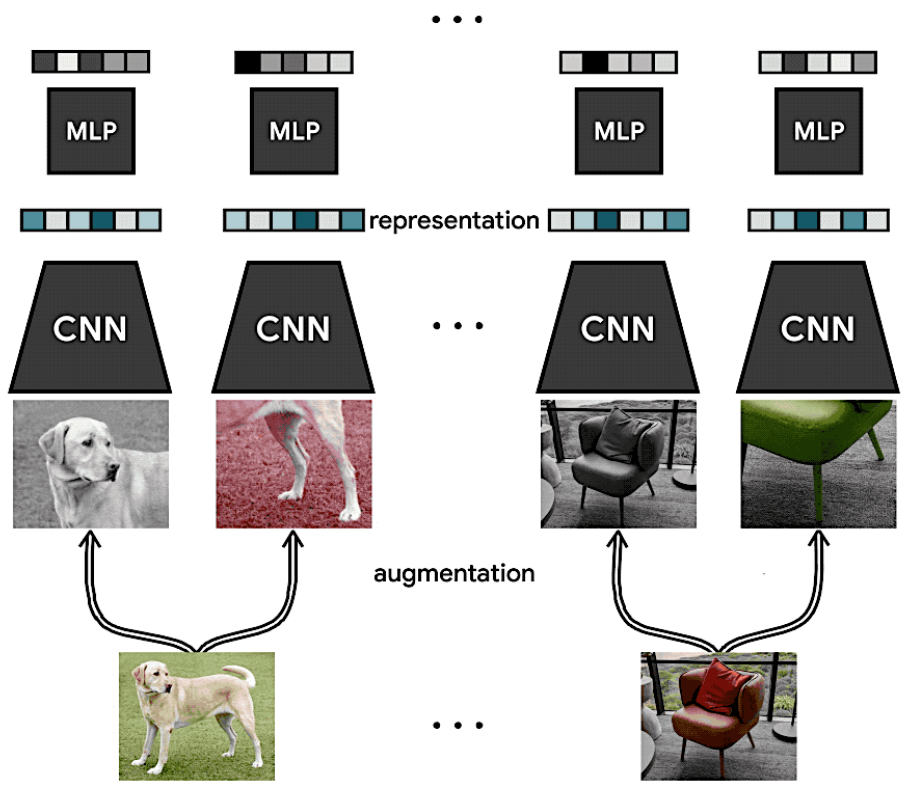
Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart.



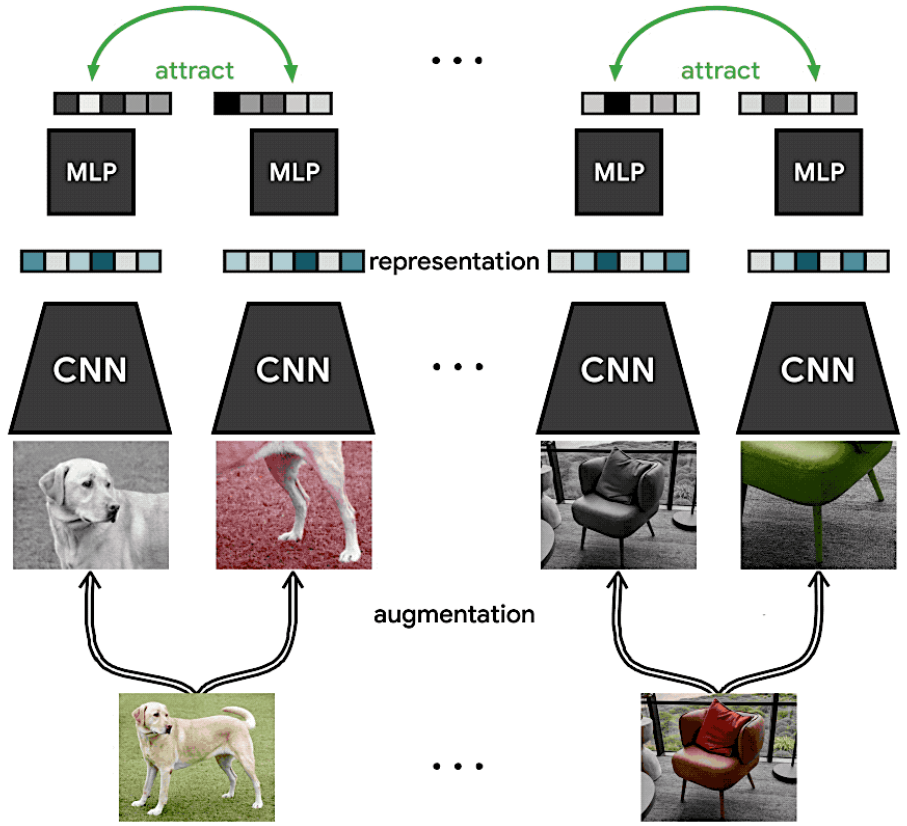
Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart.



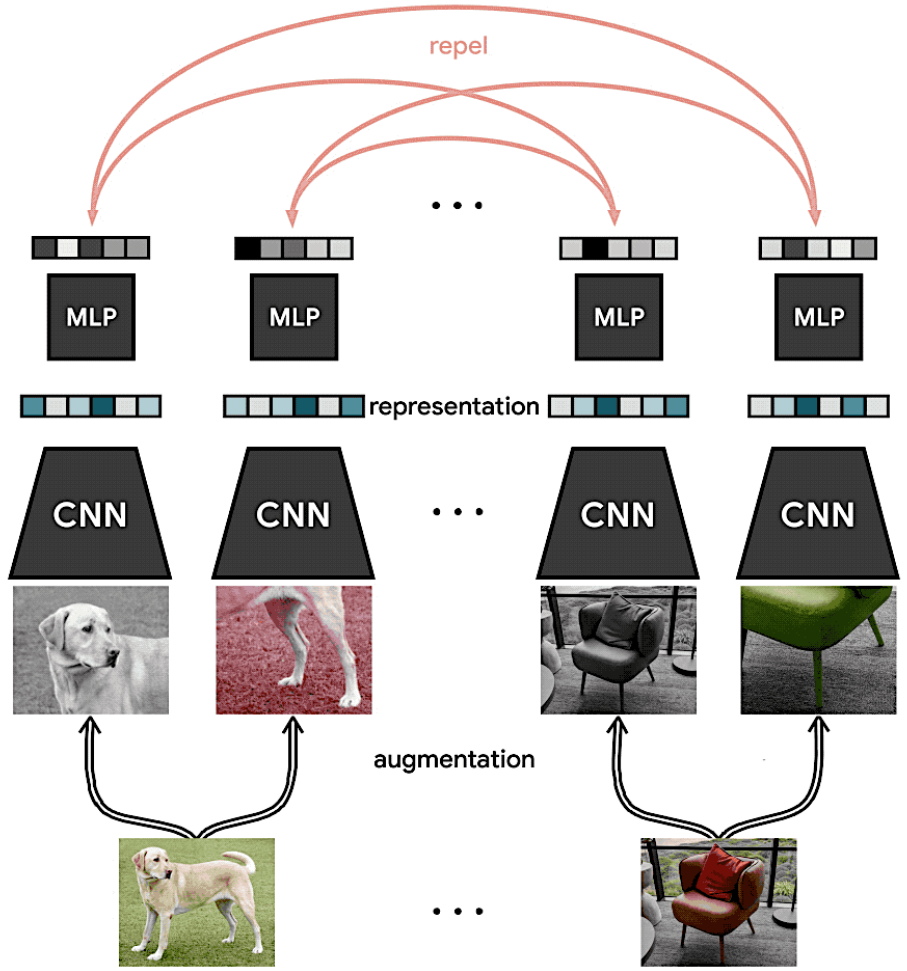
Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart.



Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart.



Self / Semi-Supervised Learning

Self/Semi-Supervised Learning

Self/Semi-Supervised Learning is a semi-supervised learning method that consists of two stages:

1) Self-supervised pre-training based on contrastive learning
2) Semi-supervised finetuning based on augmentation consistency regularization.

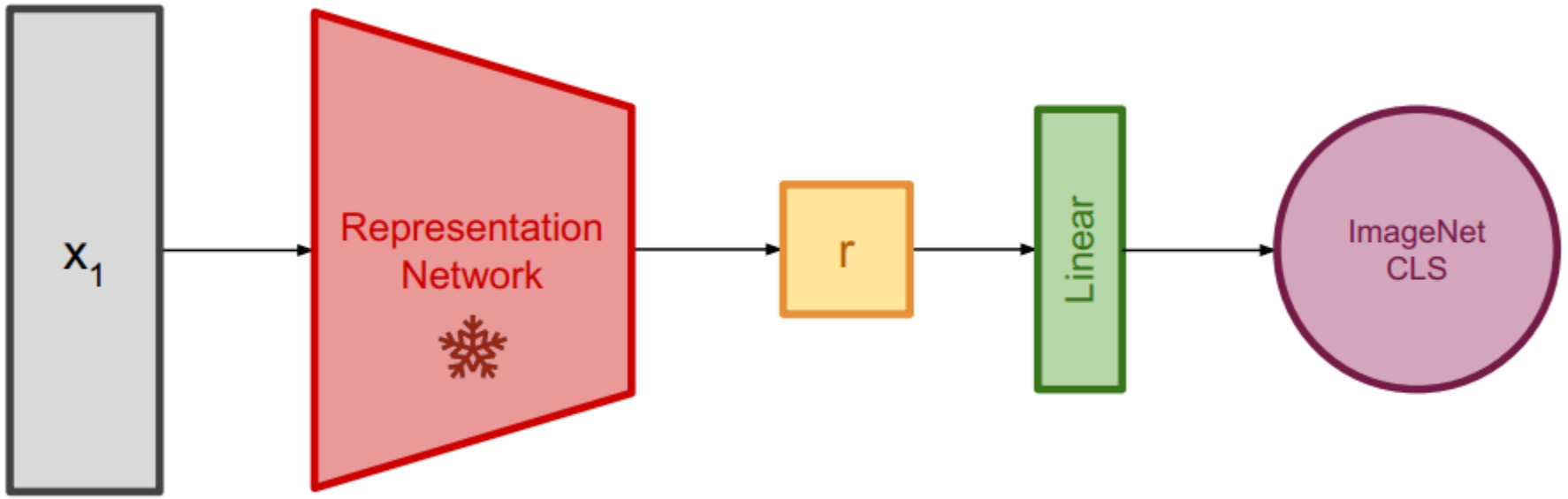
37th International Conference on Machine Learning (ICML2020), Vienna, Austria

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen [1] Simon Kornblith [1] Mohammad Norouzi [1] Geoffrey Hinton [1]

Google Research, Brain Team

Simple framework for Contrastive Learning (SimCLR)



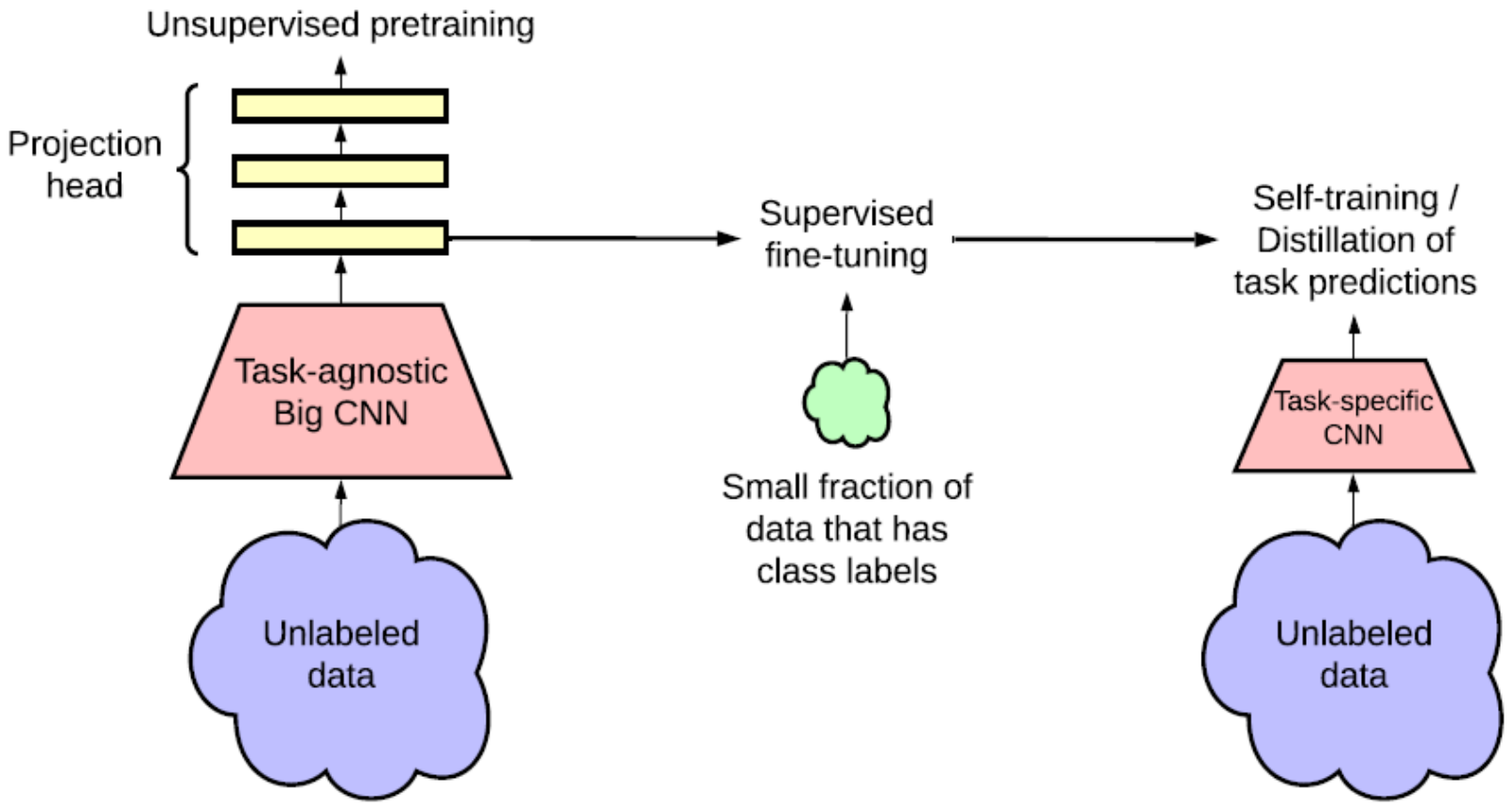
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada

Big Self-Supervised Models are Strong Semi-Supervised Learners

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey Hinton
Google Research, Brain Team

Big Self-Supervised Models are Strong Semi-Supervised Learners (SimCLRv2)

1) Unsupervised or self-supervised pretraining
2) Supervised fine-tuning
3) Distillation using unlabeled data



Other works

ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring
[Google Research]
International Conference on Learning Representations (ICLR 2020)

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence
[Google Research]
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

SelfMatch: Combining Contrastive Self-Supervision and Consistency for Semi-Supervised Learning
[Samsung]
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.