Adaptive Poincaré Point to Set Distance for Few-Shot Classification

Rongkai Ma1, Pengfei Fang2,3*, Tom Drummond4, Mehrtash Harandi1,3

1Monash University
2The Australian National University
3DATA61-CSIRO, Australia
4The University of Melbourne
{rongkai.ma, mehrtash.harandi}@monash.edu, pengfei.fang@anu.edu.au, tom.drummond@unimelb.edu.au

Abstract
Learning and generalizing from limited examples, i.e., few-shot learning, is of core importance to many real-world vision applications. A principal way of achieving few-shot learning is to realize an embedding where samples from different classes are distinctive. Recent studies suggest that embedding via hyperbolic geometry enjoys low distortion for hierarchical and structured data, making it suitable for few-shot learning. In this paper, we propose to learn a context-aware hyperbolic metric to characterize the distance between a point and a set associated with a learned set to set distance. To this end, we formulate the metric as a weighted sum on the tangent bundle of the hyperbolic space and develop a mechanism to obtain the weights adaptively, based on the constellation of the points. This not only makes the metric local but also dependent on the task in hand, meaning that the metric will adapt depending on the samples that it compares. We empirically show that such metric yields robustness in the presence of outliers and achieves a tangible improvement over baseline models. This includes the state-of-the-art results on five popular few-shot classification benchmarks, namely mini-ImageNet, tiered-ImageNet, Caltech-UCSD Birds-200-2011 (CUB), CIFAR-FS, and FC100.

Figure 1: (a): For a query sample in “red” class, outliers (i.e., yellow and white circles) drag the prototype (i.e., the black circle) far away from the real cluster center in the embedding space such that the nearest neighbor classifier mis-classifies the query point into “green” class. (b): Our method computes an adaptive point to set distance on the manifold, which is more robust to outliers than prototypes. Best viewed in color.

In many cases, FSL methods deem to learn an embedding space to distinguish samples from different classes. Therein, the embedding space is a multidimensional Euclidean space and is realized via a deep neural network.

Employing hyperbolic geometry to encode data has been shown rewarding, as the volume of space expands exponentially (Ganea, Bécigneul, and Hofmann 2018; Khrulkov et al. 2020). Recent works have shown that a hierarchical structure exists within visual datasets and that the use of hyperbolic embeddings can yield significant improvements over Euclidean embeddings (Khrulkov et al. 2020; Fang, Harandi, and Petersson 2021).

Most existing FSL solutions learn a metric through comparing the distance between a query sample and the class prototypes, often modeled as the mean embeddings of each class. However, this does not take the adverse effects of outliers and noises into consideration (Sun et al. 2019). This severely limits the representation power of embedding-based methods since the outliers may drag the prototype away from the true center of the cluster (see Fig. 1(a)). For a more robust approach, we require an adaptive metric, which can faithfully capture the distribution per class, while being robust to outliers and other nuances in data (Fig. 1(b)).

With this in mind, we propose learning a context-
aware hyperbolic metric that characterizes the point to set (dis)similarities. This is achieved through employing a Poincaré ball to model hyperbolic spaces and casting the (dis)similarity as a weighted-sum between a query and a class that is learned adaptively. In doing so, each sample (from the support and query sets) is modeled by a set itself (i.e., a feature map). Therefore, we propose to make use of pairwise distances between elements of two sets, along with a refinement mechanism to disregard uninformative parts of the feature maps. This leads to a flexible and robust framework for the FSL tasks. We summarize our contributions as follows:

- We propose a novel adaptive Poincaré point to set (APP2S) distance metric for the FSL task.
- We further design a mechanism to produce a weight, dependent on the constellation of the point, for our APP2S method.
- We conduct extensive experiments across five FSL benchmarks to evaluate the effectiveness of the proposed method.
- We further study the robustness of our method, which shows our method is robust against the outliers compared to competing baselines.

Preliminaries

In what follows, we use $\mathbb{R}^n$ and $\mathbb{R}^{m \times n}$ to denote the $n$-dimensional Euclidean space and space of $m \times n$ real matrices, respectively. The $n$-dimensional hyperbolic space is denoted by $\mathbb{H}^n$. The $\text{arctanh} : (-1, 1) \to \mathbb{R}$, $\text{arctanh}(x) = \frac{1}{2} \ln(\frac{1+x}{1-x})$, $|x| < 1$ refers to the inverse hyperbolic tangent function. The vectors and matrices (or 3-D tensors) are denoted by bold lower-case letters and bold upper-case letters throughout the paper.

Riemannian Geometry

In this section, we will give a brief recap of Riemannian geometry. A manifold, denoted by $\mathcal{M}$, is a curved surface, which locally resembles the Euclidean space. The tangent space at $x \in \mathcal{M}$ is denoted by $T_x \mathcal{M}$. It contains all possible vectors passing through point $x$ tangentially. On the manifold, the shortest path connecting two points is a geodesic, and its length is used to measure the distances on the manifold.

Hyperbolic Space

Hyperbolic spaces are Riemannian manifolds with constant negative curvature and can be studied using the Poincaré ball model (Ganea, Bécigneul, and Hofmann 2018; Khrulkov et al. 2020). The Poincaré ball ($\mathbb{D}^n_c, g^c$) is a smooth $n$-dimensional manifold identified by satisfying $\mathbb{D}^n_c = \{x \in \mathbb{R}^n : c||x|| < 1, c \geq 0\}$, where $c$ is the absolute value of the curvature for a Poincaré ball, while the real curvature value is $-c$. The Riemannian metric $g^c$ at $x$ is defined as $g^c = \lambda_x^c g^F$, where $g^F$ is the Euclidean metric tensor and $\lambda_x^c$ is the conformal factor, defined as:

$$\lambda_x^c := \frac{2}{1 - c||x||^2}.$$  (1)

Since the hyperbolic space is a non-Euclidean space, the rudimentary operations, such as vector addition, cannot be applied (as they are not faithful to the geometry). The Möbius gyrovector space provides many standard operations for hyperbolic spaces. Essential to our developments in this work is the Möbius addition of two points $x, y \in \mathbb{D}^n_c$, which is calculated as:

$$x \oplus_c y = \frac{(1 + 2c(x, y) + c||y||^2)x + (1 - c||x||^2)y}{1 + 2c(x, y) + c^2||x||^2||y||^2}.$$  (2)

The geodesic distance between two points $x, y \in \mathbb{D}^n_c$ can be obtained as:

$$d_c(x, y) = \frac{2}{\sqrt{c}} \text{arctanh}(\sqrt{c}||x \oplus_c y||).$$  (3)

Another essential operation used in our model is the hyperbolic averaging. The counterpart of Euclidean averaging in hyperbolic space is the Einstein mid-point which has the most simple form in Klein coordinates (another model of the hyperbolic space which is isometric to the Poincaré ball). Thus, we transform the points from Poincaré (i.e., $x_{2s}$) ball model to Klein model (i.e., $x_k$) using the transformation:

$$x_k = \frac{2x_{2s}}{1 + c||x_{2s}||^2}.$$  (4)

Then, the hyperbolic averaging in Klein model is obtained as:

$$\text{HypAve}(x_1, \ldots, x_N) = \sum_{i=1}^{N} \gamma_i x_i / \sum_{i=1}^{N} \gamma_i,$$  (5)

where $\gamma_i = \frac{1}{\sqrt{1-c||x_i||^2}}$ are the Lorentz factors. Finally, we transform the coordinates back to Poincaré model using:

$$x_{2s} = \frac{x_k}{1 + \sqrt{1 - c||x_k||^2}}.$$  (6)

In our work, we make use of the tangent bundle of the $\mathbb{D}^n_c$. The logarithm map defines a function from $\mathbb{D}^n_c \to T_x \mathbb{D}^n_c$, which projects a point in the Poincaré ball onto the tangent space at $x$, as:

$$\pi_x^c(y) = \frac{2}{\sqrt{c}\lambda_x^c} \text{arctanh}(\sqrt{c}||x \oplus_c y||) - \frac{x \oplus_c y}{||x \oplus_c y||}.$$  (7)

Point to Set Distance

Let $S = \{s_1, \ldots, s_b\}$ be a set. The distance from a point $p$ to the set $S$ can be defined in various forms. The min and max distance from a point $p$ to the set $S$ are two simple metrics, which can be defined as:

$$d^i_{2s}(p; S) = \inf \{d(p, s_i) | s_i \in S\},$$  (8)

$$d^h_{2s}(p; S) = \sup \{d(p, s_i) | s_i \in S\},$$  (9)

where $\inf$ and $\sup$ are the infimum and supremum functions, respectively. Given their geometrical interpretation, $d^i_{2s}$ and $d^h_{2s}$ define the lower and upper pairwise bounds, and fail to encode structured information about the set. Therefore, we opt for a weighted-sum formalism to measure the distance between a point and a set in $S$.
Method

This section will give an overview of the proposed method, followed by a detailed description of each component in our model.

Problem Formulation

We follow the standard protocol to formulate few-shot learning (FSL) with episodic training. An episode represents an \( N \)-way \( K \)-shot classification problem (i.e., the training set, named support set, includes \( N \) classes where each class has \( K \) examples). As the name implies, \( K \) (i.e., the number of examples per class) is small (e.g., \( K = 1 \) or 5). The goal of learning is to realize a function \( \mathcal{F} : \mathcal{X} \to \mathbb{R}^n \) to embed the support set to a latent and possibly lower-dimensional space, such that query samples can be recognized easily using a nearest neighbour classifier. To be specific, an episode or task \( \mathcal{E}_i \) consists of a query set \( \mathcal{X}^q = \{(X^q_i, y^q_i)\}_{i=1}^{N} \), where \( X^q_i \) denotes a query example sampled from class \( y^q_i \), and a support set \( \mathcal{X}^s = \{(X^s_{ij}, y^s_{ij})\}_{i=1}^{N} \), where \( X^s_{ij} \) denotes the \( j \)-th sample in the class \( y^s_{ij} \). The embedding methods for FSL, our solution being one, often formulate training as:

\[
\mathcal{F}^* := \arg\min_{\mathcal{F}} \sum_{X^q_i \in \mathcal{X}^q} \delta(\mathcal{F}(X^q_i), \mathcal{F}(X^s_i)) \quad \text{s.t.} \quad y^q_i = y^s_{ij},
\]

where \( \delta \) measures a form of distance between the query and the support samples.

Model Overview

We begin by providing a sketch of our method (see the conceptual diagram in Fig. 2(a) and Fig. 2(b)). The feature extractor network, denoted by \( \mathcal{F} \), maps the input to a hyperbolic space in our work. We model every class in the support set by its signature. The signature is both class and episodic-aware, meaning that the signature will vary if the samples of a class or samples in the episodic vary. This will enable us to calculate an adaptive distance from the query point to every support-class while being vigilant to the arrangement and constellation of the support samples. We stress that our design is different from many prior works where class-specific prototypes are learned for FSL. For example, in (Khrulkov et al. 2020; Snell, Swersky, and Zemel 2017; Sung et al. 2018), the prototypes are class-specific but not necessarily episodic-aware.

To obtain the signatures for each class in the support set, we project the support samples onto the tangent space of the query point and feed the resulting vectors to a signature generator \( f_s \). The signature generator realizes a permutation-invariant function and refines and summarizes its inputs into a signature per class. We then leverage a relational network \( f_r \) to contrast samples of a class against their associated signature and produce a relational score. To obtain the adaptive P2S distance, we first compute a set to set (S2S) distance between the query feature map and each support feature map using the distance module \( f_c \). Moreover, a weighted-sum is calculated using the relational score acting as the weight on the corresponding S2S distance, which serves as the P2S distance.

Given P2S distances, our network is optimized by minimizing the adaptive P2S distance between the query and its corresponding set while ensuring that the P2S distance to other classes (i.e., wrong classes) is maximized.

Adaptive Poincaré Point to Set Distance

In FSL, we are given a small support set of \( K \) images, \( \mathcal{X}^s_i = \{X^s_{ij}\}_{j=1}^{N} \) per class \( y^s_i \) to learn a classification model. We use a deep neural network to first encode the input to a multi-channel feature map, as \( \mathcal{S}_i = \mathcal{F}(\mathcal{X}^s_i) \), with \( \mathcal{S}_i = \{S^1_{ij}, \ldots, S^K_{ij} \mid s^r_{ij} \in \mathbb{R}^{H \times W \times C} \} \), where \( H, W, \) and \( C \) indicate the height, width, and channel size of the instance feature map. Each feature map consists of a set of patch descriptors (local features), which can be further represented as \( \mathcal{S}_i = \{s^1_{ij}, \ldots, s^K_{HW} \mid s^r_{ij} \in \mathbb{R}^C \} \).

In our work, we train the network to embed the representation in the Poincaré ball; thus, we need to impose a constraint on patch descriptors at each spatial location \( s^r_{ij} \) as follows:

\[
s^r_{ij} = \begin{cases} s^1_{ij} & \text{if } ||s^r_{ij}|| \leq \mu \\ \mu s^1_{ij}/||s^r_{ij}|| & \text{if } ||s^r_{ij}|| > \mu, \end{cases}
\]

where \( \mu \) is the norm upper bound of the vectors in the Poincaré ball. In our model, we choose \( \mu = (1 - \epsilon)/c \), where \( c \) is the curvature of the Poincaré ball and \( \epsilon \) is a small value that makes the system numerically stable. The same operation applies to the query sample, thereby obtaining an instance feature map for the query sample \( \mathcal{Q} = \{q^1, \ldots, q^{HW}\} \). Then the P2S distance between the query sample \( \mathcal{Q} \) and the support set per class \( \mathcal{S}_i \) can be calculated using Eq. (8) or Eq. (9). However, those two metrics only determine the lower or upper bound of P2S distance, thereby ignoring the structure and distribution of the set to a great degree. To make better use of the distribution of samples in a set, we propose the adaptive P2S distance metric as:

\[
d^\text{adp}_{P2S}(\mathcal{Q}, \mathcal{S}_i) := \frac{\sum_{j=1}^{K} w_{ij} d(\mathcal{Q}, \mathcal{S}_j)}{\sum_{j=1}^{K} w_{ij}},
\]

where \( w_{ij} \) is the adaptive factor for \( d(\mathcal{Q}, \mathcal{S}_j) \). We refer to the distance in Eq. (12) as Adaptive Poincaré Point to Set (APP2S) distance, hereafter.

In Eq. (12), we need to calculate the distance between two feature maps (i.e., \( d(\mathcal{Q}, \mathcal{S}_j) \)). In doing so, we formulate a feature map as a set (i.e., \( \{q^1, \ldots, q^{HW}\} \) and \( \{s^1_{ij}, \ldots, s^K_{HW}\} \)), such that a set to set (S2S) distance can be obtained. One-sided Hausdorff and two-sided Hausdorff distances (Huttenlocher, Klanderman, and Rucklidge 1993) are two widely used metrics to measure the distance between sets. However, these two metrics are sensitive to
Our proposed set-signature generator \( f_\zeta \) is similar to the set-to-set function in FEAT (Ye et al. 2020), in the sense that both functions perform self-attention over the input features. However, the fundamental difference is that our module exploits the relation between the spatial feature descriptors of all samples in a support set (e.g., \( \tilde{s}_{ij} \)), instead of prototypes as proposed in FEAT (Ye et al. 2020), which possibly gives the model more flexibility to encode meaningful features.

Given sample features in a class \( \tilde{s}_i = \{ \tilde{s}_{i1}, \ldots, \tilde{s}_{iK} \} \) and the corresponding class signature \( \tilde{s}_i \), we use a relation generator (i.e., \( f_\phi \) in Fig. 2(a)) to compare the relationship between an individual feature map and the class signature. In doing so, we first concatenate the individual feature maps and their class signature along the channel dimension to obtain a hybrid representation, as:

\[
G_{ij} = \text{CONCAT}(\tilde{s}_{ij}, \tilde{s}_i).
\]

Given the hybrid representation \( G_{ij} \), the relation generator produces a relation score as:

\[
w_{ij} = f_\phi(G_{ij}).
\]

This score
Remark 2: The point to set distance defined by Eq. (12) is different from that in MatchingNet (Vinyals et al. 2016). MatchingNet formulates all the samples in the support set as a set. In contrast, we treat the samples in a class as the set, which makes our adaptive point to set distance fully contextual aware of the whole support set (by the set-signature) and encodes the distribution of each class.

Algorithm 1: Train network using adaptive Poincaré point to set distance

**Input:** An episodes \( E \), with their associated support set \( \mathcal{X}^s = \{ (X^s_i, y^s_i) \mid i = 1, \ldots, N \} \) and \( \mathcal{X}^q \) and a query sample \( X^q \)

**Output:** The optimal parameters for \( \mathcal{F}, f_\omega, f_\xi, \) and \( f_\phi \)

1. Map \( \mathcal{X}^s \) and \( \mathcal{X}^q \) into Poincaré ball
2. Obtain the tangent support set \( S \) using Eq. (7)
3. \( \tilde{S} = f_\omega(S) \) \( \triangleright \) the refined support set
4. for \( i \in \{1, \ldots, N\} \) do
5. \( \tilde{S}_i = \sum_{j=1}^{K} \tilde{S}_{ij} / K \) \( \triangleright \) the set signature
6. \( G_{ij} = \text{CONCAT}(\tilde{S}_{ij}, \tilde{S}_i) \) \( \triangleright \) the hybrid representation
7. \( \omega_{ij} = f_\phi(G_{ij}) \) \( \triangleright \) the weight
8. Compute point to set distance and set to set distance using Eq. (12) and Eq. (13)
9. end for
10. Optimize the model using Eq. (10)

**Related Work**

In this section, we discuss the literature on few-shot learning and highlight those that motivate this work. Generally, there are two main branches on the few-shot learning literature, optimization-based and metric-based methods. The optimization-based methods (Antoniou, Edwards, and Storkey 2019; Chen et al. 2019; Finn, Abbeel, and Levine 2017; Flennerhag et al. 2019; Franceschi et al. 2018; Nichol, Achiam, and Schulman 2018), such as MAML and Reptile (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018), aim to learn a set of initial model parameters that can adapt to new tasks quickly using backpropagation in the episodic regime, without severe overfitting. However, this group of methods usually adopt a bi-level optimization setting to optimize the initial parameters, which is computationally expensive during inference.

On the other hand, our proposed method is closer to metric-based methods (Simon et al. 2020; Snell, Swersky, and Zemel 2017; Sung et al. 2018; Vinyals et al. 2016; Ye et al. 2020; Zhang et al. 2020; Tang et al. 2020; Ma et al. 2021), which target to realize an embedding: \( \mathbb{R}^M \rightarrow \mathbb{R}^D \) to represent images in semantic space equipped with an appropriate distance metric such that different categories are distinctive. Matching Network (Vinyals et al. 2016) determines the query labels by learning a sample-wise distance along with a self-attention mechanism that produces a fully contextualized embedding over samples. Prototypical Network (Snell, Swersky, and Zemel 2017) takes a step further from a sample-wise to a class-wise metric, where all the samples of a class are averaged into a prototype to represent the class in the embedding space. Relation Network (Sung et al. 2018) and CTM (Li et al. 2019a) replace the handcrafted metric with a network to encode the non-linear relation between the class representations and the query embedding. Ye et al. (Ye et al. 2020) propose adopting a transformer to learn the task-specific features for few-shot learning. Zhang et al. (Zhang et al. 2020) adopt the Earth Mover’s Distance as a metric to compute a structural distance between representation to obtain the labels for the query images. Simon et al. (Simon et al. 2020) propose to generate a dynamic classifier via using subspace. Along this line of research, most of the previous methods utilize the global feature vectors as representations. However, several recent works have demonstrated that utilizing the local feature maps can further boost performance. Therefore, we follow these works (Doersch, Gupta, and Zisserman 2020; Zhang et al. 2020; Wertheimer, Tang, and Hariharan 2021; Lifchitz et al. 2019; Li et al. 2019b) to develop our model.

However, the majority of the aforementioned metric-based works employ various metrics within Euclidean space. Ganea et al. (Ganea, Bécigneul, and Hofmann 2018) have proved that embedding via hyperbolic geometry enjoys low distortion for hierarchical and structured data (e.g., trees) and developed the hyperbolic version of the feed-forward neural networks and recurrent neural networks (RNN). Moreover, a recent work (Khrulkov et al. 2020) has shown that the vision tasks can largely benefit from hyperbolic embeddings, which inspires us to further develop algorithms with hyperbolic geometry.
### Experiments

#### Datasets

In this section, we will empirically evaluate our approach across five standard benchmarks, i.e., mini-ImageNet (Ravi and Larochelle 2016), tiered-ImageNet (Ren et al. 2018), Caltech-UCSD Birds-200-2011 (CUB) (Wah et al. 2011), CIFAR-FS (Bertinetto et al. 2018) and Few-shot-CIFAR100 (FC100) (Oreshkin, López, and Lacoste 2018). Full details of the datasets and implementation are described in the supplementary material. In the following, we will briefly describe our results on each dataset.

#### Main Result

We evaluate our methods for 100 epochs, and in each epoch, we sample 100 tasks (episodes) randomly from the test set, for both 5-way 1-shot and 5-way 5-shot settings. Following the standard protocol (Simon et al. 2020), we report the mean accuracy with 95% confidence interval.

**mini-ImageNet.** As shown in Table 1, we evaluate our model using ResNet-12 and ResNet-18 as the backbones on mini-ImageNet. Between them, ResNet-12 produces the best results. In addition, our model also outperforms recent state-of-the-art models in most of the cases. Interestingly, our model further outperforms hyperbolic ProtoNet by 7.77% and 7.11% for 5-way 1-shot and 5-way 5-shot with ResNet-18, respectively. With ResNet-12, we outperform the hyperbolic ProtoNet by 5.60% and 7.29% for 5-way 1-shot and 5-way 5-shot, respectively.

**tiered-ImageNet.** We further evaluate our model on tiered-ImageNet with ResNet backbones. The results in Table 1 indicate that with ResNet-12, our model outperforms the hyperbolic ProtoNet by 4.62% and 7.12% for 5-way 1-shot and 5-way 5-shot, respectively, and achieves state-of-the-art results for inductive few-shot learning.

**CIFAR-FS and FC100.** As the results in Table 2 suggested, our model also achieves comparable performance with the relevant state-of-the-art methods on this dataset, with ResNet-12 backbone, which vividly shows the superiority of our method.

**CUB.** We use ResNet-18 as our backbone to evaluate our method on the CUB dataset. Table 3 shows that our model improves the performance over baseline by 3.94% and 4.88% for 5-way 1-shot and 5-way 5-shot settings, respectively. Besides, our model achieves 77.64% and 90.43% for 5-way 1-shot and 5-way 5-shot settings on this dataset, which outperforms state-of-the-art models (i.e., DeepEMD (Zhang et al. 2020) and P-transfer (Shen et al. 2021)) and achieve competitive performance on this dataset.

#### Robustness to Outliers

To further validate the robustness of our method, we conduct experiments in the presence of outliers in the form of mislabelled images. In the first study, we add a various number of outliers (e.g., 1, 2, 3, 4), whose classes are disjoint to the support-class, to each class of the support set. We performed this study with ResNet-12 backbone on the 5-way 5-shot setting on tiered-ImageNet. Fig. 4(a) shows that the performances of hyperbolic ProtoNet degrade remarkably.

On the contrary, both our APP2S and Euclidean AP2S are robust to outliers, which shows the superiority of our adaptive metric. Comparing to Euclidean AP2S, APP2S is even more robust (see the slope of Fig. 4(a)) and performs consistently even in the presence of 20 outliers. This suggests that integrating our proposed adaptive metric and hyperbolic geometry can further bring robustness to our framework. In the second study (shown in Fig. 4(b)), we conduct the same experiments on mini-ImageNet. The results show a similar trend as the previous one, which further proves the effectiveness of our proposed method.

#### Ablation Study

We further conduct the ablation study to verify the effectiveness of each component in our method on the tiered-ImageNet dataset using the ResNet-12 backbone.

**Experiments Set-Up.** For setting (ii) in Table 4, we disable the relation module $f_\Omega$ and signature generator $f_\sigma$. The P2S distance can be obtained by Eq. (12) and Eq. (13) with equal weights (i.e., 1). Moreover, we enable the relation generator $f_\Omega$ but not the signature generator in setting (iii). We use the class prototype instead of the signature for this experiment.

We disable both $f_\Omega$ and $f_\sigma$ and use the Euclidean distances for setting (iv). In the end, we enable the Poincaré ball but disable the $f_\Sigma$ for setting (v). In terms of implementation of (v), the backbone is designed to output a feature vector instead of a feature map, such that the P2S distance can be directly computed by Eq. (3) and Eq. (12).

**Effectiveness of Point to Set Distance.** In this experiment, we first evaluate the effectiveness of the P2S distance by comparing to its point to point (P2P) distance counterpart (i.e., hyperbolic ProtoNet). From Table 4, we could observe that the P2S distance can learn a more discriminative embedding space than P2P distance (i.e., (i) vs. (ii)), and the adaptive P2S can further bring performance gain to our application (i.e., (ii) vs. (iii)). This observation shows the potential of our P2S distance setting in the FSL task.

**Effectiveness of Signature Generator.** We further evaluate another essential component in our work, i.e., the signature generator, which refines the entire support set and produces a signature per class. As shown in Table 4 (i.e., (iii) and (vi)),
Effectiveness of Hyperbolic Geometry. We also implement our model in the Euclidean space to verify the effectiveness of our method. The row (iv) and (vi) in Table 4 vividly show that the representation in the Poincaré ball has a richer embedding than that in Euclidean spaces.

Effectiveness of Set to Set Distance. The comparison between (v) and (vi) shows that our set to set distance generator associated with the feature map outputs richer information than using a feature vector to directly compute the APP2S.

Table 4: The ablation study of our model, we start from the hyperbolic ProtoNet (Khrulkov et al. 2020) towards APP2S.

**Conclusion**

In this paper, we propose a novel adaptive Poincaré point to set (APP2S) distance metric for the few-shot learning, which can adapt depending on the samples at hands. Empirically, we showed that this approach is expressive with both hyperbolic geometry and Euclidean counterpart. Our model improves the performances over baseline models and achieves competing results on five standard FSL benchmarks.
References


