A Method of Moments Embedding Constraint and its Application to Semi-Supervised Learning

Anonymous arξiv submission

Paper ID 20

Abstract

Discriminative deep learning models with a linear+softmax final layer have a problem: the latent space only predicts the conditional probabilities $p(Y|X)$ but not the full joint distribution $p(Y, X)$, which necessitates a generative approach. The conditional probability cannot detect outliers, causing outlier sensitivity in softmax networks. This exacerbates model over-confidence impacting many problems, such as hallucinations, confounding biases, and dependence on large datasets. To address this we introduce a novel embedding constraint based on the Method of Moments (MoM). We investigate the use of polynomial moments ranging from 1st through 4th order hypercovariance matrices. Furthermore, we use this embedding constraint to train an Axis-Aligned Gaussian Mixture Model (AAGMM) final layer, which learns not only the conditional, but also the joint distribution of the latent space. We apply this method to the domain of semi-supervised image classification by extending FlexMatch with our technique. We find our MoM constraint with the AAGMM layer is able to match the reported FlexMatch accuracy, while also modeling the joint distribution, thereby reducing outlier sensitivity. We also present a preliminary outlier detection strategy based on Mahalanobis distance and discuss future improvements to this strategy. Future work explores potential applications for the AAGMM layer and MoM embedding constraint, and how/why this MoM technique can overcome theoretical limitations of other existing methods including the approximate KL-divergence constraint of variational autoencoders. Code is available at: https://github.com/*******

1. Introduction

The majority of deep classifiers rely on a softmax final activation layer which predicts the conditional probability $p(Y|X)$. When that layer receives input X, the model predicts a soft pseudo-distribution of labels Y which argmax can convert into a hard label. If X is far from the decision boundary, then by definition, softmax assigns a prediction Y with high confidence. This works well for inlier samples, well represented by the training distribution. However, when presented with an outlier X, it is likely X will be far from the decision boundary (Figure 1, top). Therefore, softmax perceptrons, by definition, over-confidently hallucinate when given unexpected inputs [42]. Most deep classifiers use softmax without a safety net and as such over-confidently predict Y. Ideally, when input X is far from the decision boundary and training exemplars, the model should not be confident about the output class label Y.

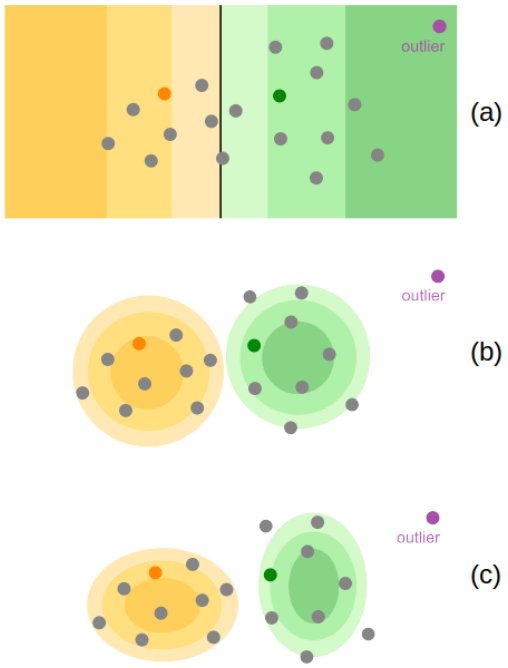

Figure 1. Schematic of the outlier problem, and how generative modeling of the joint probability can improve the situation. Prediction with (a) fully-supervised softmax, (b) semi-supervised KMeans, and (c) semi-supervised AAGMM.

Replacing softmax with a generative method that models

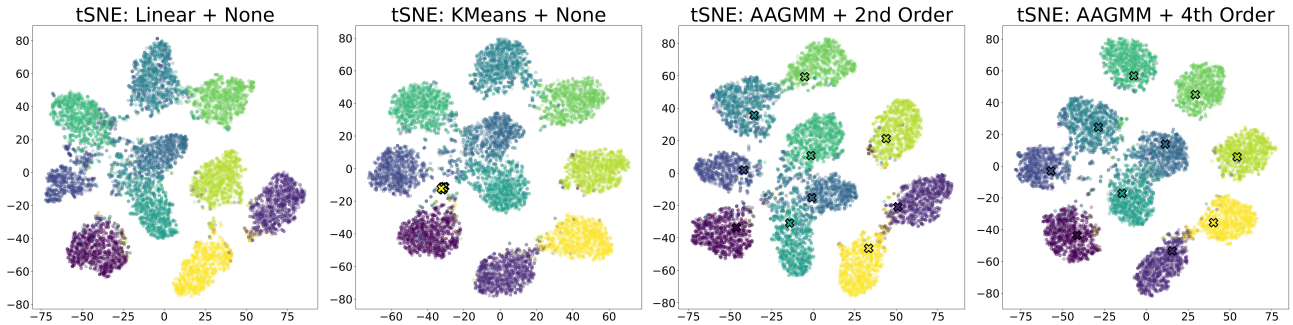

Figure 2. t-SNE[37] plot of the latent embedding space for various final layers with different MoM embedding constraints.

the joint probability $p(Y, X)$ can improve the capability of deep classifiers. Models using a final layer capable of learning the joint probability $p(Y, X)$ can infer the conditional $p(Y|X)$. More importantly, such a layer can also infer the prior probability $p(X)$. Thus, if X is an unexpected input, then such a layer can flag the input as a low-probability outlier, rather than confidently predicting a label.

Prior work has explored generative modeling for image classification [15, 17, 22]. How to best train and utilize generative modeling within a deep learning context remains an open question. The naive approach of minimizing cross entropy between Y and Y_pred will not work. Figure 2 (b) shows t-SNE plots for semi-supervised CIFAR-10 [18] image classification which illustrate why the naive approach will not work as intended. The t-SNE [37] Figure 2(b) plot shows the latent space of a 93% accurate CIFAR-10 classification model. However, the explicitly modeled cluster centers (shown as X's) do not align with the underlying data. While that model has acceptable predictive performance, it does not accurately learn and represent the underlying training data. To construct a robust model, one cannot simply fit a decision boundary. The model needs to learn the full joint distribution of the latent space. Figure 2 (d) demonstrates that with our proposed AAGMM final layer with 4th order MoM embedding constraints, the exact same model can achieve comparable (if not better) accuracy, but with the added benefit of modeling the underlying data clusters in the latent space.

We apply this work to the domain of Semi-Supervised Learning (SSL), because over-confident label predictions can cause confounding issues with pseudo-labeling methods [1]. SSL leverages an abundance of unlabeled data to improve deep learning based model performance under limited training data regimes [13, 23, 52]. Contrastive learning methods leverage the intuition that similar instances should be close in the representation space, while different instances are farther apart [20, 45]. Consistency regularization borrows the intuition that modified views of the same instance should have similar representations and predictions [14, 19, 35, 48]. This work contributes:

1. A novel Method of Moments (MoM) based embedding constraint that enables the model to not only learn the decision boundary but also the latent joint distribution. Moreover, this constraint ensures that each latent cluster exhibits a well-behaved Gaussian shape.
2. A replacement of the final linear+softmax activation layer of the neural network with either an axis-aligned differentiable Gaussian Mixture Model (AAGMM) or an equal variance version named KMeans trained via back propagation, both of which have explicit modeling of class cluster centroids.
3. A preliminary outlier removal strategy based on Mahalanobis distance that is compatible with AAGMM and MoM techniques.

We apply this methodology to the task of semi-supervised image-classification using CIFAR-10 [18] and STL-10 [**?**] with 40 training labels. The embedding constraint penalties are applied to all unlabeled data and not just the valid pseudo-labels. As such our method fits the latent joint distribution across all of the unlabeled data points, an improvement on baseline pseudo-labeling methods (like FlexMatch [48]) which only fit the conditional distribution to the high confidence pseudo-labels while removing low confidence pseudo-labels.

2. Related Work

SSL has shown great progress in learning high quality models, in some cases matching fully supervised performance for a number of benchmarks [48]. The goal of SSL is to produce a trained model of equivalent accuracy to fully supervised training, with vastly reduced data annotation requirements. Doing so relies on accurately characterizing inlier vs outlier unlabeled samples.

2.1. Pseudo-Labeling

Self-training was among the initial approaches employed in the context of SSL to annotate unlabeled images. This in-

volves the initial training of a classifier with a limited set of labeled samples which incorporates pseudo-labels exceeding a predefined threshold into the gradient descent process [9, 24, 25, 27, 34, 46, 47]. 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, 31]. A notable approach to pseudo-labeling is the Mean Teacher algorithm [36], which leverages exponential moving averages of model parameters to acquire a notably more stable target prediction. This refinement substantially enhances the convergence of the algorithm.

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. [33] 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 [33] where better network logit calibration might improve results [44]. Improvements to the pseudo-labeling process have been demonstrated by imposing curriculum [48] or by including a class-aware contrastive term [45]. Leveraging the concept of explicit class cluster centers for conditioning semantic similarity improves final model accuracy [50]. Additionally, improvements have been found in incorporating purely clustering based methods like DINO [7] into semi-supervised methods [11].

2.2. Consistency Regularization

Consistency Regularization is a branch of techniques that have been instrumental toward many of the state of the art techniques in semi-supervied learning within the last several years [2, 3, 14, 20, 35, 48, 50, 51]. The idea being that augmentation does not typically change the meaning of images. MixMatch is a semi-supervised pseudolabeling that greatly popularized the use of consistency regularization to ensure that augmentation does not affect the predicted label [2, 3]. FixMatch further extended this method by introducing the notion of weak and strong augmentations including the cutout operator to increase the robustness of the regularization [35]. FlexMatch is a further improvement that introduces a curriculum pseudo-labeling strategy for flexible threshold values [48]. Co-Match made use of a form of consistency regularization to ensure that strong augmentations shared not only a similar pseudolabel, but furthermore a similar embedding space. Moreover, a neighborhood graph was constructed for embeddings and pseudolabels and refined via co-learning [20]. Con-match introduced a confidence metric based on the similarity of a basket of augmented embeddings [14]. SimMatch introduced a graph-based label propagation algorithm through a low-dimensional latent projection, and utilized multiple forms of consistency regularization including both semantic-level and instance-level consistency terms [50, 51].

2.3. Latent Embedding Constraints

A notable latent embedding constraint that is related, yet substantially different from our approach is the Evidence Lower Bound (ELBO). ELBO approximates a latent sample with a variational distribution and constrains the KL-divergence between the variational distribution and a target shape which is typically a multivariate standard normal distribution [15]. The main drawback of this approach is that the true KL-divergence is intractable to calculate. As such, the posterior must take on a simplified form. Most practical implementations use a diagonal posterior which can only penalize simple differences in shape such as mean and standard deviation. Arbitrarily complex posteriors are nevertheless possible using the method of Normalizing flows, which provides an iterative framework based on change of variables although this method is quite involved [8, 16, 32]. Our MoM constraint is relatively simple but can also penalize complex differences in shape by constraining 2nd, 3rd, and 4th order hyper-covariance matrices, although we do so by comparing the moments directly. This greatly simplifies implementation as we do not need to explicitly construct a posterior distribution.

Another notable embedding constraint is the Maximum Mean Discrepancy (MMD), also known as the two-sample test [12]. MMD was used in the Generative Moment Matching Network [21] and has since been used extensively for the problem of domain adaptation [38, 39], in order to constrain the latent projections of the source and target distributions to follow the same distribution. MMD is a moment-matching constraint based on the kernel trick and can therefore constrain any difference in shape between two samples including very high order moments. Due to the kernel trick requiring proper inner products, MMD can only be used to constrain one sample to another sample. It cannot directly constrain sample statistics to population statistics, although it is possible to approximate populations numerically via monte-carlo sampling [49]. Like MMD, our method is based on MoM, but it does not involve the kernel trick, and instead penalizes polynomial moments explicitly thereby enabling the sample embedding to be constrained to an exact target distribution.

3. Methodology

In this section, we explore our proposed replacement final activation layers and our embedding space constraints. Our methodology is based upon the published FlexMatch [48] algorithm as implemented in the USB framework [40], with identical hyper-parameters unless otherwise stated. We ex-

tend FlexMatch with a few minor training algorithm modifications explored in Section 4.1. FlexMatch [48] is a simple, well performing SSL algorithm. As such, it serves as a good comparison point for exploring the effect of our contributions.

Both the linear layer replacements and the embedding constraints explored herein represent increasing levels of prescription about how the final latent embedding space should be arranged compared to a traditional linear layer. The idea of leveraging clusters in embedding space is not new [5, 6, 10], but we extend the core idea with a novel differentiable model with learned cluster centroids and MoM based constraints. The MoM constraints do not impose any assumptions outside of applying L2 penalties as described in Section 3.2.

3.1. Alternate Final Layers

As we discussed in the introduction traditional final activation layers such as linear+softmax are fully discriminative in that they directly estimate the conditional probability $p(Y|X)$. These layers do not estimate $p(X)$ or the joint probabilities $p(Y, X)$. To overcome this limitation, we present two generative 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 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 D-dimensaional latent representation Z_i as follows,

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

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,

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

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 given by the normal pdf of the k^{th} cluster,

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

By Bayesian identity, the conditional probability $\hat{Y}_{i,k} = p(Y_{i,k}|X_i)$ can therefore be inferred from Eq. 1 and 3 as follows,

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

The AAGMM layer is implemented as a normal PyTorch [29] module. It has two parameters updated by backprop. (1) the explicit cluster centers, a matrix $K \times D$ initialized randomly, and (2) the diagonal elements of the $D \times D$ matrix Σ_k are randomly initialized in the range $[0.9, 1.1]$, which contains the diagonal elements of the GMM Sigma matrix for each cluster.

3.1.2 KMeans Layer

We also implement a KMeans final layer which is a more restrictive form of the AAGMM layer. The KMeans layer is additionally constrained such 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. See the published codebase for implementation details about the AAGMM and KMeans layers.

3.1.3 Relation between K-means, AAGMM and Softmax layers

The AAGMM, KMeans and Softmax layers are theoretically related because Softmax is the conditional distribution $p(Y|X)$ that arises when the joint distributions $p(Y, X)$ follow the equivariate normal distributions as modeled by

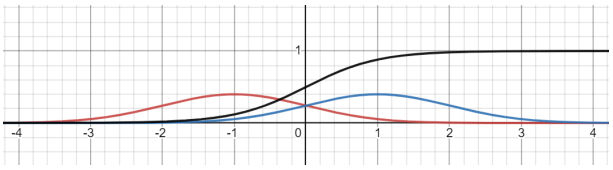

Figure 3. Illustrated relationship between K-means and Softmax Layers for 1D binary classification. If the joint distributions (blue curve and red curve) follow equivariate normal probability densities, then the conditional distribution (black curve) is softmax.

KMeans. Figure 3 shows a simple example of this relationship in one dimension with two labels A (blue curve) and B (red curve), with joint distributions as follows,

$$\begin{aligned} p(Y = A, X) &= \frac{1}{2}\mathcal{N}(X, \mu_A, \sigma) \\ p(Y = B, C) &= \frac{1}{2}\mathcal{N}(C, \mu_B, \sigma) \end{aligned} \tag{5}$$

In this case the conditional distribution can be described by sigmoid which is a special case of Softmax,

$$\begin{aligned} p(Y = A \mid X) &= \textit{sigmoid}(mX + b) \\ &\text{where} \\ m = \frac{\mu_A - \mu_B}{\sigma^2} &\quad \text{and} \quad b = \frac{\mu_A^2 + \mu_B^2}{2\sigma^2} \end{aligned} \tag{6}$$

The AAGMM layer is a generalization of the KMeans layer to allow different diagonal covariance matrices for each cluster. This gives the AAGMM somewhat greater expressive power than Softmax and KMeans as it can model the joint distribution of latent spaces with different cluster sizes and non-spherical shapes.

3.2. Method of Moments Embedding Constraints

We introduce and evaluate a series of embedding constraints based on the Method of Moments (MoM) [30]. For each sample i and each cluster k, the joint $p(Y_{i,k}, X_i)$ is calculated as in Eq. 3 and then used to infer the prior $p(X_i)$ and the conditional $p(Y_{i,k}|X_i)$. As usual, the conditional probability is trained using cross entropy loss. When embedding constraints are omitted, it is possible for the model to learn an accurate decision boundary for the conditional probability without modeling the latent joint distribution. MoM solves these problems and is an appropriate strategy for semi-parametric models.

The MoM relies on the use of *consistent estimators*, which 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 P population statistic as follows,

$$\forall p = 1 \dots P \quad \frac{1}{n}\sum_{i=1}^{n} g_p(z_i) \approx E(g_p(Z)) \tag{7}$$

We can therefore constrain the latent representation of our model to approximate a multivariate standard normal distribution.

The centralized moments are a classical choice for the consistent estimator g_p representing the terms of a power series around the mean μ

$$g_p(Z) = (Z - \mu)^p \tag{8}$$

In the univariate standard normal case, the p^{th} order centralized moment constraint is the following.

$$E\left[(Z-\mu)^p\right] = \begin{cases} 0 & \text{if } p \text{ is odd} \\ \sigma^p(p-1)!! & \text{if } p \text{ is even} \end{cases} \tag{9}$$

Where '!!' represents the double factorial operator. By this formula, the univariate unit Gaussian has mean 0, standard deviation 1, skew 0, and kurtosis 3.

The multivariate standard normal distribution is the marginal product of the univariate standard normal distributions. As such, if we redefine Z, μ, and p to be all D dimensional, then the centralized marginal product moment can be defined as follows,

$$E\left[g_p(Z-\mu)\right] = E\left[\prod_{d=1}^{D}(Z_d - \mu_d)^{p_d}\right] \tag{10}$$

Due to independence of the axes, this multivariate population moment can be represented as a product of univariate moments of the individual standard normal distributions as follows,

$$E\left[\prod_{d=1}^{D}(Z_d - \mu_d)_d^p\right] = \prod_{d=1}^{D} E\left[(Z_d - \mu_d)^{p_d}\right] \tag{11}$$

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

$$\varepsilon_p = \left(\frac{1}{n}\sum_{i=1}^{n} g_p(z_i) - E(g_p(Z))\right)^2 \tag{12}$$

Some moments are more important than others, and must be weighted more heavily. 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. To prevent over-representation of off-diagonal terms and encourage representation of on-diagonal terms, the loss function we use for any given moment term is inversely proportional to the number of 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. This weighting scheme supports using 0 to 4th order MoM constraints seamlessly and is not a hyper-parameter we expect to require tuning.

3.3. Manhabalobis Outlier Removal

The AAGMM layer allows us to detect and remove outliers based on Mahalanobis distance in the latent feature space. By *outlier*, we are referring to the problem that the pseudolabel learner (i.e. FlexMatch) is simply not yet ready to learn a given unlabeled sample, because the model has only attempted to learn the distribution of labeled and previously pseudolabeled samples up until that point. Due to small labeled sample size, the labeled and pseudolabeled samples do not fully represent the distribution of the unlabeled samples in early iterations. Thus, unlabeled samples far from the learned distribution are considered to be ouliers in a given iteration.

In order to implement *outlier removal*, in the context of pseudolabeling, we exclude the unlabeled *detected outlier* sample from gradient updates for any the given iteration of the semi-supervised procedure. For FlexMatch in particular, all unlabeled samples are augmented with weak and strong RandAugment. As such we use the weakly augmented samples as input to outlier detection.

Mathematically, we consider an unlabeled sample x to be an outlier if it is far from all cluster centers $\mu_1 \dots \mu_K$ in terms of Mahalanobis distance based on the cluster covariances $\Sigma_1 \dots \Sigma_K$. Mahalanobis distance of a given point x to cluster k is defined as follows,

$$d_{M,k}(x) = \sqrt{(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)} \tag{13}$$

Define X_L as the labeled population and X_U as the unlabeled population, with $x \in X_U$ as an unlabeled sample, and P_{90} as the 90th percentile. As such, we detect outliers as follows,

$$\begin{aligned} &x \in X_U \ \text{ is an outlier iff.} \\ &\quad max_k\ d_{M,k}(x) > \tau \\ &\text{where } \ \tau = P_{90}\left(max_k\ d_{M,k}\left(X_L\right)\right) \end{aligned} \tag{14}$$

4. Experiments

We evaluate both AAGMM and KMeans linear layer replacements and the embedding space constraints using our modified FlexMatch [48] on the common SSL benchmarks CIFAR-10 [18] at 40 labels (4 labels per class) and STL-10 [**?**] at 40 labels. We randomly selected 5 seeds a priori for evaluation. For each algorithm configuration tested, one model was trained per seed. During each run, the required number of labeled samples are drawn without replacement from the training population of the dataset in a deterministic manner (reproducible with the same seed). All data not in this labeled subset are used as unlabeled data (i.e., the labels are discarded).

4.1. Hyper-Parameters

All CIFAR-10 models were trained with the standard benchmark WideResNet28-2 architecture. All STL-10 models were trained with the standard benchmark WideResNet37-2 architecture. This work leverages the published FlexMatch [48] code, hyper-parameters, and training configurations within the USB Framework [40]. However, due to the higher training instability of the AAGMM layers compared to a linear layer, the gradient norm was clipped to 1.0. Despite specific attention to computing the AAGMM and embedding constraints in a numerically stable manner, they are still less stable during backprop than a simple linear layer. Gradient clipping was especially important when there were latent points that were multiple standard deviations away from the cluster centers. In this case, the gradient of the Gaussian probability density function converges rapidly toward zero, which can affect the division step of equation 4. We have almost entirely overcome this issue in the code by using laws of exponents in order to normalize the denominator to be greater or equal to 1 prior to division, but in extreme cases of latent points far from cluster centers, the gradient clipping is still necessary to achieve stable gradient descent.

We believe that the rapid decrease in slope of the Gaussian distribution PDF for points that are multiple standard deviations away from the mean. As the conditional distribution involves calculating a division step, it is very likely that such a division may become less stable for points that are far from the mean.

This work includes an exploration of how various latent embedding dimensionalities affect the generative linear layer replacement. As such, the model architecture was modified with a single additional linear layer before the output to project the baseline model embedding dimension (128 for WideResNes28-2) down to a reduced 8 dimensional space. Results listed with an embedding dimensionality of 128 do not include the additional linear layer which reduces the latent dimensionality.

Due to exponential GPU memory requirements with

Mean Test Accuracy Per Method and Dataset

Last Layer	Emb Dim	CIFAR-10 at 40 Labels (5 trials)		
Embedding Constraint		None	$1st$ Order	$2nd$ Order
Linear	128	95.03 ±0.06		
(i.e. FullyConnected)	8	90.89 ±3.24		
AAGMM	8	94.98 ±0.07	94.64 ±0.27	93.58 ±2.74
KMeans	8	90.15 ±6.40	93.50 ±0.62	93.44 ±0.50
Last Layer	Emb Dim	STL-10 at 40 Labels (3 trials)		
Embedding Constraint		None	$1st$ Order	$2nd$ Order
Linear	128	70.85 ±4.16		
AAGMM	8	58.79 ±11.00	71.11 ±7.60	70.40 ±6.39

Table 1. Mean test accuracy % for CIFAR-10 and STL-10 SSL benchmarks comparing various configurations of our method. The FlexMatch results in the table is drawn from the publication. For CIFAR-10, the WideResNet model used by FlexMatch has an embedding size of 128 dimension. Results for a given order of embedding constraint include all lower constraints.

each successive MoM moment, only the 8D embedding can operate with higher order MoM embedding constraints. While our method can place constraints on any number of moments, we only explored MoM constraints up to the second order in this paper. Results for any given order of embedding constraint include all lower constraints.

4.2. CIFAR-10 and STL-10 Results

While current SOTA on CIFAR-10 at 250 labels is close to fully supervised accuracy, the 40 label case provides a far more challenging task. Table 1 summarizes the relative performance of our various configurations. As we simply extended FlexMatch, our hyper-parameter selection is likely sub-optimal, and further experimentation might yield improvements. The 'Linear (i.e. FullyConnected)' rows in table 1 represent the baseline fully connected linear last layer with and without additional embedding dimensionality projection.

For CIFAR-10 at 40 labels, the highest mean-accuracy configuration that we have achieved was 94.98% which was obtained using the AAGMM technique with no embedding constraints and an 8 dimensional embedding.

We see that the AAGMM final layer consistently out performs the KMeans final layer for the CIFAR-10 Test Accuracy with 40 labels. We furthermore see that the KMeans layer performs significantly better with the 1st and 2nd order MoM constraint, as compared with no constraints.

For STL-10 the AAGMM final layer (71.11%) improved upon the FlexMatch [48] result (70.85%), though within the margin of error. Additionally, both 1st and 2nd order MoM constraints significantly improved the upon the AAGMM with no embedding constraints.

The modeled cluster centers vary in quality between individual model runs of the AAGMM layer due to the stochasticity of the training process. Figure 4 (a & b) showcases degenerate cluster centers. The Figure 4 (c) AAGMM model learned cluster centers that are an adequate approximation of the underlying data. However, the embedding constraints encourage cluster centers which are better aligned with the underlying data, Figure 4 (d). It is worth noting that we did not observe the KMeans layer learning non-degenerate cluster centers without an embedding constraint. In contrast, the AAGMM layer can, under some circumstances, learn viable cluster centers.

Table 2 puts these results in context with the current SSL SOTA for CIFAR-10 and STL-10 at 40 labels and demonstrates that this methodology is almost competitive with for CIFAR-10, but still requires improvement for STL-10.

Method	CIFAR-10 (40 Labels)	STL-10 (40 Labels)
FixMatch[35]	13.81 ±3.37	35.97 ±4.14
FlexMatch[48]	4.97 ±0.06	29.15 ±4.16
FreeMatch[41]	4.90 ±0.29	15.56 ±0.55
SimMatchV2[51]	4.90 ±0.04	15.85 ±2.62
AAGMM+None	5.02 ±0.07	41.21 ±11.00
AAGMM+1stOrder	5.36 ±0.27	28.89 ±.60

Table 2. Error rate % for CIFAR-10 and STL-10 SSL benchmarks with 40 labels, comparing to state of the art results. Results for previously published methods are drawn from USB [40] except for FreeMatch [41] and SimMatchV2 [51] publications.

5. Discussion

Although our preliminary results with the proposed AAGMM and MoM achieve high accuracy relative to SOTA, this was achieved without Mahalanobis outlier detection as documented in table 3. When the outlier detection was enabled, we believe the reduced accuracy is due to: 1. the 90th percentile distance threshold being too aggressive and filtering too much signal relative to noise, and

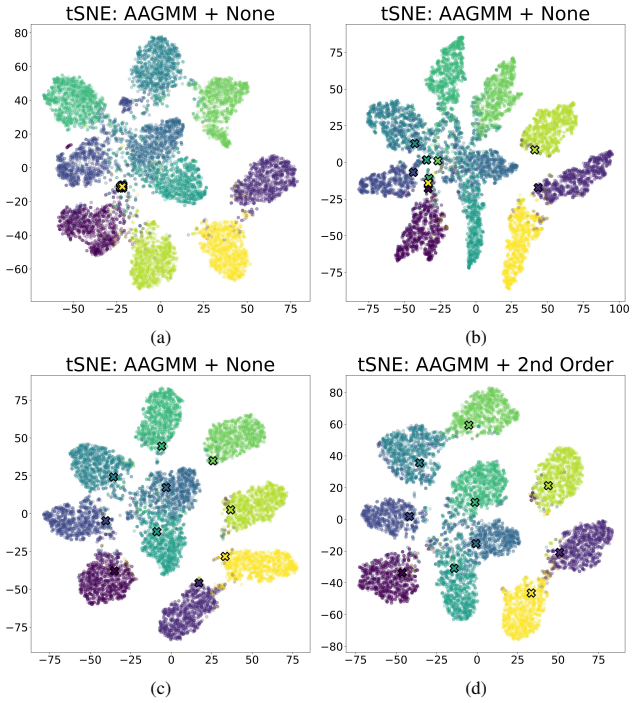

Figure 4. t-SNE plot of fully trained and reasonably accurate AAGMM model's latent embedding space (with the learned cluster centers marked with X's). Depending on the run, the AAGMM cluster centers will be degenerate (top left), non-degenerate but still mis-aligned with the clusters (top right), acceptably aligned (bottom left), or well aligned with the underlying clusters when a 2nd order constraint is employed (bottom right).

2. the need for an adaptive outlier detection threshold. In early epochs, an aggressive outlier detection threshold is viable because the model will not adequately fit many of the unlabeled samples, but as the model converges the fit improves reducing the need for outlier removal. As such, we believe that the aggressive outlier filtering is removing too many inliers, particularly in later epochs.

The proposed MoM embedding constraint has at least one significant downside, by requiring exponentially increasing amounts of GPU memory for each successive moment penalty included. This limits the current practicality of these MoM constraints. Additional optimization and/or avoiding the explicit creation of both the n^{th} order moment and its target value on GPUs would likely improve the usability.

Semi-supervised learning is highly sensitive to both which samples are selected form the labeled population [35] and the stochasticity of the training process itself.

Future work in this area will explore alternate outlier removal strategies, including thresholding the unlabeled samples based on their latent sample probability $p(x)$ as opposed to latent Mahalanobis distance. The Mahalanobis distance is part of the exponential term in the calculation of the multivariate Gaussian PDF. As such, we expect that a removing outliers based on low $p(X)$ is likely to perform perform comparably to removing samples based on far Mahalanobis distance, in the special case that all clusters share a similar determinant $det\,(\Sigma_k)$. Additionally, we plan to explore how to best take advantage of the better behaved latent embedding space to improve data efficiency for model training.

AAGMM (8D) Mean Test Accuracy With Outlier Removal

CIFAR-10		
Outlier Threshold	None	90th Percentile
MoM: None	94.98 ±0.07	94.9 ±0.125
MoM: $1st$ Order	94.64 ±0.27	87.70 ±2.96
MoM: $2nd$ Order	93.58 ±2.74	87.25 ±2.51
STL-10		
Outlier Threshold	None	90th Percentile
MoM: None	58.79 ±11.00	57.50 ±12.75
MoM: $1st$ Order	71.11 ±7.60	64.18 ±3.82
MoM: $2nd$ Order	70.40 ±6.39	65.90 ±4.09

Table 3. Mean test accuracy % for CIFAR-10 and STL-10 SSL benchmarks comparing an 8D embedding AAGMM final layer with and without outlier removal during training. Results for a given order of embedding constraint include all lower constraints.

We demonstrate a novel fully differentiable Axis-Aligned Gaussian Mixture Model with Method of Moments based latent embedding space constraints can be applied to semi-supervised learning. The combination of these techniques enables outlier detection strategies that would otherwise not be possible with a traditional softmax discriminator approach. This preliminary work constructs these novel layers with the associated constraints and demonstrates reasonable performance on challenging benchmark semi-supervised learning tasks, while opening the door for future outlier detection strategies that can make semi-supervised learning more robust to large and diverse unlabeled sample distributions.

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, 3

[2] 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. 3

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching

and augmentation anchoring. In *International Conference on Learning Representations*, 2020. 3

[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. 3

[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. 4

[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. 4

[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. 3

[8] Anthony Caterini, Rob Cornish, Dino Sejdinovic, and Arnaud Doucet. Variational inference with continuously-indexed normalizing flows. In *Uncertainty in Artificial Intelligence*, pages 44–53. PMLR, 2021. 3

[9] 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. 3

[10] 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. 4

[11] 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. 3

[12] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007. 3

[13] Mohamed Farouk Abdel Hady and Friedhelm Schwenker. Semi-supervised learning. *Handbook on Neural Information Processing*, pages 215–239, 2013. 2

[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. 2, 3

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 3

[16] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016. 3

[17] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. 2

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 2, 6

[19] 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. 2

[20] 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. 2, 3

[21] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International conference on machine learning*, pages 1718–1727. PMLR, 2015. 3

[22] Yang Li, Quan Pan, Suhang Wang, Haiyun Peng, Tao Yang, and Erik Cambria. Disentangled variational auto-encoder for semi-supervised learning. *Information Sciences*, 482:73–85, 2019. 2

[23] Yu-Feng Li and De-Ming Liang. Safe semi-supervised learning: a brief introduction. *Frontiers of Computer Science*, 13: 669–676, 2019. 2

[24] Ioannis E Livieris, Konstantina Drakopoulou, Vassilis T Tampakas, Tassos A Mikropoulos, and Panagiotis Pintelas. Predicting secondary school students' performance utilizing a semi-supervised learning approach. *Journal of educational computing research*, 57(2):448–470, 2019. 3

[25] 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. 3

[26] Sumeet Menon. *Semi-Supervised Expectation Maximization with Contrastive Outlier Removal*. PhD thesis, 2022. AAI29167191.

[27] Sumeet Menon, David Chapman, Phuong Nguyen, Yelena Yesha, Michael Morris, and Babak Saboury. Deep expectation-maximization for semi-supervised lung cancer screening. 2019. 3

[28] 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.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[30] Karl Pearson. Method of moments and method of maximum likelihood. *Biometrika*, 28(1/2):34–59, 1936. 5

[31] V Jothi Prakash and Dr LM Nithya. A survey on semi-supervised learning techniques. *arXiv preprint arXiv:1402.4645*, 2014. 3

[32] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 3

[33] 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. In *International Conference on Learning Representations*, 2021. 3

[34] 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. 3

[35] 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. 2, 3, 7, 8

[36] 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. 3

[37] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 2

[38] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[39] Wei Wang, Haojie Li, Zhengming Ding, Feiping Nie, Junyang Chen, Xiao Dong, and Zhihui Wang. Rethinking maximum mean discrepancy for visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 34 (1):264–277, 2023. 3

[40] 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. 3, 6, 7

[41] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. 7

[42] 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

[43] 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.

[44] Chen Xing, Sercan Arik, Zizhao Zhang, and Tomas Pfister. Distance-based learning from errors for confidence calibration. In *International Conference on Learning Representations*, 2020. 3

[45] 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. 2, 3

[46] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 3

[47] 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. 3

[48] 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. 2, 3, 4, 6, 7

[49] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, pages 5885–5892, 2019. 3

[50] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14471–14481, 2022. 3

[51] Mingkai Zheng, Shan You, Lang Huang, Chen Luo, Fei Wang, Chen Qian, and Chang Xu. Simmatchv2: Semi-supervised learning with graph consistency. pages 16432–16442, 2023. 3, 7

[52] Xiaojin Zhu and Andrew B Goldberg. *Introduction to semi-supervised learning*. Springer Nature, 2022. 2