arXiv:1903.02164v1 [cs.LG] 6 Mar 2019

Semi-Supervised Few-Shot Learning with Local and Global Consistency

Ahmed Ayyad [1] **Nassir Navab** [1 2] **Mohamed Elhoseiny** [* 3] **Shadi Albarqouni** [* 1]

Abstract

Learning from a few examples is a key characteristic of human intelligence that AI researchers have been excited about modeling. With the web-scale data being mostly unlabeled, few recent works showed that few-shot learning performance can be significantly improved with access to unlabeled data (Zhang et al., 2018; Ren et al., 2018), known as semi-supervised few shot learning (SS-FSL). We introduce a SS-FSL approach that we denote as Consistent Prototypical Networks (CPN), which builds on top of Prototypical Networks (Ren et al., 2018). We propose new loss terms to leverage unlabelled data, by enforcing notions of local and global consistency. Our work shows the effectiveness of our consistency losses in semi-supervised few shot setting. Our model outperforms the state-of-the-art in most benchmarks, showing large improvements in some cases. For example, in one mini-Imagenet 5-shot classification task, we obtain 70.1% accuracy to the 64.59% state-of-the-art. Moreover, our semi-supervised model, trained with 40% of the labels, compares well against the vanilla prototypical network trained on 100% of the labels, even outperforming it in the 1-shot mini-Imagenet case with 51.03% to 49.4% accuracy. For reproducibility, we make our code publicly available.[1]

1. Introduction

Humans are capable of learning rich hypotheses from 'sparse, noisy, and ambiguous' input data, posing a grand and longstanding challenge to scientist and philosophers: "How do our minds get so much from so little?" (B Tenenbaum et al., 2011). In contrast, artificial learners require a massive amount of labeled data to achieve a comparable performance in complex tasks (Dodge & Karam, 2017). Thereby bringing the challenge to the machine learning community: How can we build models that can "get so much from so little?". Building on the success of supervised training, much of the work addressing the aforementioned question took the form of modified supervised training with some additional challenge, meant to take us a step closer to models which learn from 'sparse, noisy, and ambiguous' input data. The two relevant paradigms here are few-shot learning (FSL) and semi-supervised learning (SSL). Those paradigms have been largely independent, with most work addressing one challenge or the other, we work on semi-supervised few-shot learning (SS-FSL) (Ren et al., 2018), a new paradigm combining the challenges of SSL and FSL.

The **FSL paradigm** is a testing grounds for the ability to learn from few examples. Concretely, a learner is presented with a support set $\mathcal{S}$ containing K examples from each of N classes, in order to *learn* to distinguish between the classes. The learner is then required to classify some query examples into the N classes. The FSL problem may be formulated as a form of meta-learning (Thrun, 1998; Hochreiter et al., 2001), where the learner trains on a collection of classification tasks, generated from large quantities of available data, to generalize well on classification tasks over unseen classes. This formulation is analogous to supervised learning, where each instance is a classification task, rather than a single sample. In the meta-learning formulation, there are two levels of learning; **meta-training** is learning the shared model parameters to be used on future tasks, **adaptation** is *learning* from the support set of a given task to classify its query set.

The **SSL paradigm** is a testing ground for the ability of discriminative models to leverage unlabelled data. Semi-supervised learning techniques come in many shapes, some requiring specialized models (Rasmus et al., 2015) or additional architecture components (Zhang et al., 2018; Dai et al., 2017). We focused on discriminative methods which come in the form of additional loss terms; they require little change to our models, no additional architecture and are state-of-the art for SSL on image classification datasets (Oliver et al., 2018) . In order to leverage unla-

[*]Shared senior authorship. [1]Dept. of Informatics, TU Munich [2]Computer Aided Medical Procedures, Johns Hopkins University [3]Facebook AI Research. Correspondence to: Ahmed Ayyad <a.3ayad@gmail.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML).

[1]Github repository

belled data in discriminative settings, SSL methods must make some assumptions about the data.

The three most popular such assumptions are the *smoothness* assumption, the *cluster* assumption, and the *manifold* assumption (Chapelle et al., 2010). The smoothness assumption states that points close together should hold similar labels (*local consistency*). The cluster assumption states that points of the same class tend to form clusters. The manifold assumption states that data lies on a lower dimensional manifold. It is usually combined with the cluster assumption to state that points forming tight structures over the manifold should hold similar labels (*global consistency*). These *consistency* notions have so far not been introduced to the SS-FSL setting.

In SS-FSL, the learner is presented with additional unlabelled examples in the support set. The challenge is to design few-shot models which perform better when such data is provided. This unlabelled data may be leveraged for **meta-training**, **adaptation**, or both. Currently, there are only two published approaches to semi-supervised few-shot learning (Ren et al., 2018; Zhang et al., 2018). The method proposed in (Ren et al., 2018) for semi-supervised PN, exploits unlabeled data on both learning levels (meta-training and adaptation), however, we hypothesize that while it is powerful for adaptation, it falls short for meta-training. On the other hand, MetaGAN (Zhang et al., 2018) has powerful semi-supervised meta-training, however, it doesn't leverage the unlabeled data in the adaptation step. This motivates us to design a model which can leverage unlabeled data in both meta-training, and adaptation.

Contributions. We present Consistent Prototypical Networks (CPN) where we boost the capabilities of Prototypical Networks by enforcing *local consistency* and *global consistency* for our classifier. Local consistency is encouraged by enforcing the network prediction to be less sensitive to an added noise. Global inconsistency is alleviated by encouraging each prototype to reach itself after a random walk through the points in the given mini batch. Since the local consistency loss could be maximized by producing a high entropy distribution given an arbitrary input, the global consistency loss brings a needed balance to the prototypical network by discouraging it to make high entropy predictions as we detail later.

The rest of this paper is organized as follows. Sec.2 present our approach, then we situate our approach into the wider literature in Sec.3. We present then our experiments and results in Sec.4 where we set the state-of-the-art on most benchmarks.

2. Approach

Our CPN approach is built on top of Prototypical Networks (Snell et al., 2017) with an additional semi-supervised losses to leverage the unlabelled data during the meta-training phase. To formulate local consistency loss, we were inspired to by successful approaches in SSL, namely virtual adversarial training (VAT) (Miyato et al., 2018). To formulate our global consistency loss, we were inspired by a random walk (RW) loss from (Kamnitsas et al., 2018; Haeusser et al., 2017) to smooth our classifier w.r.t. the data manifold.

Concretely, our semi-supervised loss $\mathcal{L}_{SSL} = \mathcal{L}_{VAT} + \mathcal{L}_{RW}$. $\mathcal{L}_{VAT}$ and $\mathcal{L}_{RW}$ are defined in Sections 2.2.1 and 2.2.2 respectively. For adaptation, CPN can effectively leverage the unlabeled data using the refinement step in (Ren et al., 2018).

Next, we define the few-shot learning setup including the episodic training and the prototypical networks, before we explain our additional contributions in details.

2.1. Problem Definition.

Given a dataset $\mathcal{D} = \{\mathcal{D}_L, \mathcal{D}_U\}$, where $\mathcal{D}_L = \{(x_1, y_1), \cdots, (x_L, y_L)\}$ consists of tuples of x_i as an input instance and y_i as the corresponding class label, and $\mathcal{D}_U = \{x_{L+1}, \cdots, x_{L+U}\}$ denote all labeled and unlabeled points, respectively, we aim to build a meta-learner, i.e. N-shot K-way classifier, that can utilize both labeled and unlabeled points during meta-training and adaptation.

In the context of **Few-Shot learning**, an N-shot K-way classifier is tested on episodes E consisting of a support set $\mathcal{S} = \{(x_1, y_1), \cdots, (x_{S_T}, y_{S_T})\}$, and a query set $\mathcal{Q} = \{x_1, \cdots, x_{Q_T}\}$, where S_T and Q_T are the number of instances, provided at meta-testing phase, in the support and query sets, respectively. The classifier then uses the support set to *learn*, and predict the classes of given instances in the query set. Training is performed in the same episodic fashion as testing.

For the **semi-supervised Few-Shot learning**, additional unlabelled examples are provided within each episode. The unlabelled examples mainly come from the available classes in the episode, however additional *distractor* classes might be added to make the setting more challenging and realistic.

In the next subsections, we briefly explain the SS-FSL episode construction, then we overview prototypical networks and present our approach.

2.1.1. SS-FSL Episode construction

For simplicity, we follow the same construction appeared in (Ren et al., 2018), however, we added few more nota-

Algorithm 1 Construct a semi-supervised episode E, optionally with distractors.
RANDOMSAMPLE(S, N) denotes a set of N elements chosen randomly from set S, without replacement. Items sampled from the labeled split are tuples (x_i, y_i), while items sampled from the unlabeled split are simply (x_i)

Require: N_c {The number of classes or *way*}
N_s {The number of examples per class or *shot*}
N_q {The number of query images per class}
N_u {The number of unlabeled examples per class in the support set}
N_d {The number of distractor classes per episode}
$V \leftarrow$ RANDOMSAMPLE($\{1 \cdots K\}$, N_c)
for $k \in V$ **do**
  $\mathcal{S}_{k,L} \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{k,L}$, N_s)
  $\mathcal{S}_{k,U} \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{k,U}$, N_u)
  $\mathcal{Q}_k \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{k,L}$, N_q)
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{Q}_k$
  $\mathcal{S} \leftarrow \mathcal{S} \cup (\mathcal{S}_{k,L} \cup \mathcal{S}_{k,U})$
end for
if DISTRACTOR **then**
  $L \leftarrow$ RANDOMSAMPLE($\{1 \cdots K\}/V$, N_d)
  for $k \in L$ **do**
    $\mathcal{S} \leftarrow \mathcal{S} \cup$ RANDOMSAMPLE($\mathcal{D}_{k,U}$, N_u)
  end for
end if

tions for SS-FSL. For instance, $\mathcal{D}_{k,L}$ denote all labeled point $x \in class(k)$, and $\mathcal{D}_{k,U}$ be all unlabeled point $x \in class(k)$. Analogous notation holds for our support and query set, $\mathcal{S}$ and $\mathcal{Q}$. Pseudo-code for the construction of a semi-supervised episode is provided in algorithm 1.

2.1.2. Prototypical Networks

Prototypical networks (Snell et al., 2017) aim to train a neural network as an embedding function mapping from input space to a latent space where points of the same class tend to cluster. The embedding function $\Phi(\cdot)$ is used to compute a *prototype* for each class, by averaging the embeddings of all points in the support belonging to that class,

$$\mathbf{p}_c = \frac{1}{|\mathcal{S}_{c,L}|} \sum_{x_i \in \mathcal{S}_{c,L}} \Phi(x_i), \tag{1}$$

where $\mathbf{p}_c$ is the prototype for our cth class. Once prototypes of all classes are obtained, query points are also embedded to the same space, and then classified based on their distances to the prototypes, via a softmax function. For instance, for a point x_i, with an embedding $h_i = \Phi(x_i)$, the probability of belonging to class c is computed by

$$z_{i,c} = p(y_c|x_i) = \frac{\exp\left(-d(h_i, \mathbf{p}_c)\right)}{\sum_{j=1}^{N_c} \exp\left(-d(h_i, \mathbf{p}_j)\right)}, \tag{2}$$

where $d(\cdot, \cdot)$ is the Euclidean distance.

In the semi-supervised variant (Ren et al., 2018), PN use the unlabelled data to *refine* the class prototypes. This is achieved via a soft K-means step. First, the class probabilities for the unlabelled data $z_{i,c}$ are computed as in Eq.2, and the labelled points have a hard assignment, i.e. $z_{i,c}$ is 1 if $x_i \in class(c)$ and 0 otherwise. Then the updated prototype $\tilde{\mathbf{p}}_c$ is computed as the weighted mean of the points assigned to it,

$$\tilde{\mathbf{p}}_c = \frac{\sum_{x_i \in S_U \cup S_L} h_i \cdot z_{i,c}}{\sum_{i=1}^{N} z_{i,c}}. \tag{3}$$

We can see that this is a task adaptation step, which does not directly propagate any learning signal from the unlabelled points to our model parameters. In fact, it may be used only at inference time, and results from (Ren et al., 2018) show that it is provides a significant improvement when used as such. When used during training, information from the unlabelled data flows to the network parameters through the classification loss, and performance is improved even further. However, our original motivation in building on this work, was our belief that this approach while powerful as a task adaptation step, it fails to fully exploit the unlabelled data for meta-training.

SS-FSL with Adaption at test time. Our approach also allows using the former K-means refinement step at inference time, but not during training. This is analogous to the 'Semi-supervised inference' model from (Ren et al., 2018). We perform experiments with adaptation where the adaptation unlabeled data may/may not include samples from a distractor class (samples of classes that are not included in the training episode).

2.2. Semi-Supervised Meta-Training

2.2.1. Virtual Adversarial Training Loss

The first term is the VAT loss $\mathcal{L}_{VAT}$ taken from (Miyato et al., 2018). Underlying this loss is the assumption of *smoothness* (Chapelle et al., 2010) or *local consistency*; two points which are close together should get similar labels. This idea is translated to the practical notion that adding small perturbations to a point should not change its label much. Concretely, let f_θ be our classifier which outputs a probability distribution over classes, $D(\cdot, \cdot)$ some distance function and ϵ a small perturbation: we want $D(f_\theta(x),\ f_\theta(x + \epsilon))$ to be small.

In PN we train a feature extractor rather than a full classifier. Thus to compute the VAT loss for an episode we first generate the prototypes using the labelled points $\mathcal{S}_L$, with this we have a classifier and we can compute the VAT loss

for an episode, $\mathcal{L}_{VAT} = \mathcal{L}_{VU} + \beta_{VAT} \cdot \mathcal{L}_{VL}$, where

$$\mathcal{L}_{VU} = \sum_{x \in \mathcal{S}_U} D(f_\theta(x),\ f_\theta(x+\epsilon)),$$

$$\mathcal{L}_{VL} = \sum_{x \in \mathcal{Q}} D(f_\theta(x),\ f_\theta(x+\epsilon)),$$

and β_{VAT} is a hyperparameter.

To actualize this loss, we use the KL-divergence as our distance function $D(\cdot,\cdot)$, and we choose the adversarial noise vector, proposed in VAT, $\epsilon_{adv} = \arg\max_{\epsilon,\ \|\epsilon\|<r} D(f_\theta(x),\ f_\theta(x+\epsilon))$ as our noise vector. The choice of this noise was motivated by the work in (Goodfellow et al., 2014b) which showed that training classifiers, to be robust in the adversarial directions, improved generalization. An approximation of ϵ_{adv} is provided in (Miyato et al., 2018).

It is worth mentioning that Miyato et al. 2018 found empirically that VAT works better when coupled with an entropy minimization loss (ENT), however no reasoning is provided for this synergy. Intuitively, it is clear that there is some *counterbalancing* effects between the two losses, since the VAT loss may favor smooth functions and the entropy loss favors sharp ones. For instance, let us imagine our network maps all class prototypes to the same point, producing a uniform class distribution for any point, the VAT loss would go to zero, the entropy loss would be maximized. In the upcoming section, we introduce a loss which implicitly requires low entropy outputs, but does much more to leverage the unlabelled data.

2.2.2. Random Walk Loss

The VAT and entropy losses are *local*, it is easy to see that each point's loss is calculated independent of other points. We introduce a *global* loss inspired by (Kamnitsas et al., 2018; Haeusser et al., 2017), where the overall structure of the embeddings manifold is considered, based on random walks over similarity graphs.

Given an episode, we first need to compute the prototypes, and embed the unlabeled points in our latent space. Then we construct a similarity graph between the unlabeled points' embeddings and the prototypes. Our goal is to construct a graph where the points of a class form a tight neighborhood, well separated from other classes. This notion is translated into the idea that a random walker over the graph rarely crosses class decision boundaries. Here, we do not know the labels for our points or the right decision boundaries, so we can not optimize for this directly. Analogous to (Haeusser et al., 2017), we basically imagine our walker starting at a prototype, taking a step to an unlabeled point, and then stepping back to a prototype. The objective is to increase the probability that the walker returns to the same prototype it started from. Additionally, we can imagine our walker taking some steps between the unlabelled points, before taking a step back to a prototype.

Concretely, for an episode with N classes, and M unlabeled points overall, let $A \in \mathbb{R}^{M \times N}$ be the similarity matrix, such that each row contains the negative Euclidean distances between the embedding of an unlabelled point and the class prototypes,

$$A_{i,j} = -\|h_i - \mathbf{p}_j\|^2,$$

where $h_i = \Phi(x_i)$ is the embedding of the ith unlabeled sample, and $\mathbf{p}_j$ is the jth class prototype. Let $B \in \mathbb{R}^{M \times M}$ be the similarity matrix for the unlabelled points $B_{i,j} = -\|h_i - h_j\|^2$. [2]

Transition probability matrices for our random walker are calculated by taking a softmax over the rows of similarity matrices. For instance, the transition matrix from prototypes to points is obtained by softmaxing A^T,

$$\Gamma^{(\mathbf{p}\to x)} = \text{softmax}(A^T),$$

such that $p(x_i|\mathbf{p}_j) = \Gamma^{(\mathbf{p}\to x)}_{i,j}$. Similarly, transition from points to prototypes $\Gamma^{(x\to\mathbf{p})}$, and transitions between points $\Gamma^{(x\to x)}$, are computed by softmaxing A, and B, respectively.

Now, we define our random walker as

$$\Gamma^{(\tau)} = \Gamma^{(\mathbf{p}\to x)} \cdot (\Gamma^{(x\to x)})^\tau \cdot \Gamma^{(x\to\mathbf{p})},$$

where τ denotes the number of steps taken between the unlabelled points, before stepping back to a prototype. An entry $\Gamma_{i,j}$ denotes the probability of ending a walk at prototype j **given** that we have started at prototype i, and the jth row is a probability distribution over ending prototypes, given that we started at prototype j.

Our objective is to maximize the probability in the diagonal entry of the random walker. This can be achieved by minimizing the cross-entropy loss between the identity matrix I and our random walker Γ,

$$\mathcal{L}_{walker} = \sum_{i=0}^{\tau} \alpha_i \cdot H(I, \Gamma^{(i)}),$$

where $H(I,\Gamma) = -\frac{1}{N_c}\sum_{i=0}^{N_c} \log\Gamma_{i,i}$, since those are probability distributions, and α_i are hyperparameters to weigh longer walks. [3]

[2] To avoid our walker taking steps from a node back to itself, the diagonal entries $B_{i,i}$ need to be set to a sufficiently small number.

[3] To be exact, this is the average cross-entropy between the individual rows of I and Γ

Note that, for this equation to be minimized, we need a configuration of points and prototypes such that if our random walker has a non-trivial probability to going from $\mathbf{p}_j$ to x_i, then the probability of going from x_i to $\mathbf{p}_j$ needs to be 1. Of course, the transition probabilities from points to prototypes are also the class probabilities PN output for this point. So implicit in this loss, is that the our classifier should output low entropy distributions, that is the goal of entropy minimization loss.

However, one issue with this loss, is that we could end up graphs where our random walker only visits a small subset of the unlabelled points. To remedy this problem, (Haeusser et al., 2017) introduce a 'visit loss', pressuring the walker to visit all unlabeled points. To do this, we assume that our walker is equally likely to start at any prototype, then we compute the overall probability that each point would be visited when we step from prototypes to points $P = \frac{1}{N_c}\sum_{i=0}^{N_c}\Gamma_i^{(\mathbf{p}\to x)}$, where $\Gamma_i^{(\mathbf{p}\to x)}$ represents a row of the matrix . Then we minimize the standard cross-entropy between this probability distribution and the uniform distribution $\mathcal{L}_{visit} = H(\mathcal{U}, P)$. For stability reasons, our transition matrices $\Gamma^{(\mathbf{p}\to x)}$ and $\Gamma^{(x\to\mathbf{p})}$ are computed as $(1-\eta)\cdot\Gamma + \eta\cdot\mathcal{U}$ where $\mathcal{U}$ is the uniform transition matrix. Our total random walker loss is thus

$$\mathcal{L}_{RW} = \mathcal{L}_{walker} + \mathcal{L}_{visit}.$$

3. Related Works

3.1. Semi-Supervised Learning

3.1.1. Graph-Based SSL

There is a large number of graph-based SSL approaches(Zhu et al., 2005; Zhou et al., 2004). These methods operate over an adjacency matrix W, where $W_{i,j}$ is the similarity between samples $x_i, x_j \in \mathcal{D}_L \cup \mathcal{D}_U$. The similarity graph may be used either to propagate labels from the labeled points to the unlabelled ones, or used to compute some regularization term. The key equation of graph-based approaches is

$$\mathbf{E}(f, W) = \sum_{x_i, x_j \in \mathcal{D}_L \cup \mathcal{D}_U} (f(x_i) - f(x_j))^2 W_{i,j} = \mathbf{f}^t \Delta \mathbf{f} \tag{4}$$

where Δ is the graph Laplacian, and **f**=$[f(x_1)\cdots f(x_n)]$ a vector of labels we attach to our points. Intuitively, we see that minimizing this equation entails having an **f** which gives similar labels to similar points(i.e large $W_{i,j}$), or a W which assigns low weight to points with differing labels. Note that this is also the key equation for spectral clustering [4].

[4]with possible variations depending on the version of the Laplacian being used.

In the inductive setting, we seek to choose a function f which minimizes eq. 4, i.e. it is added as a regularization term. Belkin et al. 2006 show that, under certain conditions, this regularization is a proxy for minimizing the gradient of f *on* the data manifold

$$\int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{P}(x) \tag{5}$$

In the transductive setting, where **f** is a vector and we wish to infer the labels for $x_i \in \mathcal{D}_U$ given the labels for $x_i \in \mathcal{D}_L$. Zhu et al. 2003 show that the unknown labels can be inferred by minimizing eq. 4. This is related to label propagation approaches, where labels *flow* in graph from labelled nodes to unlabelled ones, through high density regions, until equilibrium is reached (Zhou et al., 2004).

Some graph-based SSL approaches formulate their problem in terms of random walks or Markov chains over the similarity graph(Kamnitsas et al., 2018; Haeusser et al., 2017; Szummer & Jaakkola, 2001). These approaches can be inductive(Kamnitsas et al., 2018) or transductive(Szummer & Jaakkola, 2001). In a classification context, they typically aim to produce graphs, and associated labels, such that transition probabilities are high between similarly labelled nodes, and low otherwise. Zhu et. al 2003 note a close relationship between this random walk approach and minimizing eq. 4. It has also been shown in the spectral clustering literature, that minimizing eq. 4 is equivalent to finding a graph, and associated labels, such that a random walker seldom crosses class decision boundaries(Zhang & You, 2011; Von Luxburg, 2007). We consider our objective, stipulating that a random walker starting at a prototype should end up in the same prototype, to be a practical proxy for the aforementioned random walker objective.

It is important to note that, while most methods discussed above assume a given similarity matrix, and optimize for the **f**. That is, the structure of the manifold is fixed, and we optimize **f** over that structure. Our approach, inspired by (Kamnitsas et al., 2018; Haeusser et al., 2017), optimizes the **f** and W matrix jointly; in fact both are products of the same parameters.

3.1.2. Perturbation-Based SSL

The use of perturbations in machine learning is a fundamental technique to improve generalization, and it extends well beyond SSL. Dropout (Sajjadi et al., 2016b) is used in supervised learning as a regularizer to prevent feature co-adaptation and improve performance. It has also been show to operate as a Variational approximation of the posterior of neural network parameters (Gal & Ghahramani, 2016). Gaussian noise is also used in Variational Autoencoders to sample from the posterior over latent encodings

(Kingma & Welling, 2013). In all those cases, the system is required to output the *right* answer while being subjected to noise. In SSL, where the answer is not available, noise is used to enforce consistency. The general penalty here is of the form

$$D(f(x),\ f_{perturb}(x))$$

Where D is a distance function, $f(x)$ the output of our function on x, and $f_{perturb}(x)$ is the output with an added perturbation. The perturbation could be to the model parameters(Park et al., 2018), intermediate representations(Bachman et al., 2014), or to the input points(Miyato et al., 2018; Sajjadi et al., 2016a).

There are a few interpretations for this kind of approach in the literature. Bachman et al. 2014 relate their method to the notion that noise robustness improves generalization. Wager et al. 2013 state the intuition behind their approach is to find model weights that make confident predictions on the labelled as well as the unlabelled data. Temporal ensembling (Laine & Aila, 2016) presumes that the predictions of an ensemble, are more accurate than those of a single model under taken from the ensemble. Thus ensemble predictions can be used to train single instances from the ensemble.

Lecouat et al. 2018 take an interesting perturbation-based approach which minimizes eq. 5, as with numerous graph-based approaches. The main idea is to estimate eq. 5 by perturbing points on the data manifold. In order to estimate the norm of the gradient *on* the data manifold, noise must be added in the latent space of the data. To achieve this, a generative adversarial network (Goodfellow et al., 2014a) to model the data manifold, and the latent space of the generator is then considered to be a faithful model of the data distribution and manifold. Giving rise to the approximation

$$\int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{P}(x) \approx \frac{1}{n}\sum_{i=1}^{n} \|J_z f(g(z^i))\|^2 \qquad (6)$$

Where g is the generator, and J is the Jacobian and its norm is then approximated with $\|f(g(z)) - f(g(z+\epsilon))\|^2$ where ϵ is Gaussian noise.

From this work, we see some connection between the perturbation-based and the graph-based approaches. We can now consider the approach where noise is added directly in the input space, as in VAT. This means that our perturbed points can be 'knocked off' the data manifold, and if we were to perform the same estimation, as above, we would be estimating the norm of the gradient *around* the manifold, rather than *on* the manifold as in 5.

In VAT, we are not using Gaussian noise, but adding the adversarial noise, and this has been shown to be a big component in its effectiveness. Moreover, we are using the KL-divergence to measure change, rather than the norm. However, it is clear that in VAT, functions would be penalized for changing rapidly *around* the manifold. In contrast to graph-based approaches, where change strictly off the manifold is ignored.

3.2. Few-Shot Learning

The few-shot learning problem may be defined as training over a distribution of tasks $\mathcal{P}_{train}(\mathcal{T})$ where each task is an episode as described in 2.1, and at test time tasks are drawn from a related distribution of tasks $\mathcal{P}_{test}(\mathcal{T})$. Much current wave of FSL models can be broadly classified into three strategies, learning a shared metric (Vinyals et al., 2016; Snell et al., 2017; Satorras & Estrach, 2018; Sung et al., 2018) , a shared initialization (Finn et al., 2017), a shared optimizer (Ravi & Larochelle, 2017), or a generic inference model (Santoro et al., 2016; Mishra et al., 2018).

Generic inference models aim for versatility and generality, by relying on a recurrent neural network, which take as input the task's training data, and out output the solution. For few-shot learning, SNAIL(Mishra et al., 2018) takes as input the support set as sequence of example-label pairs, plus a query point, and it outputs the label for that query point.

Optimization based techniques aim to learn an optimizer, which then handles the task adaptation step. In (Ravi & Larochelle, 2017), the model is comprised of a recurrent neural network, playing the role of the optimizer, and a convolutional neural network which does the actual classification. Two things are learned during meta-training, the parameters of the optimizer, and suitable shared initialization for the classifier. Given the task support set as input, the optimizer network then adapts the weights of the classifier for the task. Initialization based techniques , such as MAML (Finn et al., 2017), strip down this approach and only learn a suitable initialization for the classifier, the adaptation step is then done via gradient descent.

Metric based approaches seek to embed the data into a space where points of the same class are close together. After the embedding, there are several ways to infer labels for query points. Prototypical networks, as discussed in 2.1, create a prototype for each class and then perform classification based on the distance between a query point and the class prototypes. Relation networks (Sung et al., 2018) perform pairwise comparisons between query and support points. In (Satorras & Estrach, 2018) the embeddings of the support and query sets are pushed into a graph convolutional neural network which propagates labels to the query points.

4. Experiments

In this section, we investigate the performance of our approach on two well established benchmarks on the semi-

Model	Omniglot	Mini-Imagenet	
	1-shot	1-shot	5-shot
PN (Ren et al., 2018)	94.62 ± 0.09	43.61 ± 0.27	59.08 ± 0.22
Our CPN: (PN + VAT)	95.66 ± 0.21	44.63 ± 0.21	64.02 ± 0.20
Our CPN: (PN + VAT + ENT)	97.14 ± 0.16	44.48 ± 0.22	66.94 ± 0.20
Our CPN (PN + RW)	97.96 ± 0.07	50.33 ± 0.27	66.99 ± 0.24
Our CPN (final:PN+RW+VAT)	$\mathbf{98.03 \pm 0.11}$	$\mathbf{51.03 \pm 0.23}$	$\mathbf{67.78 \pm 0.20}$

Table 1. Ablation Study

Model	Omniglot	Mini-Imagenet	
	1-shot	1-shot	5-shot
PN_{all}(Snell et al., 2017)	98.8	49.4	68.2
PN(Ren et al., 2018)	94.62 ± 0.09	43.61 ± 0.27	59.08 ± 0.22
MetaGAN	97.58 ± 0.07	50.35 ± 0.23	64.43 ± 0.27
Our CPN	$\mathbf{98.03 \pm 0.11}$	$\mathbf{51.03 \pm 0.23}$	$\mathbf{67.78 \pm 0.20}$

Table 2. Semi-supervised meta-learning (without adaptation)

supervised few-shot learning setting. We start by covering some details about datasets and the benchmarks, followed by results and discussions.

4.1. Semi-Supervised Few-Shot Learning Benchmarks

Omniglot (Lake et al., 2011) is a dataset of 1,623 handwritten characters from 50 alphabets. Each character was drawn by 20 human subjects. We follow the few-shot setting proposed by (Vinyals et al., 2016), in which the images are resized to 28 28 pixels and rotations in multiples of 90° are applied, yielding 6,492 classes in total. These are split into 4,112 training classes, 688 validation classes, and 1,692 testing classes.

Mini-ImageNet (Vinyals et al., 2016) is a modified version of the ILSVRC-12 dataset (Russakovsky et al., 2015), in which 600 images for each of 100 classes were randomly chosen to be part of the dataset. We rely on the class split used by (Ravi & Larochelle, 2017). These splits use 64 classes for training, 16 for validation, and 20 for test. All images are of size 84×84 pixels.

For all experiments, our labeled/unlabeled split follows previous works (Ren et al., 2018; Zhang et al., 2018). For Omniglot, we sample 10% of the points in each class to form the labeled split, for MiniImagenet 40%. All the results presented here are for 5-way classification.

Hyper-parameter Selection. The hyper-parameters of our CPN approach are selected based on the performance on the validation classes provided with Omniglot and Mini-ImageNet datasets. The key hyper-parameters of our approach are λ and VAT $\|\epsilon\|$.

4.2. Ablation Study.

In order to understand the contribution of the local and global consistency components of our approach, we did an ablation study by training our CPN with the individual components of our semi-supervised loss, namely PN + $\mathcal{L}_{VAT}$ and PN + $\mathcal{L}_{RW}$. This contrasts the effects of enforcing local or global consistency. Table 1 illustrates our ablation studies. Note that PN denotes the standard PN trained on the labelled split of the data. In addition, we present results for VAT coupled with a Shannon entropy minimization loss, as this has been shown to boost the performance(Miyato et al., 2018). We denote these two variations as VAT and $VAT + ENT$, respectively. Note that although the ENT encourage confident predictions when integrated with VAT, it does not encourage smoothness over the data manifold like our RW loss. This explains the advantage of integrating VAT loss with RW loss instead of ENT loss as shown in table 1 making it the best performing model in our results on both Ominiglot and Mini-imagenet datasets.

All Our mini-Imagenet results are on models trained on 5-shot episodes. Note that PN+VAT performs poorly in the 1-shot setting, when the model is trained on 5-shot, however, with 1-shot training we get a higher accuracy(46.4%). This behavior has also been reported in (Snell et al., 2017), so it is interesting that PN+RW and CPN perform well in 1-shot tests when trained on 5-shots. Note that all the experiments in Table 1 where there are performed without using unlabeled data at test time (no adaptation) which we study in the following sections.

Model	Omniglot	Mini-Imagenet	
	1-shot	1-shot	5-shot
PN	94.62 ± 0.09	43.61 ± 0.27	59.08 ± 0.22
PN+ Semi-supervised inference	97.45 ± 0.05	49.98 ± 0.34	63.77 ± 0.20
PN+ Soft K-means	97.25 ± 0.10	50.09 ± 0.45	64.59 ± 0.28
PN+ Soft K-means + cluster	97.68 ± 0.07	49.03 ± 0.24	63.08 ± 0.18
PN+ Masked soft K-means	97.52 ± 0.07	50.41 ± 0.24	64.39 ± 0.24
Ours: CPN	98.03 ± 0.11	51.03 ± 0.23	67.78 ± 0.20
Ours: CPN + semi-supervised inference	$\mathbf{99.30 \pm 0.04}$	$\mathbf{56.91 \pm 0.25}$	$\mathbf{70.11 \pm 0.19}$

Table 3. Adaptation Experiments without distractor classes

Model	Omniglot	Mini-Imagenet	
	1-shot	1-shot	5-shot
PN	94.62 ± 0.09	43.61 ± 0.27	59.08 ± 0.22
PN+ Semi-supervised inference	95.08 ± 0.09	47.42 ± 0.33	62.62 ± 0.24
PN+ Soft K-means	95.01 ± 0.09	48.70 ± 0.32	63.55 ± 0.28
PN+ Soft K-means + cluster	97.17 ± 0.04	48.86 ± 0.32	61.27 ± 0.24
PN+ Masked soft K-means	$\mathbf{97.30 \pm 0.30}$	49.04 ± 0.31	62.96 ± 0.14
Ours: CPN	96.44 ± 0.11	50.2 ± 0.23	64.1 ± 0.26
Ours: CPN + semi-supervised inference	96.76 ± 0.09	$\mathbf{53.76 \pm 0.23}$	$\mathbf{66.17 \pm 0.21}$

Table 4. Adaptation Experiments with distractor classes

4.3. Semi-supervised meta-learning

We evaluate the effectiveness of CPN and compare it to the state-of-the art semi-supervised few-shot learning methods (PN (Ren et al., 2018) and MetaGAN (Zhang et al., 2018). CPN outperform MetaGAN(Zhang et al., 2018) in all tests, despite having less than half the trainable parameters; as both models use the same architecture as a discriminator/feature extractor, but MetaGAN employs an additional generator that is larger than the discriminator. Our model also significantly improves on the PN baseline (Ren et al., 2018). As an upper bound, we also report the performance of Prototypical Networks (Snell et al., 2017) where the labels of the designated unlabeled set are used for training; denoted as PN_{all}.; see Table 2. Note that our method performs closely to PN_{all}, even outperforming it in the case of 1-shot Mini-Imagenet. The CPN results are all for models trained on 5-shot episodes, with with the number of unlabelled points $N_u = 10$.

4.4. Semi-supervised meta-learning with Adaptation

We evaluate the effectiveness of our proposed approach where additional unlabelled data is present during test time; also known as adaptation. Our results presented in tables 3 and 4, are from the same trained model from the previous section 4.3 but with the additional 'semi-supervised' inference step, applied at test time. The training and testing episodes are constructed exactly as in (Ren et al., 2018) for fair comparison. At test time, the number of unlabelled points per class $N_u = 20$ for Mini-imagenet and $N_u = 5$ for ominiglot. The first five rows are results from (Ren et al., 2018) in tables 3 and 4 which represents different adaptation techniques to exploit the additional unlabelled points for adaptation. Soft-Kmeans model performs a prototype refinement step as described in Section 3, during training and testing. The semi-supervised inference model performs that refinement during testing only. The Soft Kmean+cluster and Masked soft Kmeans models are models with additional components to better handle distractor classes, however they may provide benefits in the absence of distractors. We leverage unlabled data at test time with our approach by a simple semi-supervised inference step adapted from (Ren et al., 2018). We also tried other variants but we found "semi-supervised inference" both simple and effective.

Learning without distractor classes. In table 3, we can see that CPN can leverage unlabelled data during adaptation with an advantage over competing methods on both mini-Imagenet and omniglot. For example, in the more challenging mini-Imagenet, our performance (CPN+semi-supervised inference) is 70.11% in the 5-shot setting, which is 5.5% better than the runner-up method (64.59%). For reference, we also report CPN without semi-supervised inference which performs 6% lower indicating the effectiveness of the combination of our CPN approach

Learning with distractor classes. In table 4, we evaluate our model on a more challenging and realistic setting where distractors labels are included (unlabeled data that does not belong to any of the classes in the episode). For all the adap-

tation experiments and only during test time, the number of distractor classes $N_d = 5$.At training time for the distractor case, the number of unlabelled points per class $N_u = 5$ for Mini-imagenet and for ominiglot. We can see that even with distractors, CPN improves significantly on the baseline. With the additional semi-supervised inference at test time, CPN are state-of-art on Mini-imagenet.

5. Conclusion

SS-FSL is relatively unexplored yet challenging and important task that aims learning from few-examples while leverages unlabeled data. In this paper, we investigated the value of local and global consistency losses to learn the the data distribution efficiently and hence facilitate few-shot recognition. Local consistency is achieved by promoting the neural network prediction stability against noise with inspiration from VAT (Miyato et al., 2018). More importantly, we proposed a global consistency random-walk loss that encourages the data to be magnetized around the class prototypes.

While the local consistency loss has an improvement on the performance, we found out that our global consistency loss significantly improves the performance in SS-FSL. Our experiments and results set the state-of-the-art on most benchmarks.

References

B Tenenbaum, J., Kemp, C., L Griffiths, T., and D Goodman, N. How to grow a mind: Statistics, structure, and abstraction. *Science (New York, N.Y.)*, 331:1279–85, 03 2011. doi: 10.1126/science.1192788.

Bachman, P., Alsharif, O., and Precup, D. Learning with pseudo-ensembles. In *NIPS*, 2014.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 11 2006.

Chapelle, O., Schlkopf, B., and Zien, A. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. ISBN 0262514125, 9780262514125.

Dai, Z., Yang, Z., Yang, F., Cohen, W. W., and Salakhutdinov, R. Good semi-supervised learning that requires a bad gan. In *NIPS*, 2017.

Dodge, S. and Karam, L. A study and comparison of human and deep learning recognition performance under visual distortions. In *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*, pp. 1–7. IEEE, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014b.

Haeusser, P., Mordvintsev, A., and Cremers, D. Learning by association a versatile semi-supervised training method for neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 626–635, July 2017. doi: 10.1109/CVPR.2017.74.

Hochreiter, S., Younger, A. S., and Conwell, P. R. Learning to learn using gradient descent. In *IN LECTURE NOTES ON COMP. SCI. 2130, PROC. INTL. CONF. ON ARTI NEURAL NETWORKS (ICANN-2001*, pp. 87–94. Springer, 2001.

Kamnitsas, K., Castro, D. C., Folgoc, L. L., Walker, I., Tanno, R., Rueckert, D., Glocker, B., Criminisi, A., and Nori, A. V. Semi-supervised learning via compact latent space clustering. In *ICML*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

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

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011.

Lecouat, B., Foo, C. S., Zenati, H., and Chandrasekhar, V. R. Semi-supervised learning with gans: Revisiting manifold regularization. *CoRR*, abs/1805.08957, 2018.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1DmUzWAW.

Miyato, T., Maeda, S.-i., Ishii, S., and Koyama, M. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., and Goodfellow, I. Realistic evaluation of semi-supervised learning algorithms. 2018. URL https://arxiv.org/pdf/1804.09170.pdf.

Park, S., Park, J.-K., Shin, S.-J., and Moon, I.-C. Adversarial dropout for supervised and semi-supervised learning. In *AAAI*, 2018.

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pp. 3546–3554, 2015.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations*, 2017.

Ren, M., Ravi, S., Triantafillou, E., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJcSzz-CZ.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1163–1171, 2016a.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 1163–1171, 2016b.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. P. Meta-learning with memory-augmented neural networks. In *ICML*, 2016.

Satorras, V. G. and Estrach, J. B. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJj6qGbRW.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Szummer, M. and Jaakkola, T. S. Partially labeled classification with markov random walks. In *NIPS*, 2001.

Thrun, S. Learning to learn. chapter Lifelong Learning Algorithms, pp. 181–209. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-8047-9. URL http://dl.acm.org/citation.cfm?id=296635.296651.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Wager, S., Wang, S., and Liang, P. Dropout training as adaptive regularization. *Advances in Neural Information Processing Systems*, 07 2013.

Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., and Song, Y. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 2371–2380, 2018.

Zhang, X. and You, Q. An improved spectral clustering algorithm based on random walk. *Frontiers of Computer Science in China*, 5(3):268, 2011.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. Learning with local and global consistency. In *Advances in neural information processing systems*, pp. 321–328, 2004.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, 2003.

Zhu, X., Lafferty, J., and Rosenfeld, R. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, language technologies institute, school of , 2005.