Expert Systems With Applications 238 (2024) 122173

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

SSL-ProtoNet: Self-supervised Learning Prototypical Networks for few-shot learning

Jit Yan Lim, Kian Ming Lim *, Chin Poo Lee, Yong Xuan Tan

Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450, Melaka, Malaysia

ARTICLE INFO

Keywords:
Few-shot classification
Self-supervised learning
Prototypical networks
Few-shot learning
Knowledge distillation

ABSTRACT

Few-shot learning is seeking to generalize well to unseen tasks with insufficient labeled samples. Existing works have achieved generalization by exploring inter-class discrimination. However, their performance is limited because sample discrimination is neglected. In this work, we propose a metric-based few-shot approach that leverages self-supervised learning, Prototypical networks, and knowledge distillation, referred to as SSL-ProtoNet, to utilize sample discrimination. The proposed SSL-ProtoNet consists of three stages: pre-training stage, fine-tuning stage, and self-distillation stage. In the pre-training stage, self-supervised learning is leveraged to cluster the samples with their augmented variants to enhance the sample discrimination. The learned representation is then served as an initial point for the next stage. In the fine-tuning stage, the model weights transferred from the pre-training stage are fine-tuned to the target few-shot tasks. A self-supervised loss and a few-shot loss are integrated to prevent overfitting during few-shot task adaptation and to maintain the embedding diversity. In the self-distillation stage, the model is arranged in a teacher–student architecture. The teacher model will serve as a guidance in student model training to reduce overfitting and further improve the performance. The experimental results show that the proposed SSL-ProtoNet outshines the state-of-the-art few-shot image classification methods on three benchmark few-shot datasets, namely, *mini*ImageNet, *tiered*ImageNet, and CIFAR-FS. The source code for the proposed method is available at https://github.com/Jityan/sslprotonet.

1. Introduction

Generally, deep learning algorithms require a large amount of samples to obtain optimal performance. However, it is painful to obtain a large amount of samples for most real world problems. This scenario is referred to as few-shot learning (FSL) problems where there are only a few samples available for each category. In FSL, all tasks are divided into disjoint training and testing sets. Each task consists of a limited number of samples which are categorized into support samples and corresponding query samples. The objective of FSL is to classify the query sample to the correct support sample. The few-shot tasks consist of a N number of classes where each class contains a K number of samples, which is denoted as N-way K-shot. For instance, 5-way 1-shot refers to 5 classes where each class contains 1 support sample. One of the challenges in most machine learning methods is that they are not able to learn a well generalized embedding with only a small amount of samples.

Most of the existing works are proposed to learn a generic model representation across different tasks via meta-learning or metric learning. Meta-learning approaches (Finn et al., 2017; Raghu et al., 2020) are popular in solving few-shot tasks. The idea of meta-learning or learning to learn (Antoniou et al., 2019; Finn et al., 2017; Rusu et al., 2019; Schmidhuber, 1987; Thrun & Pratt, 1998; Vilalta & Drissi, 2005) is to mimic the human learning capability by learning a new task with a small number of samples and integrating it with prior knowledge. Recent works (Dhillon et al., 2020; Tian et al., 2020) focus on learning a good representation to solve few-shot tasks. In metric learning approaches (Allen et al., 2019; Koch et al., 2015; Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016), center point representation (Allen et al., 2019; Snell et al., 2017; Sung et al., 2018) has proven its effectiveness in comparing the query sample to multiple support samples.

Although the prior works managed to achieve few-shot generalization by seeking inter-class discrimination, the learned features were limited in adapting to a wide range of instances. They only focused on constructing the boundaries between each of the classes based on a rough observation of the class samples. The sample discrimination between each instance is neglected. This motivates this work to

* Corresponding author.
E-mail addresses: 1141124378@student.mmu.edu.my (J.Y. Lim), kmlim@mmu.edu.my (K.M. Lim), cplee@mmu.edu.my (C.P. Lee), 1141124379@student.mmu.edu.my (Y.X. Tan).

https://doi.org/10.1016/j.eswa.2023.122173
Received 2 January 2023; Received in revised form 26 June 2023; Accepted 13 October 2023
Available online 19 October 2023

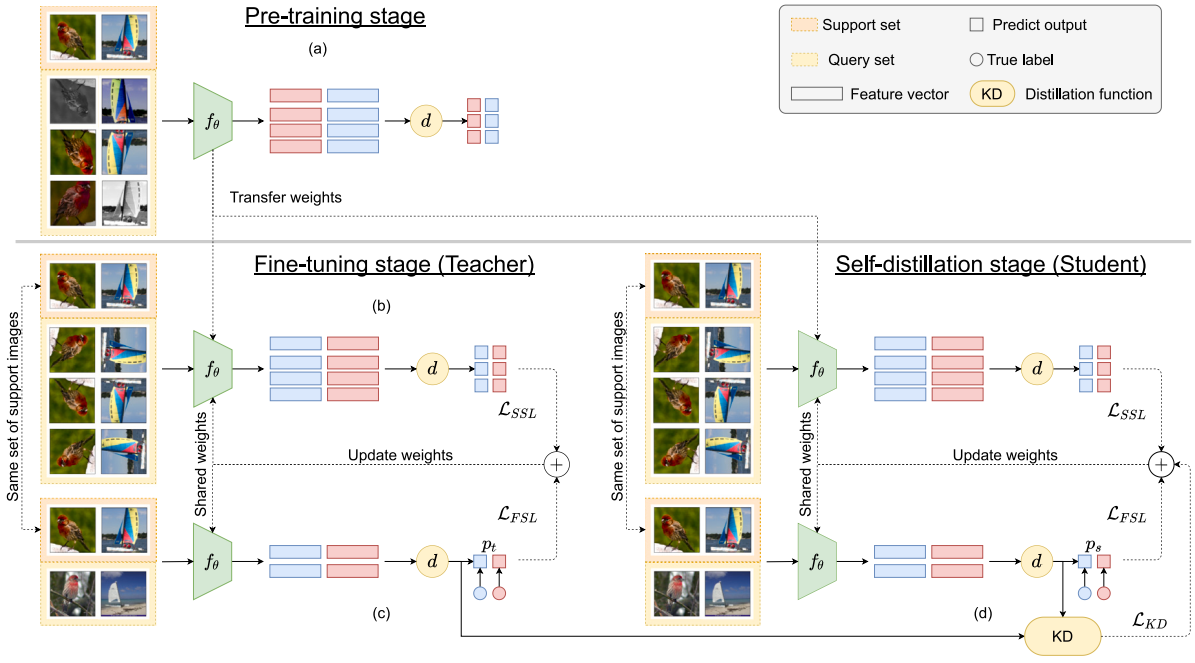

Fig. 1. The overall architecture of the proposed SSL-ProtoNet in a 2-way 1-shot paradigm. The model embedding backbone is denoted by f_θ and Euclidean distance function is denoted by d. $\mathcal{L}_{SSL}$ and $\mathcal{L}_{FSL}$ denote the self-supervised loss signal and few-shot loss in the fine-tuning stage. p_t and p_s denote the predicted logits/output from the teacher model and the student model. $\mathcal{L}_{KD}$ denotes the distillation loss in the self-distillation stage.

obtain a generalized representation via enhancing sample discrimination by using self-supervised learning (SSL) instead of only focusing on inter-class discrimination. SSL is managed to enhance the sample discrimination in the feature space (Chen et al., 2020; Chopra et al., 2005). In recent years, SSL has been introduced into FSL and it has achieved outstanding performance. Prior works (Gidaris et al., 2019; Rajasegaran et al., 2020) proved that SSL is suitable for the situation with limited training samples and able to effectively learn a good representation for each class. Inspired by these works, this paper proposed a novel metric-based few-shot approach, named as Self-Supervised Learning Prototypical Networks (SSL-ProtoNet) for few-shot image classification. The SSL-ProtoNet is separated into 3 stages: pre-training, fine-tuning, and self-distillation. The overall architecture of the proposed SSL-ProtoNet is presented in Fig. 1.

In Fig. 1(a), the pre-training stage demonstrates f_θ is trained on a large number of unlabeled images. Each image is augmented into another three images, which serve as query samples. After completing the pre-training stage, the learned weights are transferred into the fine-tuning stage. The fine-tuning stage consists of a self-supervised process (Fig. 1(b)) and a few-shot process (Fig. 1(c)). The self-supervised process aims to produce self-supervised loss signal $\mathcal{L}_{SSL}$, while the few-shot process aims to produce few-shot loss $\mathcal{L}_{FSL}$. The self-supervised process utilizes the same support samples with the few-shot process. The support samples are rotated to produce the query samples that are only used in the self-supervised process. They compute $\mathcal{L}_{SSL}$ by minimizing the distance between original sample and rotated sample. The few-shot process aims to minimize the distance between the support sample and query samples from the same classes to produce $\mathcal{L}_{FSL}$ loss. Finally, both losses are combined and used to fine-tune f_θ.

Fig. 1(d) illustrates the self-distillation stage of the proposed SSL-ProtoNet. In the self-distillation stage, the student model is initialized based on the learned weights from the pre-training stage, thereby, the student model is able to contain a preliminary level of knowledge instead of being trained from scratch (Tian et al., 2020). After that, the model in the fine-tuning stage serves as a teacher model to guide the training of the student model by providing predicted output as the soft target. The student model utilized the soft target to compute $\mathcal{L}_{KD}$ and update f_θ during the training. The soft target acts as a dark magic to further improve the student model's performance instead of only relying on the hard target (truth label). The self-distillation process provides the model an extra performance boost by reducing the model overfitting and increasing task generalization (Lim et al., 2021; Tian et al., 2020).

In the pre-training stage, the input images serve as the support samples and its augmented variants serve as the query samples. The proposed SSL-ProtoNet aims to cluster the augmented images around the original image and minimizes its pairwise distance loss, as depicted in Fig. 2(a). The distance loss (self-supervised loss) can be viewed as a self-supervised version of prototypical loss in ProtoNet (Snell et al., 2017). To ensure the model to learn useful representation, noisy transformation is proposed to create the augmented variants, instead of simple transformation in Gidaris et al. (2019) and Rajasegaran et al. (2020) (details in Section 4.8). By doing so, the distance loss not only enhances the discriminability between different samples but also gives prominence to the essential features.

Fine-tuning stage consists of two processes: self-supervised and few-shot. In the fine-tuning stage, the learned weights from the pre-training stage are further fine-tuned to adapt to the few-shot tasks. However, directly fine-tuning the pre-trained weights on the few-shot tasks will cause overfitting and reduce the representation diversity which resulted in performance degradation. To ensure the pre-trained model can well adapt to the few-shot tasks, a self-supervised loss signal is proposed in the self-supervised process during the model weight update, as shown in Fig. 2(b). The self-supervised process in the fine-tuning stage is similar to the pre-training stage, as both aim to minimize the distance between the original sample and the augmented sample. The purpose of using a similar SSL approach is to diversify the model representation through sample discrimination, thus achieving better generalization. By doing so, it is useful in exploring the preliminary level of generalization during pre-training, and also helps prevent overfitting of the model during the fine-tuning stage. In the self-supervised process, each support sample in the few-shot tasks is treated as a single class. Given the support sample, 3 rotated samples are generated as their associated query samples in the self-supervised process. The self-supervised loss signal is computed based on the distance of the support sample to its rotated variants. The few-shot process is the task adaptation process

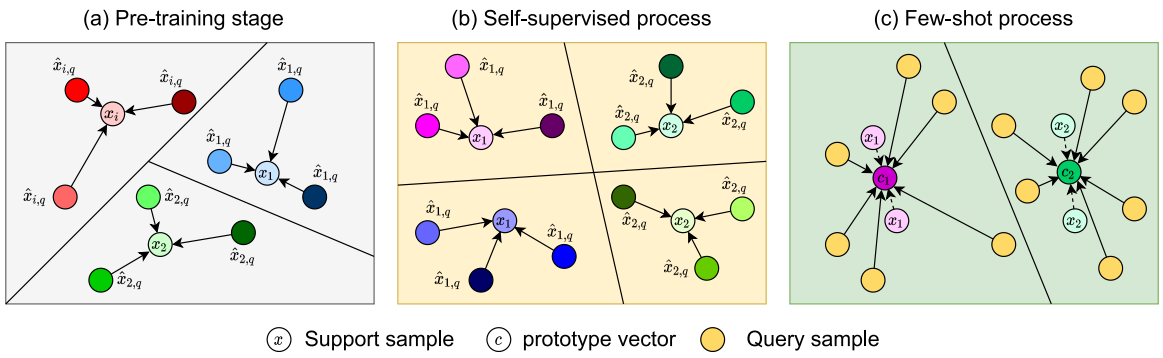

Fig. 2. The goal of the proposed SSL-ProtoNet is different in each stage (a,b,c). (a) Pre-training stage: this is an i-way 1-shot paradigm. The goal is to cluster the augmented query samples $\hat{x}_{i,q}$ to the target support sample x_i. (b) and (c) are in the fine-tuning stage under the 2-way 2-shot paradigm. (b) Self-supervised process: the 2-way 2-shot problem is converted into 4-way 1-shot problem where each support sample is treated as a single class. In (b), the self-supervised process aims to cluster the rotated query samples around their own support sample. (c) Few-shot process: this process computes the prototype vector for each class and clusters their 6 query samples together.

which has the same objective function as ProtoNet. It aims to cluster the query samples around the corresponding support samples in the few-shot tasks to form the few-shot loss, as presented in Fig. 2(c). The self-supervised loss is then integrated with the few-shot loss to fine-tune the model.

In contrast to the existing works (Gidaris et al., 2019; Rajasegaran et al., 2020) that used linear classifiers, this paper performs SSL by bringing the support sample closer to its augmented samples using Euclidean distance. The proposed SSL-ProtoNet aims to obtain center points that well represent each class by learning more diverse model representation to achieve better generalization via SSL. The primary distinction between SSL-ProtoNet and ProtoNet is the inclusion of SSL. To date, utilizing SSL during fine-tuning is rarely seen in the existing works. With the proposed SSL during the fine-tuning stage, it can ensure the model better adapts to the few-shot tasks. Although SSL-ProtoNet employs the same distance function as ProtoNet, it could theoretically use other distance functions in place of Euclidean distance. Additionally, ProtoNet requires class labels during training, whereas SSL-ProtoNet only requires them during the few-shot process in the fine-tuning stage and self-distillation stage. The main contributions of this paper are as follows:

- Introduce SSL in the pre-training stage to obtain preliminary generalized weights via enhancing sample discrimination with noisy transformation.
- A self-supervised loss signal is proposed in the fine-tuning stage to preserve the diversity and generalization of the embedding representation and stabilize the adaptation process.
- A self-distillation process is introduced into the proposed model to reduce overfitting and improve the model performance.
- The proposed SSL-ProtoNet leverages SSL strategies to obtain a better performance as compared to the state-of-the-art methods on three benchmark few-shot image classification datasets.

2. Related work

This section presents the related works of SSL and FSL.

2.1. Self-supervised learning

Self-supervised learning (SSL) enhances the model learning capability without extra data labeling and hence it is widely used in various domains. Dosovitskiy et al. (2015) performed SSL by sampling different patches of images from the same classes and applied random transformation to the patches. In Gidaris et al. (2018), the model was trained to predict rotation degree of the images to learn useful representation. Doersch et al. (2015) proposed to learn the object spatial context by predicting the relative position of two random patches from a single image. Noroozi and Favaro (2016) extended the idea from Doersch et al. (2015) by utilizing all patches from a single image. Later, object counting and image colorization were implemented to generate meaningful representation (Noroozi et al., 2017; Zhang et al., 2016). Chen et al. (2020) proposed to learn useful visual representation by utilizing heavy data augmentation.

In this work, a noisy transformation is proposed to create the self-supervised tasks in the pre-training stage in order to optimize the learned representation. The model generates the query sample from each support sample via a set of random transformations and further rotated with different degrees. With the rich transformations, the embedding model in the proposed SSL-ProtoNet is able to learn a good representation and achieve better generalization. In the fine-tuning stage, the rotation transformation is leveraged to maintain the model representation diversity by reducing the distance between the support sample and its rotated variants. In doing so, the distance between the samples within the same classes is minimized.

2.2. Few-shot learning

Few-shot learning (FSL) aims to classify unseen tasks effectively with only a few samples and has been widely used in several domains (Wang et al., 2022; Xu & Xiang, 2021; Zhou et al., 2021). In Koch et al. (2015), a Siamese network was used to learn the similarity between images from an image pair. Vinyals et al. (2016) utilized two different models to embed support samples and query samples and calculate their similarity using cosine distance. Snell et al. (2017) represented every class with a mean point and classified query samples based on the Euclidean distance. Similarly, Sung et al. (2018) utilized the centroid representation (Snell et al., 2017) and used convolutional neural network (CNN) for similarity learning. Oreshkin et al. (2018) further improved (Snell et al., 2017) by introducing an extra task embedding network for a wide range of task adaptation. Different from Snell et al. (2017), Sung et al. (2018) and Allen et al. (2019) used a set of points to represent a single class.

MAML (Finn et al., 2017) used a meta-learner to learn the initial parameters of the base-learner for faster adaptation to new tasks. To achieve simplicity, Nichol et al. (2018) removed the second derivative and reduced the computation time of MAML. Later, Li et al. (2017) improved MAML by predicting the update direction and learning rate of the algorithm. Likewise, Antoniou et al. (2019) solved the overfitting and generalization problem of MAML by introducing a learnable learning rate and multi-step loss. Lee et al. (2019) replaced the linear classifier of MAML with a support vector machine (SVM) to improve performance. Later on, Finn et al. (2018) utilized probability to handle ambiguous tasks. Sun et al. (2020) integrated transfer learning into meta-learning to handle new tasks. Raghu et al. (2020) pointed out that the effectiveness of meta-learning was contributed by the strong feature

representation instead of the meta-learning itself. Besides, Rusu et al. (2019) showed that separating the high dimensional model parameters from the meta-learning algorithm can improve performance

Prior works (Dhillon et al., 2020; Tian et al., 2020) proved that learning a strong embedding representation is essential for solving few-shot tasks effectively. Chen et al. (2019) proved that training the model on the base class and fine-tuning it on the novel class outperformed most of the conventional FSL methods with shallow model backbone. Instead of learning only task-agnostic features, Ye et al. (2020) explored task-specific features via a transformer to achieve optimal performance. Lim et al. (2021) transferred knowledge from a pre-trained large scale model and fine-tuned it for the target few-shot task. Inspired by the recent effectiveness of SSL, Gidaris et al. (2019) and Lim et al. (2023) improved the embedding representation by integrating it with different self-supervised tasks. In Su et al. (2020), SSL was utilized to pre-train a deep model backbone and transferred the knowledge for few-shot adaptation to obtain optimal performance. To improve cross-domain performance, Medina et al. (2020) proposed using SSL in clustering to improve model generalization before beginning FSL. Lee et al. (2020) utilized SSL to learn the joint distribution with few-shot tasks. An et al. (2021) proposed a three-stage training paradigm to investigate SSL with FSL to explore optimal performance.

Unlike Gidaris et al. (2019) that handled self-supervised tasks together with few-shot tasks using different linear layers, the proposed SSL-ProtoNet simulates contrastive learning where the training is completely self-supervised in the pre-training stage. In doing so, the feature of each sample is more discriminable. In addition, it will be more effective for the learning in the fine-tuning stage. In contrast to Oreshkin et al. (2018) and Allen et al. (2019), this paper aims to obtain better generalization than ProtoNet (Snell et al., 2017) by leveraging SSL. ProtoNet obtained a prototype vector that was enriched with the features from the samples within the same class. The proposed SSL-ProtoNet leverages self-supervised tasks to obtain a well generalized prototype vector that encodes the features of the samples in the same class. In the pre-training stage, the SSL leverages the noisy transformation to generate the query samples. The main objective of noisy transformation is to form a primary stage of generalization via maximizing the sample discrimination. This helps the model to produce prototype vectors that are capable of generalizing to the query samples to increase the discriminating power. Subsequently, the fine-tuning stage exploits SSL to provide a secondary stage of generalization to prevent over-adaptation to the few-shot tasks. As a result, the enriched model representation improves the prototype vectors to learn all possible samples within the same classes.

3. SSL-ProtoNet

In this work, the training set is denoted as D. The support set is represented by $S = \{(x_1, y_1), \ldots, (x_I, y_I)\}$. The input image is denoted by x and the corresponding label is denoted by y where $y \in \{1, \ldots, N\}$. I represents the number of the labeled data. A set of labeled support samples for class k is denoted as S_k and the corresponding query set is denoted as Q_k. The number of samples from D in each mini batch b is denoted as M.

3.1. Pre-training stage

The goal of incorporating SSL in the pre-training is to learn the representative features that encode the prominent structural and semantic information for few-shot tasks. For each mini batch b in the pre-training stage, M number of samples are randomly sampled from the training set D. The mini batch is treated as a M-way 1-shot task. As the pre-training stage is fully self-supervised, it has no access to any label information. Thus, each sample x_i serves as a 1-shot support sample and at the same time as a prototype vector. All support samples x_i are randomly augmented to form the corresponding query samples $\hat{x}_{i,q}$ via the proposed noisy transformation $T + R$. In this paper, noisy transformation is introduced to generate query samples from the combination of random transformation set T (cropping, flipping, and coloring) and rotation R with $90°, 180°, 270°$ degrees to yield the greatest discriminability (details in Section 4.8). With the proposed noisy transformation, the embedding model f_θ manages to learn high-quality and robust representation to enhance the sample discrimination and generalize well for the downstream tasks. The model learns to minimize the distance from the augmented samples toward its support sample (self-supervised loss). The distance between the support sample x_i and the query sample $\hat{x}_{i,q}$ is denoted as d. Specifically, d is the squared Euclidean distance:

$$d\left[f_\theta\left(\hat{x}_{i,q}\right), f_\theta\left(x_i\right)\right] = \left\| f_\theta\left(\hat{x}_{i,q}\right) - f_\theta\left(x_i\right)\right\|^2 \tag{1}$$

The distribution of the augmented query samples $\hat{x}_{i,q}$ is computed using a softmax function over the distances d to its prototype vector (support sample) $f_\theta\left(x_k\right)$ in the embedding space. The softmax function is defined as:

$$\lambda(i,q) = -\log \frac{\exp\left(-d\left[f_\theta\left(\hat{x}_{i,q}\right), f_\theta\left(x_i\right)\right]\right)}{\sum_{k=1}^{M} \exp\left(-d\left[f_\theta\left(\hat{x}_{i,q}\right), f_\theta\left(x_k\right)\right]\right)} \tag{2}$$

The model prediction process can be viewed as a linear model by expanding the exponential term:

$$-\|q - c\|^2 = -q^\top q + 2c^\top q - c^\top c \tag{3}$$

where $q = f_\theta\left(\hat{x}_{i,q}\right)$ and $c = f_\theta\left(x_i\right)$. The pre-training stage serves as a preparation for the actual FSL of the target few-shot tasks in the next stage. The pre-trained weights obtained from this stage are transferred to the fine-tuning stage for better adaptation to the few-shot tasks.

The training procedure of the pre-training stage is presented in Algorithm 1. Each sample x_i in the mini batch b is utilized to create its own augmented query samples $\hat{x}_{i,q}$ (lines 4–10). A random transformation t is applied to transform the query samples. The transformed query samples are then rotated with a degree r (lines 6–9). In lines 12–13, the pre-training loss $\mathcal{L}$ is computed by clustering the augmented query samples $\hat{x}_{i,q}$ to its own support sample x_i in the embedding space using Euclidean distance d. The cross-entropy loss over M classes is minimized with respect to the embedding parameters θ in the embedding model f_θ.

Algorithm 1 The training procedure of the pre-training stage for the proposed SSL-ProtoNet.

Input: Mini batch b from the training dataset D, set of random transformation T, rotation degrees R, embedding backbone f_θ, learning rate α

1: randomly initialize θ
2: **for** $b \sim D$ **do**
3: $x_i \leftarrow b$
4: **for** all $i \in \{1, \ldots, M\}$ **do**
5: **for** all $q \in \{1, \ldots, 3\}$ **do**
6: draw a random transformation $t \sim T$
7: $\hat{x}_{i,q} \leftarrow t\left(x_i\right)$
8: select a rotation degree $r \sim R(q)$
9: $\hat{x}_{i,q} \leftarrow r\left(\hat{x}_{i,q}\right)$
10: **end for**
11: **end for**
12: $\mathcal{L} \leftarrow \frac{1}{3M} \sum_{i=1}^{M} \sum_{q=1}^{3} \lambda(i,q)$
13: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
14: **end for**
15: return f_θ

3.2. Fine-tuning stage

The fine-tuning stage utilizes the pre-trained weights transferred from the pre-training stage to better adapt to the target few-shot tasks.

The fine-tuning stage comprises self-supervised and few-shot processes. In general, few-shot tasks are organized in the N-way K-shot structure with N support samples from K classes. In the self-supervised process, the N-way K-shot tasks are converted into $N \times K$-way 1-shot tasks where each support sample (1-shot) is treated as a single class (1-way). This conversion is only applied in the self-supervised process since it has no access to any label information. Each support sample is rotated with $\{90°, 180°, 270°\}$ to generate 3 corresponding query samples with different rotation degrees for SSL. The model learns to cluster the rotated samples closer to their original sample by minimizing the distance loss. In this way, the diversified samples generated from the rotation in the SSL maintain the diversity of feature representation via the self-supervised loss signal $\mathcal{L}_{SSL}$. Furthermore, it helps to prevent overfitting in the few-shot tasks adaptation. The $\mathcal{L}_{SSL}$ is formulated as:

$$\mathcal{L}_{SSL} = \frac{1}{3 \times |S_k|} \sum_{i=1}^{|S_k|} \sum_{q=1}^{3} \lambda(i, q) \tag{4}$$

where $|S_k|$ denotes the number of support samples.

In the few-shot process, the model adjusts the prototype vectors output from the support samples to get closer to their query samples obtained from the query set Q_k. The prototype vector c_k is computed as the mean of the support samples from the same class k via:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\theta(x_i) \tag{5}$$

The few-shot loss $\mathcal{L}_{FSL}$ is computed as:

$$\mathcal{L}_{FSL} = \frac{1}{|S_k Q_k|} d\left[f_\theta(x_i), c_k\right] + \log \sum_{k'} \exp\left(-d\left[f_\theta(x_i), c_{k'}\right]\right) \tag{6}$$

$\mathcal{L}_{FSL}$ enables the model to utilize the class label to well adapt the preliminary generalized representation toward the few-shot instances.

The final loss from the fine-tuning stage is computed as:

$$\mathcal{L} = \beta \cdot \mathcal{L}_{SSL} + \mathcal{L}_{FSL} \tag{7}$$

where β denotes the coefficient of $\mathcal{L}_{SSL}$ that is used to adjust $\mathcal{L}_{SSL}$'s influence on the final loss. The coefficient β is set within the range of $[0, 1]$.

After these two stages, the proposed SSL-ProtoNet manages to adapt to the target few-shot tasks. In addition, the proposed SSL-ProtoNet is also able to maintain the diversity of the model representation. The generated prototype vectors are closer to the target query samples thus boost the performance of the proposed SSL-ProtoNet. The training procedure of the fine-tuning stage is presented in Algorithm 2.

From Algorithm 2, lines 5–12 compute the self-supervised loss for each sample x_i with its rotated samples $\hat{x}_{i,q}$. The N-way K-shot samples are re-organized into $N \times K$-way 1-shot samples. Each sample is rotated with the selected rotation degree from $R \in \{90°, 180°, 270°\}$ (lines 8–9). Lines 14–18 describe the few-shot process in this stage. The prototype vectors c_k are produced for each class (N-way) based on the number of support samples (K-shot). Each query sample $x_i \in Q_k$ is then compared with the corresponding prototype vector c_k to compute the few-shot loss $\mathcal{L}_{FSL}$ using Euclidean distance $\|x_i - c_k\|^2$ (line 17–18). The final loss $\mathcal{L}$ combines both $\mathcal{L}_{SSL}$ and $\mathcal{L}_{FSL}$ and it is used to update the model (line 20–21).

3.3. Self-distillation stage

In the self-distillation stage, the model obtained from the fine-tuning stage is served as a teacher model. A student model is generated with the same architecture as the teacher model. Similar to the teacher model, the student model is initialized with the pre-trained weights from the pre-training stage for faster adaptation. The process inside the student model is the same as the teacher model but with an extra distillation process. The distillation process is computed as below:

$$\mathcal{L}_{KD} = KL\left[\sigma\left(\frac{p_s}{\tau}\right), \sigma\left(\frac{p_t}{\tau}\right)\right] \tag{8}$$

Algorithm 2 The training procedure of the fine-tuning stage for the proposed SSL-ProtoNet.

Input: The training dataset D, rotation degrees R, embedding backbone f_θ, Euclidean distance $d[\cdot, \cdot]$, learning rate α

1: receive θ from pre-training stage
2: **for** $b \sim D$ **do**
3: $S_k, Q_k \leftarrow b$
4: *compute self-supervised loss:*
5: $\mathcal{L}_{SSL} \leftarrow 0$
6: **for** x_i in S_k **do**
7: **for** all $q \in \{1, ..., 3\}$ **do**
8: select a rotation degree $r \sim R(q)$
9: $\hat{x}_{i,q} \leftarrow r(x_i)$
10: **end for**
11: $\mathcal{L}_{SSL} \leftarrow \mathcal{L}_{SSL} + \frac{1}{3 \times |S_k|} \sum_{i=1}^{|S_k|} \sum_{q=1}^{3} \lambda(i, q)$
12: **end for**
13: *compute few-shot loss:*
14: $\mathcal{L}_{FSL} \leftarrow 0$
15: $c_k \leftarrow \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\theta(x_i)$
16: **for** (x_i, y_i) in Q_k **do**
17: $d \leftarrow d\left[f_\theta(x_i), c_k\right] + \log \sum_{k'} \exp\left(-d\left[f_\theta(x_i), c_{k'}\right]\right)$
18: $\mathcal{L}_{FSL} \leftarrow \mathcal{L}_{FSL} + \frac{1}{|S_k Q_k|} d$
19: **end for**
20: $\mathcal{L} \leftarrow \mathcal{L}_{FSL} + \beta \cdot \mathcal{L}_{SSL}$
21: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
22: **end for**
23: return f_θ as the teacher model

where KL denotes the Kullback Leibler divergence between the student prediction p_s and soft target p_t from the teacher model. σ denotes the softmax function. The predicted outputs p_t and p_s are softened via a parameter τ, which denotes the temperature. In this work, temperature τ is set to 4 (Hinton et al., 2015) for a softer class distribution. With the distillation loss $\mathcal{L}_{KD}$, the final loss $\mathcal{L}_s$ in the self-distillation stage is computed as below:

$$\mathcal{L}_s = (1 - \gamma) \cdot \mathcal{L}_{FSL} + \gamma \cdot \mathcal{L}_{KD} + \beta \cdot \mathcal{L}_{SSL} \tag{9}$$

where γ is the coefficient of $\mathcal{L}_{KD}$ to control the impact of the teacher's soft target toward $\mathcal{L}_s$. Same as the teacher model, $\mathcal{L}_{FSL}$ and $\mathcal{L}_{SSL}$ are obtained from Eqs. (4) and (6). The teacher model produces predicted output as a soft target to guide the training of the student model for extra performance improvement.

The overall training procedure for the self-distillation stage is presented in Algorithm 3. The processes for computing $\mathcal{L}_{SSL}$ (line 5) and $\mathcal{L}_{FSL}$ (line 10) are the same with Algorithm 2. For computing distillation loss $\mathcal{L}_{KD}$, two prediction outputs p_t and p_s are generated based on the teacher model and student model (line 11–12) with corresponding prototype vectors $c_{k,t}$ and c_k (line 7–8). The outputs are used to compute $\mathcal{L}_{KD}$ (line 13) and update the student model together with other losses (line 15–16).

4. Experiments

The datasets, evaluation protocols, implementation details, and experimental results of the proposed SSL-ProtoNet are all covered in this section.

4.1. Datasets

The performance of the proposed SSL-ProtoNet is assessed using three benchmark few-shot image classification datasets. ***mini*ImageNet** (Vinyals et al., 2016) contains 60,000 RGB images originating from the ImageNet (Deng et al., 2009). It is splitted into 100 classes, each

Algorithm 3 The training procedure for the student model in the self-distillation stage of proposed SSL-ProtoNet.

Input: The training dataset D, rotation degrees R, embedding backbone f_θ, Euclidean distance $d[\cdot,\cdot]$, learning rate α, temperature τ

receive θ from pre-training stage as student model
receive θ from fine-tuning stage as teacher model θ_t
for $b \sim D$ **do**
  $S_k, Q_k \leftarrow b$
  $\mathcal{L}_{SSL} \leftarrow$ *same as Algorithm 2 (line 5-12)*
  $\mathcal{L}_{FSL}, \mathcal{L}_{KD} \leftarrow 0$
  $c_{k,t} \leftarrow \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} f_{\theta_t}(x_i)$
  $c_k \leftarrow \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} f_{\theta}(x_i)$
  for (x_i, y_i) in Q_k **do**
    $\mathcal{L}_{FSL} \leftarrow$ *same as Algorithm 2 (line 17-18)*
    $p_t \leftarrow -d\left[f_{\theta_t}(x_i), c_{k,t}\right]$
    $p_s \leftarrow -d\left[f_{\theta}(x_i), c_k\right]$
    $\mathcal{L}_{KD} \leftarrow KL\left[\sigma\left(\frac{p_s}{\tau}\right), \sigma\left(\frac{p_t}{\tau}\right)\right]$
  end for
  $\mathcal{L}_s \leftarrow (1-\gamma)\cdot\mathcal{L}_{FSL} + \gamma\cdot\mathcal{L}_{KD} + \beta\cdot\mathcal{L}_{SSL}$
  $\theta \leftarrow \theta - \alpha\nabla_\theta\mathcal{L}_s$
end for

of them having 600 images. The train–test split is adopted from Ravi and Larochelle (2017) with 64 training classes, 16 validation classes, and 20 test classes respectively. ***tiered*ImageNet** (Ren et al., 2018) is derived from ImageNet dataset with larger subset. It contains 608 classes with a total of 779,165 images. The dataset is split into 351, 97, and 160 classes for training, validation, and testing, respectively. **CIFAR100 few-shots (CIFAR-FS)** (Bertinetto et al., 2018) is constructed from CIFAR100 (Krizhevsky, 2009) dataset with the same train–test split criteria as *mini*ImageNet. The images from *mini*ImageNet and *tiered*ImageNet are resized into 84 × 84 pixels and the images from CIFAR-FS are resized into 32 × 32 pixels.

4.2. Evaluation protocols

In this paper, the common 5-way 1-shot and 5-way 5-shot settings are used to evaluate the performance of the proposed SSL-ProtoNet. In each task, 5 test classes (5-way) are randomly selected, each of them containing 1 or 5 support samples (K-shot) along with 15 query samples. To evaluate the performance, 2000 unseen tasks are randomly constructed from the test classes. Then, with 95% confidence intervals, the average 5-way K-shot classification accuracies are reported.

4.3. Implementation details

The ConvNet embedding backbone of the proposed SSL-ProtoNet is following Snell et al. (2017) and Vinyals et al. (2016). Adaptive Moment Estimation (Adam) optimizer is used to train the proposed SSL-ProtoNet with 20,000 mini-batches b in the pre-training stage and 20,000 tasks in both fine-tuning and self-distillation stages. The initial learning rate α is set to 0.001 and decayed into half for every 2000 tasks with 0.001 L2 penalty to prevent the model from overfitting. These hyperparameter settings are applied to all datasets for performance consistency except *tiered*ImageNet is trained with 50,000 tasks (mini-batch b) and for every 5000 tasks, α is reduced in all stages. The details of coefficients β and γ for self-supervised loss and distillation loss can be found in Sections 4.5 and 4.10. The hyperparameters settings of the proposed SSL-ProtoNet and other few-shot approaches can be found in Table 1.

4.4. Discussion of results

The 5-way 1-shot and 5-way 5-shot experiments are carried out on three few-shot datasets. For a fair comparison, ConvNet is adopted as the embedding backbone for all methods. The performance comparison of all datasets are reported in Table 2. It is notable that the SSL in each stage and knowledge distillation improve the learned representation in the embedding backbone of the proposed SSL-ProtoNet. In the 1-shot setting, the proposed SSL-ProtoNet outperforms the state-of-the-art methods with the highest accuracy of 52.25 ± 0.45, 55.07 ± 0.49, and 59.18 ± 0.50 on *mini*ImageNet, *tiered*ImageNet, and CIFAR-FS, respectively. In the 5-shot setting, the proposed SSL-ProtoNet outshines the methods in comparison with the highest accuracy of 70.60 ± 0.36, 74.02 ± 0.40, and 76.05 ± 0.38 on *mini*ImageNet, *tiered*ImageNet, and CIFAR-FS, respectively. Before the self-distillation stage (w/o KD), the proposed SSL-ProtoNet has outperformed the highest method in comparison for approximately 1% in the 1-shot setting and 1%–2% in the 5-shot setting. Noticeably, the proposed SSL-ProtoNet achieves more improvements in 5-shot settings and this demonstrates that the proposed SSL-ProtoNet works better when it receives more samples. After integrated with knowledge distillation (self-distillation stage), the performance of SSL-ProtoNet gains further improvement for around 0.07%–1.23% on 1-shot setting and 0.21%–0.47% on 5-shot setting across all datasets.

The performance of the proposed SSL-ProtoNet is further compared with recent few-shot approaches involving SSL (An et al., 2021; Gidaris et al., 2019; Lee et al., 2020; Medina et al., 2020; Su et al., 2020). On the *mini*ImageNet, the proposed SSL-ProtoNet achieves around 2% improvement in both 1-shot and 5-shot settings as compared to CSS (An et al., 2021). The improvements can also be observed in the *tiered*ImageNet and CIFAR-FS datasets. A higher performance of more than 8% and more than 2% in accuracy has been recorded on *tiered*ImageNet and CIFAR-FS datasets, respectively. The results have shown that SSL with knowledge distillation plays an important role in enhancing the model performance of the few-shot classification tasks. The improvements in the performance corroborate the effectiveness of SSL to form a diverse representation in the embedding model for better generalization capability. In the pre-training stage, SSL is introduced in the proposed SSL-ProtoNet to minimize the distance between support image and augmented query images. This has provided a primary stage of generalization to the model to adapt to a wide range of instances with the enhanced sample discrimination. In the fine-tuning stage, the self-supervised task is introduced as another objective for the model to maintain the representation diversity. This enables the smoothness of the adaptation process thus improving the model performance. In the self-distillation stage, the model representation is further distilled by knowledge distillation to obtain a more generic model.

4.5. Ablation study

An ablation study is conducted to analyze several components of the proposed SSL-ProtoNet to determine its effectiveness. Table 3 presents the results of the ablation study with different combinations of the components for 5-way 1-shot and 5-way 5-shot classification tasks on all datasets. The components are as below: the pre-trained weights θ_{T+R} from the pre-training stage, the few-shot loss $\mathcal{L}_{FSL}$, the self-supervised loss signal $\mathcal{L}_{SSL}$ from the fine-tuning stage, and the distillation loss $\mathcal{L}_{KD}$ from the self-distillation stage. From Table 3, it shows that the performance of the model is improved when receiving the pre-trained weights from the pre-training stage or the self-supervised loss signal $\mathcal{L}_{SSL}$ in the fine-tuning stage. Moreover, the performance of the proposed model is further improved after incorporating both components together with $\mathcal{L}_{KD}$.

The baseline model only contains the few-shot loss $\mathcal{L}_{FSL}$. When the model is initialized with the pre-trained weights θ_{T+R} derived from the pre-training stage, it achieves 1.22% improvement on 5-way 1-shot and

Table 1
The hyperparameters settings of the existing few-shot approaches on *mini*ImageNet.

Method	Classifier	Episode/Iteration	Batch size	Optimizer	Learning rate
MatchingNet (Vinyals et al., 2016)	Cosine	50 000	10	Adam	0.001
MAML (Finn et al., 2017)	Linear	60 000	4	Adam	0.01
ProtoNet (Snell et al., 2017)	Euclidean	20 000	–	Adam	0.001
RelationNet (Sung et al., 2018)	Linear	500 000	–	Adam	0.001
Reptile[a] (Nichol et al., 2018)	Linear	100 000, 8	5,10	Adam	0.001
ANIL (Raghu et al., 2020)	Linear	30 000	4	Adam	0.01
IMP (Allen et al., 2019)	Euclidean	100 000	–	RMSprop	0.001
Baseline++ (Chen et al., 2019)	Cosine	400	16	Adam	0.001
FEAT (Ye et al., 2020)	Euclidean	20 000	–	Adam	0.0001
ProtoTransfer[b] (Medina et al., 2020)	Euclidean	1 000 000, 15	50,100	Adam	0.001
CC/PN+rot (Gidaris et al., 2019)	Cosine & Euclidean	60	128	SGD	0.1
SLA (Lee et al., 2020)	Linear	80 000	128	SGD	0.1
ProtoNet+jig/rot (Su et al., 2020)	Euclidean	600	16	Adam	0.001
CSS[c] (An et al., 2021)	Cosine	400	16	Adam	0.001, 0.000001
SSL-ProtoNet (Ours)	Euclidean	20 000	–	Adam	0.001

[a] Denotes the inner loop and outer loop used different parameters.

[b] Denotes the number of iteration and batch size are different in different stages.

[c] Denotes use 0.000001 learning rate for projector head during pre-training stage.

Table 2
Performance comparison of the existing FSL methods on all three datasets.

Method	*mini*ImageNet, 5-way		*tiered*ImageNet, 5-way		CIFAR-FS, 5-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet (Vinyals et al., 2016)	47.06 ± 0.73	62.75 ± 0.75	–	–	51.32 ± 0.85	68.93 ± 0.74
MAML (Finn et al., 2017)	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75	58.90 ± 1.90	71.50 ± 1.00
ProtoNet (Snell et al., 2017)	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74	55.50 ± 0.70	72.00 ± 0.60
RelationNet (Sung et al., 2018)	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.32 ± 0.78	55.00 ± 1.00	69.30 ± 0.80
Reptile (Nichol et al., 2018)	49.97 ± 0.32	65.99 ± 0.58	48.97 ± 0.21	66.47 ± 0.21	–	–
ANIL (Raghu et al., 2020)	46.7 ± 0.4	61.5 ± 0.5	46.63	63.96	–	–
IMP (Allen et al., 2019)	49.6 ± 0.8	68.1 ± 0.8	–	–	–	–
Baseline (Chen et al., 2019)	42.11 ± 0.71	62.53 ± 0.69	–	–	47.01 ± 0.78	68.43 ± 0.73
Baseline++ (Chen et al., 2019)	48.24 ± 0.75	66.43 ± 0.63	–	–	51.55 ± 0.80	72.85 ± 0.72
FEAT (Ye et al., 2020)	47.44 ± 0.72	67.07 ± 0.71	35.48 ± 0.18	44.92 ± 0.18	50.69 ± 0.86	69.39 ± 0.77
ProtoTransfer (Medina et al., 2020)	38.20 ± 0.73	63.47 ± 0.68	–	–	42.46 ± 0.77	67.95 ± 0.76
CC+rot (Gidaris et al., 2019)	48.19 ± 0.77	64.71 ± 0.68	–	–	52.02 ± 0.87	69.75 ± 0.76
PN+rot (Gidaris et al., 2019)	47.46 ± 0.79	64.66 ± 0.68	–	–	48.53 ± 0.88	70.52 ± 0.72
SLA (Lee et al., 2020)	44.95 ± 0.79	63.32 ± 0.68	–	–	45.94 ± 0.87	68.62 ± 0.75
ProtoNet+jig (Su et al., 2020)	42.24 ± 0.77	66.04 ± 0.72	39.52 ± 0.76	62.07 ± 0.78	43.13 ± 0.83	63.30 ± 0.78
ProtoNet+rot (Su et al., 2020)	45.78 ± 0.77	64.77 ± 0.69	44.93 ± 0.86	65.68 ± 0.74	46.81 ± 0.87	69.46 ± 0.75
CSS (An et al., 2021)	50.85 ± 0.84	68.08 ± 0.73	39.32 ± 0.80	66.12 ± 0.77	56.49 ± 0.93	74.59 ± 0.72
SSL-ProtoNet (w/o KD) (Ours)	52.25 ± 0.45	70.60 ± 0.36	55.07 ± 0.49	74.02 ± 0.40	59.18 ± 0.50	76.05 ± 0.38
SSL-ProtoNet (Ours)	**52.58 ± 0.45**	**70.87 ± 0.36**	**55.14 ± 0.49**	**74.23 ± 0.40**	**60.41 ± 0.52**	**76.52 ± 0.38**

Table 3
The ablation study of the proposed SSL-ProtoNet for 5-way K-shot classification tasks.

Model	β	*mini*ImageNet		*tiered*ImageNet		CIFAR-FS	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
$\mathcal{L}_{FSL}$	–	50.30 ± 0.46	68.30 ± 0.35	53.21 ± 0.48	72.54 ± 0.40	57.35 ± 0.52	74.34 ± 0.40
$\theta_{T+R} + \mathcal{L}_{FSL}$	–	51.52 ± 0.45	68.92 ± 0.36	54.11 ± 0.49	73.42 ± 0.40	58.66 ± 0.52	74.70 ± 0.39
$\mathcal{L}_{FSL} + \mathcal{L}_{SSL}$	0.1	50.75 ± 0.45	69.38 ± 0.36	54.17 ± 0.49	73.82 ± 0.40	58.60 ± 0.52	75.65 ± 0.40
$\theta_{T+R} + \mathcal{L}_{FSL} + \mathcal{L}_{SSL}$	0.01	51.71 ± 0.43	69.45 ± 0.35	54.24 ± 0.50	73.49 ± 0.40	58.36 ± 0.48	75.00 ± 0.39
	0.05	52.03 ± 0.43	70.15 ± 0.35	54.67 ± 0.49	73.93 ± 0.40	59.09 ± 0.49	75.82 ± 0.38
	0.1	52.25 ± 0.45	70.60 ± 0.36	55.07 ± 0.49	74.02 ± 0.40	59.18 ± 0.50	76.05 ± 0.38
	0.5	52.24 ± 0.44	70.36 ± 0.37	54.45 ± 0.48	73.06 ± 0.40	59.17 ± 0.47	75.90 ± 0.38
	1.0	51.95 ± 0.44	69.54 ± 0.36	53.53 ± 0.46	72.19 ± 0.40	58.78 ± 0.47	75.89 ± 0.38
$\theta_{T+R} + \mathcal{L}_{FSL} + \mathcal{L}_{SSL} + \mathcal{L}_{KD}$	0.1	**52.58 ± 0.45**	**70.87 ± 0.36**	**55.14 ± 0.49**	**74.23 ± 0.40**	**60.41 ± 0.52**	**76.52 ± 0.38**

0.62% on 5-way 5-shot *mini*ImageNet. The improvement proves that using SSL to increase the sample discriminability is able to improve the generalization in the few-shot task. However, the improvements are not obvious when the number of support samples increases to 5 (shot). This scenario may be caused by the overfitting in the few-shot task thus losing the diversity of the representation. In order to preserve the well generalized representations in the task adaptation process, a self-supervised loss signal $\mathcal{L}_{SSL}$ is proposed in the fine-tuning stage. With this loss signal, the baseline model gains 0.45% improvement in 5-way 1-shot and 1.08% on 5-way 5-shot *mini*ImageNet. Similar observations are discovered for both *tiered*ImageNet and CIFAR-FS datasets. After integrating both components together in the baseline model, the performance receives the optimal results in both 5-way settings. This proves that the $\mathcal{L}_{SSL}$ further diversifies the representation thus improving the generalization as well as alleviating the model overfitting. Different values are tested for the β coefficient to control the $\mathcal{L}_{SSL}$. As presented in Table 3, β with the value of 0.1 achieves highest performance in both 5-way 1-shot and 5-way 5-shot tasks. The same β value is applied to the student model in the self-distillation stage. The accuracies are further increased for all datasets after incorporated with $\mathcal{L}_{KD}$.

To gain better insight of SSL-ProtoNet, the testing losses and accuracies for 5-way 1-shot *mini*ImageNet are plotted in Fig. 3. As depicted

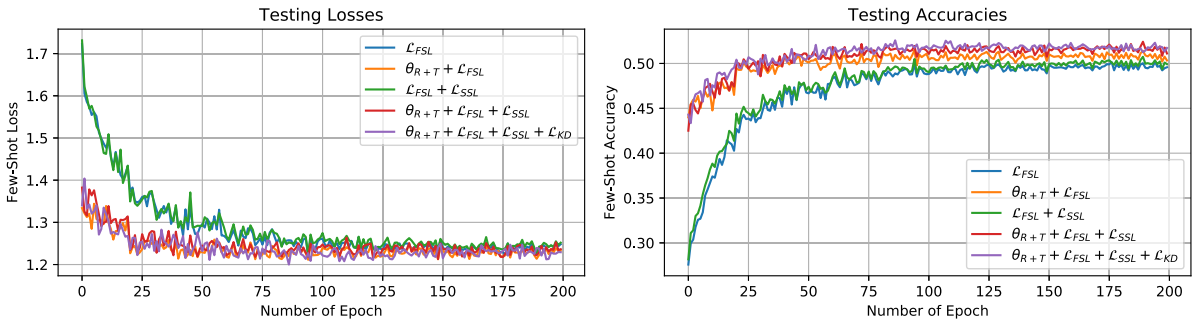

Fig. 3. The losses and accuracies of the baseline models and the proposed SSL-ProtoNet on 5-way 1-shot *mini*ImageNet.

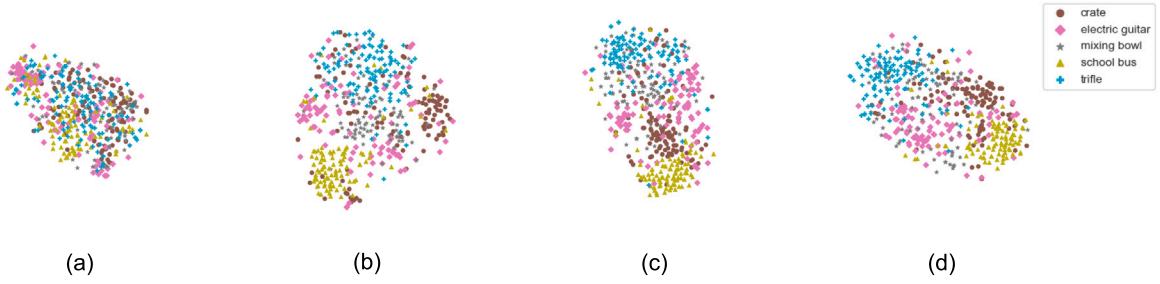

Fig. 4. The t-SNE visualization of sample embeddings of *mini*ImageNet under 5-way 1-shot setting.

in Fig. 3, the testing losses from all combinations of SSL-ProtoNet are gradually decreased, and the testing accuracies are increasing over the epoch. In each epoch, 2000 tasks are randomly selected from the testing set to compute the loss and accuracy. $\theta_{T+R}+\mathcal{L}_{FSL}$, $\theta_{T+R}+\mathcal{L}_{FSL}+\mathcal{L}_{SSL}$, and the proposed SSL-ProtoNet ($\theta_{T+R} + \mathcal{L}_{FSL} + \mathcal{L}_{SSL} + \mathcal{L}_{KD}$) start with a lower error rate as compared to the baseline model $\mathcal{L}_{FSL}$ and $\mathcal{L}_{FSL} + \mathcal{L}_{SSL}$. This provides an insight that pre-trained weights θ_{T+R} and distillation loss $\mathcal{L}_{KD}$ help to speed up the adaptation process during FSL. The error rates of $\theta_{T+R}+\mathcal{L}_{FSL}$ and the proposed SSL-ProtoNet are highly similar but the proposed SSL-ProtoNet starts with slightly higher loss due to the additional self-supervised loss signal $\mathcal{L}_{SSL}$. Eventually all models converge to very small loss values. Compared to the baseline model in terms of accuracy, the $\theta_{T+R}+\mathcal{L}_{FSL}$, $\theta_{T+R}+\mathcal{L}_{FSL}+\mathcal{L}_{SSL}$, and the proposed SSL-ProtoNet begin with higher accuracy rates due to the pre-trained weight θ_{T+R}. The accuracies of $\theta_{T+R}+\mathcal{L}_{FSL}$, $\theta_{T+R}+\mathcal{L}_{FSL}+\mathcal{L}_{SSL}$, and the proposed SSL-ProtoNet are similar but the proposed SSL-ProtoNet obtains higher accuracies due to the benefit from the self-supervised loss signal $\mathcal{L}_{SSL}$ and distillation loss $\mathcal{L}_{KD}$.

4.6. Visualization

To determine the generalization of the model, the learned embedding space of the proposed SSL-ProtoNet are visualized via t-SNE. In *mini*ImageNet, 5 testing classes with 100 samples in each class are randomly selected. The randomly selected 5 novel classes are: (1) *electric guitar*, (2) *trifle*, (3) *mixing bowl*, (4) *crate*, and (5) *school bus*. The t-SNE visualization of the proposed SSL-ProtoNet under 5-way 1-shot setting is depicted in Fig. 4.

Fig. 4(a) demonstrates a very compact feature space of original distribution. The proposed SSL-ProtoNet leverages SSL by training the model in a self-supervised way to learn a preliminary generalization. The results are illustrated in Fig. 4(b). It is observable that the learned embedding f_θ from the pre-training stage clearly shows that the sample embedding from the same class tends to get closer to each other. This demonstrates that the generalization ability from the SSL environment helps to form a better distribution by enhancing sample discrimination in the pre-training stage of the SSL-ProtoNet. The enhanced sample discrimination allows the model to learn across a wide range of samples. In Fig. 4(c), it shows that the proposed SSL-ProtoNet manages to better cluster the original distribution after the fine-tuning stage. This is because the model gets access to the class label during the few-shot process. The adaptation process refines the feature space for all classes. Furthermore, the self-supervised process helps to diversify the model embedding for a smoother adaptation process, thus forming better sample distribution. In Fig. 4(d), the class distribution in the feature space is further refined after the student model obtained the guidance from the teacher model in the self-distillation stage.

Table 4
The performance analysis of different self-supervised tasks using different batch size M in the pre-training stage of the proposed SSL-ProtoNet for *mini*ImageNet 5-way 1-shot classification tasks.

Support	Query	M		
		30	50	100
–	R	38.61 ± 0.39	39.29 ± 0.40	39.53 ± 0.41
T	R	37.98 ± 0.40	39.15 ± 0.41	39.76 ± 0.40
–	$T+R$	43.86 ± 0.41	**44.74 ± 0.42**	44.61 ± 0.41

4.7. Training-way

The number for training-way N is determined for all datasets where $N \in \{5, 10, 15, 20, 25, 30\}$. The value N that yields the highest accuracy is selected for the training-way throughout all the experiments. The results are presented in Fig. 5. In the 5-way 1-shot setting, 30-way is selected for *mini*ImageNet and *tiered*ImageNet while 20-way is chosen for CIFAR-FS. In the 5-way 5-shot setting, 15-way, 25-way, and 10-way are selected for *mini*ImageNet, *tiered*ImageNet, and 10-way CIFAR-FS, respectively. Same number of training-way N is applied in both fine-tuning and self-distillation stages.

4.8. Transformation analysis

The pre-training stage performs SSL to cluster the augmented sample to its original sample. In this section, we present an experiment where different augmentations are applied to the support samples and query samples with batch size $M \in \{30, 50, 100\}$ during the pre-training

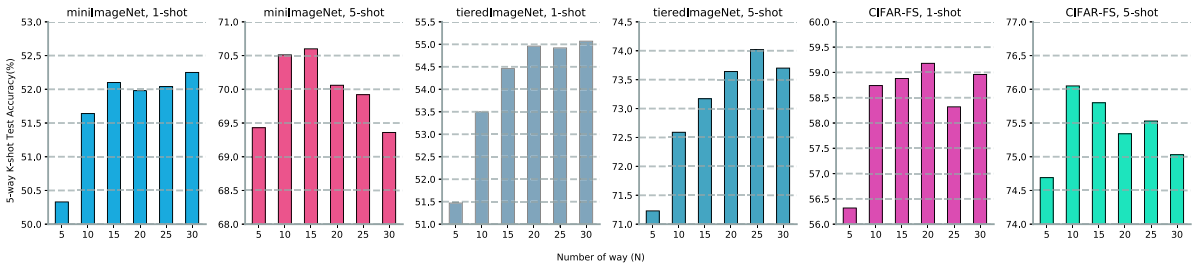

Fig. 5. The results of different training-way N in the proposed SSL-ProtoNet's fine-tuning stage on 5-way K-shot classification tasks.

Table 5
The comparison results for baseline model that only applied augmentation T and the proposed SSL-ProtoNet without self-distillation stage (w/o KD).

Method	*mini*ImageNet, 5-way		*tiered*ImageNet, 5-way		CIFAR-FS, 5-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
T	48.12 ± 0.43	68.49 ± 0.36	47.05 ± 0.47	66.87 ± 0.41	53.88 ± 0.53	72.23 ± 0.42
SSL-ProtoNet (w/o KD)	**52.25 ± 0.45**	**70.60 ± 0.36**	**55.07 ± 0.49**	**74.02 ± 0.40**	**59.18 ± 0.50**	**76.05 ± 0.38**

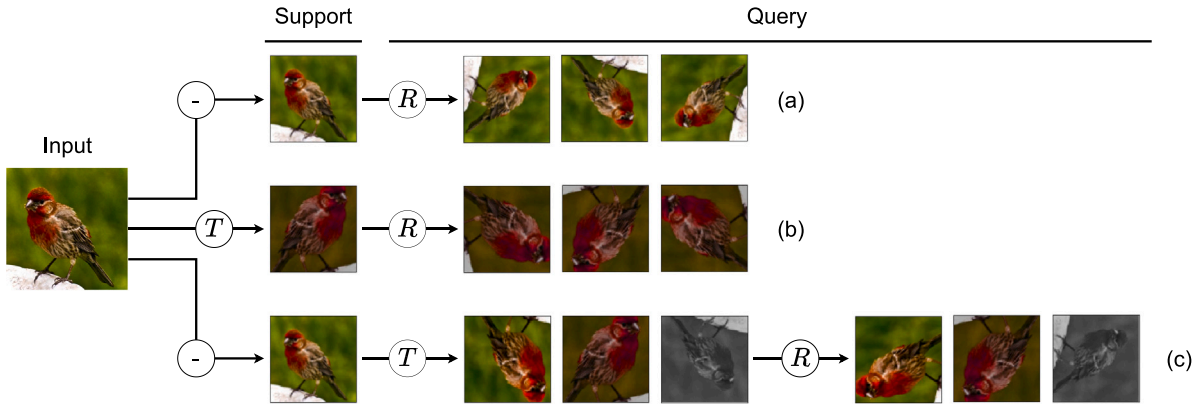

Fig. 6. The different combinations of augmentations applied to the support image to form its query images. (a) only resize the input image as support image without rotation and transformation (–) and apply rotation to form its query images (R). (b) apply random transformation (T) to generate the support image and apply rotation to form the query images (R). (c) only resize the input image to obtain support image (–), and apply noisy transformation ($T + R$) to form its query images.

Table 6
Different distance function of SSL-ProtoNet on CIFAR-FS.

Distance	5-way 1-shot	5-way 5-shot
Cosine	49.82 ± 0.57	58.00 ± 0.45
Euclidean	**59.18 ± 0.50**	**76.05 ± 0.38**

Table 7
Different γ values in the self-distillation stage of SSL-ProtoNet on CIFAR-FS.

γ	5-way 1-shot	5-way 5-shot
0.1	59.95 ± 0.52	**76.52 ± 0.38**
0.3	60.36 ± 0.52	76.37 ± 0.38
0.5	60.36 ± 0.52	76.05 ± 0.38
0.7	**60.41 ± 0.52**	76.07 ± 0.38
0.9	60.32 ± 0.50	76.02 ± 0.38

stage. The augmentation combination includes: no augmentation –, rotation R and a set of random transformations T. The details of the random transformation set T includes:

1. Random resized crop with scale 0.5 to 1.0 and bilinear interpolation.
2. Random (50%) horizontal flip and (50%) vertical flip.
3. Random (80%) color jittering (brightness, contrast, saturation = 0.4 and hue = 0.1).
4. Random (20%) grayscaling.

The illustration of the noise transformations is depicted in Fig. 6 and the analysis of performance is presented in Table 4. Three different combinations in Fig. 6 are: (a) directly rotate the support image as query images (–, R), (b) augment the support image and rotate it to form query images (T, R), and (c) produce different augmented query images and rotate it without augmenting support images (–, $T + R$).

Noticeably, applying the noisy transformation (both random transformations T and rotation R) to the query samples yields the highest performance (as shown in Table 4). The improvement shows that when the model is given more diverse query samples in the learning process, it obtains richer representation and better generalization. The experimental results in Table 4 also demonstrate that larger batch size M has better performance for the proposed SSL-ProtoNet. Based on the results, M is set to 50 for *mini*ImageNet and CIFAR-FS datasets, and set to 100 for *tiered*ImageNet.

To further investigate if the performance of the proposed SSL-ProtoNet is benefited from SSL instead of augmentation itself, the baseline model that utilizes random transformation T in all samples and the proposed SSL-ProtoNet (without self-distillation stage) are compared. The results are reported in Table 5. Based on the results, it is notable that the performance of the baseline model with augmentation decreased significantly as compared to the proposed SSL-ProtoNet. The baseline model is not able to directly handle the visual complexity introduced from the complex transformations. However, SSL-ProtoNet successfully utilizes SSL with these complex transformations to enhance the performance.

Table 8
The time complexity of SSL-ProtoNet on CIFAR-FS.

Method	Stage	5-way 1-shot	5-way 5-shot
FEAT (Ye et al., 2020)	2	08h58m15s	09h23m39s
SSL-ProtoNet (w/o KD)	2	**06h59m02s**	**07h21m36s**
CSS (An et al., 2021)	3	12h22m55s	15h35m59s
SSL-ProtoNet	3	**09h38m12s**	**10h23m20s**

4.9. Cosine vs. Euclidean distance

In this section, different distance functions have been examined in the fine-tuning stage of the proposed SSL-ProtoNet including Euclidean and cosine distance on CIFAR-FS. From Table 6, it is notable that Euclidean distance outperformed the cosine distance in both 1-shot and 5-shot results. Specifically, an improvement of approximately 9% in the 1-shot setting and 18% in the 5-shot setting are observed.

4.10. Distillation analysis

In this section, different γ values are examined to determine the impact of the distillation loss $\mathcal{L}_{KD}$ in the self-distillation stage on CIFAR-FS. The examined values include {0.1, 0.3, 0.5, 0.7, 0.9} and the results are reported in Table 7. The 1-shot setting demonstrates a preference for larger γ values, whereas the 5-shot setting tends to favor smaller γ values. Based on the result, γ is set to 0.7 in 1-shot and 0.1 in 5-shot for CIFAR-FS. For *mini*ImageNet and *tiered*ImageNet, the γ values are set to 0.1 for all settings.

4.11. Time complexity

In this section, we compare the time complexity of the proposed SSL-ProtoNet with FEAT (Ye et al., 2020) and CSS (An et al., 2021). FEAT is implemented with 2 stage training. Thus, we compare it with our proposed SSLProtoNet without the distillation stage (w/o KD), the time complexity is $\mathcal{O}(S_1 + S_2)$. CSS and proposed SSL-ProtoNet are implemented with 3 stage training. Therefore, their time complexity is $\mathcal{O}(S_1 + S_2 + S_3)$. To compute the total execution time, we trained the methods with ConvNet backbone on the CIFAR-FS dataset using a single NVIDIA Quadro P6000 GPU. Based on Table 8, we observed that the total execution time for FEAT is around 9 h for 1-shot and around 9.5 h for 5-shot settings, while the total execution time for CSS is around 12.5 h for 1-shot and around 15.5 h for 5-shot settings. In comparison, the total execution time for our proposed SSL-ProtoNet is significantly lower in both 2 stage and 3 stage training, taking only around 7 h for 1-shot and around 7.5 h for 5-shot settings in 2 stage training, around 9.5 h for 1-shot and around 10.5 h for 5-shot settings in 3 stage training. The lower time complexity of our proposed SSL-ProtoNet is due to the efficient utilization of SSL during the training process, which enhances the generalization ability of the model and reduces the amount of training required. Overall, the results demonstrate that our proposed SSL-ProtoNet is an effective and efficient approach to FSL.

5. Conclusion

This paper presents a 3-stage FSL method referred to as SSL-ProtoNet. It leverages the SSL and knowledge distillation to enhance the performance in few-shot image classification tasks. In the pre-training stage, SSL is utilized to provide a primary stage of generalization via improving sample discrimination. This is more suitable for adaptation across a wide range of instances by mapping different augmented query samples to its support sample. The empirical results demonstrate that the greater the difference between augmented samples, the better the generalization capability of the model. In the fine-tuning stage, an additional objective function (self-supervised process) is introduced into the proposed SSL-ProtoNet. This objective function maps the rotated query image to its corresponding support image. By doing so, the few-shot process maintains the representation diversity and avoids overfitting during few-shot task adaptation. With these SSL strategies, the proposed SSL-ProtoNet outshines other FSL methods on all three benchmark few-shot image classification datasets. The knowledge distillation in the self-distillation stage further boosts the performance of the proposed SSL-ProtoNet.

CRediT authorship contribution statement

Jit Yan Lim: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft. **Kian Ming Lim:** Supervision, Methodology, Funding acquisition, Writing – review & editing. **Chin Poo Lee:** Validation, Writing – review & editing. **Yong Xuan Tan:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research is supported by the Fundamental Research Grant Scheme (FRGS), Malaysia under Grant No. FRGS/1/2021/ICT02/MMU/02/4. We gratefully acknowledge the support of NVIDIA Corporation, United States with the donation of the A100 80 GB GPU used for this research under the NVIDIA Academic Hardware Grant Program.

References

Allen, K. R., Shelhamer, E., Shin, H., & Tenenbaum, J. (2019). Infinite mixture prototypes for few-shot learning. In *ICML*.

An, Y., Xue, H., Zhao, X., & Zhang, L. (2021). Conditional self-supervised learning for few-shot classification. In Z.-H. Zhou (Ed.), *Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI-21* (pp. 2140–2146). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2021/295, Main Track.

Antoniou, A., Edwards, H., & Storkey, A. (2019). How to train your MAML. In *International conference on learning representations*.

Bertinetto, L., Henriques, J. F., Torr, P., & Vedaldi, A. (2018). Meta-learning with differentiable closed-form solvers. In *International conference on learning representations*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607). PMLR.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C., & Huang, J.-B. (2019). A closer look at few-shot classification. In *International conference on learning representations*.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1* (pp. 539–546). http://dx.doi.org/10.1109/CVPR.2005.202, vol. 1.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).

Dhillon, G. S., Chaudhari, P., Ravichandran, A., & Soatto, S. (2020). A baseline for few-shot image classification. In *International conference on learning representations*.

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision* (pp. 1422–1430).

Dosovitskiy, A., Fischer, P., Springenberg, J. T., Riedmiller, M., & Brox, T. (2015). Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(9), 1734–1747.

Finn, C., Abbeel, P., & Levine, S. (2017). In D. Precup, & Y. W. Teh (Eds.), *Proceedings of machine learning research*: *Vol. 70*, *Model-agnostic meta-learning for fast adaptation of deep networks* (pp. 1126–1135). International Convention Centre, Sydney, Australia: PMLR.

Finn, C., Xu, K., & Levine, S. (2018). Probabilistic model-agnostic meta-learning. In *NeurIPS*.

Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., & Cord, M. (2019). Boosting few-shot visual learning with self-supervision. In *2019 IEEE/CVF international conference on computer vision (ICCV)* (pp. 8058–8067).

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. In *International conference on learning representations*.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS deep learning and representation learning workshop*.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop, Vol. 2*. Lille.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Lee, H., Hwang, S. J., & Shin, J. (2020). Self-supervised label augmentation via input transformations. In *International conference on machine learning* (pp. 5714–5724). PMLR.

Lee, K., Maji, S., Ravichandran, A., & Soatto, S. (2019). Meta-learning with differentiable convex optimization. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 10649–10657).

Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-SGD: Learning to learn quickly for few shot learning. ArXiv, abs/1707.09835.

Lim, J. Y., Lim, K. M., Lee, C. P., & Tan, Y. X. (2023). SCL: Self-supervised contrastive learning for few-shot image classification. *Neural Networks*, *165*, 19–30. http://dx.doi.org/10.1016/j.neunet.2023.05.037.

Lim, J. Y., Lim, K. M., Ooi, S. Y., & Lee, C. P. (2021). Efficient-PrototypicalNet with self knowledge distillation for few-shot learning. *Neurocomputing*, *459*, 327–337. http://dx.doi.org/10.1016/j.neucom.2021.06.090.

Medina, C., Devos, A., & Grossglauser, M. (2020). Self-supervised prototypical transfer learning for few-shot classification. arXiv preprint arXiv:2006.11325.

Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. ArXiv, abs/1803.02999.

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision* (pp. 69–84). Springer.

Noroozi, M., Pirsiavash, H., & Favaro, P. (2017). Representation learning by learning to count. In *Proceedings of the IEEE international conference on computer vision* (pp. 5898–5906).

Oreshkin, B., Rodríguez López, P., & Lacoste, A. (2018). Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, *31*, 721–731.

Raghu, A., Raghu, M., Bengio, S., & Vinyals, O. (2020). Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *International conference on learning representations (ICLR)*.

Rajasegaran, J., Khan, S., Hayat, M., Khan, F. S., & Shah, M. (2020). Self-supervised knowledge distillation for few-shot learning. https://arxiv.org/abs/2006.09785.

Ravi, S., & Larochelle, H. (2017). Optimization as a model for few-shot learning. In *ICLR*.

Ren, M., Ravi, S., Triantafillou, E., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., & Zemel, R. S. (2018). Meta-learning for semi-supervised few-shot classification. In *International conference on learning representations*.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., & Hadsell, R. (2019). Meta-learning with latent embedding optimization. In *International conference on learning representations*.

Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook* (Diploma thesis).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in neural information processing systems* (pp. 4077–4087).

Su, J.-C., Maji, S., & Hariharan, B. (2020). When does self-supervision improve few-shot learning? In *European conference on computer vision* (pp. 645–666). Springer.

Sun, Q., Liu, Y., Chen, Z., Chua, T. S., & Schiele, B. (2020). Meta-transfer learning through hard tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1. http://dx.doi.org/10.1109/TPAMI.2020.3018506.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 1199–1208). http://dx.doi.org/10.1109/CVPR.2018.00131.

Thrun, S., & Pratt, L. Y. (1998). Learning to learn: Introduction and overview. In *Learning to learn*.

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B., & Isola, P. (2020). Rethinking few-shot image classification: A good embedding is all you need? In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer vision – ECCV 2020* (pp. 266–282). Cham: Springer International Publishing.

Vilalta, R., & Drissi, Y. (2005). A perspective view and survey of meta-learning. *Artificial Intelligence Review*, *18*, 77–95.

Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In *NIPS*.

Wang, Z., Ma, P., Chi, Z., Li, D., Yang, H., & Du, W. (2022). Multi-attention mutual information distributed framework for few-shot learning. *Expert Systems with Applications*, *202*, Article 117062. http://dx.doi.org/10.1016/j.eswa.2022.117062.

Xu, S., & Xiang, Y. (2021). Frog-GNN: Multi-perspective aggregation based graph neural network for few-shot text classification. *Expert Systems with Applications*, *176*, Article 114795. http://dx.doi.org/10.1016/j.eswa.2021.114795.

Ye, H.-J., Hu, H., Zhan, D.-C., & Sha, F. (2020). Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8808–8817).

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision* (pp. 649–666). Springer.

Zhou, F., Cao, C., Zhong, T., & Geng, J. (2021). Learning meta-knowledge for few-shot image emotion recognition. *Expert Systems with Applications*, *168*, Article 114274. http://dx.doi.org/10.1016/j.eswa.2020.114274.