Journal of Machine Learning Research 23 (2022) 1-27 Submitted 5/21; Revised 6/22; Published 9/22

A Closer Look at Embedding Propagation for Manifold Smoothing

Diego Velazquez DIEGOALEJANDRO.VELAZQUEZ@UAB.CAT
Visual Tagging Services and Computer Vision Center, Barcelona, Spain

Pau Rodríguez PAU.RODRIGUEZ@SERVICENOW.COM
ServiceNow Research, Montreal, Canada

Josep M. Gonfaus PEP.GONFAUS@GMAIL.COM
Visual Tagging Services, Barcelona, Spain

F.Xavier Roca XAVIER.ROCA@UAB.CAT
Computer Vision Center and Univ. Autònoma de Barcelona, Barcelona, Spain

Jordi Gonzàlez XAVIER.ROCA@UAB.CAT
Computer Vision Center and Univ. Autònoma de Barcelona, Barcelona, Spain

Editor: Stefan Harmeling

Abstract

Supervised training of neural networks requires a large amount of manually annotated data and the resulting networks tend to be sensitive to out-of-distribution (OOD) data. Self- and semi-supervised training schemes reduce the amount of annotated data required during the training process. However, OOD generalization remains a major challenge for most methods. Strategies that promote smoother decision boundaries play an important role in out-of-distribution generalization. For example, embedding propagation (EP) for manifold smoothing has recently shown to considerably improve the OOD performance for few-shot classification. EP achieves smoother class manifolds by building a graph from sample embeddings and propagating information through the nodes in an unsupervised manner. In this work, we extend the original EP paper providing additional evidence and experiments showing that it attains smoother class embedding manifolds and improves results in settings beyond few-shot classification. Concretely, we show that EP improves the robustness of neural networks against multiple adversarial attacks as well as semi- and self-supervised learning performance.

Keywords: regularization, semi-supervised learning, self-supervised learning, adversarial robustness, few-shot classification

1. Introduction

Deep convolutional networks have achieved state-of-the-art performance in many machine learning tasks, like image classification (Krizhevsky et al., 2012), object detection (Long et al., 2015), and image segmentation (Ren et al., 2015). However, due to the huge number of parameters of these models, a considerably large amount of training data is typically required to prevent overfitting and achieve generalization. Despite having visual data easily available in large quantities, the number of reliably and precisely annotated images is relatively scarce and, in most cases, such annotations performed by humans are very expensive to obtain (Kuznetsova et al., 2020). In order to alleviate the lack of enough labelled data, researchers have designed different machine learning frameworks that minimize the need of annotations: transfer learning where the knowledge acquired by a deep learning model trained on one task is used to retrain the model on another similar task (Yosinski et al., 2014); few-shot learning (FSL) which uses prior knowledge in order to generalize to unseen samples from a minimal amount of annotated training data (Fei-Fei et al., 2006); and other popular methodologies like semi-supervised learning (Chapelle and Zien, 2005) and unsupervised learning (He et al., 2020) which leverage unlabeled data to improve a model's performance on a downstream task.

However, one of the key challenges still present in all of the aforementioned training frameworks is their ability to generalize beyond the original training distribution, or out-of-distribution (OOD) generalization. In other words, when the test distribution differs too much from the training distribution, models tend to provide incorrect predictions with a strikingly high confidence. This weakness of neural models is exploited by the so-called adversarial attacks (Szegedy et al., 2013). Adversarial attacks generate perturbations on the input image that are imperceptible to the human eye but make trained models with near perfect performance yield to incorrect predictions with high confidence. Fortunately, different strategies have been proposed to increase the robustness of deep models against adversarial attacks, like promoting smoother decision boundaries with techniques such as manifold mixup (Verma et al., 2019a). The idea behind this technique is to push the decision boundaries far away from the data, and it has proven to work well against adversarial perturbations while improving results in semi- and self-supervised settings (Chapelle and Zien, 2005).

A recent work on few-shot learning proposes an embedding propagation (EP) technique (Rodríguez et al., 2020) for improving OOD generalization. EP computes a set of interpolations from the network output features based on their similarity in a graph. This graph is built taking into account pairwise similarities of the features using the radial basis function (RBF). Thus, EP is non-parametric and it can be applied on top of any feature extractor but also as part of a network, referred to EPNet in (Rodríguez et al., 2020). This way, EP allows any neural network to generate a regularized manifold for both training and testing. Since using interpolated embeddings result in smoother class embedding manifolds and an increased robustness to noise, and these properties have been shown to be important for generalization (Bartlett and Shawe-Taylor, 1999; Lee et al., 1995; Verma et al., 2019a) and semi-supervised learning (Chapelle and Zien, 2005), the question is whether EP would be also beneficial beyond the few-shot learning scenario, and how EP could be efficiently applied in other learning frameworks.

This paper aims to answer the previous question by extending (Rodríguez et al., 2020) showing that, in addition to few-shot learning, EP can be used to improve adversarial robustness and semi-supervised/self-supervised learning performances of classifiers. We also present additional evidence showing how EP achieves smoother class embedding manifolds. Therefore, the scope of this work differs from (Rodríguez et al., 2020) in that we do not aim to achieve state-of-the-art few-shot learning performance but to improve our understanding of EP. In summary, the contributions of this paper are: (i) we show that EP increases the smoothness of the classification surface as measured with the Laplacian; (ii) EP increases adversarial robustness; (iii) EP improves the performance of self- and semi-supervised learning algorithms by acting as a natural hard negative mining method; and (iv) EP improves few-shot learning classification performance.

2. Related Work

In this paper, we study the effect of EP as a manifold regularization method and extend its use beyond few-shot learning to adversarial attacks, self- and semi-supervised learning. So we next review the literature for each one of these fields.

Regularization is a major area of research in machine learning (Srivastava et al., 2014; Wan et al., 2013). Whether it is directly enforcing constraints on networks weights (Ioffe and Szegedy, 2015; Salimans and Kingma, 2016; Rodríguez et al., 2016), or regularizing the embedding manifold (Belkin et al., 2006; Tokozume et al., 2018; Zhang et al., 2018), regularization has been shown to aid models generalize. For example, TPN (Liu et al., 2019b) introduces a meta-learning approach to label propagation by learning a graph construction module that exploits the manifold structure in the data. EPNet (Rodríguez et al., 2020) attempts to smooth the class embedding manifold by applying an embedding propagation operation on extracted features. Similarly, manifold mixup (Verma et al., 2019a) leverages interpolations of the hidden layers of the network as an additional training signal and it has been shown to improve adversarial robustness and work well in a self-supervised setting. Techniques similar to embedding propagation have also been applied as a message passing algorithms in graph neural networks (Klicpera et al., 2018; Xhonneux et al., 2020), here we recast it as a regularization technique.

Adversarial attacks were introduced by (Szegedy et al., 2013). The authors showed that convolutional neural networks are extremely sensitive to small perturbations in the input image. So visual perturbations imperceptible to the human eye are sufficient to cause the model to incorrectly misclassify an example with very high confidence.

Adversarial attacks can be classified into three categories depending on the knowledge of the attacker about the targeted model: white-box, gray-box and black-box. In a white box setting the adversary has full knowledge of the target model, including its parameters and architecture, so the attacker can easily craft adversarial examples by any means. In a gray-box threat model, only the structure of the target model is known to the attacker. Lastly, in a black-box threat model only the task of the target model is known (Papernot et al., 2016a), thus the attacker has to resort to query-level access to the black-box model in order to generate adversarial examples (Chakraborty et al., 2018). Since the introduction of FGSM (Szegedy et al., 2013), many adversarial attacks have been proposed, such as projected gradient descent (PGD) (Madry et al., 2017), Jacobian-based saliency map attack (JSMA) (Papernot et al., 2016b) and Fast Adaptive Boundary attack (FAB) (Croce and Hein, 2020).

Semi-supervised learning aims to leverage a set of unlabeled data in order to improve the performance on a downstream task (Chapelle and Zien, 2005). (Berthelot et al., 2019) categorize the different semi-supervised learning methods into three categories: consistency regularization, entropy minimization, and traditional regularization, as detailed next.

Consistency regularization consists of performing extensive data augmentation (Cireşan et al., 2010; Simard et al., 2003) to expand the decision boundaries of classifiers, so that they remain consistent on unlabeled data (Laine and Aila, 2016; Sajjadi et al., 2016; Tarvainen and Valpola, 2017; Miyato et al., 2018).

Entropy minimization methods ensure that decision boundaries only pass through low-density regions, which is a common assumption in semi-supervised learning (Chapelle and Zien, 2005). This property is here enforced by minimizing the entropy of the model outputs on unlabeled data (Grandvalet et al., 2005; Miyato et al., 2018; Berthelot et al., 2019). Likewise, pseudo-label methods can reduce the entropy by directly discretizing the predictions of the model on unlabeled data (Lee, 2013).

Regularization techniques for semi-supervised learning constrain models to increase their bias in order to improve their generalization on unlabeled data (Zhang et al., 2016). The MixUp technique (Zhang et al., 2018) is a popular regularization method that has been leveraged in multiple works to improve semi-supervised learning performance. In essence, the prediction at an interpolation of unlabeled points is forced to be consistent with the interpolation of the predictions at those points, thus moving the decision boundary to low-density regions of the data distribution. This strategy has been applied for interpolation consistency training (ICT) (Verma et al., 2019b) and manifold mixup (Verma et al., 2019a). Similarly, the Mixmatch technique (Berthelot et al., 2019) uses MixUp to mix labeled and unlabeled data to produce pseudo-labels.

Embedding propagation is also considered as a MixUp strategy, since it leverages embedding interpolations based on a similarity graph, and intersects with the family of transductive semi-supervised learning methods (Vapnik, 1999). These methods consider the relationship between instances in the test set to predict them as a whole, improving the performance of classifiers in the low-data regime. Likewise, embedding propagation also considers the relationships between instances by forming a graph from the query samples in an episode.

Self-supervised learning methods train models on unlabeled data by minimizing a contrastive learning loss or by learning to solve a pretext task. Current state-of-the-art self-supervised learning methods such as MoCo (He et al., 2020; Chen et al., 2020c) and SimCLR (Chen et al., 2020a,b), are based on contrastive learning: these methods use contrastive losses to measure the similarities of sample pairs in representation space. Approaches based on pretext tasks propose to solve an artificially designed proxy task. The underlying assumption is that by solving this proxy task, the model will acquire general knowledge required to solve the downstream tasks. A wide range of pretext tasks have been proposed, e.g., colorization (Zhang et al., 2016; Ye et al., 2019), recovering corrupted input (denoising) (Vincent et al., 2008), forming pseudo-labels by transformations of a single image (Dosovitskiy et al., 2014), patch ordering (Doersch et al., 2015; Noroozi and Favaro, 2016) or tracking (Wang and Gupta, 2015).

Recent works (Cao et al., 2020; Chuang et al., 2020; Iscen et al., 2018; Ho and Vasconcelos, 2020; Wu et al., 2020; Xie et al., 2020) also investigate approaches around the selection of negative examples in self-supervised learning. (Kalantidis et al., 2020) proposed using the hardest existing negatives to synthesize additional hard negatives on the fly, i.e., directly in the feature space, by mixing two of the hardest negatives or by mixing the query itself with one of the hardest negatives. Similarly, applying embedding propagation in a self-supervised setting, also creates hard negatives and positives on the fly as a byproduct of the EP algorithm.

Few-shot learning denotes those methods that learn to solve a task from a very reduced set of labelled data. For example, one-shot classification methods learn a classifier with just one example per class. Most few-shot learning methods are included in two broad categories: meta-learning and transfer-learning. Meta-learning aims to learn a representation that can be robustly adapted to a new problem with few samples. For instance, authors in (Snell et al., 2017; Vinyals et al., 2016; Sung et al., 2018; Oreshkin et al., 2018) embed input data into a Hilbert space and perform distance-based classification. Another examples are those optimization-based approaches (Ravi and Larochelle, 2016; Finn et al., 2017; Yoon et al., 2019) which learn a good initialization that can be adapted to solve a specific problem in few optimization steps.

On the other hand, transfer-learning (Chen et al., 2019; Mensink et al., 2012) aims to learn generalisable representations from training data so that any new task can easily be solved with a simple classifier. Many of these approaches build on top of a pre-trained feature extractors (Rusu et al., 2018; Wang et al., 2019). For example, authors in (Mangla et al., 2019) introduced self-supervision to learn more transferable representations. Also, graph-based approaches (Hu et al., 2020b; Kim et al., 2019; Liu et al., 2019b) have been proposed to leverage the relationships between the samples in each episode by forming a graph and propagating information between nodes. In particular, Embedding Propagation (Rodríguez et al., 2020) is a graph-based approach that uses a non-parametric operation (Zhou et al., 2004) to propagate information between the nodes, achieving smoother decision boundaries and better few-shot generalization. In this work we show that EP offers improvement beyond few-shot learning and we extend it to other settings such as adversarial robustness and self- and semi- supervised learning. Different from (Rodríguez et al., 2020), the goal of this work is to delve deeper into the benefits of EP rather than achieving state-of-the-art performance.

3. Proposed Method

In this paper we extend the embedding propagation method introduced by (Rodríguez et al., 2020), whose basis is described in this section. Given a set of features $Z \in \mathbb{R}$ extracted from some input $X \in \mathbb{R}$ by a feature extractor $f : X \to Z$, EP maps those features to a set of interpolated features. Finally, the output of EP is fed into a classifier to label the images. (Rodríguez et al., 2020) found that EP smooths the classification surface by pushing the boundaries away from the data, improving generalization (Bartlett and Shawe-Taylor, 1999; Lee et al., 1995). The EP approach differs from label propagation (Zhou et al., 2004) and TPN (Liu et al., 2019b) in that EP is completely unsupervised see Figure 1. Furthermore TPN is a meta-learning approach, to label propagation, hence it requires learning a graph construction module beforehand. Next we describe the EP algorithm in more detail.

In the image classification domain, embedding propagation takes a set of feature vectors $\mathbf{z}_i \in \mathbb{R}^m, i \in 1..|Z|$, obtained from applying a feature extractor (CNN) to the input images. Then, it outputs a set of embeddings $\widetilde{\mathbf{z}}_i \in \mathbb{R}^m$ through the following two steps. Firstly, for each pair of features (i, j), the model computes the distance as $d_{ij}^2 = \|\mathbf{z}_i - \mathbf{z}_j\|_2^2$ and the

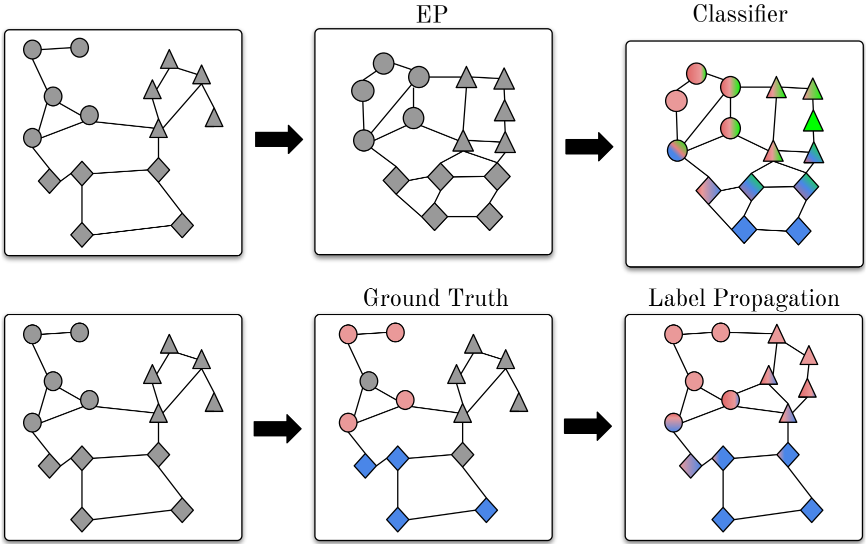


Figure 1: Illustration of the embedding propagation method, in comparison with label propagation (LP). The grey nodes represent unlabeled samples. Since EP is completely unsupervised, it does not requires labels to reorganize the manifold and create a smooth classification surface. In contrast, LP requires initial labels. Note how EP increases the similarity between all pairs of points and forces samples that are close together to have similar classification score. *Best viewed in color.*

adjacency matrix as $A_{ij} = \exp\left(-d_{ij}^2/\sigma^2\right)$, where σ^2 is a scaling factor and $A_{ii} = 0$, $\forall i$, as done in TPN (Liu et al., 2019b). The authors of the original paper chose $\sigma^2 = Var\left(d_{ij}^2\right)$ which was found to stabilize training.

Secondly, the Laplacian of the adjacency matrix is computed,

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \;\; D_{ii} = \textstyle\sum_j A_{ij}. \tag{1}$$

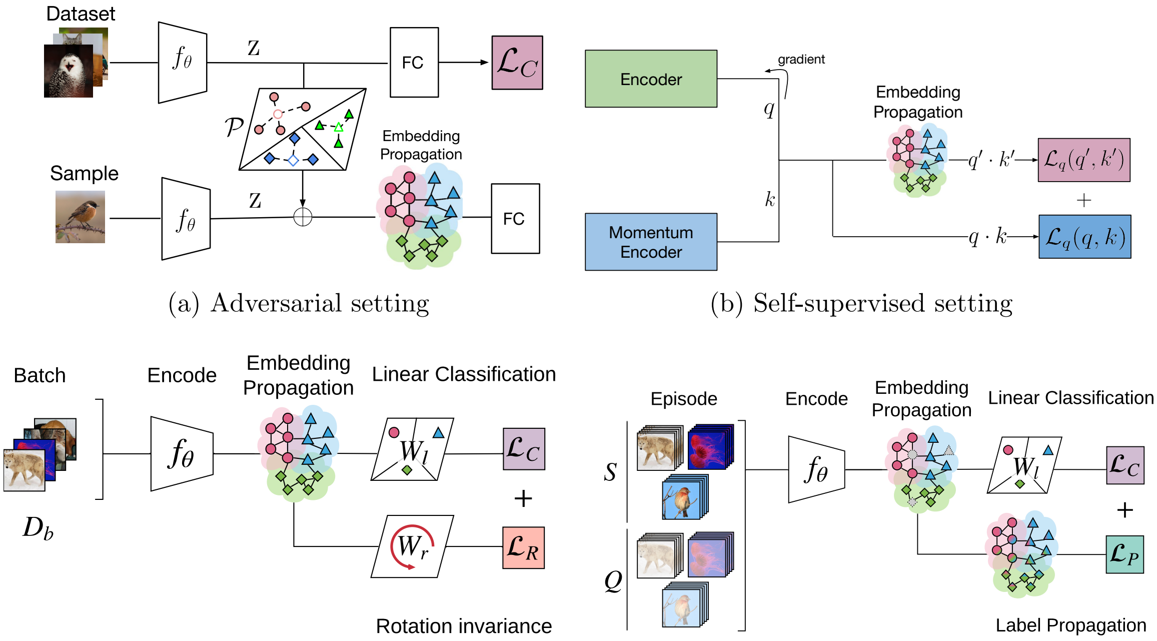
Finally, the propagator matrix P is obtained using the label propagation formula described in (Zhou et al., 2004) as,

$$P = (I - \alpha L)^{-1}, \tag{2}$$

where $\alpha \in \mathbb{R}$ is a scaling factor, and I is the identity matrix. As a result, the embeddings are obtained as follows,

$$\widetilde{\mathbf{z}}_i = \sum_j P_{ij} \mathbf{z}_j. \tag{3}$$

Notice that $\widetilde{\mathbf{z}}_i$ are now a weighted sum of their neighbors. Thus, we hypothesize that undesired noise in the feature vectors is reduced after being averaged out by the embedding propagation operation. This operation is simple to implement and compatible with a wide range of feature extractors and classifiers. Further, note that the computational complexity of our approach is $\mathcal{O}(n^2)$, which is similar to the complexity of the label propagation



(a) Adversarial setting

(b) Self-supervised setting

(c) Few-shot pretraining phase

(d) Few-shot episodic fine-tuning and evaluation

Figure 2: Overview of the EPNet training procedure across different tasks. **(a)** Non-transductive (inductive) version of the embedding propagation algorithm. First, the prototypes $\mathcal{P}$ are built during the last training epoch. Then, at test time, EP is applied on the prototypes along with a single test sample. **(b)** Integration of the embedding propagation algorithm in MoCo. During the pre-training phase EP is applied on the keys and queries and the output is used in a secondary contrastive loss $\mathcal{L}_q(q', k')$. For few-shot learning **(c, d)**, the model is trained to learn general feature representations using a standard classification loss $\mathcal{L}_C$ and an auxiliary rotation loss $\mathcal{L}_R$ (left). Then, the model is fine-tuned using episodic learning to learn to generalize to novel classes by minimizing the standard classification loss $\mathcal{L}_C$ and a label propagation loss $\mathcal{L}_P$ (right).

algorithm (Zhou et al., 2004) as discussed in (Wang et al., 2013). Further, note that the computational complexity of Eq. 2 is negligible for few-shot episodes Liu et al. (2019b) since the size of the episode is smallIscen et al. (2019).

Smoothness measure We use the Laplace operator Δ or Laplacian to measure the smoothness of the decision surface around a set of embeddings before and after applying the embedding propagation. The Laplacian is given by the sum of second partial derivatives of a function with respect to each independent variable:

$$\Delta f(x, y) = \sum_{i=1}^{n} \frac{\delta^2 f}{\delta x_i^2} + \frac{\delta^2 f}{\delta y_i^2}, \tag{4}$$

where x and y represent the two dimensions in the Cartesian coordinate frame and f is a classification function. Since we are only interested on the Laplacian around the decision boundary, we generate the minimal mesh that contains all the datapoints and use the discrete laplace operator in the form of a convolution:

$$\Delta f(x, y) = \mathbf{D}^2_{xy} * f, \tag{5}$$

where $*$ is the convolution operator and $\mathbf{D^2_{xy}}$ is the Laplacian convolution kernel (Jain et al., 1995). For each point in the grid, the absolute value of the magnitude of the Laplacian indicates a sharp change in the decision boundary. Thus, we approximate the total surface smoothness as the the definite integral of the Laplacian on the 2d grid. Since the grid is discrete, we compute smoothness $\mathcal{S}$ as the inverse of the summation over all the grid values:

$$\mathcal{S} = \sum_y \sum_x \frac{1}{1 + \left|\Delta f(x, y)\right|}. \tag{6}$$

4. Experiments

Next we present additional evidence of how EP smooths the classification surface and adapt it to different settings: adversarial attacks, self- and semi-supervised learning and few-shot learning. Although EP is applied at different stages of the machine learning pipeline for each of the following experiments (see Figure 2), the EP algorithm will remain unchanged across all experiments.

4.1 Datasets

***mini*Imagenet (Ravi and Larochelle, 2016)** consists of a subset of the Imagenet dataset (Russakovsky et al., 2015) comprised of 100 classes with 600 images per class. Classes are divided in three disjoint sets of 64 base classes, 16 for validation and 20 novel classes.

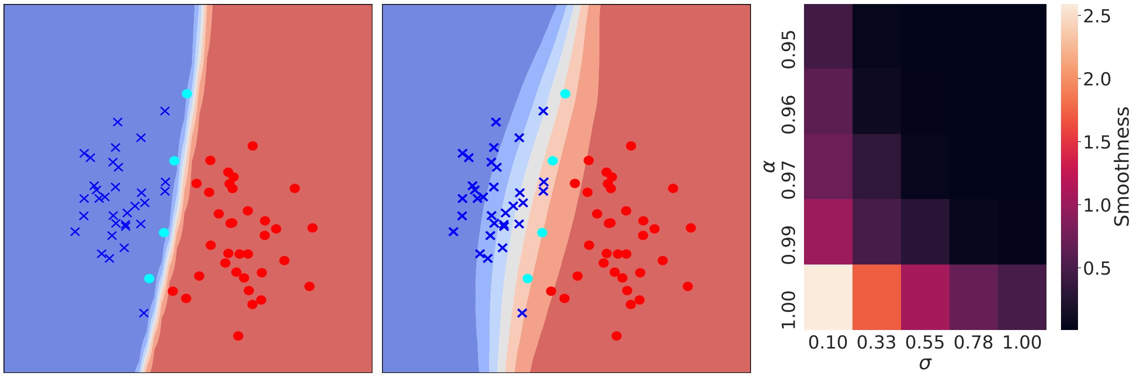
***tiered*Imagenet (Ren et al., 2018)** is a more challenging subset of the Imagenet dataset (Russakovsky et al., 2015) where class subsets are chosen from supersets of the wordnet hierarchy. The top hierarchy has 34 super-classes, which are divided into 20 base (351 classes), 6 validation (97 classes) and 8 novel (160 classes) categories.

CIFAR10 (Krizhevsky et al., 2009) is comprised of 60,000 32 × 32 colour images divided into 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.

CIFAR100 (Krizhevsky et al., 2009) is just like the CIFAR10 dataset, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class.

MNIST (LeCun and Cortes, 2010) is a dataset of 70,000 small 28 × 28 pixels grayscale images of handwritten single digits between 0 and 9 (10 classes). There are 60,000 examples in the training dataset and 10,000 in the test dataset.

Fashion-MNIST (Xiao et al., 2017) is a dataset of Zalando's article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28 × 28 grayscale image, associated with a label from 10 classes.



(a) Without EP (b) With EP (c) Manifold smoothness

Figure 3: **(a, b)** Comparison of the class embedding manifold without and with embedding propagation on a toy classification dataset. Notice how without EP **(a)** the adversarial examples (cyan) cross the decision boundary and are misclassified, the smoothness achieved by applying EP **(b)** at test time on the same classifier prevents this misclassification. **(c)** Effect of α and σ on the smoothness of the class embedding manifold. The higher the α and the smaller the value of σ the smoother the manifold becomes, notice the lower diagonal of the matrix. Smoothness is given by Equation 6

STL-10 (Coates et al., 2011) is a dataset of 96 × 96 color images, categorized into 10 classes, with 500 training images and 800 test images per class. The dataset also has 100,000 unlabeled images for unsupervised learning. Images were drawn from Imagenet labeled examples.

4.2 Manifold smoothness

The embedding propagation algorithm is based on the closed-form solution of the label propagation algorithm proposed by (Zhou et al., 2004). One of the main advantages of label propagation is that the decision boundaries are smooth with respect to the structure of the data (Zhou et al., 2004) and this is a desirable property for semi-supervised learning algorithms (Chapelle and Zien, 2005) since it encourages points that are close together in embedding space to share the same label. This is important to propagate label information from labeled to unlabeled datapoints. We have included this explanation in Section 4.2. Here, we investigate if the decision boundaries remain smooth when propagation is performed directly in embedding space (see Equation 3) instead of the output space.

Experimental setup According to (Learning, 2016) a smooth function is that in which $f(x) = f(x+\epsilon)$ for small values of ϵ. In order to assess smoothness before and after applying embedding propagation, we generate a 2D toy dataset of randomly sampled embeddings with their corresponding labels. The dataset has two classes and 50 data points per class. Samples from both classes are drawn from two different, opposing gaussians. We resorted to a low-dimensional dataset since other real datasets such as *mini*Imagenet would require dimensionality reduction techniques for visualization, resulting in loss of information.

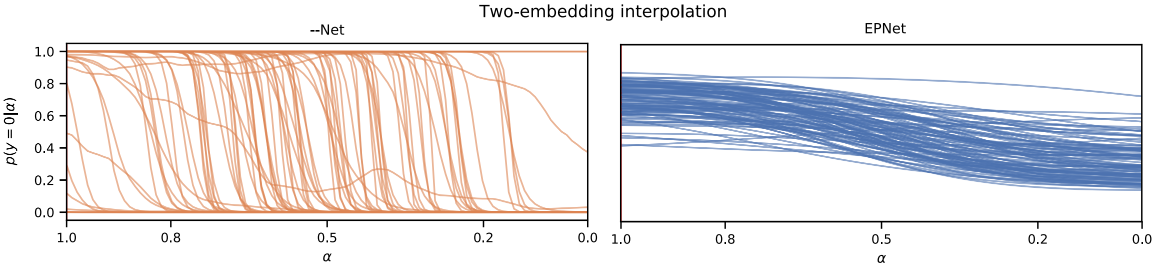


Figure 4: Interpolation of embedding pairs for two random data points of the *mini*Imagenet dataset with different classes vs probability of belonging to the first of these two classes. The right figure shows the class probability for Resnet-12 embeddings extracted from EPNet, and the left figure (--Net) from the same network trained without embedding propagation. The scalar α controls the weight of the first embedding in the linear interpolation.

Results Defining a term to empirically measure the smoothness of the classification surface allows us to measure how the hyperparameters that control embedding propagation (Sec 3) affect the smoothness of the class embedding manifold. In EP (Eq. 2), the hyperparameter α controls the amount of propagation performed in the graph and σ is the radius of the RBF function used to calculate the similarity matrix. Therefore, the value of α should be directly correlated with $\mathcal{S}$ (Eq. 6) and the value of σ inversely correlated with $\mathcal{S}$. Figure 3c shows that the former hypothesis holds, thus showing that α and σ control the smoothness of the class embedding manifold. Furthermore, we show the smoothing effect of EP on a toy classification dataset in Figure 3b (more details can be found in the Appendix). The manifold hypothesis from semi-supervised learning theory holds that smoother decision boundaries aid generalization, as shown in (Verma et al., 2019a). Hence by applying EP, encouraging smoother decision boundaries, we improve the classification of adversarial examples.

Lastly, to further reinforce the smoothness hypothesis, we visualize embedding interpolations with and without embedding propagation. We use EPNet to obtain image embeddings and select a set of random pairs $\mathbf{z}_i, \mathbf{z}_j$ that belong to different classes y_i, y_j. We then interpolate between each pair as $\widetilde{\mathbf{z}} = \alpha \cdot \mathbf{z}_i + (1-\alpha)\mathbf{z}_j$ where $\alpha \in [0..1]$, and plot this value against $p(y_i|\widetilde{\mathbf{z}})$ in Figure 4. We also plot $p(y_i|\hat{\mathbf{z}})$ where embeddings were obtained using EPNet without embedding propagation (--Net). We observe that EPNet has significantly smoother probability transitions than --Net as the embedding $\widetilde{\mathbf{z}}$ changes from $\mathbf{z}_i$ to $\mathbf{z}_j$. In contrast, --Net yields sudden probability transitions. This suggests that embedding propagation encourages smoother decision boundaries.

4.3 Adversarial robustness

Few-shot learning algorithms are tested outside of the original distribution given that few-shot learning datasets use a disjoint set of test classes. Similarly, adversarial attacks try to modify a sample to move it outside of the original training distribution in order to cause unexpected behavior of the model. (Verma et al., 2019a) showed that smoother decision boundaries improve adversarial robustness. Likewise, in the previous experiment, we have shown that embedding propagation has a smoothing effect on the class embedding

Table 1: Adversarial attacks results across four different datasets. Notice that manifold mixup fails against iterative perturbations (PGD (Madry et al., 2017), FAB (Croce and Hein, 2020)), while EP, despite only being applied at test time, increases adversarial robustness considerably. Furthermore, a combination of both regularization methods improves results substantially across datasets for (Croce and Hein, 2020). We report the average of five different runs. Vanilla refers to a setting where neither EP nor mixup is applied.

	CIFAR10	CIFAR100	MNIST	FashionMNIST
No perturbation				
Vanilla	93.65 ±1.23	**76.37** ±1.41	99.37 ±0.04	94.59 ±0.18
Mixup	93.49 ±0.65	72.72 ±1.89	99.21 ±0.12	**94.78** ±0.22
EP	**94.30** ±0.39	76.13 ±1.27	99.32 ±0.08	94.66 ±0.15
EP + Mixup	92.71 ±0.46	71.76 ±1.07	**99.39** ±0.06	94.53 ±0.20
FGSM (Goodfellow et al., 2014)				
Vanilla	17.24 ±1.09	6.59 ±0.23	61.77 ±20.37	45.07 ±1.73
Mixup	18.78 ±2.55	5.46 ±0.57	84.234 ±10.98	38.88 ±10.64
EP	**31.24** ±0.67	**9.59** ±0.47	84.44 ±7.40	61.74 ±4.65
EP + Mixup	20.89 ±2.69	6.38 ±0.64	81.34 ±10.68	57.92 ±7.74
PGD (Madry et al., 2017)				
Vanilla	0.006 ±0.004	0.01 ±0.01	22.82 ±12.15	0.81 ±0.46
Mixup	0.03 ±0.01	0.02 ±0.01	29.09 ±14.20	2.45 ±0.45
EP	**11.70** ±0.90	**3.01** ±0.73	43.6 ±23.90	22.99 ±5.97
EP + Mixup	8.79 ±0.75	0.85 ±0.09	52.86 ±24.04	19.01 ±7.22
FAB (Croce and Hein, 2020)				
Vanilla	0.58 ±0.07	0.09 ±0.02	0.03 ±0.01	0.09 ±0.02
Mixup	0.69 ±0.11	0.09 ±0.01	1.91 ±2.04	3.16 ±2.20
EP	5.64 ±0.33	5.85 ±0.21	14.20 ±6.28	10.07 ±0.94
EP + Mixup	**9.95** ±0.98	**8.24** ±1.35	**61.99** ±20.41	**35.12** ±3.16

manifold, similar to the effect caused by manifold mixup (Verma et al., 2019a). Therefore, in this section we explore whether similar benefits are observed from applying embedding propagation.

Experimental setup Adversarial attacks exploit the linear nature of neural networks and their difficulty generalizing to OOD data. They imperceptibly modify an input sample as to cause missclassification with high confidence. In this work we focus on white box attacks, where the attacker has full access to the model gradients. Concretely, we evaluate our method on: FGSM (Goodfellow et al., 2014), PDG (Madry et al., 2017) and FAB (Croce and Hein, 2020) attacks.

In this setting, we only consider one test sample at a time in order to make embedding propagation non-transductive and decouple its performance from the ordering of the batch. However, EP requires multiple embeddings in order to build a graph (Eq. 1). To address this issue, we compute class prototypes from the training dataset and use those prototypes

Table 2: Self-Supervised results for STL-10 and CIFAR100 datasets where both Manifold mixup and embedding propagation are applied in the same way during MoCo pre-training. We report the average of five different runs.

	STL-10	CIFAR100
MoCo	85.28 ±0.75	74.68 ±0.18
MoCo + Manifold Mixup	85.75 ±0.48	74.85 ±0.31
MoCo-EP	86.02 ±0.65	75.02 ±0.56

to form a graph for each test sample. Let $Z \in \mathbb{R}^k$ be the output of a feature extractor, $\mathcal{C}$ the set of classes in our dataset and N the total of samples in our training set. Then the prototypes matrix is defined as $\mathcal{P} \in \mathbb{R}^{k \times \mathcal{C}}$ and it is computed as:

$$\mathcal{P}_c = \frac{1}{N_c} \sum_{i \in c}^{N} z_i, \ \forall c \in \mathcal{C} \tag{7}$$

where N_c is the number of examples belonging to class c. Notice that obtaining the prototype matrix $\mathcal{P}$ does not require any additional training, a forward pass on the training dataset is all that is required. At test time, we apply EP on the concatenation of $\mathcal{P}$ with the embedding of single data point z_i, resulting in $\widetilde{z}_i$ (Eq. 2). Then the classifier is applied to $\widetilde{z}_i$. This process is illustrated in Figure 2a. Note that we only apply the embedding propagation operation at test time, when the adversarial attacks are performed, since we observed similar results when applying it both at train and at test time.

Results As seen in Table 1 EP increases adversarial robustness against strong iterative perturbations, with an average improvement with respect to manifold mixup of 12.17% against (Madry et al., 2017) and 7.85% against (Croce and Hein, 2020). Furthermore, we show that EP and manifold mixup are not mutually exclusive, and they can be combined to improve performance against FAB attacks (Croce and Hein, 2020) with an improvement of up to 47%. It is worth noticing that manifold mixup, improves adversarial robustness against single step attacks, but fails against iterative perturbations, despite being applied during training. Conversely, EP improves robustness against both single and multiple step attacks, while being applied at inference time only with no additional training required.

4.4 Self-supervised Learning

We experiment on the self-supervised and semi-supervised learning scenarios, where the model has to learn from an unlabeled set of examples. Mixup (Zhang et al., 2018) has been shown to improve results in these scenarios. Works such as (Verma et al., 2020; Kalantidis et al., 2020) leverage embedding interpolations to create hard positives and hard negatives, resulting in improved self-supervised learning performance.

Manifold mixup (Verma et al., 2019a) and EP (Rodríguez et al., 2020), also have a smoothing effect on the classification surface. In this section, we explore how embedding propagation compares to manifold mixup in a self-supervised scenario. Notice that this effect is a natural byproduct of Equation 3 and thus the embedding propagation algorithm itself

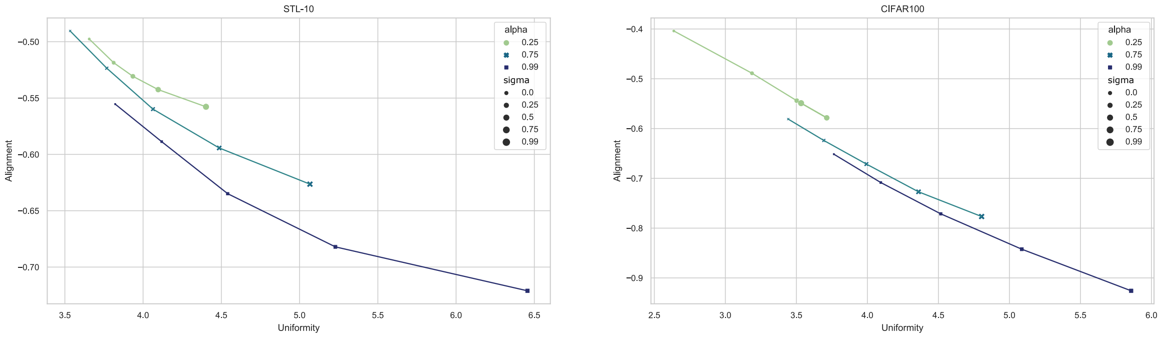


Figure 5: *Alignment* and *Uniformity* values obtained for different values of α and σ on the STL-10 (left) and CIFAR100 (right) datasets. The x and y axis correspond to $-\mathcal{L}_{Uniform}$ and $-\mathcal{L}_{align}$ of (Wang and Isola, 2020), respectively. Harder positives and negatives make the embedding space more uniform and less aligned.

requires no modifications from its original implementation to be applied in self-supervised learning.

Experimental setup We adapt the original MoCo (He et al., 2020) implementation to integrate embedding propagation. The pre-training process is shown in Figure 2b. In this setting we use EP to generate new embeddings and use them in an additional contrastive loss. Consider an encoder that outputs an encoded query q and a momentum encoder that outputs coded samples k that are keys of a dictionary. Letting only one key k_+ to match the query q, the contrastive loss function used in MoCo can be defined as

$$\mathcal{L}_q(q,k) = -log\frac{exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} exp(q \cdot k_i/\tau)}, \tag{8}$$

where τ is a temperature hyper-parameter. We introduce an additional loss where the embedding propagation is applied; $EP(q \oplus k)$ to obtain new queries q' and keys k'. Thus, the criteria to optimize becomes:

$$\mathcal{L}_q = \alpha\mathcal{L}_q(q,k) + (1-\alpha)\mathcal{L}_q(q',k') \tag{9}$$

where α is a weighting hyper-parameter set to 0.6 (found through random search) for all experiments. In contrast to label propagation (Zhou et al., 2004), embedding propagation is completely unsupervised, which makes it possible to apply it during MoCo's pre-training phase. For comparison, we also provide results with manifold mixup (Verma et al., 2019a) applied to q and k in the same way as EP. The main difference between the two methods is that manifold mixup considers random pairs of samples while EP takes into account the topology of the data. The hyperparameters manifold mixup's Dirichlet distribution are the best found through random search.

Results As seen in Table 2 embedding propagation increases the validation accuracy with respect to a MoCo baseline by 0.74% in STL-10 and by 0.34% in CIFAR100. EP also outperforms manifold mixup in both datasets by 0.27% and 0.17%, respectively. We hypothesize that the improvement is due to the creation of artificial hard negatives and positives by the embedding propagation operation during the training process. In fact, (Verma et al., 2020) showed that mixup can be used in a self-supervised setting to synthesize hard positives. Similarly, EP naturally synthesizes hard positives and negatives taking into account the topology of the data.

Recently (Wang and Isola, 2020) proposed two losses or metrics for assessing the quality of contrastive learning representations. The first one ($\mathcal{L}_align$) measures the absolute distance between representations with the same label, while the second one ($\mathcal{L}_{Uniform}$) measures how uniformly distributed are the representations in the hyper-sphere. In Figure 5, we show how the alignment decreases and the uniformity increases as the α and σ in the EP operation increase. Indicating that EP helps the proxy task to obtain a better representation of the embedding space as shown in (Kalantidis et al., 2020) by creating hard-negatives and positives.

4.5 Few-Shot and Semi-supervised Learning

In this section we review the use of EP in the few-shot learning scenario. First we describe the experimental setup, then we provide implementation details, and finally we report the results. Note that we consider transductive few-shot as a form of semi-supervised learning.

Experimental setup We follow the common few-shot learning setup (Vinyals et al., 2016; Ren et al., 2018) where three datasets are given: a *base* dataset ($\mathcal{D}_b$), a *novel* dataset ($\mathcal{D}_n$), and a *validation* dataset ($\mathcal{D}_v$). The base dataset is composed of a large amount of labeled images, the novel dataset is composed of labeled images from previously unseen classes and it is used to evaluate the transfer learning capabilities of a model. Lastly, the validation dataset $\mathcal{D}_v$ contains classes not present in either $\mathcal{D}_b$ or $\mathcal{D}_n$ and is used to conduct hyperparameter search.

Furthermore, we have access to episodes. Each episode consists of n classes sampled uniformly without replacement from the set of all classes, a support set S (k examples per class) and a query set Q (q examples per class). This is referred to as n-way k-shot learning.

Given an episode, inference is performed by sequentially performing embedding and label propagation on features extracted from the input image. More formally, this is performed as follows. Let $\widetilde{Z} \in \mathbb{R}^{(k+q)\times m}$ be the matrix of propagated embeddings obtained by jointly applying Eq. 1-3 to the support and query sets. Let $P_{\widetilde{Z}}$ be the corresponding propagator matrix. Further, let $Y_S \in \mathbb{R}^{k\times n}$ be a one-hot encoding of the labels. We compute the logits for the query set ($\hat{Y}_Q$) by performing label propagation as described in (Zhou et al., 2004).

For few-shot learning, we train the model in two phases. During the first phase we train two linear classifiers parametrized by W_l and W_r, respectively. The first classifier is trained to predict the class labels of examples in $\mathcal{D}_b$. It is optimized by minimizing the cross-entropy loss,

$$\mathcal{L}_c(\mathbf{x}_i, y_i; W_l, \theta) = -\ln p(y_i|\widetilde{\mathbf{z}}_i, W_l), \tag{10}$$

where $y_i \in \mathcal{Y}_b$ and the probabilities are obtained by applying softmax to the logits provided by the neural network. For fair comparison with recent literature (Mangla et al., 2019; Gidaris et al., 2019) we also add a self-supervision loss. Hence, the second classifier is trained to predict image rotations, minimizing the following loss,

Table 3: Comparison of test accuracy against state-of-the art methods for Few-shot classification using *mini*Imagenet and *tiered*Imagenet with the 1-shot and 5-shot settings. The second column shows number of parameters per model in thousands (K). --Net is identical to EPNet but without EP. We report the average of 600 episodes.

		***mini*Imagenet**		***tiered*Imagenet**	
	Params	1-shot	5-shot	1-shot	5-shot
CONV-4					
Matching (Vinyals et al., 2016)	112K	43.56 ±0.84	55.31 ±0.73	-	-
MAML (Liu et al., 2019b)	112K	48.70 ±1.84	63.11 ±0.92	51.67 ±1.81	70.30 ±0.08
ProtoNet (Snell et al., 2017)	112K	49.42 ±0.78	68.20 ±0.66	53.31 ±0.89	72.69 ±0.74
ReNet (Sung et al., 2018)	223K	50.44 ±0.82	65.32 ±0.70	54.48 ±0.92	71.32 ±0.78
GNN (Garcia and Bruna, 2017)	1619K	50.33 ±0.36	66.41 ±0.63	-	-
TPN (Liu et al., 2019b)	171K	53.75 ±0.86	69.43 ±0.67	57.53 ±0.96	72.85 ±0.74
CC+rot (Gidaris et al., 2019)	112K	54.83 ±0.43	71.86 ±0.33	-	-
SIB (Hu et al., 2020a)	112K	58.00 ±0.60	70.70 ±0.40	-	-
EGNN (Kim et al., 2019)	5068K	-	**76.37** ±*N/A*	-	**80.15** ±*N/A*
--Net (ours)	112K	57.18 ±0.83	72.57 ±0.66	57.60 ±0.93	73.30 ±0.74
EPNet (ours)	112K	**59.32** ±0.88	72.95 ±0.64	**59.97** ±0.95	73.91 ±0.75
RESNET-12					
ProtoNets++ (Xing et al., 2019)	7989K	56.52 ±0.45	74.28 ±0.20	58.47 ±0.64	78.41 ±0.41
TADAM (Oreshkin et al., 2018)	7989K	58.50 ±0.30	76.70 ±0.30	-	-
MetaOpt-SVM (Lee et al., 2019)	12415K	62.64 ±0.61	78.60 ±0.46	65.99 ±0.72	81.56 ±0.53
TPN (Liu et al., 2019b)	8284K	59.46 ±*N/A*	75.65 ±*N/A*	-	-
Robust-20++ (Dvornik et al., 2019)	11174K	58.11 ±0.64	75.24 ±0.49	70.44 ±0.32	85.43 ±0.21
MTL (Sun et al., 2019)	8286K	61.20 ±1.80	75.50 ±0.80	-	-
CAN (Hou et al., 2019)	8026K	**67.19** ±0.55	80.64 ±0.35	73.21 ±0.58	84.93 ±0.38
BD-CSPN (Liu et al., 2019a)	7989K	65.94 ±*N/A*	79.23 ±*N/A*	-	-
--Net (ours)	7989K	65.66 ±0.85	**81.28** ±0.62	72.60 ±0.91	85.69 ±0.65
EPNet (ours)	7989K	66.50 ±0.89	81.06 ±0.60	**76.53** ±0.87	**87.32** ±0.64
WRN-28-10					
LEO (Rusu et al., 2018)	37582K	61.76 ±0.08	77.59 ±0.12	66.33 ±0.05	81.44 ±0.09
Robust-20++ (Dvornik et al., 2019)	37582K	62.80 ±0.62	80.85 ±0.43	-	-
wDAE-GNN (Gidaris and Komodakis, 2019)	48855K	62.96 ±0.15	78.85 ±0.10	68.18 ±0.16	83.09 ±0.12
CC+rot (Gidaris et al., 2019)	37582K	62.93 ±0.45	79.87 ±0.33	70.53 ±0.51	84.98 ±0.36
Manifold mixup (Mangla et al., 2019)	37582K	64.93 ±0.48	83.18 ±0.72	-	-
FEAT (Ye et al., 2020)	37582K	65.10 ±0.20	81.11 ±0.14	70.41 ±0.23	84.38 ±0.16
SimpleShot (Wang et al., 2019)	37582K	65.87 ±20	82.09 ±0.14	70.90 ±0.22	85.76 ±0.15
SIB (Hu et al., 2020a)	37582K	70.00 ±0.60	79.20 ±0.40	-	-
BD-CSPN (Liu et al., 2019a)	37582K	70.31 ±0.93	81.89 ±0.60	78.74 ±0.95	86.92 ±0.63
LaplacianShot (Ziko et al., 2020)	37582K	74.86 ±0.19	84.13 ±0.14	80.18 ±0.21	87.56 ±0.15
TIM-GD(Boudiaf et al., 2020)	37582K	**77.80** ±*N/A*	**87.40** ±*N/A*	82.10 ±*N/A*	**89.80** ±*N/A*
--Net (ours)	37582K	65.98 ±0.85	82.22 ±0.66	74.04 ±0.93	86.03 ±0.63
EPNet (ours)	37582K	70.74 ±0.85	84.34 ±0.53	78.50 ±0.91	88.36 ±0.57

$$\mathcal{L}_r(\mathbf{x}_i, r_j; W_r, \theta) = -\ln p(r_j|\widetilde{\mathbf{z}}_i, W_r), \tag{11}$$

where $r_j \in \{0°, 90°, 180°, 270°\}$, and $p(r_j|\widetilde{\mathbf{z}}_i, W_r)$ is the probability of the input being rotated by r_j as predicted by a softmax classifier with weights W_r.

Thus the criteria to optimize in this first phase becomes:

$$\mathcal{L}_c(\mathbf{x}, y; W_l, \theta) + \mathcal{L}_r(\mathbf{x}, r; W_r, \theta). \tag{12}$$

In the second phase we use episodic training in order to generalize to novel classes. This process is illustrated in Figure 2d. In this phase, the model uses two classifiers. The first one is based on label propagation, and it computes class probabilities by applying a softmax to the query set logits $\hat{Y}_Q$.

$$\mathcal{L}_p(\mathbf{x}_i, y_i; \theta) = -\ln p(y_i|\widetilde{\mathbf{z}}_i, \widetilde{Z}, Y_S). \tag{13}$$

The second classifier is used to predict the base classes as during the pre-training phase, and thus, it is identical to the W_l-based classifier used in pre-training. It is included to preserve a discriminative feature representation. Hence, the criteria to optimize becomes:

$$\underset{\theta, W_l}{\operatorname{argmin}} \left[\frac{1}{|Q|} \sum_{(\mathbf{x}_i, y_i) \in Q} \mathcal{L}_p(\mathbf{x}_i, y_i; \theta) + \frac{1}{|S \cup Q|} \sum_{(\mathbf{x}_i, y_i) \in S \cup Q} \frac{1}{2} \mathcal{L}_c(\mathbf{x}_i, y_i; W_l, \theta) \right]. \tag{14}$$

Implementation details For fair comparison with previous work, we used three common feature extractors: (i) a 4-layer convnet (Vinyals et al., 2016; Snell et al., 2017) with 64 channels per layer, (ii) a 12-layer resnet (Oreshkin et al., 2018), and (iii) a wide residual network (WRN-28-10) (Rusu et al., 2018; Zagoruyko and Komodakis, 2016). For *mini* and *tiered*Imagenet, images are resized to 84 × 84. Results for Imagenet-FS and few-shot semi-supervised learning can be found in (Rodríguez et al., 2020). We denote as EPNet the model resulting of combining these feature extractors with the EP procedure.

We evaluate 2 variations of our method: (i) EPNet as described in Eq. 1-3; (ii) --Net, which is identical to EPNet but without applying EP.

We also consider the few-shot semi-supervised learning scenario, where we have access to an unlabeled set of images U. We use the unlabeled set as follows. First, we use the same inference procedure as previously described to predict the labels $\hat{c}_U$ for the unlabeled set as pseudo-labels. Then, we augment the support set with U using their pseudo-labels as the true labels. Finally, we apply the aforementioned inference procedure on the new support set to predict the labels for the query set.

Results are shown in Table 3. We compare the performance of the same neural network with and without EP (--Net and EPNet) against different few-shot classification methods across different backbones. EGNN uses a graph neural net on top of conv-4, hence the large amount of parameters. We observe that EP consistently improves the performance with respect to the same backbone without EP (--Net). We observe that the improvement is most significant in the one-shot scenarios, since EP leverages unlabeled queries to improve the classification performance. Specifically, with the largest backbone (WRN-28-10), EP improves up to 5% and 2% in 1-shot and 5-shot respectively in *mini*Imagenet. Moreover, note that EP becomes more effective on higher capacity backbones, with an average improvement of 4% across datasets with a WRN-28-10 backbone. We hypothesize that these backbones provide more accurate embeddings that result in accurate graphs that attain more consistent information propagation between nodes.

Table 4 shows results in the SSL setting where 100 additional unlabeled samples are available (Ren et al., 2018; Liu et al., 2019b) (EPNet$_{SSL}$). Notice that including unlabeled samples increases the accuracy of EPNet for all settings, surpassing the state of the art by a wide margin of up to 16% accuracy points for the 1-shot WRN-28-10. Similar to previous experiments, removing EP from EPNet (`--Net`) is detrimental for model performance, supporting our hypotheses. Furthermore, in Figure 6, we show that the improvements of

Table 4: Semi-Supervised Learning (SSL) results with 100 unlabeled samples. --Net is identical to EPNet but without embedding propagation. *Re-implementation of (Yu et al., 2020). We report the average of five different runs

		***mini*Imagenet**		***tiered*Imagenet**	
	Backbone	1-shot	5-shot	1-shot	5-shot
TPN$_{SSL}$ (Liu et al., 2019b)	CONV-4	52.78	66.42	55.74	71.01
k-Means$_{masked,soft}$ (Ren et al., 2018)	CONV-4	50.41 ±0.31	64.39 ±0.24	-	-
--Net (ours)	CONV-4	57.18 ±0.83	72.57 ±0.66	57.60 ±0.93	73.30 ±0.74
EPNet (ours)	CONV-4	59.32 ±0.88	72.95 ±0.64	59.97 ±0.95	73.91 ±0.75
--Net$_{SSL}$ (ours)	CONV-4	63.74 ±0.97	75.30 ±0.67	65.01 ±1.04	74.24 ±0.80
EPNet$_{SSL}$ (ours)	CONV-4	**65.13** ±0.97	**75.42** ±0.64	**66.63** ±1.04	**75.70** ±0.74
LST (Li et al., 2019)	RESNET-12	70.10 ±1.90	78.70 ±0.80	77.70 ±1.60	85.20 ±0.80
--Net (ours)	RESNET-12	65.66 ±0.85	81.28 ±0.62	72.60 ±0.91	85.69 ±0.65
EPNet (ours)	RESNET-12	66.50 ±0.89	81.06 ±0.60	76.53 ±0.87	87.32 ±0.64
--Net$_{SSL}$ (ours)	RESNET-12	73.42 ±0.94	83.17 ±0.58	80.26 ±0.96	88.06 ±0.59
EPNet$_{SSL}$ (ours)	RESNET-12	**75.36** ±1.01	**84.07** ±0.60	**81.79** ±0.97	**88.45** ±0.61
*k-Means$_{masked,soft}$ (Ren et al., 2018)	WRN-28-10	52.78 ±0.27	66.42 ±0.21	-	-
TransMatch (Yu et al., 2020)	WRN-28-10	63.02 ±1.07	81.19 ±0.59	-	-
--Net (ours)	WRN-28-10	65.98 ±0.85	82.22 ±0.66	74.04 ±0.93	86.03 ±0.63
EPNet (ours)	WRN-28-10	70.74 ±0.85	84.34 ±0.53	78.50 ±0.91	88.36 ±0.57
--Net$_{SSL}$ (ours)	WRN-28-10	77.70 ±0.96	86.30 ±0.50	82.03 ±1.03	88.20 ±0.61
EPNet$_{SSL}$ (ours)	WRN-28-10	**79.22** ±0.92	**88.05** ±0.51	**83.69** ±0.99	**89.34** ±0.59

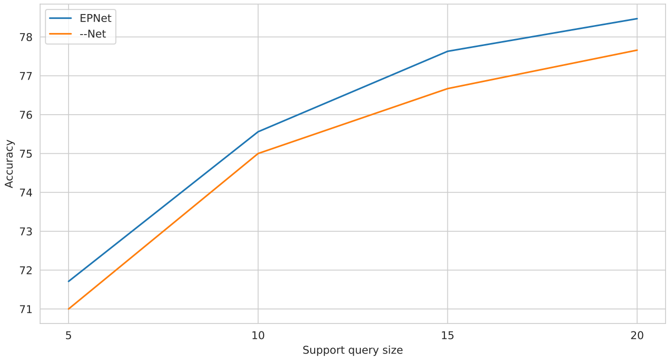


Figure 6: Performance of a small convolutional network (CONV-4) on the *mini*Imagenet dataset with and without Embedding Propagation. Notice how the improvement obtained by EP is consistent for all query sizes, showing that EP remains effective in high-shot classification settings.

EP remain consistent even in high shot settings. Additional results for few-shot SSL and Imagenet-FS (Hariharan and Girshick, 2017) and ablations from (Rodríguez et al., 2020) can be found in the Appendix.

5. Conclusions

The embedding propagation (EP) procedure has been shown to improve few-shot learning performance. The main hypothesis is that EP smooths the class embedding manifold, acting as a regularizer. In fact, smooth class embedding manifolds are a known requisite for semi-supervised learning (Chapelle and Zien, 2005), and to improve adversarial robustness (Verma et al., 2019a). In this work we provided additional quantitative and qualitative insights showing that EP yields smoother classification surface as measured with the Laplacian. In addition, we extended (Rodríguez et al., 2020) showing that besides few-shot classification, EP also improves adversarial robustness and self/semi-supervised learning performance.

Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) and the European Regional Development Fund (ERDF) under Grant PID2020-120311RB-I00 funded by MCIN/AEI/10.13039/501100011033, and by the Generalitat de Catalunya under the Industrial Doctorate Program (grant number 2020DI62).

References

Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel methods—support vector learning*, 1999.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7(Nov):2399–2434, 2006.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, Jose Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *arXiv preprint arXiv:2006.14618*, 2020.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64, 2005.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12): 3207–3220, 2010.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. Citeseer, 2014.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *ICCV*, pages 3723–3731, 2019.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. JMLR. org, 2017.

Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *ICLR*, 2017.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018.

Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2019.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *CVPR*, pages 8059–8068, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. In *CAP*, pages 281–296, 2005.

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, pages 3018–3027, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. *arXiv preprint arXiv:2010.12050*, 2020.

Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, pages 4005–4016, 2019.

Shell Xu Hu, Pablo Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. In *International Conference on Learning Representations (ICLR)*, 2020a. URL `https://openreview.net/forum?id=Hkg-xgrYvH`.

Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Exploiting unsupervised inputs for accurate few-shot classification, 2020b.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7642–7651, 2018.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, pages 5070–5079, 2019.

Ramesh Jain, Rangachar Kasturi, and Brian G Schunck. *Machine vision*, volume 5. McGraw-hill New York, 1995.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*, 2020.

Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Deep Learning. Ian goodfellow, yoshua bengio, aaron courville. *The reference book for deep learning models*, 2016.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pages 10657–10665, 2019.

Wee Sun Lee, Peter L Bartlett, and Robert C Williamson. Lower bounds on the vc dimension of smoothly parameterized function classes. *Neural Computation*, 7(5):1040–1053, 1995.

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*, pages 10276–10286, 2019.

Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. *ECCV*, 2019a.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. Learning to propagate labels: transductive propagation network for few-shot learning. In *ICLR*, 2019b.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold mixup for few-shot learning. *arXiv preprint arXiv:1907.12087*, 2019.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *European Conference on Computer Vision*, pages 488–501. Springer, 2012.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 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.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, pages 721–731, 2018.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning, 2017. In *ACM Asia Conference on Computer and Communications Security*, 2016a.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016b.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016.

Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

Pau Rodríguez, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations. *ICLR*, 2016.

Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *European Conference on Computer Vision*, pages 121–138. Springer, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCC*, 2015.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2018.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *arXiv preprint arXiv:1606.04586*, 2016.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NeurIPS*, pages 901–909, 2016.

Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Citeseer, 2003.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1): 1929–1958, 2014.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, pages 403–412, 2019.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204, 2017.

Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *CVPR*, pages 5486–5494, 2018.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 1999.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019a.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *stat*, 1050:19, 2019b.

Vikas Verma, Minh-Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc V Le. Towards domain-agnostic contrastive learning. *arXiv preprint arXiv:2011.04419*, 2020.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066. PMLR, 2013.

Bo Wang, Zhuowen Tu, and John K Tsotsos. Dynamic label propagation for semi-supervised multi-class multi-label classification. In *Proceedings of the IEEE international conference on computer vision*, pages 425–432, 2013.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, pages 7278–7286, 2018.

Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*, 2020.

Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *International Conference on Machine Learning*, pages 10432–10441. PMLR, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. *arXiv preprint arXiv:2008.11702*, 2020.

Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro OO Pinheiro. Adaptive cross-modal few-shot learning. In *NeurIPS*, pages 4848–4858, 2019.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pages 7115–7123. PMLR, 2019.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328, 2014.

Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *CVPR*, pages 12856–12864, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NeurIPS*, pages 321–328, 2004.

Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning*, pages 11660–11670. PMLR, 2020.

Appendix A. Additional Results

In the semi-supervised setting we consider the scenario proposed by Garcia and Bruna (Garcia and Bruna, 2017). In this scenario the model is trained to perform 5-shot 5-way classification but only 20% to 60% of the support set is labeled. As shown by (Lee, 2013), this scenario is equivalent to entropy regularization, an effective method for semi-supervised learning. Entropy regularization is particularly effective in cases where the decision boundary lies in low-density regions. With embedding propagation we achieve a similar decision boundary by smoothing the class embedding manifold. Following the same setting described in (Garcia and Bruna, 2017; Kim et al., 2019), we trained our model in the 5-shot 5-way scenario where the support samples are partially labeled. In Table 5, we report the test

Table 5: SSL results for the 5-shot 5-way scenario with different amounts of unlabeled data. The percentages refer to the amount of supports that are labeled in a set of 5 images per class.

	Params	20%	40%	60%	100%
GNN (Garcia and Bruna, 2017)	112K	52.45	58.76	-	66.41
EGNN (Kim et al., 2019)	5068K	**63.62**	64.32	66.37	**76.37**
--Net$_{SSL}$ (ours)	112K	58.52 ±0.97	64.46 ±0.79	67.81 ±0.74	57.18 ±0.83
EPNet$_{SSL}$ (ours)	112K	60.66 ±0.97	**67.08** ±0.80	**68.74** ±0.74	59.32 ±0.88

accuracy with `conv-4` when labeling 20%, 40%, 60% and 100% of the support set. EPNet obtains up to 2.7% improvement over previous state-of-the-art when 40% of the support are labeled. Moreover, EPNet also outperforms EGNN (Kim et al., 2019) in the 40% and 60% scenarios, although EPNet has 45× less parameters. On the large-scale Imagenet-FS, EP improves all benchmarks by approximately 2% accuracy, see Table 7. These results demonstrate the scalability of our method and the orthogonality with other embedding transformations such as denoising autoencoders (Gidaris and Komodakis, 2019). Table 6, shows that EPNet outperforms models that use more parameters or higher resolution images on the CUB-200-2011 dataset for the 1-shot and 5-shot benchmarks.

A.1 Ablation Studies

In this section we investigate the impact of the rotation loss (ROT), embedding fine-tuning (EFT), label propagation (LP), and embedding propagation (EP) on the 1-shot *mini*Imagenet accuracy. As seen in Table 8, when label propagation is deactivated, we substitute it with a prototypical classifier. Interestingly, it can be seen that the improvement is larger when using LP in combination with EP (Table 8; columns 2-4, and 10-12). This finding is in accordance with the hypothesis that EP promotes smoother decision boundaries, and this is beneficial for transductive and SSL algorithms.We included a rotation loss for fair comparison with other SotA (Gidaris et al., 2019; Mangla et al., 2019), however, we see that the main improvement is due to the combination of EP with LP. We also find that episodic fine-tuning successfully adapts our model to the episodic scenario (Table 8; line 2).

Table 6: Comparison with the state of the art on CUB-200-2011. $^{*}Robust - 20 + +$ uses an 18-layer residual network, and Accuracies obtained with 224 × 224 images appear in gray.

	backbone	1-shot	5-shot
*Robust-20++ (Dvornik et al., 2019)	RESNET-18	68.68 ±0.69	83.21 ±0.44
EPNet (ours)	RESNET-12	**82.85** ±0.81	**91.32** ±0.41
Manifold mixup (Mangla et al., 2019)	WRN-28-10	80.68 ±0.81	90.85 ±0.44
EPNet (ours)	WRN-28-10	**87.75** ±0.70	**94.03** ±0.33

Table 7: Top-5 test accuracy on Imagenet-FS.

	Novel Classes					All classes				
Approach	K=1	2	5	10	20	K=1	2	5	10	20
Batch SGM (Hariharan and Girshick, 2017)	-	-	-	-	-	49.3	60.5	71.4	75.8	78.5
PMN (Wang et al., 2018)	45.8	57.8	69.0	74.3	77.4	57.6	64.7	71.9	75.2	77.5
LwoF (Gidaris and Komodakis, 2018)	46.2	57.5	69.2	74.8	78.1	58.2	65.2	72.7	76.5	78.7
CC+ Rot (Gidaris et al., 2019)	46.43 ±0.24	57.80 ±0.16	69.67 ±0.09	74.64 ±0.06	77.31 ±0.05	57.88 ±0.15	64.76 ±0.10	72.29 ±0.07	75.63 ±0.04	77.40 ±0.03
wDAE-GNN (Gidaris and Komodakis, 2019)	48.00 ±0.21	59.70 ±0.15	70.30 ±0.08	75.00 ±0.06	77.80 ±0.05	59.10 ±0.13	66.30 ±0.10	73.20 ±0.05	76.10 ±0.04	77.50 ±0.03
wDAE-GNN + EP (ours)	**50.07** ±0.27	**62.16** ±0.16	**72.89** ±0.11	**77.25** ±0.07	**79.48** ±0.05	**60.87** ±0.16	**68.53** ±0.10	**75.56** ±0.07	**78.28** ±0.04	**78.89** ±0.03

Table 8: Algorithm ablation with conv-4 on 1-shot *mini*Imagenet. EFT: Episodic Finetuning, ROT: Rotation loss, LP: Label Propagation, EP: Embedding Propagation

EXP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
EFT									✓	✓	✓	✓	✓	✓	✓	✓
ROT					✓	✓	✓	✓					✓	✓	✓	✓
LP			✓	✓			✓	✓			✓	✓			✓	✓
EP		✓		✓		✓		✓		✓		✓		✓		✓
ACC	49.57	52.83	53.40	55.75	50.83	53.63	53.38	55.55	54.29	56.38	56.93	58.35	54.92	56.46	57.35	58.85

Appendix B. Toy Experiment

The decision boundaries were for the toy experiment were obtained by training an MLP on the toy data **without the cyan points**. Then, we illustrate how for a new set of points (cyan) fall in the uncertain region with EP while they are classified as blue without EP. We set the value of σ to 0.5 to exaggerate smoothness while we kept α to 0.1 (note that the purpose of this figure is merely illustrative).

Appendix C. Adversarial Setting

In order to ensure reproducibility and since adversarial attacks hyperparameter choice greatly influences the results this section details the setting used for the various adversarial perturbations methods used throughout the paper.

In the case of PGD (Madry et al., 2017) and FAB (Croce and Hein, 2020) attacks, we use a perturbation ϵ of 0.03 and for 40 and 100 steps respectively. For FGSM (Goodfellow et al., 2014) we use a higher value of $\epsilon = 0.3$ since it is not an iterative attack. All methods use distance measure ℓ_∞ as a distance measure.

For the manifold mixup experiments we follow the setting proposed in (Verma et al., 2019a) with a mixing coefficient α of 2. All of the attacks were conducted on a Resnet-18 architecture trained for 20 epochs using the AdamW algorithm (Loshchilov and Hutter, 2017) with a learning rate of 0.01 and weight decay of 5×10^{-4}. This setting remains fixed across datasets.