Cosine Transformer for Transductive Few-Shot Image Classification

Quang-Huy Nguyen[1], Quoc-Cuong Nguyen[1], Dung D. Le[2], Hieu H. Pham[1,2,*], and Minh Do[1,3]

[1] VinUni-Illinois Smart Health Center, VinUniversity, Hanoi, Vietnam;
{huy.nt,cuong.nq,hieu.ph,dung.ld}@vinuni.edu.vn
[2] College of Engineering & Computer Science, VinUniversity, Hanoi, Vietnam;
[3] University of Illinois at Urbana-Champaign, US;
minhdo@illinois.edu
*Corresponding author

Abstract. This paper addresses the few-shot image classification problem. Given a pair of labeled support and unlabeled query sets, the approaching method is to obtain the heatmap between two sets to label the latter one in the transductive setting manner. One effective method is to utilize the cross-attention with scale dot-product mechanism. However, the difference in magnitude between two different sets of embedding vectors can affect the output attention map a great deal. We tackle this problem by improving the attention mechanism with cosine similarity. With the proposed cosine attention, we develop a few-shot image classification method named FS-CT (Few-shot Cosine Transformer). The proposed cosine attention improves FS-CT performances by nearly 5% to over 20% in accuracy compared to the scaled dot-product attention in various scenarios on mini-ImageNet, CUB-200, and CIFAR-FS datasets. The FS-CT method along with the cosine attention mechanism is simple to implement, with the implementation code available at https://github.com/vinuni-vishc/Few-shot-transformer.

1 Introduction

In this paper, we seek to tackle the few-shot image classification problem, which classifies unlabeled data based on a few amount of labeled samples that share the same categories. A standard few-shot learning problem, generally, includes two major sets: a labeled support set that provides just small samples to perform the problem's task—in this case, categorical classification—on the unlabeled query set. Both sets are then split into two groups, disjoint in categories, for training and testing the approach method. A few-shot learning problem is sometimes referred with the term "*n-way k-shot*", where *n-way* indicates the number of categories that the task performs, and *k-shot* indicates the number of provided support samples per category. There are two major challenges for a few-shot

learning problem. First, the number of labeling samples per category for each training/testing step is very small; Second, the proposed method must be able to shift into other unseen domain tasks, where the categoricals are disjoined. As one of the basic problems in computer vision few-shot learning, studies often be conducted to tackle the problem, which are splited into three main approaches: (i) using a distance or similarity metric to label query samples on a learnable embedding space [32], (ii) utilizing an additional memory for an extra learning method [26], and (iii) fine-tunning learnable parameters to transfer to other tasks with few update steps [23]. Since the provided data for performing tasks is limited, methods are often developed with complex architecture to obtain a good embedding to tackle the task.

To develop an effective method for the classification task, the pivotal step is obtaining a well-represented relationship between the support and query images, then classified queries based on the information. In this work, we make two main contributions. For *our first contribution*, based on provided support and query set, we learned a good embedding model that can represent well individual categories from the labeled data. We choose prototypical embedding [29] as this comes with effective performance with low computational cost, simply calculating the arithmetic mean of embedding features from the same categories. However, as samples with the same label are small in quantity, the average representation is not well reliable. Therefore, we applied the weight averaging mean instead and made the weight learnable, which can represent better the categorical distributions.

For *our second contribution*, based on the learned representations, we obtained a good relationship map between two sets to provide information for the classification tasks. We used the cross-attention mechanism from Transformer [31], to tackle the step. However, the standard scaled dot-product attention is limited in learning ability, mainly by the difference in magnitudes. This problem is somewhat tackled by dividing the initial multiplied matrix with a fixed scaling factor. However, dividing an entire matrix with a fixed scalar helps reduce the effect, not solve them completely. Therefore, to remove the effect, we applied Cosine Similarity to the attention mechanism, replacing the scale dot-product. This helps enhance the learning ability of the few-shot method.

The two main contributions are incorporated in our proposed Few-Shot TransFomer (FS-CT). To prove the effectiveness of our method, we developed a detailed evaluation scheme with various settings, configurations, and backbone networks. Our FS-CT with two attention variants using either the standard scaled-dot product attention (which is called Softmax Attention in this study) and the improved Cosine Attention are evaluated on three few-shot datasets mini-ImageNet, CUB-200, and CIFAR-FS with various backbones and few-shot settings. We further utilized another attention-based few-shot image classifica-

tion CrossTransformer (CTX) [6] for a fine comparison under the same evaluation scheme, also validating our proposed Cosine Attention on the other method. Furthermore, we developed a few-shot dataset for Yoga poses with 50 categories with arbitrary image size and noise, which somewhat reflect the actual data, as a customized dataset to evaluate methods' performance and served as an initial step of conducting research for heath-related Yoga monitoring problem. Our performances on various configurations and datasets prove the effectiveness of the proposed Cosine Attention in various conditions.

In summary, our contributions in this paper are: (i) We modify the prototypical network with learnable averaging weight for better embedding features ;(ii) We proposed the Cosine Attention, a mechanism to learn attention weight without the effect of the difference in feature's magnitudes, which improves learning performance; We (iii) proposed Few-Shot Cosine Transformer, which tackles the few-shot image classification task using Transformer encoder and the proposed Cosine Attention mechanism. Finally (iv), we evaluated and ablated how the improved Cosine Attention impacts the few-shot algorithms, using a detailed evaluation scheme on various datasets, backbones, and configurations.

In the following, Section 2 provides mathematical formulatuon of the few-shot task and reviews previous studies; Section 3 specifies our proposed FS-CT with Cosine Attention mechanism; Section 4 presents our experimental settings and evaluation; Section 5 discusses limitations and future works; and Section 6 concludes our contributions.

2 Background and Related Work

2.1 Problem setting

We first formalize a standard few-shot classification problem while introducing some notations. Given a base set D_{base} and meta-learner algorithm $\boldsymbol{A}(.|\theta)$. The objective is to learn the optimal parameter θ^* so that it can achieve a good performance of algorithm $\boldsymbol{A}(.|\theta)$ on a novel set D_{novel}. D_{base} categories do not overlap with D_{novel} categories.

A few-shot learning problem is usually trained with an episodic learning strategy, where the proposed approach is trained and tested on different tasks with different sets of categories. The episodic learning with m tasks is generally described in Figure 1. Individual task $\mathcal{T} = \{S,\ Q\} \sim p(D)$ is derived randomly from data set D, where D can be either the training set D_{base} or testing set D_{novel} depending on the training scheme and $p(D)$ is the distribution over D. Each task $\mathcal{T}$ consists of two sets: labeled support set S and unlabeled query set Q that share the same n ground-truth categories. Task $\mathcal{T} = \{S,\ Q\}$ is also called *episodic batch* or episode.

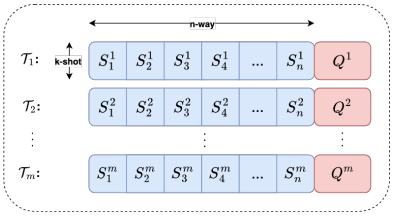

Fig. 1. Episodic learning for Few-shot Learning

Let $(x,\ y)$ be defined as the input image sample and its ground-truth respectively. The objective for a Few-shot Learning problem is to recognizing labels for the Query set $Q = \left\{\boldsymbol{x}_q^i\right\}_{i=1}^{q}$, given the Support set $S = \left\{(\boldsymbol{x}_s,\ y_s)^{i,j}\right\}_{i=1,\cdots,n}^{j=1,\cdots,k}$, where $y_s \in C$ for a set of n categories (n-*way*), k is the number of training samples per category (k-*shot*), and q is the total number of Q samples. The number of k must be small in a few-shot setting. In this paper, we explore two configurations: 5-way 5-shot and 5-way 1-shot. The few-shot classification problem can be formulated with the following optimization function:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(S,Q)\sim p(D)} \mathcal{L}(S, Q;\ \boldsymbol{A}(.|\theta)) \tag{1}$$

$$\mathcal{L}(S, Q;\ \boldsymbol{A}(.|\theta)) = \frac{1}{q} \sum_{(\boldsymbol{x}_q, y)\in Q} -\log\left[\ \boldsymbol{p}(y\big|\boldsymbol{A}(\boldsymbol{x}_q|\theta);\ S)\right] + \lambda R(\theta)$$

where $\boldsymbol{p}(y\big|\boldsymbol{A}(\boldsymbol{x}_q|\theta);\ S)$ is the probabilistic prediction of sample $\boldsymbol{x}_q \in Q$ on true label y using few-shot algorithm $\boldsymbol{A}(.|\theta)$ given for $\boldsymbol{x}_q$ and support set S. $\lambda R(\theta)$ is an optional regularization with factor λ. The loss function $\mathcal{L}(S, Q;\ \boldsymbol{A}(.|\theta))$ is dependent on the few-shot problem and method, in this case, Categorical Cross-Entropy loss for categorical classification.

2.2 Related Work

Few-shot Learning Few-shot learning is a subset of Meta-Learning, which develops models with the ability of "learning to learn" with only a few training samples. Few-shot learning algorithms obtain task-level knowledge from seen data and effectively generalize them for solving tasks from unseen data. Based on the learning method, Few-shot learning algorithms can be divided into three learning methods. *Metric-based learning* [4,13,22,29,30,32] learns an embedding space that samples from the same category are mapped to points that close together and vice versa under a distance metric. *Memory-based learning* [19,20, 26] involves external memory to store and update data, thus updating parameters rapidly to solve a few-shot learning problem. *Optimization-based learning* [8,9, 14,23,25,35] fine-tune model's parameters to quickly adapt to the new task with

only a few effective gradient descent steps. Metric-based learning is the most simple yet effective method among three, which usually consists of two stages: *feature extraction* to obtain extracted feature from both labeled (support) and unlabeled (query) samples under the same embedding space, and *metric function* to utilize a similarity or distance metric for categorizing unlabeled samples by comparing [30] or clustering [29] the embedded features.

Base on the test settings, there are two learning approaches for few-shot algorithm, including *inductive learning* [8, 28, 30, 32] and *transductive learning* [1, 6, 13, 16, 21]. Inductive learning classifies each query sample individually, while transductive learning classifies every query sample collectively. The latter learning method allows additional information in data distribution or visual resemblance can be obtained and leveraged among query samples, thus potentially improving the overall performance. In this study, we considered transductive learning for our method due to the characteristic of the attention mechanism.

Transformer Attention Mechanism After being introduced in [31] for natural language processing tasks, Transformer soon rising to the dominance in Computer Vision [3, 5, 7]. The core of Transformer is the attention mechanism, which calculates an attention weight relationship between features for solving tasks. Attention mechanism comes in two variants, self-attention determines the inner relationships within a feature set, and cross-transformer calculates the outer relationships between two feature sets. Several few-shot learning studies are inspired by Transformer and its attention mechanism [6, 11, 15, 34], where the methods, in general, involve attention mechanism to align labeled and unlabeled feature for classification. Our method is inspired by Transformer architecture and attention mechanism to solve the problem.

3 Proposed Method

In this section, we describe our Few-Shot Cosine Transformer (FS-CT) architecture, which utilizes episodic learning to learn the relationship between label Support set and unlabled Query set via attention representation for classifying Query set follows the transductive learning method. Figure 2 presents the overall architecture of FS-CT, which share the similar pipeline with the Transformer encoder.

3.1 Few-shot Cosine Transformer

Overall, FS-CT shares a similar architecture with the transformer encoder architecture. Given two input support set S and query set Q, their images are fed into a backbone feature extractor to obtain two feature tensor Z_S and Z_Q, then

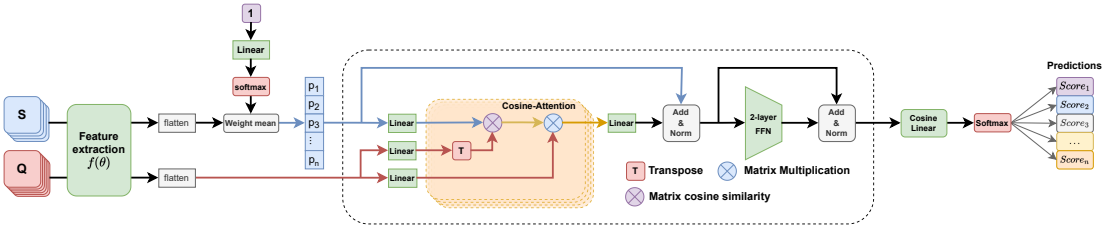

Fig. 2. Proposed method with Cosine attention mechanism

features from Z_S is averaged along individual categories to obtain the prototypical representation Z_P like the Prototypical Network [29]. Unlike conventional transformer-based architectures, positional encoding is not applied, since the order of features is not an impactful factor for the classification task. After that, Z_P and Z_Q are brought into three linear layers to split into a multi-head of three features $\langle \mathbf{q}^*, \mathbf{k}, \mathbf{v} \rangle$[4], then go through a multi-head cross-attention mechanism to obtain the weight attention features of Z_Q on Z_P. Instead of using the vanilla softmax attention, we propose a variation of the attention mechanism named Cosine Attention, which utilizes Cosine Similarity to calculate the attention weight. The outcome attention values between heads are then fed to an output linear layer to combine heads together, followed by a two-layer MLP with GELU activation function. Two skip-connected layers are used between steps to prevent losing information and layer normalization is applied before linear layer for smoothing values throughout the FS-CT. Finally, the outcome feature is brought through a Cosine linear layer followed by the softmax to yield probabilistic scores on individual categories to predict queries' labels. In the following subsections, we will describe in detail essential modules within our FS-CT method:

3.2 Learnable prototypical representation

Given support set $S = (X_S, Y_S)$ and query set $Q = (X_Q, Y_Q)$ follows few-shot setting (n-way, k-shot, q query samples) with $Y_S,\ Y_Q \in n$ categories; $X_S,\ X_Q \in \mathbb{R}^{c \times h \times w}$; X_S and X_Q are first fed to a backbone feature extraction $f(.|\theta)$ to obtain the feature representations $Z_S \in \mathbb{R}^{n \times k \times d}$, $Z_Q \in \mathbb{R}^{q \times d}$. Then, all k learned features $\{z_c^i\} \in Z_S,\ i \in [1, k]$ that share the same category c are then average equally obtain the prototypical representation $Z_P \in \mathbb{R}^{n \times d}$, with $z_c \in Z_P$ represented for the centroid of category c. In the context of Few-shot learning,

[4] To avoid the conflict from double usage of the term "query", we denote the queries $\mathbf{q}^*$ of the attention mechanism to distinguish with the Query set Q with sample size q in the Few-shot tasks

we noticed that the conventional averaging equally method cannot obtain well the categorial representatives, as there might exist samples with noise or on the edge of the categorial feature distribution that affects the proto-representative values. Therefore, we developed a learnable averaging method to obtain the proto feature for each category that represents better with given data as follows:

$$z_P = \sum_{i=1}^{k} z_s^i \; . \; \text{Softmax}\left[g(a|\theta_P)\right] \tag{2}$$

Initially, the averaging weights $W_{avg} = g(a|\theta_P)$ for calculating categories' centroid are obtained by feeding a fixed scalar input a into the linear layer $g(.|\theta_P)$ with learnable weight $\theta_P \in \mathbb{R}^{n\times k\times 1}$ and no bias. We used $a = 1$ so that the output W_{avg} can have the same value with the weight θ_P. After achieving the averaging weight, Softmax is applied along the k axis to ensure that components within that axis form the weight distribution with the sum of 1. Then, the prototypical representation $Z_P \in \mathbb{R}^{n\times d}$ is obtained by element-wise multiplication between $Z_s \in \mathbb{R}^{n\times k\times d}$ and $W_{avg} \in \mathbb{R}^{n\times k\times 1}$, then summarize values along the k axis to achieve the weight averaging representation on each category, which is the prototypical representation.

In our implementation, instead of feeding a fixed scalar into a linear layer, we directly initiate a learnable parameter with value $a = 1$ that shares the same dimensional space $\mathbb{R}^{n\times k\times d}$ and fed it to a softmax function for W_{avg}. We used the initial value of 1 for the learnable parameter as we want to begin with averaging equally among feature vectors within the same category at first, but then the averaging weight values will vary through the training process to obtain a better prototypical representation that fits best with given data. In the experiment, this learnable averaging approach helps us achieve a better representation, and thus comes with higher performances in many scenarios.

3.3 Cosine Attention

Cosine Similarity between two matrices Given two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, the Cosine Similarity between $\mathbf{a}$ and $\mathbf{b}$ is calculated by the dot-product of two vectors divided by the product of their magnitudes, as follows:

$$S_C(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\cdot\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\cdot\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{3}$$

From the definition for vectors, we extended the above definition for Cosine Similarity on matrices. Specifically, the cosine similarity $S_C(A, B) \in \mathbb{R}^{n\times m}$ between matrix $A \in \mathbb{R}^{n\times k}$ and $B \in \mathbb{R}^{k\times m}$ is the Hadamard division between the multiplication of A and $B^\top$ and the outer product between vectors $M_A \in \mathbb{R}^n$ and $M_{B^\top} \in \mathbb{R}^m$, where M_A and $M_{B^\top}$ are the vectors of the magnitude values

of the row vectors of two matrices A and $B^\top$, respectively. The definition is described by the formula:

$$\begin{aligned} S_C(A,B) &= (A \cdot B) \oslash (M_A \otimes M_{B^\top}) \\ &= (A \cdot B) \oslash \left(\left[\sqrt{\textstyle\sum_{i=1}^{k} a_{1,i}^2}, \cdots, \sqrt{\textstyle\sum_{i=1}^{k} a_{n,i}^2} \right] \otimes \left[\sqrt{\textstyle\sum_{i=1}^{k} b_{i,1}^2}, \cdots, \sqrt{\textstyle\sum_{i=1}^{k} b_{i,m}^2} \right] \right) \end{aligned} \tag{4}$$

With that definition, individual element $S_C(A,B)_{i,j}$ where $i \in [1,\ n],\ j \in [1,\ m]$ is the cosine similarity between the row vector a_i of matrix A and the column vector b_j of matrix B.

Cosine Attention mechanism Initially, as n and q are different in values, we reshaped the proto-feature $Z_P \in \mathbb{R}^{n \times d}$ into the $1 \times n \times d$ tensor and $Z_Q \in \mathbb{R}^{q \times d}$ into a $q \times 1 \times d$ tensor so we can maintain both n and q dimensions in the attention output $H_{out} \in \mathbb{R}^{q \times n \times d}$, providing the attention relationship between Z_P and Z_Q. With the two reshaped tensors, sets of three representations $\langle \mathbf{q}^*,\ \mathbf{k},\ \mathbf{v} \rangle$ are obtained by linear layers: $\mathbf{q}^* = Z_P \cdot \theta_{q^*}^\top;\ \mathbf{k} = Z_Q \cdot \theta_k^\top;\ \mathbf{v} = Z_Q \cdot \theta_v^\top$ where $\theta_{\mathbf{q}^*},\ \theta_{\mathbf{k}},\ \theta_{\mathbf{v}} \in \mathbb{R}^{d \times d_h}$ are the weight matrices, d_h is the dimension inside attention. The output attention head H_{out} is calculated by multiplying the attention weight matrix $\mathbf{A} \in \mathbb{R}^{q \times n \times 1}$ between q^* and $k^\top$ with $v \in \mathbb{R}^{q \times 1 \times d_h}$, using the scaled dot-product or "softmax attention":

$$\textbf{\textit{Softmax Attention}} = \text{Softmax}\left[(\mathbf{q}^* \cdot \mathbf{k}^\top)/\sqrt{d} \right] \cdot \mathbf{v} \tag{5}$$

Specifically, the matrix multiplication performs dot-product operation between every pair of feature vectors between $\mathbf{q}^*$ and $\mathbf{k}^\top$, then divided by a scaling factor $\sqrt{d}$ before feeding to a softmax function for a attention weight matrix $A \in \mathbb{R}^{q \times n}$. The involvement of the scaling factor helps counter the effect of enlarging in magnitude by the dot product that makes the softmax function produce an extremely small gradient output, thus causing gradient vanishing. However, by fixing a determined number for scaling the whole attention matrix, the differential in magnitude between features is still maintained, which not only makes it unfair to estimate the similarities between features with different lengths but also makes the attention map dependent solely on the multiplication operation.

Therefore, to eliminate the effect of the differences in vectors' magnitude, we replaced the dot-product with cosine similarity for calculating $\mathbf{A}$. We clarify that the concept of replacing the dot-product operation with cosine similarity is not new and has been studied in [18] and applied in few-shot image classification in [4, 10] in terms of a cosine similarity-based classification/recognition model. In this study, the replacement of cosine similarity in the attention mechanism helps us highlight the alignment between two representations by features' con-

tent. Specifically, instead of using a fixed number for scaling the entire weight matrix, individual components of the multiplicated matrix will be divided with the product of their corresponding vector's magnitude. We refer to this attention mechanism as "Cosine Attention" with the general formula described as follows, based on the definition of cosine similarity for matrices in the above section:

$$\textbf{\textit{Cosine Attention}} = \left[(\mathbf{q}^* \cdot \mathbf{k}^\top) \oslash (M_{q^*} \otimes M_k)\right] \cdot \mathbf{v} \tag{6}$$

With cosine similarity, the output attention map $\mathbf{A}$ focuses more on the features' content and determines better the similarity relationship between every pair of features $\langle q_i^* \in \mathbf{q}^*,\ k_j \in \mathbf{k}^\top \rangle$. By the nature of cosine similarity to output values within the range of $[-1, 1]$, the Softmax function is no longer necessary for normalizing values. Without the softmax, attention weight matrix $\mathbf{A}$ still maintains the probabilistic distributions in the row vectors as well as its components' ratio. Moreover, as cosine similarity does not scale the weight distributions into the sum of 1, $a_{i,j} \in \mathbf{A}$ possesses a wider range of value. This helps emphasize query features on aligned categories in H_{out} and vice versa, hence boosting the model's performances. In the experiment, removing Softmax in our proposed Cosine Attention helps increase our method's performance greatly.

Furthermore, the multi-head mechanism is also applied for Cosine Attention. The initial three linear layers split Z_P and Z_Q into sets of $\langle \mathbf{q}^*{}_t,\ \mathbf{k}_t,\ \mathbf{v}_t \rangle$ where $t \in [1, 8]$. For each set, a corresponding h_t is computed, represented for the projection output in different perspectives. Then, the $H_{out} \in \mathbb{R}^{q \times n \times d}$ is obtained with the output weight matrix $\theta_\circ \in \mathbb{R}^{d_h \times d}$ by:

$$H_{out} = \text{concat}(h_1, \cdots, h_8) \cdot \theta_\circ^\top \tag{7}$$

Cosine Linear Layer After the attention block, two skip connections are performed on $H_{out} = (Z_P + H_{out}) + \text{FFN}(Z_P + H_{out})$ with layer normalization before each step. The feed-forward network FFN is a simple two linear layers with GELU (Gaussian Error Linear Unit) activation [12] between. With the final outcome feature $H_{out} \in \mathbb{R}^{q \times n \times d}$, a linear layers with weight $\theta_{out} \in R^{d \times 1}$ is applied follows by softmax for $P_{out} \in \mathbb{R}^{q \times n}$, which represent the probabilistic prediction for every Query features on n categories. Instead of using a conventional linear layer, we used a cosine linear layer, delivered from [4]. Instead of performing the dot-product between H_{out} and θ_{out}, cosine similarity is replaced between two L2-normalized tensors, using Equation (3). The replacement of cosine similarity instead of the convention dot-product operation helps us achieve a better prediction score for P_{out}. Overall, the probabilistic prediction $\boldsymbol{p}(c|\mathbf{h}_{q,c};\ \theta_{out})$ for representation score $\mathbf{h}_{q,c}$ on label c for query sample $\boldsymbol{x}_q$ over n categories and

the predicted label $\hat{y}$ are calculated by:

$$\boldsymbol{p}(c|\mathbf{h}_{q,c},\ \theta_{out}) = \frac{exp[S_C(\mathbf{h}_{q,c},\ \theta_{out}^{\top})]}{\sum_{i=1}^{n} exp[S_C(\mathbf{h}_{q,i},\ \theta_{out}^{\top})]},\ \hat{y} = \arg\max_{c}\ \boldsymbol{p}(c|\mathbf{h}_{q,c},\ \theta_{out}) \quad (8)$$

3.4 Episodic Training for FS-CT

We training our FS-CT method with an episodic learning strategy. For each training step, task $\mathcal{T} = \{S,\ Q\}$ are selected randomly from the base set D_{base} which share the same N categorial labels. For convenience, all learning parameters are referred to as a unified parameter θ. With FS-CT, includes backbone feature extraction $f(.|\theta_f)$, is performed, we obtain the probabilistic prediction $p(y|\boldsymbol{x}_Q, S; \theta)$ of sample $x_Q \in Q$ on label y given S. Finally, Categorical Cross-Entropy loss is applied to update parameters at the end of each training step by Equation (1). Details of the training strategy are presented in Algorithm 1:

Algorithm 1: FS-CT Episodic Training per epoch

Input : Training set D_{base}. Number of learning tasks $N_{\mathcal{T}}$. Learnable parameters θ

Output : Updated parameters θ

for t in $\{1, ..., N_{\mathcal{T}}\}$ **do**

- Randomly chosen task $\mathcal{T}_t = \{S,\ Q\} \sim p(D_{\text{base}})$.
- Obtain feature representations $Z_S,\ Z_Q$ from S and Q by $f(.|\theta_f)$
- Calculate the proto-representation Z_P using Equation (2)
- Obtain $\langle \mathbf{q}^*,\ \mathbf{k},\ \mathbf{v} \rangle$ for attention by:
  $\mathbf{q}^* \leftarrow Z_P \cdot \theta_{q^*}^{\top};\ \mathbf{k} \leftarrow Z_Q \cdot \theta_k^{\top};\ \mathbf{v} \leftarrow Z_Q \cdot \theta_v^{\top}$
- Calculate H_{out} by Cosine Attention by Equations (6) and (7)
- Perform two skip-connections:
  $H_{out} \leftarrow (Z_P\ +\ H_{out})\ +\ \text{linear}(\text{GELU}[\text{linear}(Z_P\ +\ H_{out})])$
- Obtain the prediction score for query set P_{out} with Equation (8)
- Compute the Loss function as in Equation (1)
- Perform gradient descent step to update parameters θ

end

4 Experiments

4.1 Datasets

To evaluate our method, we adapts three standardized few-shot image classification datasets *mini-ImageNet* [32], CIFAR-FS [2], and CUB-200 [33]. The

mini-ImageNet dataset is the subset of ImageNet dataset (ILSVRC-2012) [24] that consists of 100 different categories with 600 image samples per each, each image have the size 84×84 pixels. In our implementation, we used the splits by Ravi and Laroche [23] including 64 based (training) categories, 16 validation categories, and 20 testing categories. *CIFAR-FS* including 100 categories containing 600 images for each label with the size of 32×32 pixels. The splits of this dataset is similar to *mini-ImageNet*. The *CUB-200* dataset contains 200 categories of bird species with 11,788 images of 84×84 pixels, which are divided into 100 based categories for training, 50 categories for validating, and 50 novel categories for testing. We resiZe

Fig. 3. Overview of several examples of the Yoga poses dataset

Yoga Poses Dataset Besides using the three standardized datasets above, we also created a custom dataset for Yoga Pose Scoring, including 50 categories of famous Yoga poses with 2,480 images. We developed this small-scale dataset as an initial step for making a smart monitoring and studying scheme for Yoga participants. The dataset is partially derived from Kaggle [27] and storage in our implementation code on GitHub[5]. The number of images are ranging from 30 to 81 samples per category with arbitrary size. We splited the dataset into 25 categories for training, 13 categories for validating, and 12 categories for testing. Some image samples are presented in Figure 3. Details of the dataset distribution are presented in Table 1.

[5] `https://github.com/vinuni-vishc/Few-shot-transformer/tree/main/dataset/Yoga`

Table 1. Yoga Dataset description

	Train	Val	Test	**Total**
Number of Categories	25	13	12	**50**
Min number of samples per category	30	39	31	**30**
Max number of samples per category	80	81	67	**81**
Average of samples per category	50.2	49.9	48.0	**49.6**
Total samples	1,255	649	576	**2,480**

4.2 Implementation details

We implemented our method and conducted experiments on PyTorch, using CPU Intel Core i9-10900X 3.7GHz with a GPU NVIDIA GeForce RTX 3090 24GB and 16GB RAM memory. Methods were trained and experimented with the learning rate 0.001, weight decay 1×10^{-5}, and AdamW [17] optimization function, and no dropout. We performed two configurations: 5-way 5-shot and 5-way 1-shot, with 16 query samples for each category, making a total of 80 queries. All training steps are trained on 50 training epochs with 200 episodic batches (episodes) for each. Each training epoch is followed by a validating step with 200 episodes to select the best-performed model for the testing phase on 600 episodes. All training, validating, and testing sets are disjoined in categories. To increase training data samples for training models, we applied augmentation, including randomly resizing, cropping, horizontal flipping, color jittering, and image normalizing. All experiments were conducted in a same common ground of code, setting, and environment for a fair evaluation and comparison.

For backbone feature extraction, we mainly utilized four backbone models Conv4, Conv6, ResNet-18, and ResNet-32 for the experiments. Conv4 and Conv6 are the lightweight CNN model with 4 and 6 layers respectively, have been used before in [4, 29, 32] that are trained from scratch. On the other hand, the ResNet backbone networks and pretrained on mini-ImageNet are available on Torchvision. However, as the mini-ImageNet is a subset of the ImageNet, we deployed a special pre-trained model named FETI for evaluating the dataset, which will be described in detail later. For each type of backbone architecture, we resized sample images before training depending on the dataset and backbone model. Particularly, for CNN backbones, we resize images into 64×64 pixels for CIFAR-FS (due to its small size originally) and 84 pixels for other datasets, and for Res-Net backbones, the resized input image are 112×112 and 224×224, respectively. For experiments, we use the metric top 1-Accuracy in percentage for evaluating performance of methods on one episodic batch as follows:

$$\text{Acc} = \frac{\text{Number of correct predictions}}{\text{Number of query samples}} * 100 \tag{9}$$

and the total Accuracy is the mean average of all tasks' performance.

4.3 Ablation Study

Table 2. Our proposed method *FS-CT* performances on 5-way classification, using either the standard Softmax Attention [1] or ***our*** improved Cosine Attention. One attention-based few-shot classification method *CTX* [6] is applied for comparison. Augmentation is applied for our FS-CT method only. The best and second best results are **bolded** and underlined, respectively. Evaluation metric is accuracy in percentage. This table is best viewed with color.

Methods	*mini*-ImageNet				CUB-200				CIFAR-FS			
	Conv4		Conv6		Conv4		Conv6		Conv4		Conv6	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
FS-CT + Softmax Attn	42.55	58.08	39.11	56.52	53.44	66.14	55.41	65.87	48.57	66.22	46.66	65.43
FS-CT + Cosine Attn	**45.69**	**60.38**	**40.52**	**59.85**	**56.41**	67.12	55.27	66.90	**53.86**	**69.03**	**51.04**	**68.76**
FS-CT + Softmax Attn + Aug	38.11	57.14	33.89	54.26	51.50	69.73	48.61	71.62	40.95	64.31	43.74	60.41
FS-CT + Cosine Attn + Aug	40.73	60.21	35.33	57.81	56.34	**75.88**	**55.68**	**76.93**	47.85	68.35	45.06	67.40

Methods	ResNet-18		ResNet-34		ResNet-18		ResNet-34		ResNet-18		ResNet-34	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CTX + Softmax Attn	62.37	63.09	65.08	65.66	61.24	62.12	64.28	66.02	41.16	43.05	44.54	48.01
CTX + Cosine Attn	77.21	**92.18**	**82.09**	93.41	73.14	85.71	76.54	88.15	60.18	73.97	61.82	74.83
FS-CT + Softmax Attn	72.36	84.13	73.90	85.58	74.69	85.38	75.97	87.03	63.40	77.98	63.05	78.04
FS-CT + Cosine Attn	75.93	90.12	79.01	91.64	77.72	89.13	77.72	89.63	64.29	80.57	65.67	81.44
FS-CT + Softmax Attn + Aug	73.84	84.65	74.61	88.74	75.89	88.06	77.01	89.05	62.41	78.17	62.14	79.17
FS-CT + Cosine Attn + Aug	**77.40**	91.33	80.32	**93.82**	**81.12**	**91.96**	**81.23**	**92.35**	**64.49**	**81.46**	**67.06**	**82.89**

For experiment, besides FS-CT model, we also deploy another attention-based few-shot learning algorithm CTX [6] for a direct comparison with our FS-CT method. We utilized both Softmax Attention (Equation (5)) and Cosine Attention (Equation (6)) to compare the two attention mechanism and evaluate our proposed Cosine attention method. The main experiment results with mini-ImageNet, CIFAR-FS, AND CUB-200 dataset are presented in Table 2 using four backbones Conv4, Conv6, ResNet-18, and ResNet-32 in 1-shot and 5-shot settings. Note that we use the full ImageNet pre-trained models on ResNet for the mini-ImageNet in this table for a fair comparison in performance between datasets and we will have a separate section for a fair evaluation of this dataset in the next section. For CTX, we only conducted experiments on Res-Net backbones, and data augmentation was applied for FS-CT only.

In general, our FS-CT outperformed CTX in different ResNet backbones, datasets, and few-shot settings. The results are augmented further with Cosine Attention mechanism, with the improved performance are from nearly 5% to over 20% in various scenarios. Overall, *5-shot learning comes with better performance than 1-shot learning*. This normally happens on few-shot algorithms as more label samples come with better centroid representation for individual categories, thus classify queries better. *Using Cosine Attention achieves better performance than Softmax Attention*. With our proposed attention mechanism, the overall accuracy consistently improves across datasets, settings, and backbones. Under many scenarios, FS-CT achieves better performances than CTX using the same backbone and attention block. This highlight the effectiveness of our proposed attention mechanism thats boosting models' performances by removing the effect of magnitude in calculating attention weight. *Augmentation helps improve classifier performances* on FS-CT, mainly with 5-shot setting. There are some occasions that augmentation does not help improve performance in one-shot learning. This could be explained by augmentation that comes with the growth in noise in categorical representation, therefore affecting the performances on one-shot learning. Augmentation seems to be effective on ResNet backbones, as FS-CT using Cosine Attention with augmentation mainly achieves the best result within individual scenarios. Across scenarios, *deeper backbones comes with better performances*, as increasing the number of layers helps both CNN and ResNet backbones achieve higher results. Moreover, the models' performance is heavily affected by the choice of backbone, as in most cases, ResNet-34 feature extractor comes with the highest performance among four backbones.

For the custom Yoga Poses Dataset, the results are presented in table 3. While Cosine Attention still comes with more robust performances than the standard Softmax Attention, augmentation overall seems not to help the method in improving the outcome results. Our theory is that this phenomenon is affected by the quality of the dataset. We are aiming to invest deeper in the dataset and improve its quantity and quality in further study and application in Yoga monitoring.

Performances on mini-ImageNet with non-ovellaped pre-trained model As mini-ImageNet dataset is a subset of the ImageNet dataset, using a pre-trained ResNet on the full ImageNet as feature extraction comes with a natural advanced performance on mini-ImageNet, as shown with very high performance in Table 2. Therefore, we alleviated this issue using a specific pre-trained model that had been trained on an ImageNet subset that was separated to be non-overlapped with the mini-ImageNet test-set. We adapted this pre-trained model, which is called "Feature Extractor Trained (partially) on ImageNet" or FETI in abbreviation, from [1]. As the pre-trained model is built specifically on ResNet-

Table 3. Performance on Yoga Dataset. The best and second best results are **bolded** and underlined, respectively. Evaluation metric is accuracy in percentage

Methods	Yoga Dataset			
	ResNet-18		ResNet-34	
	1-shot	5-shot	1-shot	5-shot
CTX [6] + Softmax Attn	49.83	54.88	53.09	55.44
CTX + Cosine Attn	59.99	76.61	63.12	77.80
FS-CT + Softmax Attn	61.40	69.15	58.89	71.12
FS-CT + Softmax Attn + Aug	51.89	66.21	55.08	70.12
FS-CT + Cosine Attn	**66.38**	**80.34**	**64.32**	77.66
FS-CT + Cosine Attn + Aug	57.98	73.58	59.76	**77.92**

18, we used two backbones models ResNet-18 and ResNet-12 to evaluate our FS-CT method specifically on the mini-ImageNet dataset, as ResNet-32 is too different in layers. Table 4 shows our FS-CT performance on mini-ImageNet with FETI pre-trained model. Model performances are generally reduced in the comparison with results from Table 2, about 20 percent. This further strengthens the importance of a good learned representation in few-shot classification tasks. Still, the correlating performances still remained between using two attention mechanisms, as FS-CT using Cosine Attention still outperformed the standard one.

Table 4. Performance of FPTF models on mini-ImageNet, using a ResNet-12 and ResNet-18 backbone with a pre-trained model FETI (Feature Extractor Trained (partially) on ImageNet) that was trained with non-test-set overlapping ImageNet classes to avoid the natural advantage of ImageNet pre-trained model

Methods	mini-ImageNet			
	ResNet-12		ResNet-18	
	1-shot	5-shot	1-shot	5-shot
FS-CT + Softmax Attn (FETI)	45.70	57.03	40.06	58.20
FS-CT + Softmax Attn + Aug (FETI)	38.88	57.16	47.72	62.71
FS-CT + Cosine Attn (FETI)	**55.62**	70.36	41.84	71.34
FS-CT + Cosine Attn + Aug (FETI)	49.39	**73.00**	**55.67**	**73.42**

5 Limitation Discussion

We have proposed a transformer-based method for a few-shot classification task with an attention mechanism using cosine similarity. While our results are promising in many datasets, there are some limitations. Firstly, the method performances are highly sensitive to the choosing of backbone feature extractors, especially the ones with a pre-trained model. Secondly, the complexity of architecture may hinder FS-CT from reaching better performances. Although skip-connection was applied for preserving information, this might not be enough. There are many methods [4, 29, 32] with simple pipelines yet come with impressive performances in various datasets and we are aiming to pursue that approach in further study. Thirdly, we only considered transductive learning for our method. Finally, due to some limitations in resources, we can only compare directly our proposed Cosine Attention with the scaled dot-product Softmax Attention. There are a variety of attention mechanisms recently, and further studies should be made with a wider scope of problems (not only in the Few-shot classification task) for a comprehensive evaluation of attention mechanisms. We leave these limitations as our future work.

6 Conclusion

In this paper, we proposed a novel approach for few-shot image classification with a transductive learning approach that is heavily influenced by transformer architecture and attention mechanism. Specifically, our approach obtains centroid for individual categories within the Support set as a prototypical representation. We made this process learnable with changeable weights to improve the centroids' representation quality. Then, the two proto-feature and query features are fed to a transformer encoder architecture with few modifications for a few-shot classification. We call our method Few-shot Cosine Transformer (FS-CT). Moreover, we proposed a novel attention mechanism involving cosine similarity called Cosine Attention that removes the effect of magnitude in scaled dot-product attention. We show that our Cosine Attention outperforms the scaled dot-product attention in many scenarios, and the proposed FS-CT produces comparable results on several datasets.

References

1. Bateni, P., Barber, J., van de Meent, J.W., Wood, F.: Enhancing few-shot image classification with unlabelled examples. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2796–2805 (2022)
2. Bertinetto, L., Henriques, J.F., Torr, P.H., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. arXiv preprint arXiv:1805.08136 (2018)

3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
4. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C., Huang, J.B.: A closer look at few-shot classification. In: International Conference on Learning Representations (2019)
5. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34**, 17864–17875 (2021)
6. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. Advances in Neural Information Processing Systems **33**, 21981–21993 (2020)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
9. Finn, C., Xu, K., Levine, S.: Probabilistic model-agnostic meta-learning. Advances in neural information processing systems **31** (2018)
10. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4367–4375 (2018)
11. Han, G., Ma, J., Huang, S., Chen, L., Chang, S.F.: Few-shot object detection with fully cross-transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5321–5330 (2022)
12. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
13. Hou, R., Chang, H., Ma, B., Shan, S., Chen, X.: Cross attention network for few-shot classification. Advances in Neural Information Processing Systems **32** (2019)
14. Lee, Y., Choi, S.: Gradient-based meta-learning with learned layerwise metric and subspace. In: International Conference on Machine Learning. pp. 2927–2936. PMLR (2018)
15. Liu, L., Hamilton, W., Long, G., Jiang, J., Larochelle, H.: A universal representation transformer layer for few-shot image classification. arXiv preprint arXiv:2006.11702 (2020)
16. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. arXiv preprint arXiv:1805.10002 (2018)
17. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
18. Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., Yang, Q.: Cosine normalization: Using cosine similarity instead of dot product in neural networks. In: International Conference on Artificial Neural Networks. pp. 382–391. Springer (2018)
19. Munkhdalai, T., Sordoni, A., Wang, T., Trischler, A.: Metalearned neural memory. Advances in Neural Information Processing Systems **32** (2019)

20. Munkhdalai, T., Yu, H.: Meta networks. In: International Conference on Machine Learning. pp. 2554–2563. PMLR (2017)
21. Qiao, L., Shi, Y., Li, J., Wang, Y., Huang, T., Tian, Y.: Transductive episodic-wise adaptive metric for few-shot learning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3603–3612 (2019)
22. Qiao, S., Liu, C., Shen, W., Yuille, A.L.: Few-shot image recognition by predicting parameters from activations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7229–7238 (2018)
23. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: In International Conference on Learning Representations (ICLR) (2017)
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
25. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. arXiv preprint arXiv:1807.05960 (2018)
26. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International conference on machine learning. pp. 1842–1850. PMLR (2016)
27. Saxena, S.: Yoga pose image classification dataset, [Last accessed on 2 August 2022]. Available online: `https://www.kaggle.com/shrutisaxena/yoga-pose-image-classification-dataset`
28. Shao, S., Xing, L., Wang, Y., Xu, R., Zhao, C., Wang, Y., Liu, B.: Mhfc: Multi-head feature collaboration for few-shot learning. In: Proceedings of the 29th ACM international conference on multimedia. pp. 4193–4201 (2021)
29. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
30. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1199–1208 (2018)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
32. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)
33. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. California Institute of Technology (2011)
34. Ye, H.J., Hu, H., Zhan, D.C., Sha, F.: 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 (2020)
35. Zhang, X., Meng, D., Gouk, H., Hospedales, T.M.: Shallow bayesian meta learning for real-world few-shot recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 651–660 (2021)