Beyond the Final Linear Layer: Enhancing Decision Boundaries

Michael Majurski
Information Technology Lab, NIST
University of Maryland, Baltimore County
michael.majurski@nist.gov

Sumeet Menon
University of Maryland, Baltimore County

David Chapman
University of Miami

Abstract

[TODO: (majurski) rewrite abstract once paper is done to fill in details] *SSL leverages an abundance of unlabeled data to improve deep learning based model performance under limited training data regimes. This paper presents a novel extension to any image classification architecture which improves accuracy in low-label regimes. We extend the FixMatch [28] training scheme with our novel last layers and demonstrate test accuracy improvement. The novelty consists of 2 elements: first we replace the last linear layer with a GMM trained via backprop, and second, we impose class-wise constraints on the embedding space the GMM operates on. These methods match published SOTA 250 label CIFAR-10 [15] results and come close to matching SOTA in the 40 label regime without the significant model complexity of methods like SimMatchV2 [40]. Our method achieves 94.8% and 94.2% accuracy with 250 and 40 CIFAR-10 labels respectively.* ***[TODO: cleanup the repo]*** *Our code is available at: https://github.com/* ⋆ ***[TODO: insert link for camera ready]***

1. Introduction

SSL leverages an abundance of unlabeled data to improve deep learning based model performance under limited training data regimes [13, 18, 41]. Image classification has become a playground for exploring new SSL ideas. The early successes of deep learning based methods relied on large annotated datasets to enable models to learn the relevant features to perform the task, i.e. image classification build on top of ImageNet [9]. With data annotation becoming a significant bottleneck, especially in application domains outside of the standard benchmarks, another learning paradigm was needed.

There are several flavors of SSL. Contrastive learning methods leverage the intuition that similar instances should be close in the representation space, while different instances are farther apart [17, 34]. Consistency regularization borrows the intuition that modified views of the same instance should have similar representations and predictions [14, 16, 28, 38]. Pseudo-labeling methods like FixMatch [28] fall within the consistency regularization domain.

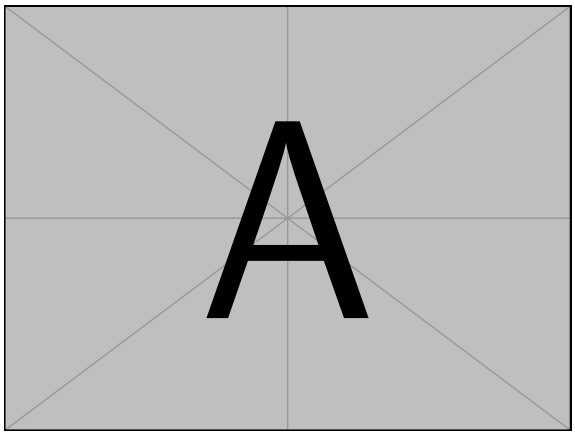

Figure 1. High level overview of our method. **[TODO: (majurski) Create this figure.]**

This work argues that pseudo-labeling methods can be improved with better calibration of the network logits used to filter the pseudo-labels into reliable and unreliable. Neural networks are known to be overconfident in their predictions [31], and this affects the pseudo-labeling process. Potentially allowing for the inclusion of more incorrect pseudo-labels any specific logit threshold would otherwise have. This work demonstrates the better calibrated replacements for a models final linear layer can improve the final accuracy of pseudo-labeling based SSL algorithms in very label scarce regimes. This work proposes:

1. Replacing the final linear (fully connected) layer of the neural network with either kmeans [10] or axis-aligned differentiable Gaussian Mixture Model (AAGMM)

trained via back prop, both of which have explicit modeling of class cluster centroids.
2. We explore various constraints on how the embedding space should be structured by adding penalties if the per-class clustering does not conform to between 0 and 4 of the first gaussian moments being identity/zero.
3. We demonstrate that increasing the specificity of how the embedding space should be structure negatively impacts model performance. [24]

This paper explores the impacts of replacing the final linear layer of a network with a generative model and embedding space constraints. This combination demonstrates improvements in final model accuracy when using pseudo-labeling methods where very few annotations are available. We demonstrate this methodology using the standard CIFAR-10 (40 and 250 labels) and CIFAR-100 (400 and 2500 label) benchmarks [15]. Additionally, we explore and demonstrate that high level prescriptive constraints on the embedding space produce significantly worse outcomes than allowing the embedding space to take on whatever emergent structure the training process produces. Finally, because the embedding constraint penalties are applied to all unlabeled data and not just the valid pseudo-labels, our method extracts training signal from every unlabeled data point, unlike FixMatch [28] and other methods which only learn from the valid pseudo-labels.

[TODO: (majurski) build t-SNE plots of the embedding spaces for the best models (1 per configuration)]

[TODO: (majurski) purge bibliography of arxiv preprints where possible, replacing with their peer reviewed equivalents]

2. Related Work

Semi-Supervised learning has shown great progress in learning high quality models, in some cases matching fully supervised performance for a number of benchmarks [38]. The goal of SSL is to produce a trained model of equivalent accuracy to fully supervised training, with vastly reduced data annotation requirements.

2.1. Pseudo-Labeling

Self-supervised learning was among the initial approaches employed in the context of semi-supervised learning to annotate unlabeled images. This technique involves the initial training of a classifier with a limited set of labeled samples and incorporates pseudo-labels into the gradient descent process, exceeding a predefined threshold [8, 19, 20, 22, 27, 35, 37]. A closely related method to self-training is co-training, where a given dataset is represented as two distinct feature sets [4]. These independent sample sets are subsequently trained separately using two distinct models, and the sample predictions surpassing predetermined thresholds are utilized in the final model training process [4, 25]. A notably advanced approach to pseudo-labeling is the Mean Teacher algorithm [29], which leverages exponential moving averages of model parameters to acquire a notably more stable target prediction. This refinement has a substantial impact on enhancing the convergence of the algorithm.

2.2. Consistency Regularization

Consistency regularization operates on the premise that when augmenting an unlabeled sample, its label should remain consistent. This approach implicitly enforces a smoothness assumption, promoting coherence between unlabeled samples and their basic augmentations [32]. In other words, the model should be able to predict the unlabeled sample x exactly the same way it predicts the class for $Augmented(x)$ [2, 3, 23, 28]. In addition to evaluating image-wise augmentations, recent research has demonstrated that incorporating class-wise and instance-based consistencies yields superior performance outcomes [17, 39]. Similarly, using consistencies between augmentations, of the predictions and low-dimensional embeddings of the strong and weak augmentations of the unlabeled images in a graph based setup has shown improvement over class-wise and instance-based consistencies [40]. Finally, pseudo-labeling filtering based on consistence between strongly augmented views, gaussian filtering and embedding based nearest neighbor filtering shows convergence improvement [14, 21].

2.3. Embedding Clustering/Constraints

Several papers have attempted to enhance the quality of pseudo-labels to either improve the final model accuracy, improve the rate of convergence, or avoid confirmation bias [1]. Rizve et al. [26] explores how uncertainty aware pseudo-label selection/filtering can be used to reduce the label noise. Incorrect pseudo-labels can be viewed as a network calibration issue [26] where better network logit calibration might improve results [33]. Other work has attempted to improve the pseudo-labeling process by imposing curriculum [38] or by including a class-aware contrastive term [34]. Previous work has leveraged the concept of explicit class cluster centers for conditioning semantic similarity [39]. Recent work has extended purely clustering based methods like DINO [7] into semi-supervised methods [12].

[TODO: (chapman) find and include citations you were reading about embedding constraints. You mentioned this in the meeting on the 17th.]

[TODO: (majurski) compute the delta in PL accuracy for two runs with same seed. Base Fixmatch, and our best.]

3. Methodology

In this section, we explore our proposed replacement final layers and our embedding space constraints. FixMatch [28] is a simple, well performing, SSL algorithm. As such, it serves as a good comparison point for exploring the effect of our contributions. Our methodology is based upon the published FixMatch [28] algorithm, with identical hyper-parameters unless otherwise stated. We extend FixMatch with an early-stopping condition when the model has not improved for 40 epochs (where epoch size is defined as 1024 batches), and with 2 learning rate reductions by a factor of 0.2 instead of configuring a fixed number of training epochs with a cosine learning rate decay. Additionally, we employ a cyclic learning rate scheduler to vary the learning rate by a factor of ± 2.0 within each epoch to make training less dependent upon exact learning rate value.

Both the linear layer replacements and the embedding constraints explored herein represent increasing levels of prescription about how the final embedding space should be arranged compared to a traditional linear layer. The idea of leveraging clusters in embedding space is not new [5, 6, 11], but we extend the core idea with a novel differentiable model with learned cluster centroids and GMMs.

3.1. Alternative Final Layers

A limitation of traditional final activation layers such as linear+softmax is that they are fully discriminative; i.e. they estimate the posterior $p(Y|X)$, but do not attempt to model the sample distribution $p(X)$ or the joint probabilities $p(Y, X)$. To overcome this limitation, we present two semi-parametric final activation layers (a) the Axis Aligned GMM (AAGMM) layer, and (b) an equal variance version of AAGMM that we henceforth call the KMeans activation layer due to the similarity of the objective function with a gradient based KMeans.

These activation layers are fully differentiable and integrated into the neural network architecture as a module in the same way as a traditional final linear layer. As such, they do not require or depend on external training and do not use expectation maximization. They are drop in replacements for the final linear layer.

Importantly, these activation layers exhibit both discriminative and generative properties. The neural network model $F(X;\theta_F)$ transforms the data X into a latent space $Z = F(X;\theta_F)$, and the final activation layer estimates the probability densities $p(X)$, $p(Y;X)$ and $p(Y|X)$ by fitting a parametric model to the latent representation Z.

3.1.1 Axis Aligned Gaussian Mixture Model Layer

The AAGMM layer defines a set of K trainable clusters, one cluster per label category. Each cluster $k = 1 \dots K$ has a cluster center μ_k and cluster covariance Σ_k. The prior probability of any given sample X_i is defined by the mixture of cluster probability densities over the latent representation Z_i as follows,

$$p(X_i) = \sum_{k=1}^{K} \mathcal{N}(Z_i, \mu_k, \Sigma_k) \quad (1)$$

$$\textit{where} \quad Z_i = F(X_i, \theta_F)$$

Where $\mathcal{N}(Z_i, \mu_k, \Sigma_k)$ represents the multivariate gaussian pdf with centroid μ_k and covariance Σ_k. AAGMM is axis aligned because Σ_k is a diagonal matrix, as such the axis-aligned multivariate normal pdf simplifies to the marginal product of Gaussians along each of the D axes as follows,

$$\mathcal{N}(X_i, \mu_k, \Sigma_k) = \prod_{d=1}^{D} \frac{1}{\sigma_{k,d}\sqrt{2\pi}} exp\Big(\frac{Z_{i,d} - \mu_{k,d}}{\sigma_{k,d}}\Big)^2 \quad (2)$$

$$\textit{where} \quad \sigma^2_{k,d} = \Sigma_{k,d,d}$$

As there is one cluster per label category, the joint probability for sample i with label assignment k, $p(Y_{i,k}, X_i)$ is the given by the normal pdf of the k^{th} cluster,

$$p(Y_{i,k}, X_i) = \mathcal{N}(Z_i, \mu_k, \Sigma_k) \quad (3)$$

By simple Bayesian identity, the posterior probability $\hat{Y}_k = p(Y_k|X_i)$ can therefore be inferred from eq 1 and 3 as follows,

$$\hat{Y}_{i,k} = p(Y_{i,k}|X_i) = \frac{p(Y_{i,k}, X_i)}{p(X_i)} \quad (4)$$

3.1.2 KMeans Layer

We also implement a KMeans final layer which is a more restrictive form of the AAGMM layer, in the sense that we impose an additional constraint that the gaussian covariance matrix Σ_k for each cluster center k is the $[D \times D]$ identity matrix. This constraint yields spherical cluster centers similar to how the traditional KMeans algorithm also assumes spherical clusters.

3.2. Method of Moments Embedding Constraints

We also introduce and evaluate a series of embedding constraints based on the Method of Moments (MoM) [24] in order to fit the parameters of our semi-parametric latent prior. As such, we calculate the latent prior $p(X_i)$ as in equation 1, which is then used to infer the posterior $p(Y_{i,k}|X_i)$. As usual, the posterior is trained using cross entropy loss. But, if one were to omit embedding constraints, then it is possible that the model could learn a good decision boundary for the posterior but without actually modeling the latent prior.

Our model is semi-parametric, because the prior is a parametric model of the latent distribution Z which is the result of a neural network feature extraction $F(X;\theta)$. As such, attempting to fit the GMM directly to Z using Maximum Likelihood (ML) or simple Expectation Maximization (EM), is not appropriate, because doing so would fail to learn an appropriate feature space for discrimination. MoM solves these problems and is an appropriate strategy for semi-parametric models including ours.

The MoM relies on the use of *consistent estimators*, as these asymptotically share sample and population statistics. Assume that z is a finite sample of n elements drawn from infinite population Z, then a series of P well-behaved sample statistics g_p should very closely approximate their k population statistic as follows,

$$\forall p = 1 \dots P \quad \frac{1}{n}\sum_{i=1}^{n} g_p(z_i) \approx E(g_p(Z)) \tag{5}$$

We can therefore constrain the latent representation of our model to approximate an independent joint Gaussian distribution. In the univariate gaussian case, the p^{th} order centralized moment constraint is the following.

$$E\left[(Z-\mu)^p\right] = \begin{cases} 0 & \text{if } p \text{ is odd} \\ \sigma^p (p-1)!! & \text{if } p \text{ is even} \end{cases} \tag{6}$$

By this formula, the univariate unit gaussian has mean 0, standard deviation 1, skew 0, and kurtosis 3.

In the joint multivariate case, each dimension is independent by definition. As such, if we redefine Z, μ, and p to be all D dimensional, then the centralized joint gaussian moment can be defined as follows,

$$E\left[g_p(Z-\mu)\right] = E\left[\prod_{d=1}^{D} (Z_d - \mu_d)_d^p\right] \tag{7}$$

Due to independence of the axes, this moment can be represented as a product of univariate moments of the individual gaussians as follows,

$$E\left[\prod_{d=1}^{D} (Z_d - \mu_d)_d^p\right] = \prod_{d=1}^{D} E\left[(Z_d - \mu_d)_d^p\right] \tag{8}$$

The error (loss) term associated with the embedding constraint for any moment p is equal to the L2 difference between the sample and population statistics as follows,

$$\varepsilon_p = \left(\frac{1}{n}\sum_{i=1}^{n} g_p(z_i) - E(g_p(Z))\right)^2 \tag{9}$$

Some moments are more important than others, and must be weighted more heavily. For example, first order moments are simply the sample mean, and should be given the greatest weight as an embedding constraint. The second order moments form a sample covariance matrix, which ideally should be equal to the identity matrix, but the diagonal terms should be given greater weight than the off-diagonal terms. This is because, in a $D \times D$ covariance matrix, there are $D(D-1)$ off diagonal terms, but only D, diagonal terms. The p^{th} order sample moments form a $p-1$ dimensional hyper-covariance matrix, with terms residing on the intersection of anywhere between 0 and $p-1$ hyper-diagonals. In order to prevent over-representation of off-diagonal terms, and encourage representation of on-diagonal terms, the loss function for any given moment term is inversely proportional to the number moment terms that share the same number of hyper-diagonals. This heuristic weighting scheme ensures that the overall contribution of each moment order is not overly influenced by the off-diagonal terms, and that the error weighting is therefore diagonally dominant.

4. Experiments

We evaluate our linear layer replacement and embedding space constraints using our modified FixMatch on common SSL benchmarks CIFAR-10/100 [15] under various label scarcities. When comparing against other SSL benchmark results (like SimMatch [39]) it is unclear how the labeled samples were selected from the fully labeled dataset. For our evaluation, for each model training run, the required number of labeled samples are drawn without replacement from the training population of the dataset. All data not in this labeled subset is used as unlabeled data (i.e. the labels are discarded). We evaluate our method against CIFAR-10 at 40 and 250 labels, and CIFAR-100 at 400 and 2500 labels. This corresponds to 4 or 25 samples per class. As prior work [28] has noted, resulting model quality is highly variable when only 4 samples are selected per class, as the quality and usefulness of the specific 4 samples can vary drastically. It can be informative to compare mean performance with max performance over N runs to see how well a method can be expected to do on average with random label sampling, vs how well it can potentially do with a more representative subset of labeled data.

4.1. Hyper-Parameters

Our training configuration for both CIFAR-10 and CIFAR-100 is identical with the exception of the model architecture. For CIFAR-10 we a WideResNet28-2, whereas the CIFAR-100 models use WideResNet28-8. Its worth noting for the record that these models, drawn from prior SSL work like FixMatch, do not match the original publication definition [36]. Somewhere along the paper chain, an additional

CIFAR-10 Mean Test Accuracy

Last Layer	Emb Dim	40 Labels (6 trials)				
Embedding Constraint		None	$1st$ Order	$2nd$ Order	$3rd$ Order	$4th$ Order
FixMatch[28] (i.e. FullyConnected)	128	83.23 $\pm$3.74				
	32	82.87 $\pm$2.56				
	8	$xx.yy$ $\pm z.zz$				
KMeans	128	84.65 $\pm$4.86	87.66 $\pm$4.45	70.94 $\pm$2.51		
	32	86.90 $\pm$5.38	**91.58** $\pm$1.39	75.24 $\pm$3.35		
	8	85.22 $\pm$4.53	87.97 $\pm$3.04	68.21 $\pm$3.09	$xx.yy$ $\pm z.zz$	$xx.yy$ $\pm z.zz$
AAGMM	128	88.21 $\pm$3.32	88.83 $\pm$4.38	85.29 $\pm$2.67		
	32	85.31 $\pm$6.31	86.89 $\pm$3.44	84.47 $\pm$2.38		
	8	87.44 $\pm$3.44	88.37 $\pm$5.21	81.59 $\pm$3.23	$xx.yy$ $\pm z.zz$	$xx.yy$ $\pm z.zz$
Last Layer	Emb Dim	250 Labels (6 trials)				
Embedding Constraint		None	$1st$ Order	$2nd$ Order	$3rd$ Order	$4th$ Order
FixMatch[28] (i.e. FullyConnected)	128	94.73 $\pm$0.18				
	32	94.07 $\pm$0.96				
	8	94.70 $\pm$0.13				
KMeans	128	94.56 $\pm$0.17	94.40 $\pm$0.18	89.77 $\pm$0.47		
	32	94.66 $\pm$0.27	94.32 $\pm$0.39	91.56 $\pm$0.16		
	8	**[TODO: update]** 94.25 $\pm$0.16	**[TODO: update]** 94.09 $\pm$0.56	89.77 $\pm$1.25	90.85 $\pm$0.76	88.8 $\pm$0.49
AAGMM	128	94.25 $\pm$0.48	**94.75** $\pm$0.29	93.05 $\pm$0.62		
	32	94.07 $\pm$0.76	94.42 $\pm$0.54	93.89 $\pm$0.62		
	8	94.36 $\pm$0.40	94.42 $\pm$0.21	92.97 $\pm$0.99	93.87 $\pm$0.46	93.29 $\pm$0.59

Table 1. Mean test accuracy % for CIFAR-10 SSL benchmark comparing various configurations of our method. The FixMatch results in the table is our reproduction of the published results, using our training pipeline modifications, which verifies the original published results. For CIFAR-10 the WideResNet model used by FixMatch has an embedding size of 128 dimension. Due to exponential GPU memory requirements only the 8D embedding can operate with higher order MoM embedding constraints. Results for a given order of embedding constraint include all lower constraints.

convolutional block was added, making the models perform better than the originally published version would.

Matching the FixMatch [28] hyper-parameters, we use SGD with Nesterov momentum and $\lambda_u = 1$, $\beta = 0.9$, $\tau = 0.95$, $\mu = 7$, $B = 64$, and epoch size = 1024 regardless of the number of images in the labeled dataset. Our $learning\_rate(\eta) = 0.01$. The replacement of a fixed number of training steps with an early stopping criteria, where the model does not improve by $1e-3$ for 40 epochs, prevents the use of a cosine decay schedule. Therefore, we replaced that with a plateau learning rate scheduler which multiplies the learning rate by 0.2 every time the early stopping criteria is met (before being reset) for a max of 2 reductions. Finally, out cyclic learning rate schedule within each epoch varies the learning rate by a factor of ± 2.0.

As part of this work we explore how various embedding dimensionalities affect the generative model linear layer replacement. As such we modify the model architectures with a single additional linear layer before the output to project the base model embedding dimension (128 for WideResNes28-2 and 512 for WideResNes28-8) down to the required 32 or 8 dimensions. We always evaluate without that linear projection. All results include rows with 128 or 512 embedding dimension, which does not require the dimensionality projection.

Finally, due to the complexity in computing the the AAGMM layer, as well as any MoM greater that $1st$ order, we employ gradient clipping by norm $= 1.0$ to stability training. We did notice during our experimentation that when gradient clipping was not necessary to ensure the loss didn't go to NaN, it reduced the final model accuracy. Despite computing the AAGMM and embedding constraints as numerically stable as we could, they are still less stable during backprop than a simple linear layer.

4.2. CIFAR-10

The CIFAR-10 SSL benchmark was used to explore the full configuration space of our method. While both 40 and 250 label counts were used, the 250 label case is a solved problem, included just to document our performance is approximately equivalent to SOTA. The 40 label case provide a far more challenging task, though recent results have claimed

to match fully-supervised performance on CIFAR-10 (similar to 250 label CIFAR-10).

Table 1 summarizes the relative performance of our various configurations for both 40 and 250 labels. We reproduced the FixMatch [28] using our hyper-parameters and ended up not quite matching the published FixMatch performance at 40 labels (250 labels matched). So its likely our selection of hyper-parameters is sub-optimal FixMatch. **[TODO: (majurski) run OG fixmatch hyper-parameters including cosine and LR to see how we turn out]** The FixMatch rows in table 1 also represent the baseline fully connected linear last layer without any additional embedding dimensionality projection.

For 250 labels both the KMeans and AAGMM last layer perform approximately equivalent to base FixMatch when using $None$, or just the $1st$ Order MoM embedding constraint. However, higher order constraints, which enforce ever stricter controls on what the embedding space should look like, produce worse overall model accuracy.

4.3. CIFAR-100

[TODO: document hyper-parameters]

4.4. Ablation Study

5. Conclusion

The exponential GPU memory requirement for the $3rd$ and $4th$ order MoM embedding constraints limits their practical application, though memory optimization is likely possible.

6. Acknowledgment

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 2

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 2

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 3

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[8] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 1

[11] Joseph Enguehard, Peter O'Halloran, and Ali Gholipour. Semi-supervised learning with deep embedded clustering for image classification and segmentation. *Ieee Access*, 7: 11093–11104, 2019. 3

[12] Enrico Fini, Pietro Astolfi, Karteek Alahari, Xavier Alameda-Pineda, Julien Mairal, Moin Nabi, and Elisa Ricci. Semi-supervised learning made simple with self-supervised clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3187–3197, 2023. 2

[13] Mohamed Farouk Abdel Hady and Friedhelm Schwenker. Semi-supervised learning. *Handbook on Neural Information Processing*, pages 215–239, 2013. 1

[14] Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision*, pages 674–690. Springer, 2022. 1, 2

[15] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 1, 2, 4

[16] Doyup Lee, Sungwoong Kim, Ildoo Kim, Yeongjae Cheon, Minsu Cho, and Wook-Shin Han. Contrastive regularization for semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3911–3920, 2022. 1

[17] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021. 1, 2

[18] Yu-Feng Li and De-Ming Liang. Safe semi-supervised learning: a brief introduction. *Frontiers of Computer Science*, 13: 669–676, 2019. 1

[19] Ioannis E Livieris, Konstantina Drakopoulou, Vassilis T Tampakas, Tassos A Mikropoulos, and Panagiotis Pintelas.

CIFAR-10 Max Test Accuracy

Last Layer	Emb Dim	40 Labels (6 trials)				
Embedding Constraint		None	$1st$ Order	$2nd$ Order	$3rd$ Order	$4th$ Order
FixMatch[28] (i.e. FullyConnected)	128	89.89				
	32	87.62				
	8	$xx.yy$				
KMeans	128	91.72	90.91	74.88		
	32	93.51	93.57	81.80		
	8	92.69	92.05	73.34	$xx.yy$	$xx.yy$
AAGMM	128	94.10	94.58	88.43		
	32	90.63	94.27	88.65		
	8	91.30	94.28	87.13	$xx.yy$	$xx.yy$

Table 2. Max run accuracy % for CIFAR-10 SSL benchmark with 40 labels comparing various configurations of our method. This table shows the best-case performance of our various methods; with the effect of poorly representative labels selected for each class. This demonstrates that for AAGMM high embedding dimensionality, the embedding constraints have no effect on the final accuracy, but with reduced embedding space dimensionality, adding a $1st$ Order penalty to the cluster centers improves model accuracy.

Method	CIFAR-10	
Label Count	40	250
Supervised	77.18 ±1.32	56.24 ±3.41
FixMatch	13.81 ±3.37	5.07 ±0.65
FlexMatch	5.29 ±0.29	4.97 ±0.07
SimMatch	5.38 ±0.01	5.36 ±0.08
SimMatchV2	4.90 ±0.16	5.04 ±0.09
Ours (KMeans)	8.42 ±1.39	5.34 ±0.27
Ours (AAGMM)	11.17 ±4.83	5.25 ±0.29

Table 3. Error rate % for CIFAR-10 SSL benchmark comparing to state of the art results. Results are copied from USB [30] unless otherwise stated. Results are based on 3 runs for USB, 5 runs for FixMatch [28], and 6 runs for ours.

Method	CIFAR-100	
Label Count	400	2500
Supervised	89.6 ±0.43	58.33 ±1.41
FixMatch	48.85 ±1.75	28.29 ±0.11
FlexMatch	40.73 ±1.44	26.17 ±0.18
SimMatch	39.32 ±0.72	26.21 ±0.37
SimMatchV2	36.68 ±0.86	26.66 ±0.38
Ours (KMeans)	$xx.yy$ ±$z.zz$	$xx.yy$ ±$z.zz$
Ours (AAGMM)	$xx.yy$ ±$z.zz$	$xx.yy$ ±$z.zz$

Table 4. Error rate % for CIFAR-100 SSL benchmark comparing to state of the art results. Results are copied from USB [30] unless otherwise stated. Results are based on 3 runs for USB, 5 runs for FixMatch [28], and 6 runs for ours.

Our Results on CIFAR-10 (40 Labels)	
Run Number	Test Accuracy
Run 1	$xx.yy$
Run 2	$xx.yy$
Run 3	$xx.yy$
Run 4	$xx.yy$
Run 5	$xx.yy$
Run 6	$xx.yy$
Run 7	$xx.yy$
Run 8	$xx.yy$
Run 9	$xx.yy$
Run 10	$xx.yy$
Run 11	$xx.yy$
Run 12	$xx.yy$

Table 5. Error rate % for CIFAR-10 SSL benchmark showing the run-to-run variance depending on the quality of the 40 labels selected from the full population.

Predicting secondary school students' performance utilizing a semi-supervised learning approach. *Journal of educational computing research*, 57(2):448–470, 2019. 2

[20] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 337–344, 2006. 2

[21] Sumeet Menon and David Chapman. Semi-supervised contrastive outlier removal for pseudo expectation maximization (scope), 2022. 2

[22] Sumeet Menon, David Chapman, Phuong Nguyen, Yelena Yesha, Michael Morris, and Babak Saboury. Deep expectation-maximization for semi-supervised lung cancer screening, 2020. 2

[23] Aamir Mustafa and Rafał K Mantiuk. Transformation consistency regularization–a semi-supervised paradigm for image-to-image translation. In *European Conference on Computer Vision*, pages 599–615. Springer, 2020. 2

[24] Karl Pearson. Method of moments and method of maximum likelihood. *Biometrika*, 28(1/2):34–59, 1936. 2, 3

[25] V Jothi Prakash and Dr LM Nithya. A survey on

semi-supervised learning techniques. *arXiv preprint arXiv:1402.4645*, 2014. 2

[26] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021. 2

[27] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, pages 29–36, 2005. 2

[28] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1, 2, 3, 4, 5, 6, 7

[29] Antti Tarvainen 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*, 30, 2017. 2

[30] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems*, 35:3938–3961, 2022. 7

[31] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022. 1

[32] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 2

[33] Chen Xing, Sercan Arik, Zizhao Zhang, and Tomas Pfister. Distance-based learning from errors for confidence calibration. In *International Conference on Learning Representations*, 2020. 2

[34] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-aware contrastive semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14421–14430, 2022. 1, 2

[35] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 2

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 4

[37] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 2

[38] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 1, 2

[39] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching, 2022. 2, 4

[40] Mingkai Zheng, Shan You, Lang Huang, Chen Luo, Fei Wang, Chen Qian, and Chang Xu. Simmatchv2: Semi-supervised learning with graph consistency. *arXiv preprint arXiv:2308.06692*, 2023. 1, 2

[41] Xiaojin Zhu and Andrew B Goldberg. *Introduction to semi-supervised learning*. Springer Nature, 2022. 1