FEWMATCH: DYNAMIC PROTOTYPE REFINEMENT FOR SEMI-SUPERVISED FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Semi-Supervised Few-shot Learning (SS-FSL) investigates the benefit of incorporating unlabelled data in few-shot settings. Recent work has relied on the popular Semi-Supervised Learning (SSL) concept of iterative pseudo-labelling, yet often yield models that are susceptible to error propagation and are sensitive to initialisation. Alternative work utilises the concept of consistency regularisation (CR), a popular SSL state of the art technique where a student model is trained to consistently agree with teacher predictions under different input perturbations, without pseudo-label requirements. However, applications of CR to the SS-FSL set-up struggle to outperform pseudo-labelling approaches; limited available training data yields unreliable early stage predictions and requires fast convergence that is not amenable for, typically slower to converge, CR approaches.

In this paper, we introduce a prototype-based approach for SS-FSL that exploits model consistency in a robust manner. Our Dynamic Prototype Refinement (DPR) approach is a novel training paradigm for few-shot model adaptation to new unseen classes, combining concepts from metric and meta-gradient based FSL methods. New class prototypes are alternatively refined 1) explicitly, using labelled and unlabelled data with high confidence class predictions and 2) implicitly, by model fine-tuning using a data selective CR loss. DPR affords CR convergence, with the explicit refinement providing an increasingly stronger initialisation. We demonstrate method efficacy and report extensive experiments on two competitive benchmarks; *mini*ImageNet and *tiered*ImageNet. The ability to effectively utilise and combine information from both labelled base-class and auxiliary unlabelled novel-class data results in significant accuracy improvements.

1 INTRODUCTION

Few-Shot Learning (FSL) has recently made steady progress in the directions of both metric learning (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018; Qiao et al., 2018), where class representative features are learned to optimise intra- and inter-class distances, and meta-gradient approaches (Finn et al., 2017; Antoniou et al., 2018; Rajeswaran et al., 2019; Rusu et al., 2018) that focus on optimising model convergence with very few training examples. Despite recent progress, FSL performance remains limited by the small available data from which to learn from. One promising direction for progress involves introducing unlabelled training examples, allowing for expansion of training set variability without increasing data labelling costs. Recent work has shown that this strategy, referred to as semi-supervised few-shot learning (SS-FSL), can substantially boost FSL performance in classification settings (Ren et al., 2018; Li et al., 2019b). These works take advantage of semi-supervised learning (SSL) techniques, that historically focus on large data regimes, to leverage information from additional unlabelled samples in combination with state of the art FSL approaches. State of the art SS-FSL (Liu et al., 2018; Li et al., 2019b) relies on popular SSL techniques of label propagation (Iscen et al., 2019), propagating label predictions to unlabelled data, and self-training (Lee, 2013) that repeatedly labels unlabelled data, based on confidence scores, and then retrains with this additional pseudo-annotated data. An important drawback of such strategies is their reliance on *iteratively extending the training set using pseudo-label predictions*. Building on pseudo-label decisions can propagate and amplify errors during training, yielding brittle methods sensitive to model initialisation and noisy data. This problem is exacerbated in few-shot scenarios, where available labelled data is highly limited and pseudo labels therefore have larger influence.

In light of these limitations, alternative work has explored the use of self-supervision techniques (Gidaris et al., 2019; Yu et al., 2020) to leverage information from unlabelled data. This involves the introduction of auxiliary tasks and artificial labels (e.g. image rotation prediction, jigsaw puzzles) or training process regularisation via a low density assumption (regularisation of consistency). These techniques are able to exploit unlabelled data without introducing reliance on pseudo-labels. Notably, Consistency Regularisation (CR) (Tarvainen & Valpola, 2017; Laine & Aila, 2016; Berthelot et al., 2019b;a) regularises models to output consistent predictions under varying input perturbations. This constitutes a state of the art SSL strategy, typically outperforming pseudo-label approaches in large data regimes. In SS-FSL settings, however, self-supervision methods struggle to outperform pseudo-labelling approaches and fail to fully exploit the benefits, especially in the lowest data regimes. This commonly results in more modest improvements from the use of unlabelled data.

In this work we propose a strategy that enables harnessing of the aforementioned strong performance of CR in standard SSL, for the SS-FSL setting. We hypothesise that CR currently fails in the SS-FSL scenario due to 1) slow convergence of CR techniques (Berthelot et al., 2019a), which is in conflict with FSL fast convergence requirements to alleviate overfitting risks and 2) poor reliability of model predictions in early stages, when training with limited data. We introduce a novel method specifically designed to address these issues and demonstrate empirically that our strategy allows successful exploitation of CR in the SS-FSL setting, outperforming state of the art techniques.

Our formulation exploits the popular concept of prototypes (Snell et al., 2017), commonly used in metric-learning based FSL. Prototypes $\mathcal{P}=\{p_1, p_2, \ldots, p_{C_b}\}$ are learned global feature representations, each describing a particular class to recognise. Class prototypes are typically defined as the average feature representation of the labelled set. They are learned using a set of base classes such that the distances between input samples, of a given class, and the respective class prototype is minimised (else maximised). Our approach builds on the imprinted weights model (Qi et al., 2018), a variant of prototypical networks, that use a simple normalisation trick to learn prototypes as classifier weights in an end-to-end manner (c.f. commonly used episode training Snell et al. (2017)). Our proposed two-stage approach comprises pre-training on base classes, followed by our key innovation, a Dynamic Prototype Refinement (DPR) on novel classes. Using the imprinted weights (IW) model we are able to seamlessly introduce an auxiliary CR loss in our base training process. This allows to leverage unlabelled data from base classes and learn a robust initialisation for our DPR stage. Our novel DPR method exploits unlabelled samples *from novel classes* towards learning prototypes of higher quality. Our approach alternates between explicit updating of prototypes using selected unlabelled samples yielding the most confident predictions (i.e. nearest to their assigned class prototype), and implicit fine-tuning of the model with CR on a second selection of unlabelled samples. We will show that alternating between typically smaller, more conservative updates (implicit refinement) and larger, often times more disruptive feature averaging based updates (explicit refinement), results in faster convergence for CR and often large performance gains, whilst at the same time affording robustness to pseudo-labelling errors. We highlight that in contrast to pseudo-labelling based approaches (Liu et al., 2018; Li et al., 2019b); *estimated labels are not propagated* and are used exclusively to strengthen prototype initialisation, prior to fine-tuning. It is this property that enables recovery from potential erroneous labels and the prevention of gradual drift.

In summary, **our contributions** are three-fold: **(a)** We present "Fewmatch"; a novel semi-supervised few-shot learning approach that robustly exploits the concept of consistency regularisation, alleviating the requirement of iterative pseudo-labelling and consistently outperforming approaches that alternatively do possess such a requirement. **(b)** We introduce a dynamic prototype refinement process, a novel training paradigm designed to harness the power of CR in few-shot regimes through the use of both implicit and explicit prototype refinement steps. **(c)** Extensive experiments demonstrate that we achieve state of the art performance on two standard benchmarks, outperforming prior CR and self-supervised methods with significant accuracy gains. Further to this, we additionally explore more realistic few-shot test conditions in terms of inequalities relating to unlabelled data availability.

2 Related work

Semi-supervised learning Existing SSL methods generally fall into two categories: (1) Pseudo-labelling and (2) Consistency Regularisation. Techniques in the former category iteratively assign pseudo labels to the unlabelled samples such that they can then be used with a supervised loss.

These include directly using the network class prediction (Lee, 2013) and graph-based label propagation (Iscen et al., 2019). A number of SSL works build on the second category of Consistency Regularisation (Sajjadi et al., 2016; Laine & Aila, 2016; Tarvainen & Valpola, 2017), and have achieved impressive results. The crux of the idea of CR is to encourage invariant (stable) predictions for a given sample under different perturbations towards improving class decision boundaries. CR ideas were first explored in (Sajjadi et al., 2016; Laine & Aila, 2016) and extended in (Tarvainen & Valpola, 2017) where the authors propose a mean teacher framework to perform CR between a student and teacher model in a learning paradigm involving models that share the same architecture and teacher parameters are updated as an exponential moving average of the student weights. Several works such as ICT (Verma et al., 2019), Mixmatch (Berthelot et al., 2019b) and Remixmatch (Berthelot et al., 2019a) have then enabled sample perturbations by creating variants of mixup samples (Zhang et al., 2017) that can then be further perturbed. Encouraged by the benefits that result from representing class information using prototypes (Snell et al., 2017; Qi et al., 2018), we take an alternative approach to CR in the context of SS-FSL and influence model prediction by considering a measure of distance between unlabelled data and class prototypes.

Few-Shot Learning Existing FSL approaches can be broadly divided into two categories (1) Metric based (Snell et al., 2017; Vinyals et al., 2016; Qi et al., 2018) and (2) Gradient based (Finn et al., 2017; Antoniou et al., 2018; Rajeswaran et al., 2019; Rusu et al., 2018). Metric based methods aim to learn global class feature representations (i.e. prototypes) whose distance is minimal to samples of the same class. In this paper, we take advantage of one such method; Imprinted weights (Qi et al., 2018) in order to provide per class prototypes. One of the main advantages of this approach is that it does not require the standard, restrictive episode training strategy. Episode training is framed as a sequence of artificially designed FSL tasks with fixed category and labelled sample counts and also imposes an identical test time set-up. This in theory affords us greater flexibility with the learning problem definition, allows for consideration of more practical problem setups, and for easier combination with techniques from other fields such as integration of auxiliary losses.

Semi-Supervised Few-Shot Learning (SS-FSL) Existing SS-FSL approaches are based on the pseudo-labelling strategy that was discussed in the context of SSL. Ren et al. (2018) propose mask soft K-means, based on the metric learning approach, ProtoNets (Snell et al., 2017). The authors use a soft K-means and iteratively assign pseudo labels to tune prototypes. More recently (Liu et al., 2018) propose a Transductive Propagation Network (TPN) that propagates labels from unlabelled data through a graph of samples and meta-learns key hyperparameters. Li et al. (2019b) proposed a Learning to Self-Train (LST) approach that is based on self-training and meta-learns a soft weighting network to control the influence of pseudo labelled samples and reduces label-noise during training. Another set of approaches explore the use of self-supervision to leverage unlabelled data. Gidaris et al. (2019) introduce auxiliary tasks, exploiting image rotations and jigsaw puzzles to learn better feature representations. More aligned with SSL approaches and closer to our work, Yu et al. (2020) pre-trains a classification model on base classes (in the standard FSL setting) using the imprinted weights model and fine-tunes (without prototypes) on novel classes using the CR based mixmatch algorithm (Berthelot et al., 2019b). While these approaches alleviate the error propagation problem that is common when pseudo labelling is employed (Laine & Aila, 2016), their performance gains remain limited; the techniques are not specifically adapted to the few-shot setting. Conversely, we propose a unique training scheme that iteratively refines prototypes using both explicit average feature representation and implicit CR refinement. We will show that this enables more flexible feature adaptation to novel tasks and obtains more accurate class prototypes.

3 METHODOLOGY

We consider a base training dataset $D_{base}=\{X_b^l, X_b^u\}$ comprising Labelled Data (LD) $X_b^l=\{\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_n^l\}$ with labels $Y_b=\{y_1, \ldots, y_n\}$, as well as an additional set of Unlabelled Data (UD) $X_b^u=\{\boldsymbol{x}_1^u, \ldots, \boldsymbol{x}_m^u\}$. All examples in D_{base} belong to one of C_b base categories. Our novel dataset D_{novel} contains C_n disjoint novel classes each with only a handful of labelled samples (e.g. ≤ 5) as well as a further limited set of unlabelled samples per class (e.g. ≤ 100) with which to fine-tune the model. D_{novel} further comprises UD used for evaluation. Our objective, similarly to standard few-shot settings, is to learn a classifier capable of accurately recognising novel classes, despite having only a limited amount of available LD. However in contrast to standard FSL, we possess additional UD for both base and novel classes, which we aim to leverage to maximise per-

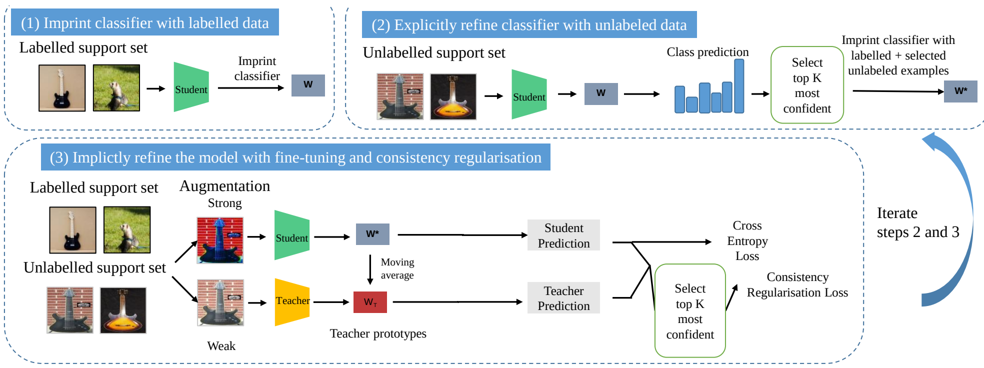

Figure 1: Overview of the Dynamic Prototype Refinement process. See main text for details.

formance. To formalise our setting, we consider that D_{novel} comprises of a fixed *support set* of K_n^l labelled and K_n^u unlabelled examples per class, and refer to the remaining unlabelled test images as the query set Q_n. This C_n-way K_n^l-shot classification problem defines a standard SS-FSL setting.

Our proposed "FewMatch" method first trains a classification model on D_{base} by exploiting the concept of imprinted weights (IW) (Sec 3.1). IW allow end-to-end model training, while at the same time learning global class feature representations (commonly referred to as prototypes (Snell et al., 2017)) utilised as classifier weights. This is achieved by computing predictions as the cosine similarity between input features and classifier. End-to-end training allows seamless introduction of a CR loss, effectively leveraging UD to train a strong feature extractor and learn high quality prototypes. The second stage involves model fine-tuning on D_{novel} in order to leverage UD for novel classes. We introduce a novel training scheme: our Dynamic Prototype Refinement (DPR) process (Sec 3.2). Our iterative strategy alternates between explicit prototype refinement using feature averaging and implicit parameter updates using fine-tuning and CR. The strong initialisation provided by the explicit averaging and longer training times afforded by the iterative scheme enable to successfully harness the power of CR in even the lowest data regimes. An overview of the proposed DPR is provided in Figure 1 with an analogous overview of the base training process in Appendix A.6.

3.1 Base Training: Prototype Driven Consistency Regularisation

In this section we firstly introduce our imprinted weight formulation and then describe the integration of this within a teacher-student framework, enabling the introduction of our CR loss.

Imprinted weights formulation. Our classification model uses a standard architecture, comprising a feature extraction network θ_f, and a classifier defined by a fully connected layer without bias $W \in F \times C_b$, where F is the output dimension of θ_f. The main idea of imprinted weights is to train the model such that, for a given class c, the cosine similarity between the embedding vector $\theta_f(\mathbf{x})$ of input image $\mathbf{x}$ and the corresponding column w_c of W is maximised. By normalising the classifier and embedding vectors, the model can be trained end-to-end using a standard cross entropy loss. In this setting, w_c is regarded as the *prototype* representation of class c and can be learned implicitly without the, typically required, episode training strategy and support set averaging. More formally, for input sample $\mathbf{x}$, the set of classification scores output by the model is $f(x) = \{f^1(x), f^2(x), \ldots, f^c(x), \ldots, f^{C_b}(x)\}$ and the score for a given class c is computed as:

$$f^c(\mathbf{x}) = \frac{\exp\left(\gamma\left(\mathbf{w_c^T}, \theta_\mathbf{f}(\mathbf{x})\right)\right)}{\sum_{\mathbf{i=1}}^{\mathbf{C_b}} \exp\left(\gamma\left(\mathbf{w_i^T}, \theta_\mathbf{f}(\mathbf{x})\right)\right)} \tag{1}$$

where $\mathrm{w_i}$ is the i^{th} column of weight matrix W and the prototype p_i of class i. The scaled cosine similarity is then given by $\gamma\left(\mathrm{w_i^T}, \theta_\mathrm{f}(\boldsymbol{x})\right) = s \cdot \mathrm{w_i^T}\left(\theta_\mathrm{f}(\boldsymbol{x})\right)$. $\mathrm{w_i}$ and $\theta_\mathrm{f}(\boldsymbol{x})$ are normalized using the L_2 norm, and s is a trainable scalar, as introduced by (Qi et al., 2018) to avoid the risk that the cosine distance yields distributions lacking in discriminative power. Finally, the classification loss can be calculated as: $\mathcal{L}_{\text{ce}}(\mathbf{x}) = -\sum_{\mathbf{c}=\mathbf{1}}^{\mathbf{C}_b} \delta_{\mathbf{c},\mathbf{y}} \log \mathbf{f}^{\mathbf{c}}(\mathbf{x})$ where $\delta_{c,y}$ is the Dirac delta function. Defining class prototypes as learnable model weights affords end-to-end training and enables introduction of CR to our model in a natural fashion. These decisions allow us to leverage UD and implicitly refine prototypes without explicit pseudo-labelling. Furthermore, this approach optimises the base class learning process by allowing full exploitation of the available LD without the typical requirement that necessitates simulation of the few-shot set-up (episode training) (Finn et al., 2017).

Consistency Regularisation. We highlight that the described training strategy does not yet leverage UD, available in the considered SS-FSL problem setting. Towards taking advantage of UD, we introduce a CR loss (Tarvainen & Valpola, 2017) that is driven by the learned prototypes. The idea underlying CR is to regularise predictions such that they become invariant to small input perturbations that do not affect class semantics. This strategy has been used successfully for a variety of problems and is particularly appealing in the semi-supervised context as it leverages UD without explicit pseudo-labelling. A key difference in our setting, with respect to conventional SSL, is that our CR loss directly depends on prototype instantiations, as predictions are based on the distance between input and each class prototype. This strategy drives our approach to learn more discriminative and robust prototypes towards maintaining classification accuracy under different input perturbations. Following strategies adopted in the recent SSL state of the art (Berthelot et al., 2019a), we embed our IW model within a teacher-student framework (Tarvainen & Valpola, 2017) where we seek to impose consistency between teacher and student predictions. Both teacher and student networks share the same architecture, however only student weights are optimised by back-propagation. Teacher weights θ_T are computed as an Exponential Moving Average (EMA) of the student weights θ , $\theta_T = (1-\alpha)\theta_T + \alpha\theta$. Such temporal averaging strategies have been shown to yield more robust and accurate models and are therefore desirable in often noisy few-shot settings.

Considering an unlabelled sample u_b we realise sample perturbations, as suggested in (Xie et al., 2019; Berthelot et al., 2019a), by generating $\bar{u}_b$ and $\hat{u}_b$ using *weak* and *strong* augmentations respectively. The weak augmentation sample $\bar{u}_b$ has the goal of improving prediction stability in the teacher network. This strategy helps to constrain the strong augmentation sample prediction. The consistency loss is then computed as: $\mathcal{L}_{\text{cons}}(u_b) = ||\text{Sharp}(f_t(\hat{u}_b), \mathcal{T}) - f_s(\bar{u}_b)||^2$; where f_s and f_t are predictions computed by the student and teacher networks respectively; and $\text{Sharp}(\cdot)$ is a sharpening function, parametrised by temperature $\mathcal{T}$, introduced in (Berthelot et al., 2019b) to reduce the entropy of the label distribution. In summary, the model is trained on the base classes using global loss $\mathcal{L}_{base} = \mathcal{L}_{\text{ce}} + \lambda\mathcal{L}_{\text{cons}}$, where hyperparameter λ balances the relative influence of the terms.

3.2 Dynamic Prototype Refinement

Our training stage, considering D_{base}, yields a model capable of estimating reliable class prototypes on novel, unseen categories. In a standard few-shot setting (i.e. without available UD), prototypes are often estimated directly from the support set and reliable performance can be achieved without further training. In our problem setting, we set the objective of exploiting the additionally available UD in order to obtain strong prototype initialisations that then lend themselves to further refinement. Towards this goal, the main component of FewMatch constitutes our Dynamic Prototype Refinement (DPR) strategy, taking advantage of the UD available from D_{novel}, with the aim of improving model adaption to novel categories. By design our approach is able to improve performance on novel categories despite the presence of limited data regimes. DPR comprises three stages: **(1)** Prototype Initial Inference (PII), via the introduced IW procedure **(2)** *Explicit* prototype refinement using top-K selection and **(3)** *Implicit* prototype refinement using CR. Prototypes are initially estimated during the first step and then dynamically updated using iterations of steps two and three, such that prototype quality is iteratively improved. The remainder of this section provides further detail on steps **(1)**-**(3)** and the iterative process.

Prototype Initial Inference. Given new category j from D_{novel} with support set $S_j = \{\boldsymbol{x_1^s}, y_1^s, \ldots, \boldsymbol{x_n^s}, y_n^s\} \cup \{\boldsymbol{u_1, \ldots, u_m}\}$, compute an initial prototype using the labelled support set as:

$$p_j^* = P(S_j) = \frac{1}{|S_j|} \sum_{\boldsymbol{x_i^s} \in S_j} \theta_f(\boldsymbol{x_i^s}), \tag{2}$$

The estimated prototype is then imprinted in classifier W as $w_j = p_j$ and the process is repeated for each new category (see Figure 1). This allows for recognition of new classes without model retraining and provides high quality initialisation for our dynamic refinement stage.

Explicit Prototype Refinement. We highlight that initial prototypes, computed using Eq. 2, do not make use of the additional UD available for novel classes. Exploiting UD can be considered crucial for novel classes due to the availability of only limited labelled data. Towards reducing prototype biases, we expand the support set using pseudo-labelled UD, where labels are assigned according to respective prediction scores. The prediction scores $f_s(u)$ are again obtained with Eq. 1 using updated prototype estimates and current model parameters. We mitigate the varying quality of pseudo labels by selecting the top-K samples with the most confident predictions per class which, by definition, consist of the K unlabelled samples that are closest to their assigned class' prototypes. This augmentation results in an extended annotated support set defined for each class j as $S^*_j{=}S_j \cup U_j$, where U_j=top-$K(f^j_s(u))$ is the set of unlabelled samples selected for class j. The prototype is then refined using Eq. 2 by replacing S with S^*. Crucially, we emphasise that per stage pseudo-labels are used *uniquely* to update prototypes and that samples, pseudo-labelled at this stage, are considered unlabelled again at the next iteration. Importantly *pseudo-labels are therefore not propagated*, allowing for recovery from potentially erroneous predictions during the subsequent fine-tuning stage.

Implicit Refinement using Consistency Regularisation. Our implicit refinement stage inherits ideas from gradient-based FSL, which typically adapts the entire model to novel classes via a fine-tuning stage. This stage is generally missing from prototype-based methods, which explicitly represent prototypes as an average feature representation, and thus lose the flexibility afforded by learning implicit network parameters. This fine-tuning stage is particularly desirable in our setting, where we seek to maximally leverage the available UD and our prototypes are defined as model weights. It is a natural choice to consider deploying Consistency Regularisation to fine-tune the model, noting that the refined prototypes obtained at this stage afford high quality teacher predictions. We implement the strategy described in Sec. 3.1 to fine-tune the model on novel classes with CR. To further improve robustness to noisy teacher predictions and difficult examples, we adopt a selective prototype CR strategy. By calculating teacher prediction scores $f_t(\bar{u})$ according to their prototype distance, we can select the top-K unlabelled examples with the least ambiguous label predictions to compute the CR loss. Note that this second top-K selection set V will differ from top-K set U computed during the explicit stage, as 1) prototypes were updated 2) they are computed on the teacher model subject to weak input augmentation. The model is fine-tuned for R gradient updates by minimising $\mathcal{L}(\mathbf{x}, v^u_b) = \mathcal{L}_{ce}(\mathbf{x}) + \lambda_{ft}\mathcal{L}_{cons}(v^u)$, where $\mathcal{L}_{ce}$ and $\mathcal{L}_{cons}$ are computed as described in Sec. 3.1, where labelled sample $\mathbf{x}$ is from $\mathcal{D}_{novel}$ and $v^u \in V$.

Dynamic Prototype Refinement. Our implicit and explicit refinement steps allow iterative prototype refinement towards further performance improvement. We alternate between explicit and implicit steps for M iterations, reinitialising estimated pseudo-label at each iteration. Top-K selection, for the first explicit stage, relies on student predictions since teachers are randomly initialised. Teacher predictions, presumed to be more accurate and stable, are used in subsequent iterations. Importantly, we note that teacher parameters are reinitialised before each implicit stage (after explicit selection) thus introducing stochasticity, increasing robustness to pseudo-label errors and aiding loss optimization. Algorithm details for dynamic prototype refinement are provided in Appendix A.5

4 Experiments

Experimental set-up. We evaluated Fewmatch on two standard SS-FSL benchmarks: *mini*ImageNet (Vinyals et al., 2016) and *tiered*ImageNet (Ren et al., 2018), both subsets of the ImageNet dataset (Russakovsky et al., 2015) designed specifically for FSL. *Mini*ImageNet consists of 100 classes with 600 image samples per class. We use the standard 64/16/20 classes split for train/val/test sets (Vinyals et al., 2016) and use $40\%/60\%$ of the data for labelled/unlabelled splits following previous works (Ren et al., 2018; Li et al., 2019b). *Tiered*ImageNet contains 608 classes from 34 super-level categories. These are divided into 20/6/8 coarse super-level categories for train/val/test splits and contain 351, 97 and 160 classes, respectively. We follow the standard semi-supervised split (Ren et al., 2018; Li et al., 2019b), with 10% of the images of each class forming the labelled split and the remaining 90% being the UD. We consider $K^l_n = 5$ way $N{=}1, 5$ shot classification problems and follow the strategy adopted in (Ren et al., 2018; Li et al., 2019b) to

Table 1: Mean classification accuracies of the 5-way 1/5-shot tasks. (**Bold**: Best results per set-up). SL+U setting uses all available training LD (SL setting) with additional UD vs SSL using 10% (*tiered*ImageNet) or 40% LD (*mini*ImageNet). Grey rows: methods using self-supervision.

Setting	Model	Backbone	*mini*Imagenet 1-shot	*mini*Imagenet 5-shot	*tiered*Imagenet 1-shot	*tiered*Imagenet 5-shot
SL	MTL (Sun et al., 2019)	*ResNet-12*	61.20 ±1.80	75.50 ±0.80	-	-
	CTM (Li et al., 2019a)	*ResNet-18*	62.05 ±0.55	78.63 ±0.06	64.78 ±0.11	81.05 ±0.52
	CC+rot (Gidaris et al., 2019)	*WRN-28-10*	62.93 ±0.45	79.87 ±0.33	70.53 ±0.51	84.98 ±0.36
SL + U	CC+rot+unlabelled	*WRN-28-10*	64.03 ±0.46	80.68 ±0.33	-	-
	TransMatch (Yu et al., 2020)	WRN-28-10	63.02±1.07	81.19±0.59	-	-
SSL	MS k-Means (Ren et al., 2018)	4Conv	50.4	64.4	52.4	69.9
	MS k-Means with MTL	ResNet-12	62.1	73.6	68.6	81.0
	TPN (Liu et al., 2018)	4Conv	52.8	66.4	55.7	71.0
	TPN with MTL	ResNet-12	62.7	74.2	72.1	83.3
	LST (Li et al., 2019b)	ResNet-12	70.1 ±1.9	78.7 ±0.8	77.7 ±1.6	85.2 ±0.8
	Ours	ResNet-12	**75.66±0.95**	**82.93±0.62**	**78.70±0.93**	**85.40±0.58**
			Distractor Setting			
SL + U	TransMatch	WRN-28-10	59..32±1.10	79.29±0.62	-	-
SSL	MS k-Means with MTL	ResNet-12	61.0	72.0	66.9	80.2
	TPN with MTL	ResNet-12	61.3	72.4	71.5	82.7
	LST	ResNet-12	64.1	77.4	73.4	83.4
	Ours	ResNet-12	**70.35±0.98**	**80.23±0.66**	**74.24±0.95**	**83.64±0.63**

generate test episodes: we randomly sample K_n^l classes from the test set, N labelled images from each class, 100 unlabelled images as support images and 15 query images.

The previous protocol can be regarded as a standard set-up that we follow for fair comparisons. Towards exploring more realistic few-shot testing scenarios, we consider two additional directions. Firstly, the *distractor* setting (Li et al., 2019b) introduces UD from irrelevant classes, providing a more challenging test environment. Testing involves randomly selecting 100 unlabelled images from three task-irrelevant classes to serve as *distractors* and adding these to the unlabelled set. Table 1 (lower), reports mean accuracy for 600 randomly generated test episodes in comparison to the state-of-the-art for this challenging setting. Secondly, the absence of an episode-based training requirement affords FewMatch additional flexibility and enables more realistic SS-FSL testing schemes, e.g. investigating model adaptation capabilities under varying amounts of UD per class. We provide classification accuracies for settings with unbalanced class sampling: (1) randomly selecting between 70-130 US per class; (2) 80-120 US per class. As Table 2 shows, FewMatch performance retains stability in unbalanced settings, c.f. the balanced default (exactly 100 US per class).

The method was implemented with PyTorch (Paszke et al., 2017) using the same ResNet-12 backbone as (Li et al., 2019b). For base category training, we follow parameters used in (Gidaris & Komodakis, 2018): our model is optimised using SGD with momentum 0.9, weight decay 0.0005, mini-batchsize 256 (128 LD and 128 UD) for 30 epochs. All input images were resized to 84×84. The learning rate was initialised to 0.1, and updated to 0.01 at epoch 20. Followng SSL practice (Tarvainen & Valpola, 2017), weighting parameter λ is defined as a linear ramp-up function increasing from 0 to 300 in the first 15 epochs. We set the total number of DPR iterations as $M = 3$ and each implicit refinement step fine-tunes the model for 20 steps with 0.01 learning rate. Each mini-batch comprises all LD and 40 randomly sampled UD per-category. We linearly increase weighting parameter λ_{ft} from 0 to 10 in the first 10 steps. The number of unlabelled samples selected is set to $K = 25$. We set EMA rate $\alpha = 0.5$, and $\mathcal{T} = 0.5$. Strong augmentations for the student network are computed using RandAugment (e.g. color, shear) (Cubuk et al., 2019), applying three random operations with magnitude set to 9. Teacher weak augmentations use random cropping and flipping.

Comparison to State-of-the-Art (SOTA) methods. We compared FewMatch with SOTA approaches including (a) 3 FSL and (b) 5 SS-FSL methods in Table 1. We note that several SS-FSL approaches, including FewMatch, outperform SOTA FSL approaches, highlighting the potential of using additional UD to learn more accurate models. We observe that FewMatch outperforms the SS-FSL state of the art in both standard and distractor settings and that strongest performance

Table 2: Ablation study on *mini*Imagenet. PCR: base training prototype Consistency Regularisation; ER: Explicit prototype refinment; IR: Implicit refinement using Selective Consistency Regularisation; DR: Dynamic Refinement

Model Components				*mini*ImageNet	
PCR	ER	IR	DR	1-shot	5-shot
Remixmatch				53.52	66.50
Imprinted-weights (IW)				59.09	75.59
IW + Remixmatch (no mixup)				62.20	76.31
✓				61.59	77.90
✓	✓			71.35	81.75
✓	✓	✓		72.52	82.25
✓	✓	✓	✓	75.66	82.93
Unbalanced Number of Unlabelled Samples					
min/max US 70/130				74.24	82.51
min/max US 80/120				75.14	82.82

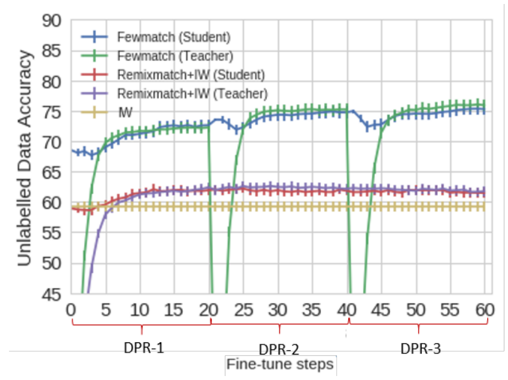

Figure 2: Accuracy on training unlabelled data with $M = 3$ iterations of the DPR stage.

gains are observed in the 1-shot setting. We further highlight that 1) we significantly outperform self-supervision methods that use a more powerful backbone encoder and were trained in a more favourable setting (SL+U: using all base LD with additional UD, vs SSL setting using a fraction of LD only) and 2) the closest SOTA method LST, requires, in contrast to FewMatch, complex episode training, requiring a fixed number of LD and UD at both training and test time.

Ablation experiments. We evaluate the influence of each model component using *mini*ImageNet under 5 way 1/5 shot settings. Specifically, we evaluate the influence of using CR in the base training stage (PCR), Explicit Prototype refinement (ER), Implicit Refinement (IR) and Dynamic Refinement (DR) which iterates between ER and IR. We additionally include three baselines: Imprinted Weights (Qi et al., 2018) (no use of unlabelled data), SOTA CR based SSL method Remixmatch (Berthelot et al., 2019a) (no accounting for the few-shot setup), and Imprinted Weights combined with Remixmatch. We highlight that the latter baseline is highly similar to the method of Yu et al. (2020) and provides context towards the performance expected in the SSL setting. We note that methods using Remixmatch use CR during *both base and novel* training stages and that the latter method is implemented without mixup (used in the Remixmatch method) as the label mixing strategy is not compatible with the prototype approach and would require the definition of infinitely many prototypes. Results are reported in Table 2 and show that each component makes a clear contribution to the performance gain; with ER (providing a strong initialisation) and DR (addressing slow CR convergence rates) yielding the strongest performance gains.

Analysis of the DPR process. Figure 2 evaluates the improved reliability of teacher predictions throughout our DPR process (M=3). We report accuracy on training UD during the DPR stage, compared to baseline imprinted weights + remixmatch (IWR) which uses CR without addressing the underlying challenges. We observe that our iterative process continuously improves performance, successfully exploiting CR towards reaching higher quality predictions. Conversely, the IWR model fails to exploit UD, obtaining a minimal performance gain with respect to baseline FSL method IW.

5 Conclusion

We introduced a novel prototype-driven approach named FewMatch, designed specifically to exploit the power of consistency regularisation in limited data regimes. In contrast with pre-existing state of the art methods, we alleviate requirements for iterative pseudo-labelling, preventing propagation of errors induced by inaccurate model predictions. We go beyond the introduction of self-supervised auxiliary losses and propose a novel training strategy: a dynamic prototype refinement that alternates between explicit pseudo label based updates and implicit model fine-tuning. Our extensive experiments demonstrate that this iterative strategy allows successful exploitation of unlabelled data within a consistency regularisation framework, yielding large performance gains.

REFERENCES

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv*, 2018.

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

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

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv*, 2019.

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

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

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. *arXiv*, 2019.

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

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

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, 2013.

Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, 2019a.

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*, 2019b.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv*, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *CVPR*, 2018.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 2018.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. *arXiv*, 2019.

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

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

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Neurips*, 2016.

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

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 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*, 2018.

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

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

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

Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Self-training with noisy student improves imagenet classification. *arXiv*, 2019.

Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12856–12864, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv*, 2017.

A Appendix

We provide additional material to supplement our work. Section A.1 evaluates the influence of the number of unlabelled samples on FewMatch's performance, and demonstrates the method's ability to leverage unlabelled examples. In Section A.2, we report a comparison between the Semi-Supervised Learning (SSL), Few Shot Learning (FSL) and Semi-Supervised Few Shot Learning (SS-FSL) settings, highlighting the challenges associated with SS-FSL. In Section A.3, we report an additional experiment, studying the influence of Dynamic Prototype Refinement (DPR) iterations M on our model performance. Please refer to our main paper for further method details. In Section A.4, we further synthesize the comparison between *FewMatch* and existing SS-FSL approaches, explicitly providing additional details to highlight the main differences between the considered methods. In Section A.5, we provide pseudocode description of our Dynamic Prototype Refinement process. Finally, Section A.6 provides detailed pseudocode for the first stage of our method (prototype-driven consistency regularisation as described in Section 3.1 of the main paper).

A.1 Influence of the number of Unlabelled samples

We test the impact of using variable amounts of US per class on classification accuracy in the 5-way 1-shot setting on *mini*ImageNet. Results are shown in Figure 3, showing a large increase in performance when including 50 US and a more modest yet consistent improvement as the number of US increases. This highlights the advantage provided by the use of US to complement the few-shot labelled examples, as well as FewMatch's ability to leverage unlabelled examples.

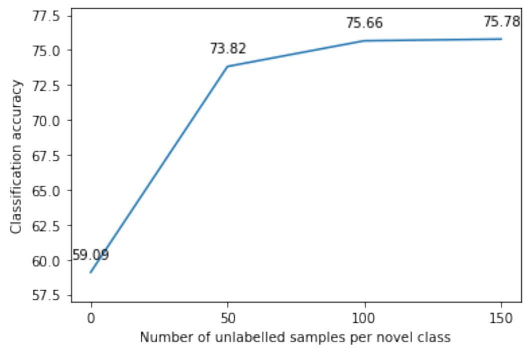

Figure 3: Mean classification accuracy on 5-way 1-shot on *mini*ImageNet with varying amounts of unlabelled samples.

A.2 COMPARISON BETWEEN SSL, FSL, AND SS-FSL SETTINGS

In Table 3, we report training sample counts (labelled and unlabelled) per category used in FSL, SSL and SS-FSL settings. The stated values follow the convention in Ren et al. (2018) (5-way 1-shot) on Mini-ImageNet and, for SSL, we report the setting comprising the minimal LS with respect to recent state of art methods Berthelot et al. (2019a) on the common benchmark, CIFAR-10. Compared to FSL, this table highlights that 1) fewer labelled data is available during the base training stage, increasing the difficulty of obtaining a strong initialisation and 2) a substantial amount of additional unlabelled data is available for novel classes. Compared to SSL, the amount of labelled and unlabelled samples is significantly reduced in the SS-FSL setting (in particular; the unlabelled samples), highlighting the challenges associated with adapting SSL methods to the SS-FSL scenario.

Table 3: Comparison of available per category training Labelled Samples (LS) and Unlabelled Samples (US) between FSL, SS-FSL, SSL)

Data Split	FSL	SS-FSL	SSL
Base classes	600 LS	240 LS + 360 US	-
Novel classes	1 LS	1LS + 100 US	25 LS + 4750 US

A.3 PARAMETER STUDY: DYNAMIC PROTOTYPE REFINEMENT (DPR) ITERATIONS

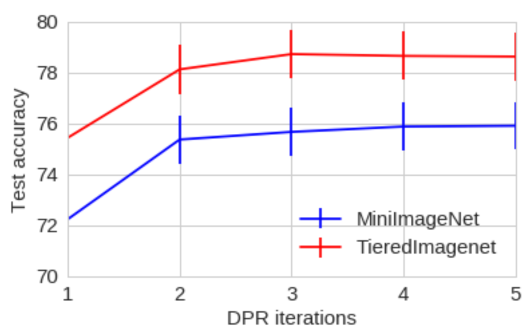

Figure 4: Dynamic Prototye Refinement (DPR) performance with respect to iterations M

We evaluate the influence of DPR iteration count M with respect to model performance in the 5-way 1-shot setting and report respective test accuracies in Figure 4. We observe similar behaviour for both datasets considered (*mini*ImageNet and *tiered*ImageNet), with performance improving and then stabilising for $M \geq 3$. Our model requires only three iterations to reach optimal performance in the investigated settings.

A.4 Comparison of FewMatch and Existing SS-FSL approaches

In Table 4, we provide an additional detailed comparison of FewMatch with state of the art SS-FSL approaches, including Masked Soft k-Means (Ren et al., 2018), Transmatch (Yu et al., 2020) and LST (Li et al., 2019b). We compare six different characteristics of the methods: Base dataset split, Training Strategy, Prototype estimation, Classifier learning approach, backbone encoder adaptation strategy (to novel task) and SSL approach used. Table 2 illustrates that 1) FewMatch provides a more flexible training strategy as it does not require episodic training. This allows consideration of different set-ups at test time in contrast to episodic training that typically enforcing a fixed K-way-N-shot setting. 2) Compared to Masked Soft k-Means, the only other method using prototypes, FewMatch adopts a more flexible prototype learning process by combining feature averaging with fine-tuning. This is enabled by the fact that prototypes are defined as classifier weights, allowing learning of high quality prototype representations. Furthermore, FewMatch adapts the feature backbone to the novel task, reducing the influence of domain shift. 3) In contrast to LST, Fewmatch combines classifier parameter updates with the concept of prototypes, allowing a stronger initialization for the fine-tuning stage to be obtained. 4) In contrast to TransMatch, FewMatch uses fewer labelled training examples in the base training stage, and fine tunes the model using a combination of feature averaging and backpropagation; affording better CR convergence.

Table 4: Comparison of FewMatch to existing SS-FSL approaches

Method	Masked Soft k-Means	Transmatch	LST	FewMatch
Base dataset	60% US+40% LS	100% LS	60% US+40% LS	60% US+40% LS
Training	Episodic	End to end	Episodic	End to end
Prototypes	Feature averaging	/	/	Iterative feature
Classifier	/	backpropagation	backpropagation	averaging and backprop
Feature	Fixed	Fixed	Adapted to novel task	Adapted to novel task
Learning	Pseudo label	CR	Pseudo label	CR

A.5 Dynamic Prototype Refinement algorithm

We provide an algorithmic description of our Dynamic Prototype Refinement (DPR) process in Algorithm 1. DPR contains three steps: 1) Prototypes initial inference; 2) Explicit prototype refinement; 3) Implicit refinement using CR. We alternate between explicit and implicit refinement for M epochs after the initial inference step.

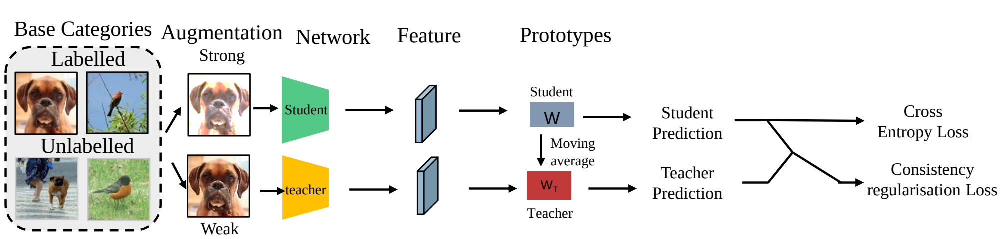

Figure 5: Base training process

Algorithm 1 Dynamic Prototype Refinement

Input: labelled examples $\mathcal{S} = \{S_1, \ldots, S_j, S_{C_n}\}$, and unlabelled examples $\mathcal{U}$; Number of novel categories: C_n ; number of iterations M; number of fine-tuning steps R; pre-trained student and teacher model parameters θ,θ_T; weighting parameters λ_{ft}, α.
Output: Prototypes of novel categories W^{**}, student model parameters θ;
Prototypes initial inference: $W \leftarrow \{p_1^*, p_2^*, \ldots, p_{C_n}^*\}$, calculate $p_j^* \leftarrow P(\mathcal{S}_j)$ by Eq equation 2
For i = 1 **to** M :
    Explicit prototype refinement
        $U_j \leftarrow$top-$K(f^j_{t,\theta_T,W_T}(u)), \forall j \in 1, \ldots, C_n$, f^j_{t,θ_T,W_T} computed by equation 1 with parameters θ_T, W_T $\triangleright$ θ_T, W_T initialised to θ, W for $i = 1$
        $S_j^* \leftarrow U_j \cup S_j \quad \forall j \in 1, \ldots, C_n$
        $W^* \leftarrow \{P(S_1^*), \ldots, P(S_{C_n}^*)\}$
    Implicit refinement using CR
    Randomly re-initialise teacher parameters θ_T
    For r = 1 to R:
        Sample a batch of unlabelled samples $\mathcal{U}_s$ from $\mathcal{U}$
        $\bar{u} \leftarrow WeakAugment(u)$, $\hat{u} \leftarrow StrongAugment(u), u \in \mathcal{U}_s$
        $V_j \leftarrow$top-$K(f^j_{t,\theta_t,W*}(\bar{u})) \quad \forall j \in 1, \ldots, C_n$
        $W^{**}, \theta^* \leftarrow \arg\min_{W,\theta} \mathcal{L}_{ce}(\mathbf{x}) + \lambda_{ft}\mathcal{L}_{cons}(v_b^u), \quad \mathbf{x} \in \mathcal{S}, v^u \in V = \{V_1, \cdots, V_{C_n}\}$
        Update teacher parameters $W_T \leftarrow (1-\alpha)W_T + \alpha W^{**}, \quad \theta_T \leftarrow (1-\alpha)\theta_T + \alpha\theta^*$
end

Algorithm 2 Prototype Driven Consistency Regularization

Input: Labelled examples and their one-hot labels $\mathcal{X} = \{(x_b, y_b) : b \in 1, \ldots, B\}$, Unlabelled examples $\mathcal{U} = \{(u_b) : b \in 1, \ldots, B\}$, weighting parameters λ, α.
Output: Optimised student model parameters θ^*, W^*
Randomly initialise Student and Teacher model parameters and prototypes: θ,θ_T,W,W_T
While not done **do**
    Sample batch of labelled $\mathcal{X}_b$ and unlabelled samples $\mathcal{U}_b$ from $\mathcal{X}$,$\mathcal{U}$
    for all $(x_b, u_b) \in (\mathcal{X}_b, \mathcal{U}_b)$ **do**
        $\hat{x}_b = StrongAugment(x_b)$
        $\bar{u}_b = WeakAugment(u_b)$
        $\hat{u}_b = StrongAugment(u_b)$
        $q_b^l \leftarrow f_{s,\theta,W}(\hat{x}_b)$, $f_{s,\theta,W}(\hat{x}_b)$ computed as in Eq (1) in main-manuscript with student parameters θ, W
        $q_b^u \leftarrow f_{t,\theta_T,W_T}(\bar{u}_b)$, $\hat{q}_b^u = f_{s,\theta,W}(\hat{u}_b)$
        $\mathcal{L}(x_b, u_b) = \mathcal{L}_{ce}(q_b^l) + \lambda||\text{Sharp}(q_b^u, \mathcal{T}) - \hat{q}_b^u||^2$ as in Eq (2) in main-manuscript
    $W^*, \theta^* \leftarrow \arg\min_{W,\theta} \sum_{\mathcal{X}_b,\mathcal{U}_b} \mathcal{L}(x_b, u_b)$
    Update teacher parameters $W_T \leftarrow (1-\alpha)W_T + \alpha W^*, \quad \theta_T \leftarrow (1-\alpha)\theta_T + \alpha\theta^*$
end