FeatMatch: Feature-Based Augmentation for Semi-Supervised Learning

Chia-Wen Kuo[†], Chih-Yao Ma[†], Jia-Bin Huang[‡], Zsolt Kira[†],
[†]Georgia Tech, [‡]Virginia Tech

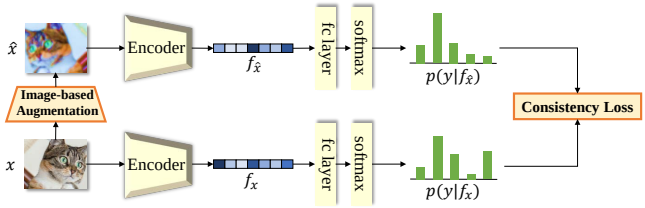
albert.cwkuo@gatech.edu, cyma@gatech.edu, jbhuang@vt.edu, zkira@gatech.edu

Abstract. Recent state-of-the-art semi-supervised learning (SSL) methods use a combination of image-based transformations and consistency regularization as core components. Such methods, however, are limited to simple transformations such as traditional data augmentation or convex combinations of two images. In this paper, we propose a novel learned feature-based refinement and augmentation method that produces a varied set of complex transformations. Importantly, these transformations also use information from both within-class and across-class prototypical representations that we extract through clustering. We use features already computed across iterations by storing them in a memory bank, obviating the need for significant extra computation. These transformations, combined with traditional image-based augmentation, are then used as part of the consistency-based regularization loss. We demonstrate that our method is comparable to current state of art for smaller datasets (CIFAR-10 and SVHN) while being able to scale up to larger datasets such as CIFAR-100 and mini-Imagenet where we achieve significant gains over the state of art (*e.g.,* absolute 17.44% gain on mini-ImageNet). We further test our method on DomainNet, demonstrating better robustness to out-of-domain unlabeled data, and perform rigorous ablations and analysis to validate the method. Code is available here: https://sites.google.com/view/chiawen-kuo/home/featmatch.

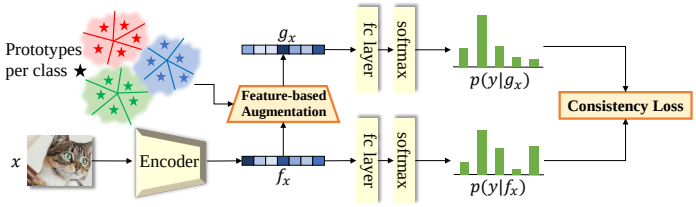
Keywords: semi-supervised learning, feature-based augmentation, consistency regularization

1 Introduction

Driven by large-scale datasets such as ImageNet as well as computing resources, deep neural networks have achieved strong performance on a wide variety of tasks. Training these deep neural networks, however, requires millions of labeled examples that are expensive to acquire and annotate. Consequently, numerous methods have been developed for semi-supervised learning (SSL), where a large number of unlabeled examples are available alongside a smaller set of labeled data. One branch of the most successful SSL methods [15,18,21,24,25,4,3] uses image-based augmentation [33,8,12,6] to generate different $transformations$ of an input image, and consistency regularization to enforce invariant representations across these transformations. While these methods have achieved great



(a) Image-Based Augmentation and Consistency



(b) Feature-Based Augmentation and Consistency

Fig. 1: Consistency regularization methods are the most successful methods for semi-supervised learning. The main idea of these methods is to enforce consistency between the predictions of different *transformations* of an input image. (***a***) Image-based augmentation method generate different views of an input image via data augmentation, which are limited to operations in the image space as well as operations within a single instance or simple convex combination of two instances. (***b***) We propose an additional learned feature-based augmentation that operates in the abstract feature space. The learned feature refinement and augmentation module is capable of leveraging information from other instances, within or outside of the same class.

success, the data augmentation methods for generating different transformations are limited to transformations in the image space and fail to leverage the knowledge of other instances in the dataset for diverse transformations.

In this paper, we propose novel feature-based refinement and augmentation that addresses the limitations of conventional image-based augmentation described above. Specifically, we propose a module that learns to refine and augment input image features via soft-attention toward a small set of representative prototypes extracted from the image features of other images in the dataset. The comparison between image-based augmentation and our proposed feature-based refinement and augmentation is shown in Fig. 1. Since the proposed module is learned and carried out in the feature space, diverse and abstract transformations of input images can be applied, which we validate in Sec. 4.4. Our approach only requires minimum computation via maintaining a memory bank and using k-means clustering to extract prototypes.

We demonstrate that adding our proposed feature-based augmentation along with conventional image-based augmentations, when used for consistency regularization, achieves significant gains. We test our method on standard SSL

datasets such as SVHN and CIFAR-10, and show that our method, despite its simplicity, compares favorably against state-of-art methods in all cases. Further, through testing on CIFAR-100 and mini-ImageNet, we show that our method is scalable to larger datasets and outperformed the current best methods by significant margins. For example, we outperformed the closest state of the art by an absolute **17%** on mini-ImageNet with 4k labels. We also propose another realistic setting on DomainNet [20] to test the robustness of our proposed method under the case where the unlabeled samples are partially coming from shifted domains, in which we improved **23%** over supervised baseline and **12%** over semi-supervised baseline when 50% unlabeled samples are all coming from shifted domains. Finally, we conduct thorough ablations and thorough analysis to highlight that the method does, in fact, perform varied complex transformations in feature space (as evidenced by t-SNE and nearest neighbor image samples). To summarize, our key contributions include:

- We develop a learned feature-based refinement and augmentation module to transform input image features in the abstract feature space by leveraging a small set of representative prototypes of all classes in the dataset.
- We propose a memory bank mechanism to efficiently extract prototypes from images of the entire dataset with minimal extra computations.
- We demonstrate thorough results across four standard SSL datasets and also propose a realistic setting where the unlabeled data partially come from domains shifted from the target labeled set.
- We perform in-depth analysis of the prototype representations extracted and used for each instance, as well as what transformations the proposed feature-based refinement and augmentation module learns.

2 Related Works

Consistency Regularization Methods. Current state-of-the-art SSL methods mostly fall into this category. The key insight of this branch of methods is that the prediction of a deep model should be consistent across different *semantic-preserving transformations* of the same data. Consistency regularization methods regularize the model to be invariant to textural or geometric changes of an image. Specifically, given an input image x and a network composed of a feature encoder $f_x = Enc(x)$ and a classifier $p_x = Clf(f_x)$, we can generate the pseudo-label of the input image by $p_x = Clf(Enc(x))$. Furthermore, given a data augmentation module $AugD(\cdot)$, we can generate an augmented copy of x by $\hat{x} = AugD(x)$. A consistency loss $\mathcal{H}$, typically KL-Divergence loss, is then applied on the model predictions of $\hat{x}$ to enforce consistent prediction: $\mathcal{L}_{con} = \mathcal{H}(p, Clf(Enc(\hat{x})))$.

Image-Based Augmentation. The core to consistency-based methods is how to generate diverse but reasonable transformations of the same data. A straightforward answer is to incorporate data augmentation, which has been widely used in the training of a deep model to increase data diversity and prevent overfitting.

Table 1: Comparison to other SSL methods with consistency regularization.

	ReMixMatch [3]	MixMatch [4]	Mean Teacher [25]	ICT [29]	PLCB [1]	**FeatMatch** (Ours)
Feature-Based Augmentation	-	-	-	-	-	✔
Image-Based Augmentation	✔	✔	✔	✔	✔	✔
Temporal Ensembling	✔	✔	✔			-
Self-Supervised Loss	✔	-	-	-	-	-
Alignment of Class Distribution	✔	-	-	-	✔	-

For example, [4,15,24,25] use traditional data augmentation to generate different transformations of semantically identical images. Data augmentation method randomly perturbs an image in terms of its texture, eg. brightness, hue, sharpness, or its geometry, eg. rotation, translation, or affine transform. In addition to data augmentation, Miyato et al. [18] and Yu et al. [31] perturbed images along the adversarial direction, and Qiao et al. [21] use multiple networks to generate different views (predictions) of the same data. Recently, several works propose data augmentation modules for supervised learning or semi-supervised learning, where the augmentation parameters can either be easily tuned [8], found by RL-training [7], or decided by the confidence of network prediction [3].

Mixup [33,32,32,12], similar to data augmentation, is another effective way of increasing data diversity. It generates new training samples by a convex combination of two images and their corresponding labels. It has been shown that models trained with Mixup is robust toward out-of-distribution data [9] and is beneficial for the uncertainty calibration of a network [26]. Given two images x_1 and x_2 and their labels (or pseudo labels) y_1 and y_2, they are mixed by a randomly sampled ratio r by $\hat{x} = r \cdot x_1 + (1-r) \cdot x_2$ and $\hat{y} = r \cdot y_1 + (1-r) \cdot y_2$. This has been done in feature space as well [28]. A standard classification loss $\mathcal{H}(\cdot)$ is then applied on the prediction of the mixed sample $\hat{x}$ and the mixed label $\hat{y}$ by $\mathcal{L}_{mix} = \mathcal{H}(\hat{y}, Clf(Enc((\hat{x}))))$. Originally, Mixup methods were developed for supervised learning. ICT [29] and MixMatch [4] introduce Mixup into semi-supervised learning by using the pseudo-label of the unlabeled data. Furthermore, by controlling the mixing ratio r to be greater than 0.5 as proposed by [4], we can make sure that the mixed sample is closer to x_1. Therefore, we can separate the mixed data into labeled mixed batch $\hat{\mathcal{X}}$ if x_1 is labeled, and unlabeled mixed batch $\hat{\mathcal{U}}$ if x_1 is unlabeled. Different loss weights can then be applied to modulate the strength of regularization from the unlabeled data.

3 Feature-Based Augmentation and Consistency

Image-based augmentation has been shown to be an effective approach to generate different views of an image for consistency-based SSL methods. However, conventional image-based augmentation has the following two limitations: (1) Operate in image space, which limits the possible transformations to textural or geometric within images, and (2) Operate within a single instance, which fails to transform data with the knowledge of other instances, either within or outside

of the same class. Some recent works that utilize Mixup only partially address the second limitation of conventional data augmentation since mixup operates only between two instances. On the other hand, Manifold Mixup [28] approaches the first limitation by performing Mixup in the feature space but is limited to a simple convex combination of two samples.

We instead propose to address these two limitations simultaneously. We proposed a novel method that refines and augments image features in the abstract feature space rather than image space. To efficiently leverage the knowledge of other classes, we condense the information of each class into a small set of prototypes by performing clustering in the feature space. The image features are then refined and augmented through information propagated from prototypes of all classes. We hypothesize that this feature refinement/augmentation can further improve the feature representations, and these refined features can produce better pseudo-labels than features without the refinement (See Sec. 4.4 for our analysis on this hypothesis). The feature refinement and augmentation are learned via a lightweight attention network for the representative prototypes and optimized end-to-end with other objectives such as classification loss. A consistency loss can naturally be applied between the prediction from the original features and the refined features to regularize the network as shown in Fig. 1b.

The final model seamlessly combines our novel feature-based augmentation with conventional image-based augmentation for consistency regularization, which is applied to data augmented from both sources. Despite the simplicity of the method, we find this achieves significant performance improvement. In summary, we compare our method with other highly relevant SSL works in Table. 1.

3.1 Prototype Selection

In order to efficiently leverage the knowledge of other classes for feature refinement and augmentation, we propose to compactly represent the information of each class by clustering in the feature space. To select representative prototypes from the dataset, we propose to use K-Means clustering in the feature space to extract p_k cluster means as prototypes for *each class*. However, there are two technical challenges: (1) in an SSL setting, most images are unlabeled; (2) even if all the labels are available, it is still computationally expensive to extract features of all the images from the entire dataset before running K-Means.

To tackle these issues, as shown in Fig. 2, we collect features f_{xi} and pseudo-labels $\hat{y}_i$ already generated by the network at every iteration of the training loop, *i.e.,* no extra computation needed. In the recording loop, the pairs of pseudo label and features are detached from the computation graph and pushed into a memory bank for later usage. The prototypes are extracted by K-Means at every epoch when we go over the whole dataset. Finally, the feature refinement and augmentation module updates the prototypes with the newly extracted ones in the training loop. Even though the prototypes are extracted from the feature computed from the model a few iterations ago, as training progresses and the model gradually converges, the extracted prototypes fall on the correct cluster and are diverse enough to compactly represent the feature distribution per class.

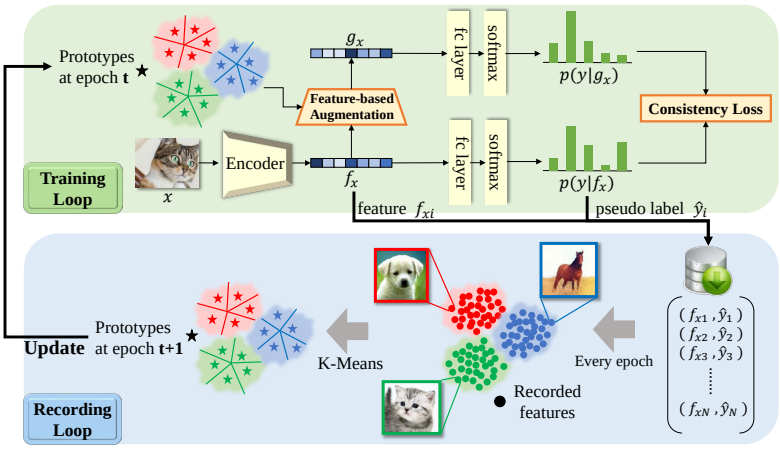


Fig. 2: A prototype recording loop that runs alongside the model training loop. The image features f_{xi} as well as their pseudo labels $\hat{y}_i$ already generated at each iteration of the training loop are collected and recorded in a memory bank as $(f_{xi}, \hat{y}_i)$ pairs. Once the training loop goes over the whole dataset, the recording loop will run K-Means to extract prototypes for each class, update the prototypes for feature-based augmentation, and clear the memory bank.

More analyses can be found in Sec. 4.4. Similar idea is concurrently explored in self-supervised learning by He et al. [30,10].

3.2 Learned Feature Augmentation

With a set of prototypes selected by the process described above, we propose a learned feature refinement and augmentation module via soft-attention [27] toward the set of selected prototypes. The proposed module refines and augments input image features in the feature space by leveraging the knowledge of prototypes, either within or outside of the same class, as shown in Fig. 3. The lightweight feature refinement and augmentation module composed of three fully connected layers is jointly optimized with other objectives and hence learns a reasonable feature-based augmentation to aid classification. We provide further analysis and discussion in Sec. 4.4.

Inspired by the attention mechanism [27], each input image feature *attends* to prototype features via attention weights computed by dot product similarity. The prototype features are then weighted summed by the attention weights and then fed back to the input image feature via residual connect for feature augmentation and refinement. Specifically, for an input image with extracted features f_x and the i-th prototype features $f_{p,i}$, we first project them into an embedding space by a learned function ϕ_e as $e_x = \phi_e(f_x)$ and $e_{p,i} = \phi_e(f_{p,i})$ respectively. We compute an attention weight w_i between e_x and $e_{p,i}$ as:

$$w_i = \text{softmax}(e_x^T e_{p,i}), \tag{1}$$

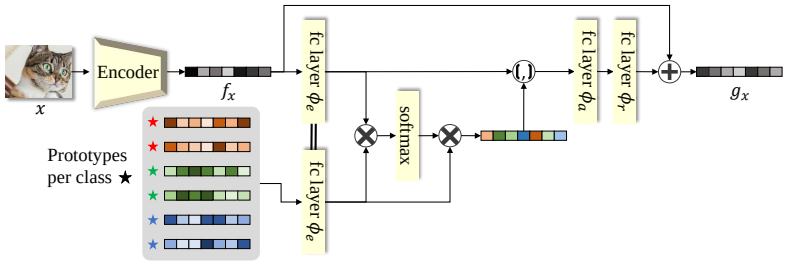


Fig. 3: **Feature-Based Augmentation:** The input image features are augmented by attention using extracted prototype features (Eq. 1), where the colors of ⋆ represent the classes of prototypes. The prototype features are calcluated via a weighted sum using the attention weights, concatenated with the image features, and then undergo a *fc* layer ϕ_a (Eq. 2) to produce attention features f_a. Finally, we use the attention features to refine and augment the input image features with a residual connection (Eq. 3).

where softmax$(\cdot)$ normalizes the dot product similarity scores across all prototypes. The information aggregated from the prototypes and passed to the image features for feature refinement and augmentation can then be expressed as a sum of prototype features weighted by the attention weights:

$$f_a = \text{relu}(\phi_a([e_x, \sum_i w_i e_{p,i}])), \tag{2}$$

where ϕ_a is a learnable function, and $[\cdot,\cdot]$ is a concatenation operation along the feature dimension. Finally, the input image features f_x is refined via a residual connection as:

$$g_x = \text{relu}(f_x + \phi_r(f_a)), \tag{3}$$

where g_x are the refined features of f_x, and ϕ_r is a learnable function.

The attention mechanism described above can be trivially generalized to multi-head attention as in [27]. In practice, we use multi-head attention, instead of single head for slightly better results. For simplicity, we define the feature refinement and augmentation process $AugF(\cdot)$ described above as $g_x = AugF(f_x)$.

3.3 Consistency Regularization

The learned $AugF$ module along with the selected prototypes provides an effective method for feature-based augmentation, which addresses the limitations of conventional data augmentation methods discussed previously. With the learned feature-based augmentation, we can naturally apply a consistency loss between the prediction of unaugmented features f_x and augmented features g_x.

However, given a classifier $p = Clf(f)$, which prediction should we use as pseudo-label, $p_g = Clf(g_x)$ or $p_f = Clf(f_x)$? We investigate this problem in

Sec. 4.4 and find that $AugF$ is able to *refine* the input features for better representation, thus generating better pseudo-labels. Therefore, we compute pseudo-label p_g on the refined feature g_x by $p_g = Clf(g_x)$. The feature-based consistency loss can be computed as: $\mathcal{L}_{con} = \mathcal{H}(p_g, Clf(f_x))$. We can easily extend $\mathcal{L}_{con}$to work seamlessly with traditional augmentation methods, *i.e.,* traditional data augmentation and Mixup. For simplicity, we will illustrate with only data augmentation, but Mixup can be easily adapted. Inspired by Berthelot et al. [3], we generate a weakly augmented image x and its strongly augmented copy $\hat{x}$. The pseudo-label is computed on the weakly augmented image x that undergoes feature-based augmentation and refinement for better pseudo-labels as $p_g = Clf(AugF(Enc(x)))$. We can then compute two consistency losses on the strongly augmented data $\hat{x}$, one with $AugF$ applied and the other without:

$$\mathcal{L}_{con\text{-}g} = \mathcal{H}(p_g, Clf(AugF(Enc(\hat{x}))) \tag{4}$$

$$\mathcal{L}_{con\text{-}f} = \mathcal{H}(p_g, Clf(Enc(\hat{x}))) \tag{5}$$

Since the pseudo-label p_g is computed on the image undergoing weak data augmentation and feature-based augmentation, the regularization signal of $\mathcal{L}_{con\text{-}g}$ and $\mathcal{L}_{con\text{-}f}$ comes from both image-based and feature-based augmentation.

3.4 Total Loss

Consistency regularization losses $\mathcal{L}_{con\text{-}g}$ and $\mathcal{L}_{con\text{-}f}$ in Eq. 4 and 5 are applied on unlabeled data. For labeled image x with label y, a regular classification loss can be applied:

$$\mathcal{L}_{clf} = \mathcal{H}(y, Clf(AugF(Enc(x)))) \tag{6}$$

Therefore, the total loss can be written as: $\mathcal{L}_{clf} + \lambda_g \mathcal{L}_{con\text{-}g} + \lambda_f \mathcal{L}_{con\text{-}f}$. Where λ_g and λ_f are weights for $\mathcal{L}_{con\text{-}g}$ and $\mathcal{L}_{con\text{-}f}$ losses respectively.

4 Experiments

4.1 Datasets

Standard SSL datasets. We conduct experiments on commonly used SSL datasets: SVHN [19], CIFAR-10 [14], CIFAR-100 [14], and mini-ImageNet [22]. Following the standard approach in SSL, we randomly choose a certain number of labeled samples as a small labeled set and discard the labels for the remaining data to form a large unlabeled set. Our proposed method is tested under various amounts of labeled samples. SVHN is a dataset of 10 digits, which has about 70k training samples. CIFAR-10 and CIFAR-100 are natural image datasets with 10 and 100 classes respectively. Both dataset contains 50k training samples. For mini-ImageNet, we follow [13,1] to construct the mini-ImageNet training set. Specifically, given a predefined list of 100 classes [22] from ILSVRC [23], 500 samples are selected randomly for each class, thus forming a training set of 50k samples. The samples are center-cropped and resized to 84x84 resolution. We

Table 2: Comparison on CIFAR-100 and mini-imageNet. Numbers represent error rate in three runs. For fair comparison, we use the same model as other methods: CNN-13 for CIFAR-100 and ResNet-18 for mini-ImageNet.

	CIFAR-100		**mini-ImageNet**	
	# Labeled samples		# Labeled samples	
Method	4,000	10,000	4,000	10,000
Π-model [24]	-	39.19 ± 0.36	-	-
SNTG [17]	-	37.97 ± 0.29	-	-
SSL with Memory [5]	-	34.51 ± 0.61	-	-
Deep Co-Training [21]	-	34.63 ± 0.14	-	-
Weight Averaging [2]	-	33.62 ± 0.54	-	-
Mean Teacher [25]	45.36 ± 0.49	36.08 ± 0.51	72.51 ± 0.22	57.55 ± 1.11
Label Propagation [13]	43.73 ± 0.20	35.92 ± 0.47	70.29 ± 0.81	57.58 ± 1.47
PLCB [1]	37.55 ± 1.09	32.15 ± 0.50	56.49 ± 0.51	46.08 ± 0.11
FeatMatch (Ours)	**31.06 ± 0.41**	**26.83 ± 0.04**	**39.05 ± 0.06**	**34.79 ± 0.22**

then follow the same standard procedure and construct a small labeled set and a large unlabeled set from the 50k training samples.

SSL under domain shift. In another realistic setting, we argue that the unlabeled data may come from a domain different from that of the target labeled data. For instance, given a small set of labeled natural images of animals, the large unlabeled set may also contain paintings of animals. To investigate the effect of domain shift in the unlabeled set, we proposed a new SSL task based on the DomainNet dataset [20], which contains 345 classes of images coming from six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

We use the *Real* domain as our target. Five percent of the data from the Real domain are kept as the target labeled set, and the rest are the target unlabeled set. We select *Clipart*, *Painting*, *Sketch*, and *Quickdraw* as shifted domains. To modulate the level of domain shift in the unlabeled data, we propose a parameter r_u that controls the ratio of unlabeled data coming from the target Real domain or the shifted domains. Specifically, r_u percent of target Real unlabeled set is replaced with data uniformly drawn from the shifted domains. By formulating the problem this way, the amount of unlabeled data remains constant. The only factor that affects the performance of the proposed method is the ratio between in-domain data and shifted domain data in the unlabeled set.

We randomly reserve 1% of data from the Real domain as the validation set. The final result is reported on the test set of the Real domain, with the model selected on the reserved validation set. The images are center-cropped and resized to 128x128 resolution, and the model we use is the standard ResNet-18 [11]. There are around 120k training samples, which is more than twice larger than the standard SSL datasets such as CIFAR-10 and CIFAR-100. For a fair comparison, we fix *all* hyper-parameters across experiments of different r_u to truly assess the robustness of proposed methods toward domain shift in the unlabeled data.

Hyper-parameters. We tune the hyper-parameters on CIFAR-10 with 250 labels with a validation set held-out from the training set. Our method is not

Table 3: Comparison between the image-based baseline with our proposed feature-based augmentation method on DomainNet with 1) unlabeled data coming from the same domain as the labeled target ($r_u = 0\%$), and 2) half of unlabeled data coming from the same domain as the labeled target and the other half from shifted domains ($r_u = 50\%$). Numbers are error rates across 3 runs.

Method (5% labeled samples)	$r_u = 0\%$	$r_u = 50\%$
(Semi-supervised) Baseline	56.63 ± 0.17	65.82 ± 0.07
FeatMatch (Ours)	40.66 ± 0.60	54.01 ± 0.66
Supervised baseline (5% labeled samples, lower bound,)	77.25 ± 0.52	
Supervised baseline (100% labeled samples, upper bound)	31.91 ± 0.15	

sensitive to the hyper-parameters, which are kept fixed across *all* the datasets and settings. Please see the supplementary for more implementation details and the values of hyper-parameters.

4.2 Results

We first show our results on CIFAR-100 and mini-ImageNet with 4k and 10k labels in Table 2. Our method consistently improves over state of the arts by large margins, with about absolute 5% on CIFAR-100 with 4k labels and 17% on mini-ImageNet with 4k labels.

In Table 3, we show our results on the larger dataset of DomainNet setting, which contains unlabeled data coming from other shifted domains. It can be clearly seen that in the setting of $r_u = 50\%$, where 50% of the unlabeled data are coming from other shifted domains, the performance drops by a large margin compared with the setting of $r_u = 0\%$, where all the unlabeled data are coming from the same domain as the target labeled set. Nevertheless, our proposed feature-based augmentation method improves over supervised baseline by absolute 36% error rate when $r_u = 0\%$ and 23% when $r_u = 50\%$. When compared to the conventional image-based augmentation baseline, we improves by 12% when $r_u = 50\%$ and 16% when $r_u = 0\%$.

In Table 4, we show the comparison of our method with other SSL methods on standard CIFAR-10 and SVHN datasets. Our method achieves comparable results with the current state of the art, ReMixMatch, even though 1) we start from a lower baseline and 2) our method is much simpler (*e.g.,* no class distribution alignment and no self-supervised loss), as compared in Table 1. Our proposed feature-based augmentation method is complementary to image-based methods and can be easily integrated to further improve the performance.

4.3 Ablation Study

In the ablation study, we are interested in answering the following questions: 1) what is the effectiveness of the two proposed consistency losses – $\mathcal{L}_{con\text{-}f}$ (Eq. 5) and $\mathcal{L}_{con\text{-}g}$ (Eq. 4). 2) how much of the improvement is from the proposed feature-based augmentation method over the image-based augmentation baseline? For

Table 4: Comparison on CIFAR-10 and SVHN. Numbers represent error rate across three runs. The results reported in the first block with CNN-13 model [15,18] are from the original paper. The results reported in the second block with wide ResNet (WRN) are reproduced by [4,3].

		CIFAR-10			**SVHN**		
		# Labeled samples			# Labeled samples		
Method	Model (param.)	250	1,000	4,000	250	1,000	4,000
SSL with Memory [5]	CNN-13 (3M)	-	-	11.91± 0.22	8.83	4.21	-
Deep Co-Training [21]		-	-	8.35 ± 0.06	-	3.29 ± 0.03	-
Weight Averaging [2]		-	15.58 ± 0.12	9.05 ± 0.21	-	-	-
ICT [29]		-	15.48 ± 0.78	7.29 ± 0.02	4.78 ± 0.68	3.89 ± 0.04	-
Label Propagation [13]		-	16.93 ± 0.70	10.61 ± 0.28	-	-	-
SNTG [17]		-	18.41 ± 0.52	9.89 ± 0.34	4.29 ± 0.23	3.86 ± 0.27	-
PLCB [1]		-	6.85 ± 0.15	5.97 ± 0.15	-	-	-
Π-model [24]	WRN (1.5M)	53.02 ± 2.05	31.53 ± 0.98	17.41 ± 0.37	17.65 ± 0.27	8.60 ± 0.18	5.57 ± 0.14
PseudoLabel [16]		49.98 ± 1.17	30.91 ± 1.73	16.21 ± 0.11	21.16 ± 0.88	10.19 ± 0.41	5.71 ± 0.07
Mixup [33]		47.43 ± 0.92	25.72 ± 0.66	13.15 ± 0.20	39.97 ± 1.89	16.79 ± 0.63	7.96 ± 0.14
VAT [18]		36.03 ± 2.82	18.68 ± 0.40	11.05 ± 0.31	8.41 ± 1.01	5.98 ± 0.21	4.20 ± 0.15
Mean Teacher [25]		47.32 ± 4.71	17.32 ± 4.00	10.36 ± 0.25	6.45 ± 2.43	3.75 ± 0.10	3.39 ± 0.11
MixMatch [4]		11.08 ± 0.87	7.75 ± 0.32	6.24 ± 0.06	3.78 ± 0.26	3.27 ± 0.31	2.89 ± 0.06
ReMixMatch [3]		**6.27 ± 0.34**	**5.73 ± 0.16**	5.14 ± 0.04	**3.10 ± 0.50**	**2.83 ± 0.30**	**2.42 ± 0.09**
FeatMatch (Ours)		7.50 ± 0.64	**5.76 ± 0.07**	**4.91 ± 0.18**	**3.34 ± 0.19**	**3.10 ± 0.06**	2.62 ± 0.08

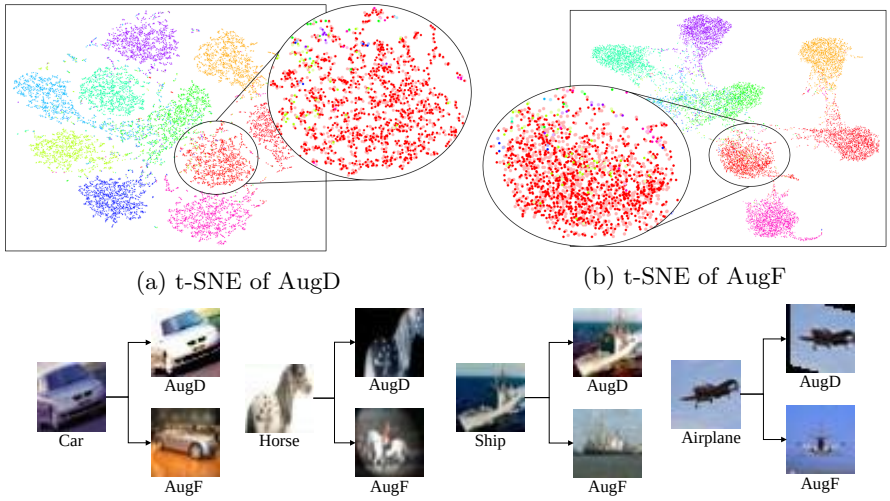
Table 5: Ablation study on CIFAR-10 with various amount of labeled samples.

						#Labeled samples		
Experiment	Image-Based Augmentation	Feature-Based Augmentation	$\mathcal{L}_{con\text{-}f}$	$\mathcal{L}_{con\text{-}g}$	$\mathcal{L}_{con}$	250	1,000	4,000
Baseline	✔	-	-	-	✔	19.55 ± 1.58	9.04 ± 1.00	6.08 ± 0.16
w/o $\mathcal{L}_{con\text{-}f}$	✔	✔	-	✔	-	18.57 ± 3.19	8.38 ± 0.35	6.09 ± 0.16
w/o $\mathcal{L}_{con\text{-}g}$	✔	✔	✔	-	-	8.19 ± 1.74	6.07 ± 0.46	5.16 ± 0.30
FeatMatch (Ours)	✔	✔	✔	✔	-	7.90 ± 0.49	5.94 ± 0.16	5.00 ± 0.21

the image-based augmentation baseline, the *AugF* module is completely removed and thus the consistency regularization comes only from image-based augmentation. This is also the same image-based augmentation baseline that our final model with feature-based augmentation builds upon. The ablation study is conducted on CIFAR-10 with various amount of labeled samples (Table 5).

We can see from Table 5 that our image-based augmentation baseline achieves good results but only on cases where there are more labeled samples. We conjecture this is because the aggressive data augmentation applied to training images makes the training unstable. Nevertheless, our baseline performance is still competitive with respect to other image-based augmentation methods in Table 4 (though slightly worse than MixMatch). By adding our proposed *AugF* module($\mathcal{L}_{con\text{-}f}$ and $\mathcal{L}_{con\text{-}g}$) for feature refinement and augmentation on top of the image-based augmentation baseline, the performance improves over baseline consistently, especially for 250 labels.

We can also see that $\mathcal{L}_{con\text{-}f}$ plays a more important role than $\mathcal{L}_{con\text{-}g}$, though our final model with both loss terms achieves the best result. In both $\mathcal{L}_{con\text{-}f}$ and



(a) t-SNE of AugD

(b) t-SNE of AugF

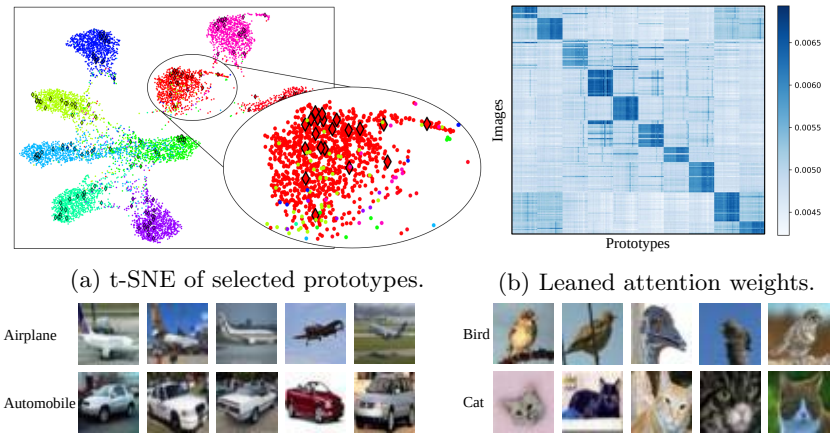
(c) Augmented images of AugD vs. AugF

Fig. 4: (***a***) We jointly compute and plot t-SNE of input unaugmented image features (dimmer color) and image-based augmented features (brighter color). (***b***) We also jointly compute and plot t-SNE of input unaugmented image features (dimmer color) and feature-based augmented features (brighter color) with the exact same t-SNE parameters with (a). (***c***) To concretely visualize the augmented feature, we find their nearest image neighbor in the feature space and compare against the image-based augmentation method side by side.

$\mathcal{L}_{con\text{-}g}$, the pseudo-labels are computed from the features undergone feature-based augmentation. The only difference is which prediction we're driving to match the pseudo-label: 1) the prediction from the feature undergone both *AugD* and *AugF* (by $\mathcal{L}_{con\text{-}g}$ loss), or 2) the prediction from the feature undergone only *AugD* (by $\mathcal{L}_{con\text{-}f}$ loss)? As claimed in Sec. 3.3 and analyzed in Sec. 4.4, *AugF* is able to refine input image features for better representation and pseudo-labels of higher quality. Therefore, matching the slightly worse prediction from the feature undergone only *AugD* (by $\mathcal{L}_{con\text{-}f}$ loss) induces a stronger consistency regularization. This explains why $\mathcal{L}_{con\text{-}f}$ improves performance more crucially.

4.4 Analysis

What augmentation does *AugF* learn? We compare the feature distribution via t-SNE 1) between input unaugmented image features and image-based augmented features in Fig. 4a, and 2) between input unaugmented image features and feature-based augmented features in Fig. 4b. In Fig. 4a, some local small clusters are captured by t-SNE and can be found in the zoomed sub-figure. This indicates that *AugD* can only perturb data locally, and fail to produce stronger augmentation for more effective consistency regularization in the fea-



(a) t-SNE of selected prototypes.

(b) Leaned attention weights.

(c) Nearest image neighbors of prototypes

Fig. 5: (***a***) In the t-SNE plot, the extracted prototypes ($\diamond$) fall on the correct clusters and are able to compactly represent the cluster. (***b***) We visualize the learned attention weights from a batch of images toward prototypes. The images and prototypes are sorted by their classes for ease of illustration. As can be seen, images have higher attention weights to the prototypes with the same class. (***c***) We find the prototypes' nearest image neighbors in the feature space. The prototypes compactly represent a diverse sets of images in each class.

ture space. In Fig. 4b, we can see $AugF$ indeed learns to augment and refine features. Furthermore, the learned augmentation preserves semantic meaning as the augmented features still fall in the correct cluster. In the zoomed figure, we can see that the perturbed features distribute more uniformly and no local small clusters could be found. This indicates that $AugF$ can produce stronger augmentation for more effective consistency regularization in the feature space.

To have a more concrete sense of the learned feature-based augmentation ($AugF$), we show the augmented feature's nearest image neighbor in the feature space. Some sample results are shown in Fig. 4c, with the comparison to image-based augmentation ($AugD$) side by side. As shown in the figure, $AugF$ is capable of transforming features in an abstract way, which goes beyond simple textural and geometric transformation as $AugD$ does. For instance, it is able to augment data to different poses and backgrounds, which could be challenging for conventional image-based augmentation methods.

What other reason does *AugF* improve model performance? We hypothesize that one other reason why our method can improve performance is that $AugF$ module is capable of refining input image features for better representation by the extracted prototypes, and thus provides better pseudo-labels. The consistency regularization losses then drive the network's prediction to match the target pseudo-labels of higher quality, leading to overall improvement. With this hypothesis, we expect classification accuracy to be higher for features after

feature refinement. To verify, we remove $\mathcal{L}_{con\text{-}f}$ loss and retrain. The accuracy of pseudo-labeling from the features refined by $AugF$ is on average $0.5 - 1.0\%$ higher. This confirms our hypothesis that $\mathcal{L}_{con\text{-}f}$ drives the feature encoder to learn a better feature representation refined by $AugF$.

The reader may wonder: why doesn't $AugF$ learn a shortcut solution of identity mapping to minimize $\mathcal{L}_{con\text{-}f}$ and $\mathcal{L}_{con\text{-}g}$? As can be seen from Fig. 4, $AugF$ does *not* learn an identity mapping. Although learning an identity mapping may be a shortcut solution for minimizing $\mathcal{L}_{con\text{-}f}$ and $\mathcal{L}_{con\text{-}g}$, it is not the case for the classification loss $\mathcal{L}_{clf}$ (Eq. 6). This finding implicitly confirms our hypothesis that there is extra information from the prototypes that $AugF$ can leverage to refine the feature representation for higher (pseudo-label) classification accuracy.

What does *Aug* do internally? In Fig. 5a and 5c, we can see that even though our proposed prototype extraction method only uses simple K-Means to extract prototypes of each class based on potentially noisy pseudo-labels, and features recorded several iterations ago, our prototype selection method can still successfully extract a diverse set of prototypes per class. Moreover, in Fig. 5b, the attention mechanism inside $AugF$ learns to attend to prototypes that belong to the same class with the input image feature. Note that there is no loss term specific for $AugF$, as it is simply jointly optimized with the standard classification and consistency regularization loss from semi-supervised learning.

5 Conclusion

We introduce a method to jointly learn a classifier and feature-based refinement and augmentations which can be used within existing consistency-based SSL methods. Unlike traditional image-based transformations, our method can learn complex, feature-based transformations as well as incorporate information from class-specific prototypical representations extracted in an efficient manner (specifically using a memory bank). Using this method, we show comparable results as the current state of the art for smaller datasets such as CIFAR-10 and SVHN, and significant improvements on datasets with a large number of categories (*e.g.*, 17.44% absolute improvement on mini-ImageNet). We also demonstrate increased robustness to out-of-domain unlabeled data, which is an important real-world problem, and perform ablations and analysis to demonstrate the learned feature transformation and extracted prototypical representations.

Acknowledgement

This work was funded by DARPA's Learning with Less Labels (LwLL) program under agreement HR0011-18-S-0044 and DARPA's Lifelong Learning Machines (L2M) program under Cooperative Agreement HR0011-18-2-0019.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. arXiv preprint arXiv:1908.02983 (2019) 4, 8, 9, 11
2. Athiwaratkun, B., Finzi, M., Izmailov, P., Wilson, A.G.: Improving consistency-based semi-supervised learning with weight averaging. arXiv preprint arXiv:1806.05594 **2** (2018) 9, 11
3. Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In: Proc. International Conference on Learning Representations (ICLR) (2020) 1, 4, 8, 11
4. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems. pp. 5050–5060 (2019) 1, 4, 11
5. Chen, Y., Zhu, X., Gong, S.: Semi-supervised deep learning with memory. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 268–283 (2018) 9, 11
6. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018) 1
7. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 113–123 (2019) 4
8. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. arXiv preprint arXiv:1909.13719 (2019) 1, 4
9. Guo, H., Mao, Y., Zhang, R.: Mixup as locally linear out-of-manifold regularization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3714–3722 (2019) 4
10. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019) 6
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 9
12. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: AugMix: A simple data processing method to improve robustness and uncertainty. Proceedings of the International Conference on Learning Representations (ICLR) (2020) 1, 4
13. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5070–5079 (2019) 8, 9, 11
14. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009) 8
15. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: Proc. International Conference on Learning Representations (ICLR) (2017) 1, 4, 11
16. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML. vol. 3, p. 2 (2013) 11
17. Luo, Y., Zhu, J., Li, M., Ren, Y., Zhang, B.: Smooth neighbors on teacher graphs for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8896–8905 (2018) 9, 11

18. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence **41**(8), 1979–1993 (2018) 1, 4, 11
19. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011) 8
20. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1406–1415 (2019) 3, 9
21. Qiao, S., Shen, W., Zhang, Z., Wang, B., Yuille, A.: Deep co-training for semi-supervised image recognition. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–152 (2018) 1, 4, 9, 11
22. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: International Conference on Learning Representations (ICLR) (2017) 8
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015) 8
24. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems. pp. 1163–1171 (2016) 1, 4, 9, 11
25. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in neural information processing systems. pp. 1195–1204 (2017) 1, 4, 9, 11
26. Thulasidasan, S., Chennupati, G., Bilmes, J.A., Bhattacharya, T., Michalak, S.: On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In: Advances in Neural Information Processing Systems. pp. 13888–13899 (2019) 4
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 6, 7
28. Verma, V., Lamb, A., Beckham, C., Courville, A., Mitliagkis, I., Bengio, Y.: Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. stat **1050**, 13 (2018) 4, 5
29. Verma, V., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825 (2019) 4, 11
30. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3733–3742 (2018) 6
31. Yu, B., Wu, J., Ma, J., Zhu, Z.: Tangent-normal adversarial regularization for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10676–10684 (2019) 4
32. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6023–6032 (2019) 4
33. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: Proc. International Conference on Learning Representations (ICLR) (2018) 1, 4, 11