HyP$^2$ Loss: Beyond Hypersphere Metric Space for Multi-label Image Retrieval

Chengyin Xu$^\dagger$
SIGS, Tsinghua University

Zenghao Chai$^\dagger$
SIGS, Tsinghua University

Zhengzhuo Xu$^\dagger$
SIGS, Tsinghua University

Chun Yuan$^\dagger$
SIGS, Tsinghua University

Yanbo Fan$^\dagger$
Tencent AI Lab

Jue Wang
Tencent AI Lab

Figure 1: Retrieval comparisons to previous methods [21, 57] and ours on the Flickr-25k [19] dataset (top) and NUS-WIDE [8] dataset (bottom). The Top-10 images are returned according to the Hamming distance between the query image and the database. The number below each retrieved image indicates the matched categories, green (red) box indicates matched categories ≥75% (≤25%). The proposed HyP$^2$ Loss outperforms state-of-the-art [21, 57] with fewer contradictions and misclassified results.

ABSTRACT

Image retrieval has become an increasingly appealing technique with broad multimedia application prospects, where deep hashing serves as the dominant branch towards low storage and efficient retrieval. In this paper, we carried out in-depth investigations on metric learning in deep hashing for establishing a powerful metric space in multi-label scenarios, where the pair loss suffers high computational overhead and converge difficulty, while the proxy loss is theoretically incapable of expressing the profound label dependencies and exhibits conflicts in the constructed hypersphere space. To address the problems, we propose a novel metric learning framework with Hybrid Proxy-Pair Loss (HyP$^2$ Loss) that constructs an expressive metric space with efficient training complexity w.r.t. the whole dataset. The proposed HyP$^2$ Loss focuses on optimizing the hypersphere space by learnable proxies and excavating data-to-data correlations of irrelevant pairs, which integrates sufficient data correspondence of pair-based methods and high-efficiency of proxy-based methods. Extensive experiments on four standard multi-label benchmarks justify the proposed method outperforms the state-of-the-art, is robust among different hash bits and achieves significant performance gains with a faster, more stable convergence speed. Our code is available at https://github.com/JerryXu0129/HyP2-Loss.

CCS CONCEPTS

• Computing methodologies → Image representations; • Information systems → Top-k retrieval in databases.

KEYWORDS

Image Retrieval, Deep Hashing, Multi-label, Metric Learning
1 INTRODUCTION

The past decades have witnessed the arrival of the era of big data, floods of images are uploaded to social platforms and search engines day and night, which calls for more efficient and accurate image retrieval in multimedia applications [9, 13, 28, 45, 52]. Real-world images typically contain more than one attribute. Hence, the multi-label retrieval task [28, 37] serves as the crucial and more challenging branch in large-scale image retrieval.

Hashing techniques [10, 56] are widely used to accelerate retrieval due to the low storage and computation costs. The retrieval system can utilize an efficient bit-wise XOR operation to estimate the distance between hash code pairs. Following the prosperous progress of Deep Neural Networks (DNNs) in visual recognition, deep hashing [5, 53] has achieved glary attention and become one of the most substantial research topics in the image retrieval community [6, 47, 48]. The target of deep hashing is to project numerous samples into the hyper metric space and then convert them into compact binary codes through hash functions. The parameterized networks are optimized such that semantically similar data (i.e., images with the same categories) are well clustered and distributed in the established metric space. Such quality and expressiveness of the metric space are optimized through elaborate-designed loss functions in a supervised manner during metric learning.

Pair-based methods [18, 24, 25, 57, 58] are predominant in multi-label retrieval, which directly consider data-to-data connections in a mini-batch. However, such approaches prohibitively confront high training complexity that require square [3, 7, 15, 16, 57] or even higher [40, 42, 43] complexity w.r.t. the number of training samples. Furthermore, the data-to-data correlations in a mini-batch could deteriorate the robustness and degrade the learned metric space because of the increasing overfitting risks and training instability.

Compared to the pair-based methods, proxy-based methods [21, 34] are more efficient and effective in single-label retrieval to embed samples into a proxy-centered hyperspace. However, in multi-label scenarios, we observe and theoretically prove that they are limited to expressing profound correlations. With the exponentially increasing combination growth among labels, the inclusive (i.e., relevant categories) and exclusive (i.e., irrelevant categories) relations cannot get well established supervised by proxy loss. Hence, proxy-based methods exhibit conflicts (see Fig. 2) and are unsatisfactory in such scenarios.

To overcome the above weaknesses, we propose Hybrid Proxy-Pair Loss (HyP$^2$ Loss) to embed samples into expressive hyperspace to establish abundant label correlations with an efficient training complexity. Concretely, we conceive the first part of HyP$^2$ Loss by setting the learnable proxy for each category such that the established metric space roughly clusters samples of similar categories. Note that simply adding the pair loss will introduce overwhelmed training complexity and is fruitless to performance gains. We creatively design additional irrelevant pair constraints as the second part to compensate for the missing multi-label data-to-data correlations, such that enables HyP$^2$ Loss to alienate irrelevant samples to avoid attribute conflicts effectively. Finally, we design the overall loss function with the learning algorithm that ensures more efficient training complexity than pair-based methods, since our predominant part during training is linear correlated to the total training dataset. The elaborate-designed loss functions and training framework guarantee that the established metric space contains expressive multi-label correlations.

To justify the effectiveness and efficiency of our framework, we conduct comprehensive experiments on four multi-label benchmark datasets, i.e., Flickr-25k [19], VOC-2007 [11], VOC-2012 [11], and NUS-WIDE [8]. Compared to existing state-of-the-art [17, 18, 55, 57], the proposed method outperforms existing techniques among different hash hits and backbones quantitatively and qualitatively with better convergence speed and stability. Additionally, in-depth ablation studies and visualization results justify the effectiveness of our mechanism and the designed loss function.

To summarize, our main contributions are three-fold:

- We prove that the hypersphere metric space established by existing proxy-based methods is limited to expressing the profound inclusive and exclusive relations in multi-label scenarios. To the best of our knowledge, this is the first work to theoretically analyze the upper bound of distinguishable hypersphere number in metric space in multi-label image retrieval task.
- We propose HyP$^2$ Loss, a novel loss function that integrates the efficient time complexity of proxy-based methods and strong data correlations of pair-based methods. Particularly, the elaborate-designed multi-label proxy loss, irrelevant pair loss, and overall learning framework contribute well to embedding multi-label images into expressive metric space.
- We conduct extensive experiments on four benchmarks to demonstrate the superiority and robustness of our proposed HyP$^2$ Loss, which is also feasible to various deep hashing methods and backbones. HyP$^2$ Loss exhibits its outperforming retrieval performance on both convergence speed and retrieval accuracy compared to the existing state-of-the-art.

2 RELATED WORK

Many hashing algorithms [10, 14, 31, 41, 50, 56] have been proposed to obtain compact binary codes, which are seminal solutions to reduce the storage and calculation overhead in large-scale image retrieval. Deep hashing [5, 6, 47, 53] has become the mainstream for the superiority of CNNs [23, 26, 44] in feature extraction, especially in scenarios where images are associated with more than one attribute. The common to all is to design the loss functions to establish powerful metric space and precise hashing positions.

Pair-based Methods. Pair-based methods [18, 25, 32, 51, 57, 58] are predominant in multi-label retrieval [37], which focus on exploring data-to-data relations from the paired samples through metric learning. In the field of image retrieval, Constrictive loss [7, 15] innovatively determines the gradient descent directions by estimating the similarity between feature vector pairs. Based on it, CNNH [53] and DPSH [27] utilize CNNs to extract the features of given images. To reveal the local optima risks in pair-wise loss [27], Triplet Loss [40, 49] associates the anchor with one positive and one negative sample for the loss calculation process.
Recently, researchers concentrate on exploring the profound attribute correlations in challenging multi-label retrieval [37]. The seminal work DSRH [58] introduces CNN-based Triplet Loss to estimate the semantic distance according to the sorted labels. IAH [25] divides hash codes into groups to separately excavate instance-aware image representations. DMSSPH [51] and RCDH [32] further improve the performance by considering the semantic similarity by supervision on grouped labels and additional regularization.

Pair-based methods fully excavate the data correlations and exhibit satisfactory performance. However, the training complexity generally requires square or cubic complexity related to the entire large-scale images. Hence, such approaches suffer high computational consumption and converge difficulty, especially are more serious and ineluctable in multi-label scenarios.

**Proxy-based Methods.** To address the challenging issues in pair-based methods, proxy-based methods are proposed to improve model robustness with efficient training complexity in single-label scenarios. Some methods [12, 17, 55] attempt to alleviate the training difficulty by fixing manually-selected or predefined hash centers. Such predefined centers are regarded as specific proxies for corresponding categories. Hence, the training complexity is affordable because each sample only interacts with a few class proxies.

However, the artificially designed proxies ignore the semantic relationship between intra-class and inter-class. To fill this gap, existing state-of-the-art [1, 21, 34] regard the hash centers as trainable parameters. Proxy NCA [34] calculates the distance between each proxy and positive & negative samples, while Manifold Proxy Loss [1] improves performance with the manifold-aware distance to measure the semantic similarity. Proxy Anchor Loss [21] integrates both advantages of pair-based and proxy-based schemes with Log-Sum-Exp function. It individually considers the distances between different samples and proxies to tackle the hard-pair challenges.

Although the proxy-based methods achieve performance better or on par with pair-based ones with faster convergence speed and promising training overhead in terms of single-label datasets [22, 39], they always fail when come into multi-label scenarios and hence haven’t been fully investigated before. We further elaborate on the reasons and propose our novel HyP² Loss solution further.

### 3 METHODOLOGY

#### 3.1 Task Definition

Given a training set $\mathcal{S}_M := \{ (x_i, y_i) \}_{i=1}^M$ composed of $M$ data points $X := \{ x_i \}_{i=1}^M \in \mathbb{R}^{D \times M}$ and corresponding label $Y := \{ y_i \}_{i=1}^M \in \{0, 1\}^{C \times M}$, where $D$ represents the resolution of images and $C$ denotes the category numbers, respectively. The image $x_i$ contains the attribute of class $y_i$ if $y_{i(j)} = 1$. In multi-label scenarios, each sample contains at least one attribute, i.e., $\sum_{j=1}^C y_{i(j)} \geq 1$. The target for deep hashing is to learn a feature extractor $F$ parameterized by $\Theta$ that encodes each data point $x_i$ into a compact $K$-bit feature vector $v_i := F_\Theta (x_i) \in \mathbb{R}^K$ in metric space, and maps into $K$-bit binary hash code $b_i := \mathcal{H}(v_i) \in \{-1, 1\}^K$ through the hashing function $\mathcal{H}$ in the Hamming space. Hence, the image-wise similarity is preserved in the Hamming space. For given query image $x_q$, we sort the hash codes for all the samples in the database according to their Hamming distance, and return the Top-$N$ images as the query results. The core challenge of this task is to learn a reliable feature extractor $F_\Theta$ to cluster images of different categories with proper and distinguishable hash positions.

#### 3.2 Motivation

**Upper Bound of Distinguishable Hypersphere Number in Metric Space.** Metric learning serves as the substantial procedure for deep hashing. It focuses on embedding the 2D images that consist of various attributes into the $K$-dimensional metric space (i.e., through their feature vectors $\sigma$), where similar samples (i.e., with closer categories) should get clustered.

For image retrieval, it is challenging to establish precise mapping that encodes the input images into the ideal metric space, especially in large-scale datasets [39] that should consider training complexity.

In single-label retrieval (e.g., ImageNet [39], CIFAR [22]), the label correlations are restricted among one positive label with negative others. Hence, proxy-based methods [1, 21, 34] are effective to embed samples around the class proxies in the metric space. Ultimately, samples tend to distribute in a hypersphere centered by predefined or learnable proxies, where similar proxies are close while heterogeneous others are well alienated. While in multi-label retrieval, the inclusive and exclusive relations (i.e., the ability of metric space to distinguish its relevant and irrelevant categories) is exponential w.r.t. category number $C$.

However, the isotropic (i.e., perfectly symmetrical) hypersphere has inherent side-effect in such scenarios. When the labels of one sample larger than two, the label correlations cannot get fully expressed in the hypersphere space. Specifically, we have the following Theorem. A1 to illustrate the upper bound of distinguishable hypersphere number $\Omega(K, C)$ (or called the maximum inclusive and exclusive relations) in multi-label scenarios. Please refer to the supplementary for proof.

**Theorem 3.1.** For the $K$-dimensional metric space $\mathbb{R}^K$ with $C$ hypersphere $S \in \mathbb{R}^K$. The upper bound of distinguishable hypersphere number $\Omega(K, C)$ cannot enumerate the ideal $\Omega^*(K, C) = \sum_{c=0}^C \binom{C}{c} = 2^C$ when $C > K + 1$. The upper bound is limited at:

$$\hat{\Omega}(K, C) = \sup_{S} \Omega(K, C) = \binom{C-1}{K-1} + \sum_{k=0}^K \binom{K}{k} < 2^C \quad (1)$$

**Embedding Position Conflicts in Multi-label Scenarios.** Another limitation of proxy-based methods in multi-label scenarios is that, some irrelevant samples (i.e., without the same categories) are inevitable to be embedded into nearby positions. The primary reason is that the proxy loss only considers the proxy-to-data distance but misses the data-to-data constraints, the irrelevant samples associated with different attributes will be potentially encoded into the close positions nearby the middle of category proxies.

The proxy loss will enforce samples with multiple attributes embedded into the middle among these proxies, because such hash position ensures images with identical multi-label retrieved by query images in priority, second by images containing partially same attributes. Hence, proxy loss encourages multi-label samples to converge nearby the middle of proxies to achieve the optimal solution. Intuitively, as illustrated in Fig. 2(c), suppose the proxy set $\mathcal{P} = \{ p_1, \cdots, p_4 \}$ is well embedded into the metric space, where the proxy loss between any sample $(x, y)$ and $\mathcal{P}$ has converged. Hence, image $x_1$ associated with $y_1, y_2$ (e.g., dog & cat) will be
3.3 Hybrid Proxy-Pair Loss

Sec. 3.2 reveals the primary reasons that proxy-based methods are unsatisfactory in multi-label scenarios, i.e., simply embedding samples distributed among a proxy-centered hypersphere cannot comprehensively introduce the combination among various categories, and the proxy-to-data supervisions ignore data-to-data attribute conflicts. As a result, some crucial label correlations may not be well-expressed, especially under large-scale datasets with limited K-bit hash codes.

The above observations motivate us to consider the data-to-data relations that contribute to a powerful metric space to represent the correlations among various attributes. To avoid constructing the metric space into an isotropic hypersphere without loss of training efficiency, we creatively propose Hybrid Proxy-Pair Loss (HyP² Loss) for metric learning to extend the proxy loss into challenging multi-label scenarios, and compensate for the local optimum and overfitting risks of pair loss in exploring data-to-data relations. The carefully designed HyP² Loss depicts a superior metric space to fully express the profound label correspondences. The overview framework of HyP² Loss is illustrated in Fig. 2, and the details of each component are elaborated as follows.

**Multi-label Proxy Loss.** Firstly, we set C learnable proxies \( \mathcal{P}_C = \{p_1, \cdots, p_C\} \), each \( p_i \in \mathcal{P}_C \) is a compact K-bit vector that is exclusive for each category. For a given feature vector \( x_i = \mathcal{F}_B(x_i) \) and corresponding label \( y_i \), the energy term between any \( x_i \) and \( p_j \) is \( \cos_\star(x_i, p_j) = -\frac{|\langle x_i, p_j \rangle|}{|x_i||p_j|} \) if they are a positive pair, i.e., \( y_{i(j)} = 1 \). Otherwise, \( x_i \) and \( p_j \) are a negative pair. The energy term is defined as \( \cos_\star(x_i, p_j) = \langle x_i, p_j \rangle - \xi \), where \( \xi = \xi(C, K) \) is a margin term that follows [54]. Then, the first term of HyP² Loss is designed as Multi-label Proxy Loss \( L_{\text{proxy}} \), which only optimizes the distance between proxies and samples, as Eq. 2 illustrates.

\[
L_{\text{proxy}}(\mathcal{D}_M, \mathcal{P}_C) = \frac{1}{|\mathcal{D}_M|} \sum_{i=1}^{M} \sum_{j=1}^{C} \mathbb{1}(y_{i(j)} = 1) \cos_\star(x_i, p_j)
+ \frac{1}{|\mathcal{D}_M|} \sum_{i=1}^{M} \sum_{j=1}^{C} \mathbb{1}(y_{i(j)} = 0) \cos_\star(x_i, p_j),
\]

where \( \mathbb{1}(\cdot) \) is the indicator function that equals to 1 \( (0) \) if \( (\cdot) \) is True (False). The denominator term balances \( \cos_\star(\cdot) \) and \( \cos_\star(\cdot) \), such that avoids the gradient bias introduced by overmuch negative pairs. The Multi-label Proxy Loss is used for establishing a primary metric space to ensure the samples are distributed among the cluster centers (cf. Fig. 4), where the correlated labels are properly clustered and irrelevant sample-proxy pairs are roughly alienated.

**Irrelevant Pair Loss.** Secondly, we focus on exploring data-to-data correlations to explicitly enforce irrelevant samples get alienated. To achieve this, we define the irrelevant pairs as: \( (x_i, y_i) \) associated with label \( (y_i, y_j) \) is an irrelevant pair iff \( |y_i|, |y_j| > 0 \) and \( |y_i| > 1, |y_j| > 1 \). Note that the total number of such irrelevant pairs is far fewer than the total \( M \times M \) pairs, the ratio is defined as \( q \) in Tab. 2. Suppose the subset \( \mathcal{D}'_M = \{(x_i, y_i)\}_{i=1}^{M} \subseteq \mathcal{D}_M \) is composed of \( M \) samples, where each sample contains more than one category. Then the second term of HyP² Loss is Irrelevant Pair Loss \( L_{\text{neg_pair}} \), which is defined as Eq. 3.

\[
L_{\text{neg_pair}}(\mathcal{D}'_M) = \frac{1}{|\mathcal{D}'_M|} \sum_{i=1}^{M} \sum_{j=1}^{M} \mathbb{1}(y_i \cdot y_j = 0) \cos_\star(x_i, x_j),
\]

where \( \cos_\star(x_i, x_j) \) indicates the pair-wise similarity of given irrelevant samples. Compared to the entire \( M \times M \) computation, the proposed \( L_{\text{neg_pair}}(\mathcal{D}'_M) \) only considers limited pairs to mine...
Algorithm 1 The training algorithm of the proposed HyP\textsuperscript{2} Loss for metric learning.

\textbf{Input}: Training dataset $\mathcal{D}_M$, hash code length $K$, mini-batch $B$. 
\textbf{Output}: Optimized network $\mathcal{F}_\Theta$ and proxy set $\mathcal{P}_C$.

1. Initialize $\Theta \leftarrow \Theta^{(0)}$, $\mathcal{P} \leftarrow \mathcal{P}^{(0)}$, Epoch $T \leftarrow 0$;
2. repeat
3. Randomly sample a mini-batch samples $\mathcal{D}_B \coloneqq \{(x_i, y_i)_t^{(1)}\}$;
4. Compute feature vector $v_i \coloneqq \mathcal{F}_\Theta(x_i)$ for each $(x_i, y_i) \in \mathcal{D}_B$ by forward propagation;
5. Compute Multi-label Proxy Loss $L_{proxy}(\mathcal{D}_B, \mathcal{P}_C^{(T)})$ via Eq. 2;
6. Compute Irrelevant Pair Loss $L_{neg\_pair}(\mathcal{P}_B)$ via Eq. 3;
7. Compute Total Loss $L_{total}(\mathcal{D}_B, \mathcal{P}_C^{(T)})$ via Eq. 4; 
8. Compute gradient $\frac{\partial L_{total}}{\partial \cos(v_i, p_j)}$ and $\frac{\partial L_{total}}{\partial \cos(o_i, a_j)}$ via Eq. 6;
9. Update $\Theta^{(T+1)}$ and $\mathcal{P}_C^{(T+1)}$ by back-propagation;
10. $T \leftarrow T + 1$;
11. until Convergence 
12. Return $\Theta^* \leftarrow \Theta^{(T)}$, $\mathcal{P}^* \leftarrow \mathcal{P}^{(T)}$.

data-to-data correlations that alienate irrelevant samples effectively without loss of efficiency.

**Overall Loss & Gradient of HyP\textsuperscript{2} Loss.** Finally, the overall HyP\textsuperscript{2} Loss is the weighted-summation of the above two loss terms to obtain $L_{total}$, as Eq. 4 illustrates.

$$L_{total}(\mathcal{D}_M, \mathcal{P}_C) = L_{proxy}(\mathcal{D}_M, \mathcal{P}_C) + \beta L_{neg\_pair}(\mathcal{P}_B)$$

(4)

where $\beta$ is a hyperparameter to balance the constraints between multi-label proxy term and irrelevant pair term.

Hence, to optimize the parameterized network $\mathcal{F}_\Theta$ and proxy set $\mathcal{P}_C$, the objective function is to minimize $L_{total}$ of the given trained set and learnable proxy set, as Eq. 5 illustrates.

$$\Theta^*, \mathcal{P}^* = \arg \min_{\Theta, \mathcal{P}} L_{total}(\mathcal{D}_M, \mathcal{P}_C)$$

To achieve this, the gradient of HyP\textsuperscript{2} Loss in Eq. 4 w.r.t. $\cos(v_i, p_j)$ and $\cos(o_i, a_j)$ is given by Eq. 6.

$$\frac{\partial L_{total}}{\partial \cos(v_i, p_j)} = \begin{cases} \frac{1 - \cos(v_i, p_j)}{\sum_{t=1}^{M} \sum_{j=1}^{K} \cos(v_i^{(t)}, p_j^{(t)})} y_{i}^{(t)} = 1, \\ 0, y_{i}^{(t)} = 0, \cos(v_i, p_j) > \zeta, \\ 0, y_{i}^{(t)} = 0, \cos(v_i, p_j) \leq \zeta \end{cases}$$

(6)

$$\frac{\partial L_{total}}{\partial \cos(o_i, a_j)} = \begin{cases} \frac{\beta - \cos(o_i, a_j)}{\sum_{t=1}^{M} \sum_{j=1}^{K} \cos(o_i^{(t)}, a_j^{(t)})} |y_{i}^{(t)} - y_{j}^{(t)}| = 0, \\ 0, y_{i}^{(t)} = 0, \cos(o_i, a_j) > \zeta, \\ 0, y_{i}^{(t)} = 0, \cos(o_i, a_j) \leq \zeta \end{cases}$$

Eq. 6 shows that minimizing the HyP\textsuperscript{2} Loss enforces $v_i$ and $p_j$ to get close if the two share the same attributes, and distinguishes the irrelevant proxy-to-data/data-to-data pairs simultaneously. When HyP\textsuperscript{2} Loss convergences, we thus construct the powerful metric space by mapping the images from the database into continuous feature vectors, and binarizing into hash codes in the Hamming space for efficient retrieval.

3.4 Overview of Learning Algorithm

**Training Algorithm.** With the novel HyP\textsuperscript{2} Loss, we ensure the constructed metric space is more powerful than existing proxy-based methods [1, 21, 34] both theoretically and experimentally.

**Table 1:** Training complexity of HyP\textsuperscript{2} Loss vs. previous state-of-the-art. Note that proxy-based methods are unqualified for multi-label metric learning, while pair-based methods require $O(M^2)$ or even $O(M^3)$ time complexity. As a comparison, HyP\textsup2 Loss costs more efficient training complexity since the irrelevant pairs are the minority to the whole dataset.

| Type   | Method                  | Time Complexity |
|--------|-------------------------|-----------------|
| Proxy  | Proxy NCA [34]          | $O(MC)$         |
|        | Proxy Anchor [21]       | $O(MC)$         |
|        | OrthoHash [17]          | $O(MC)$         |
|        | SoftTriplet [36]        | $O(MCK^2)$      |
| Pair   | Constrained [3, 7, 15]  | $O(M^2)$        |
|        | HashNet [5]             | $O(M^2)$        |
|        | DNN [59]                | $O(M^2)$        |
|        | IDNN [57]               | $O(M^2)$        |
|        | Triplet (Smart) [16]    | $O(M^2)$        |
|        | Triplet (Semi-Hard) [40]| $O(M^2)B^2$     |
|        | N-pair [42]             | $O(M^2)$        |
|        | Lifted Structure [43]   | $O(M^2)$        |
| Ours   | HyP\textsuperscript{2} Loss | $O(MC + \eta M^2)$ |

because HyP\textsuperscript{2} Loss explicitly enforces the established metric space considers the data-to-data correspondences that tackles the conflicts effectively. To achieve this, we present the training algorithm in Algo. 1. During the training process, the standard back-propagation algorithm [38] with mini-batch gradient descent method is used to optimize the network.

**Time Complexity Analysis.** The proposed method converges faster and is proven more efficient and stable than those pair-based methods [18, 25, 32, 57, 58] (as we will justify in Fig. 5). Below we analyze the training complexity of HyP\textsuperscript{2} Loss. Note that $M, C, B$, and $k$ denote the training sample number, category number, mini-batch size, and the proxy number of each category, respectively. $\eta$ is specifically defined in HyP\textsuperscript{2} Loss, which indicates the ratio of irrelevant sample pairs with multiple labels to all sample pairs in the dataset. We omit $k \equiv 1$ in single-proxy methods [21, 34] and ours for simplicity. $k$ is nontrivial for managing multiple proxies per class such as SoftTriplet Loss [36].

Tab. 1 comprehensively compares the training complexity of HyP\textsuperscript{2} Loss (ours) to state-of-the-art pair-based and proxy-based methods. The complexity of HyP\textsuperscript{2} Loss is $O(MC + \eta M^2)$ since it compares each sample with positive or negative proxies and its irrelevant samples (if exists) in a mini-batch. More specifically, in Eq. 4, the complexity of the first summation requires $M$s ($M \times M$) times calculation for positive (negative) proxy-to-data pairs, respectively. Hence the total training complexity is $O(MC + MC) = O(MC)$. Then the second term requires $\eta M^2$ times calculation for each irrelevant pair. The first term of HyP\textsuperscript{2} Loss is linear correlated to
Table 3: mAP performance by Hamming Ranking for different hash bits ($K \in \{12, 24, 36, 48\}$) in Flickr-25k (mAP@1,000) and NUS-WIDE (mAP@5,000) with AlexNet. *: reported results with the same experiment settings from [57]. †: our reproduced results through the publicly available models. Bold font (underlined) values indicate the best (second best).

| Method      | Dataset | Flickr-25k | NUS-WIDE |
|-------------|---------|------------|----------|
|             | Pub.    | 12 24 36 48 | avg. Δ  | 12 24 36 48 | avg. Δ |
| DLBHC [29]* | CVPR’15 | 0.724 0.757 0.757 0.776 | - 0.570 0.616 0.621 0.635 | - |
| DQN [4]*    | AAAI’16 | 0.809 0.823 0.830 0.827 | 0.069 0.711 0.733 0.745 0.749 0.124 |
| DHN [59]‡   | AAAI’16 | 0.817 0.831 0.829 0.851 | 0.079 0.720 0.742 0.741 0.749 0.128 |
| HashNet [5]* | ICCV’17 | 0.791 0.826 0.841 0.848 | 0.073 0.643 0.694 0.737 0.750 0.095 |
| DMSSPH [51]* | ICMR’17 | 0.780 0.808 0.810 0.816 | 0.050 0.671 0.699 0.717 0.727 0.093 |
| IDHN [57]†  | TMM’20  | 0.827 0.823 0.822 0.828 | 0.071 0.772 0.790 0.795 0.803 0.180 |
| Proxy Anchor [21]† | CVPR’20 | 0.796 0.831 0.834 0.853 | 0.075 0.767 0.802 0.809 0.815 0.188 |
| CSQ [55]†   | CVPR’20 | 0.795 0.819 0.849 0.857 | 0.077 0.692 0.754 0.757 0.769 0.132 |
| OrthoHash [17]† | NeurIPS’21 | 0.837 0.869 0.877 0.891 | 0.115 0.770 0.802 0.810 0.825 0.191 |
| DCLILH [33]† | TMM’21  | 0.852 0.879 0.884 0.888 | 0.122 0.775 0.793 0.797 0.804 0.182 |

Evaluation Protocol. We follow [20, 24, 54, 58] to employ four metrics for quantitative evaluation: 1). mean average precision (mAP@N), 2). precision w.r.t. Top-N returned images (Top-N curves), 3). the average $d_{i\text{nt}}$ distance of each sample to the corresponding cluster centers ($d_{\text{nt}}$), and 4). the average $d_{\text{nt}}$ distance of each cluster to the closest irrelevant cluster centers ($d_{\text{nt}}$). Regarding mAP@N score computation, we select the Top-N images from the retrieval ranked-list results. The returned images and the query image are considered similar iff they share at least one same label.

VOC-2007 & VOC-2012. VOC-2007 (VOC-2012) contains 9,963 (11,540) images in total, each image attaches to a label containing several of the 20 categories. We follow [18] to construct the training set and database of the two datasets for the experiment separately, with 5,011 (5,717) samples in total. The officially provided query set with 4,952 (5,823) samples is used for evaluation.

NUS-WIDE. The NUS-WIDE dataset contains 269,648 images, and each image is assigned to several 81 categories. We follow [24, 57] to select the most frequent 21 categories and 195,834 images containing these attributes. We randomly selected 10,500 and 2,100 samples as the training and query set, respectively, and the rest samples are constructed as the database.

M and $C$, while the second term is significantly degraded because $\eta$ is much less than 1 in general.

4 EXPERIMENT

In this section, we describe the datasets used for evaluation, the test protocols, and the implementation details. To evaluate our method, we fairly conduct experiments against existing state-of-the-art [17, 18, 21, 55, 57] and previous methods [4, 5, 24, 25, 29, 30, 51, 59] on four standard multi-label benchmarks, and justify the superiority of the proposed method both quantitatively and qualitatively. Finally, we explore and conduct in-depth analyses of how each component of the proposed framework contributes to the performance.

4.1 Implementation Details

We implement the proposed method in the PyTorch framework [35] and train on a single NVIDIA RTX 3090 GPU. We comprehensively adopt AlexNet [23] and GoogLeNet [44] pretrained on ImageNet [39] as the backbones to justify the robustness of the proposed method. We fine-tune the pretrained backbones for all layers up to the FC layer and map the output layer into K-dimensional hash bits. We adopt stochastic gradient descent (SGD) [2] to optimize the network with momentum 0.9 and weight decay $5e - 4$. The initial learning rates for optimizing network $F_{\theta}$/proxies $P$ are 0.01/0.001 in AlexNet and 0.02/0.02 in GoogLeNet, respectively. The learning rate decreases by 0.5 every 10 epochs with 100 epochs in total.

4.2 Dataset & Evaluation Metrics

Four standard benchmarks Flickr-25k [19], VOC-2007 [11], VOC-2012 [11], and NUS-WIDE [8] are adopted for evaluation. The statistics of the four datasets are summarized in Tab. 2, and the detailed descriptions are as follows.

Flickr-25k. The Flickr-25k dataset contains 25,000 images. We follow [24, 57] to remove the noisy images that do not contain any labels. The remaining 24,851 images contained 38 categories in total. Among them, 4,000 samples are randomly selected as the training set, 1,000 samples as the query set and the rest images to construct the database.

VOC-2007 & VOC-2012. VOC-2007 (VOC-2012) contains 9,963 (11,540) images in total, each image attaches to a label containing several of the 20 categories. We follow [18] to construct the training set and database of the two datasets for the experiment separately, with 5,011 (5,717) samples in total. The officially provided query set with 4,952 (5,823) samples is used for evaluation.

NUS-WIDE. The NUS-WIDE dataset contains 269,648 images, and each image is assigned to several 81 categories. We follow [24, 57] to select the most frequent 21 categories and 195,834 images containing these attributes. We randomly selected 10,500 and 2,100 samples as the training and query set, respectively, and the rest samples are constructed as the database.

4.3 Quantitative Comparison

Baselines & Settings. We compare the proposed method with 1). standard baselines, including HashNet [5], DMSSPH [51], DQN [4], DLBHC [29], DHN [59], NINH [24], DSH [30], and IAH [25], 2). state-of-the-art deep hashing methods, including IDHN [57], Proxy Anchor [21], CSQ [55], OrthoHash [17], OLAH [18], and DCLILH [33]. Note that OLAH and DCLILH are two state-of-the-art deep hashing methods specifically designed for multi-label image retrieval. Besides, Proxy Anchor is the state-of-the-art proxy-based method. We verify the robustness of such proxy-based methods in multi-label scenarios to elaborate on how the proposed method improves the metric space and retrieval performance effectively.

Specifically, to justify the effectiveness of the proposed method, we compare methods using AlexNet in Flickr-25k and NUS-WIDE among hash bits $K \in \{12, 24, 36, 48\}$ in Tab. 3, and further compare...
Table 4: mAP performance by Hamming Ranking for different hash bits (\(K \in \{16, 32, 48, 64\}\)) in VOC-2007 (mAP@5, 011) and VOC-2012 (mAP@5, 717) with GoogLeNet. *: reported results with the same experiment settings from [18]. †: our reproduced results through the publicly available models. Bold font (underlined) values indicate the best (second best).

| Method         | Dataset | VOC-2007 | VOC-2012 |
|----------------|---------|----------|----------|
|                |         | 16 32 48 64 | avg. \(\Delta\) | 16 32 48 64 | avg. \(\Delta\) |
| Pub.           |         |         |         |         |         |
| DHN [59]†      | AAAI’16 | 0.735 0.743 0.737 0.728 | - | 0.722 0.721 0.718 0.701 | - |
| NINH [24]†     | CVPR'15 | 0.746 0.816 0.840 0.851 | 0.077 | 0.731 0.788 0.809 0.822 | 0.072 |
| DSH [30]*      | CVPR’16 | 0.763 0.767 0.769 0.775 | 0.033 | 0.753 0.766 0.776 0.782 | 0.054 |
| IAH [25]†      | TIP’16 | 0.800 0.862 0.878 0.883 | 0.120 | 0.794 0.844 0.862 0.864 | 0.126 |
| OLAH [18]†     | TIP’18 | 0.849 0.899 0.906 0.914 | 0.156 | 0.830 0.887 0.904 0.908 | 0.167 |
| IDHN [57]†     | TMM’20 | 0.772 0.801 0.796 0.772 | 0.050 | 0.785 0.805 0.797 0.785 | 0.078 |
| Proxy Anchor [21]† | CVPR’20 | 0.752 0.802 0.836 0.841 | 0.072 | 0.722 0.795 0.804 0.823 | 0.071 |
| OrthoHash [17]† | NeurIPS’21 | 0.831 0.876 0.902 0.909 | 0.144 | 0.823 0.885 0.893 0.900 | 0.160 |
| **HyP² Loss (Ours)** | - | **0.862** 0.917 0.932 0.937 | **0.176** | **0.841** 0.903 0.917 0.925 | **0.181** |

Figure 3: Convergence comparisons in Flickr-25k (left) and NUS-WIDE (right) (48 hash bits with AlexNet). x-axis: training time (sec.), y-axis: mAP@1, 000 (left) and 5, 000 (right) performance on the query dataset. HyP² Loss achieves a faster convergence speed with a more stable training process.

Results & Analysis. As Tab. 3 and Tab. 4 illustrate, HyP² Loss outperforms existing methods over different hash bits in the four benchmarks, which justifies the robustness and effectiveness of the proposed method. Note that when the hash bits are small (e.g., 12-bit in Flickr-25k/NUS-WIDE and 16-bit in VOC-2007/VOC-2012), the proposed method achieves 10.20% performance gains on average compared to Proxy Anchor [21], the state-of-the-art proxy-based method in image retrieval.

We justify that the metric space established by proxy loss is insufficient to express the profound label correlations, which achieves unsatisfactory mAP and misclassified retrieval performance. As a comparison, HyP² Loss effectively improves the metric space by additional constraints to explicitly improve the isotropic hypersphere space, and thus improves the retrieval accuracy remarkably.

4.4 Qualitative Comparison

Convergence Comparison. To demonstrate that the convergence speed of the proposed method outperforms existing methods, we compare [17, 21, 57, 59] to HyP² Loss in Flickr-25k and NUS-WIDE datasets. The visualized results are presented in Fig. 3.

Fig. 3 shows that the proposed method achieves a more stable and faster convergence speed with higher performance compared to the previous state-of-the-art. Note that pair-based methods [57, 59]...
Figure 5: Performance of different methods in Flickr-25k (top) and NUS-WIDE (bottom) datasets. From left to right: Top-N curves (x-axis (Top-N): 100 → 1,000, y-axis (Precision): 0 → 1) w.r.t. 12, 24, 36, 48 hash bits, respectively. Our HyP$^2$ Loss outperforms previous methods in different datasets among various hash bit lengths consistently.

### Table 5: Ablation study and hyperparameter analysis of HyP$^2$ Loss.

To justify how each component of HyP$^2$ Loss contributes to a more powerful metric space, we conduct in-depth ablation studies on investigating the effectiveness of Multi-label Proxy Loss, Irrelevant Pair Loss, and the combination of them with different $\beta$, the results of mAP, $d_{\text{intra}}$, and $d_{\text{inter}}$ are shown in Tab. 5 and Tab. 6.

| Dataset | Flickr-25k | VOC-2007 |
|---------|------------|----------|
| Hash Bits | 12 | 24 | 36 | 48 |
| Pair Loss | 0.779 | 0.832 | 0.837 | 0.847 | 0.732 | 0.766 | 0.777 | 0.781 |
| Proxy Loss | 0.787 | 0.834 | 0.856 | 0.871 | 0.767 | 0.820 | 0.873 | 0.887 |
| Proxy + Pair Loss | 0.817 | 0.851 | 0.857 | 0.870 | 0.818 | 0.843 | 0.875 | 0.881 |
| HyP$^2$ Loss ($\beta = 0.50$) | 0.838 | 0.876 | 0.893 | 0.896 | 0.862 | 0.917 | 0.932 | 0.937 |
| HyP$^2$ Loss ($\beta = 0.75$) | 0.842 | 0.877 | 0.891 | 0.896 | 0.859 | 0.915 | 0.930 | 0.935 |
| HyP$^2$ Loss ($\beta = 1.00$) | 0.845 | 0.881 | 0.893 | 0.901 | 0.857 | 0.912 | 0.926 | 0.934 |
| HyP$^2$ Loss ($\beta = 1.25$) | 0.841 | 0.877 | 0.891 | 0.897 | 0.843 | 0.909 | 0.927 | 0.935 |

Top-N retrieved images in Fig. 5. We can observe that the proposed HyP$^2$ Loss consistently establishes the state-of-the-art retrieval performance with higher scores over different hash bits.

### 4.5 Ablation Study

To justify how each component of HyP$^2$ Loss contributes to a more powerful metric space, we conduct in-depth ablation studies on investigating the effectiveness of Multi-label Proxy Loss, Irrelevant Pair Loss, and the combination of them with different $\beta$, the results of mAP, $d_{\text{intra}}$ and $d_{\text{inter}}$ are shown in Tab. 5 and Tab. 6.

As Tab. 5 and Tab. 6 illustrate, either pair loss or proxy loss fails to achieve satisfactory performance due to its inherent limitations as we analyzed before, and a simple combination of the two terms with $\beta$ is invalid but confronts overwhelmed training overhead. Specifically, since smaller $d_{\text{intra}}$ indicates better cluster performance, while larger $d_{\text{inter}}$ indicates better disentangle ability on confusion samples. The pair loss constructs a sparse metric space that fails to cluster samples tightly, while the proxy loss fails to distinguish the confusing samples that introduces misclassified results. As a comparison, the proposed HyP$^2$ Loss not only achieves remarkable performance gains, but also ensures better $d_{\text{intra}}$ and

| Dataset | Flickr-25k | VOC-2007 |
|---------|------------|----------|
| Hash Bits | 12 | 24 | 36 | 48 |
| Pair Loss | 0.799 | 0.832 | 0.837 | 0.847 | 0.732 | 0.766 | 0.777 | 0.781 |
| Proxy Loss | 0.787 | 0.834 | 0.856 | 0.871 | 0.767 | 0.820 | 0.873 | 0.887 |
| Proxy + Pair Loss | 0.817 | 0.851 | 0.857 | 0.870 | 0.818 | 0.843 | 0.875 | 0.881 |
| HyP$^2$ Loss ($\beta = 0.50$) | 0.838 | 0.876 | 0.893 | 0.896 | 0.862 | 0.917 | 0.932 | 0.937 |
| HyP$^2$ Loss ($\beta = 0.75$) | 0.842 | 0.877 | 0.891 | 0.896 | 0.859 | 0.915 | 0.930 | 0.935 |
| HyP$^2$ Loss ($\beta = 1.00$) | 0.845 | 0.881 | 0.893 | 0.901 | 0.857 | 0.912 | 0.926 | 0.934 |
| HyP$^2$ Loss ($\beta = 1.25$) | 0.841 | 0.877 | 0.891 | 0.897 | 0.843 | 0.909 | 0.927 | 0.935 |

### 5 CONCLUSION

In this paper, we focus on theoretically analyzing the primary reasons that proxy-based methods are disqualified for multi-label retrieval, and propose the novel HyP$^2$ Loss to preserve the efficient training complexity of proxy loss with the irrelevant constraint term, which compensates for the limitation of the hypersphere metric space. We conduct extensive experiments to justify the superiority of the proposed method in four standard benchmarks with different backbones and hash bits. Both quantitative and qualitative results demonstrate that the proposed HyP$^2$ Loss enables fast, reliable and robust convergence speed, and constructs a powerful metric space to improve the retrieval performance significantly.

**Acknowledgment.** This work was supported by SZSTC Grant No. JCYJ20190809172201639 and WDZC20200820020655001, Shenzhen Key Laboratory ZDSYS20210623092001004.
[46] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).

[47] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. 2016. Learning to Hash for Indexing Big Data - A Survey. Proc. IEEE 104, 1 (2016), 34–57.

[48] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. 2018. A Survey on Learning to Hash. IEEE Trans. Pattern Anal. Mach. Intell. 40, 4 (2018), 769–790.

[49] Xiaofang Wang, Yi Shi, and Kris M. Kitani. 2016. Deep Supervised Hashing with Triplet Labels. In Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 10111), Shang-Hong Lai, Vincent Lepetit, Ko Nishino, and Yoichi Sato (Eds.). Springer, 70–84.

[50] Yair Weiss, Antonio Torralba, and Robert Fergus. 2008. Spectral Hashing. In Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou (Eds.). Curran Associates, Inc., 1753–1760.

[51] Dayan Wu, Zheng Lin, Bo Li, Mingzhen Ye, and Weiping Wang. 2017. Deep Supervised Hashing for Multi-Label and Large-Scale Image Retrieval. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR, Bogdan Ionescu, Nicu Sebe, Jiashi Feng, Martha A. Larson, Rainer Lienhart, and Cees Snoek (Eds.). ACM, 150–158.

[52] Haifeng Xia, Taotao Jing, Chen Chen, and Zhengming Ding. 2021. Semi-supervised Domain Adaptive Retrieval via Discriminative Hashing Learning. In MM ’21: ACM Multimedia Conference, Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo Cesar, Florian Metze, and Balakrishnan Prabhakaran (Eds.). ACM, 3409–3417.

[53] Rongkhao Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. 2014. Supervised Hashing for Image Retrieval via Image Representation Learning. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Carla E. Brodley and Peter Stone (Eds.). AAAI Press, 2156–2162.

[54] Chengyin Xu, Zhengzhao Xu, Zenghao Chai, Hongjia Li, Qiuyi Zuo, Lingyu Yang, and Chun Yuan. 2021. HHF: Hashing-guided Hinge Function for Deep Hashing Retrieval. CoRR abs/2112.02225 (2021).

[55] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis E. H. Tay, Zequn Jie, Wei Liu, and Jiashi Feng. 2020. Central Similarity Quantization for Efficient Image and Video Retrieval. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR. Computer Vision Foundation / IEEE, 3080–3089.

[56] Lei Zhang and Yong Rui. 2013. Image search - from thousands to billions in 20 years. ACM Trans. Multim. Comput. Commun. Appl. 9, 1s (2013), 36:1–36:20.

[57] Zheng Zhang, Qin Zou, Yuesei Lin, Long Chen, and Song Wang. 2020. Improved Deep Hashing With Soft Pairwise Similarity for Multi-Label Image Retrieval. IEEE Trans. Multim. 22, 2 (2020), 540–553.

[58] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. 2015. Deep semantic ranking based hashing for multi-label image retrieval. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR. IEEE Computer Society, 1556–1564.

[59] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 2415–2421.
A APPENDIX

A.1 Missing Proof

**Theorem A.1.** For the $K$-dimensional metric space $\mathbb{R}^K$ with $C$ hyperspheres $S \in \mathbb{R}^K$, the upper bound of distinguishable hypersphere number $\Omega(K, C)$ cannot enumerate the ideal $\Omega^*(K, C) = \sum_{c=0}^{\binom{C}{2}} \binom{C}{c} = 2^C$ when $C > K + 1$. The upper bound is limited at:

$$\hat{\Omega}(K, C) = \text{sup}_{S} \Omega(K, C) = \left(\binom{C}{K} - 1\right) + \sum_{k=0}^{K} \binom{C}{k} < 2^C \quad (7)$$

**Proof.** To begin, we first deduce the optimal distinguishable hyperspace number $\Omega^*(K, C)$ in the $K$-dimensional hyperspace with $C$ categories. Note that each hyperspace represents one category’s cluster center. Hence, the ideal distinguishable number of any $c \in \{0, \cdots, C\}$ hyperspheres is the combination of them that is denoted as $n(K, c)$.

$$n(K, c) = \binom{C}{c} \quad (8)$$

Hence, according to the binomial theorem, for $C$ hypersphere space, the optimal distinguishable hyperspace number is the summation of each $c \in \{0, \cdots, C\}$, as Eq. 9 illustrates.

$$\Omega^*(K, C) = \sum_{c=0}^{C} \binom{C}{c} = 2^C \quad (9)$$

Then, note that the ideal distinguishable hyperspace number can only get achieved when the hyperspace is unconstrained (i.e., it can have any intersections to each other). When it comes to $K$-dimensional isotropic hypersphere, we raise Lemma A.2 to further demonstrate the upper bound in the $C$ metric space.

**Lemma A.2.** When $K \geq 2$, two $K$-dimensional hyperspheres will intersect at one $(K-1)$-dimensional hypersphere at maximum. Specifically, two 2D circles will intersect at one 1-dimensional hypersphere at maximum. The 1-dimensional hypersphere is a pair of two points, which are the boundary of a line segment.

We denote the $K$-dimensional hypersphere with the number of $C$ has the maximum distinguishable regions as $\Omega(K, C)$. Then, the $K$-dimensional hypersphere with the number of $C-1$ has the maximum distinguishable regions $\Omega(K, C-1)$. When we add the $C$-th hypersphere into the existing $K$-dimensional hyperspheres with the number of $C-1$, according to Lemma A.2, the $C$-th hypersphere will intersect with each hypersphere in the hyperspace at one $(K-1)$-dimensional hypersphere at maximum, i.e., $(K-1)$-dimensional hyperspheres with the number of $C-1$ at maximum are added into the $K$-dimensional hyperspace. These $(K-1)$-dimensional hyperspheres thereby generate $(K-1)$-dimensional hypersurfaces with the number of $\Omega(K-1, C-1)$ at maximum correspondingly, and each of these new $(K-1)$-dimensional hypersurfaces bisects the $K$-dimensional space into two parts. So we have the following recurrence relation:

$$\Omega(K, C) = \Omega(K, C-1) + \Omega(K-1, C-1) \quad (10)$$

Note that one $K$-dimensional hypersphere can only separate hyperspace into two regions, i.e., the inside and outside of the hypersphere, respectively. Hence, we have the initial condition that for every $K \geq 2$ and $C \equiv 1$:

$$\Omega(K, 1) \equiv 2 \quad (11)$$

Without loss of generalization, we first consider the simplest particular case when $K = 2$. Then, the $K$-dimensional hypersphere is degraded into a 2D circle situation.

Suppose there are $C-1$ circles in the 2D space, when we have an additional one circle in the space, such that each $C-1$ circle should intersect with the additional one circle to achieve the maximum distinguishable hypersphere number.

Then, according to Lemma A.2, the circles will at most intersect with the additional circle with $(C-1)$ intersected 1D point pairs. Hence, the $2 \times (C - 1)$ points will introduce $2 \times (C - 1)$ lines that segment the original regions into bisects. As a result, the additional one circle will introduce additional $2 \times (C - 1)$ regions. We can obtain that:

$$\Omega(2, C) = \Omega(2, C-1) + 2 \times (C - 1) \quad (12)$$

According to Eq. 11, we have $\Omega(2, 1) = 2$, i.e., one 2D circle can represent two distinguishable regions at maximum, then:

$$\Omega(2, C) = 2 + 2 \times (C - 1) + 2 \times (C - 2) = \cdots = \Omega(2, 1) + 2 \times (C - 1 + C - 2 + \cdots + 1) \quad (13)$$

which satisfies the proposition when $K = 2$.

Then, we generalize into the $K$-dimensional situation and will prove Theorem A.1 by mathematical induction below.

- a). Considering the initial situation that for any $K \geq 2, K \in \mathbb{N}_+$ and $C = 1$. As Eq. 11 illustrates, we have:

$$\Omega(K, C) = \Omega(K, 1) = \left(\binom{C}{K} - 1\right) + \sum_{k=0}^{K} \binom{C}{k} \quad (14)$$

which satisfies the proposition when $C = 1$.

- b). For any $K \geq 2, K \in \mathbb{N}_+$ and a specific $C \geq 1, C \in \mathbb{N}_+$, we assume the upper bound of distinguishable hypersphere number $\Omega(K, C)$ satisfies:

$$\Omega(K, C) = \left(\binom{C}{K} - 1\right) + \sum_{k=0}^{K} \binom{C}{k} \quad (15)$$

- c). Then, considering the situation of $\Omega(K, C+1)$. When $K = 2$, as Eq. 13 illustrates, $\Omega(2, C+1)$ satisfies the proposition. When $K > 2$, according to Eq. 10, we have:

$$\Omega(K, C+1) = \Omega(K, C) + \Omega(K-1, C) \quad (16)$$

$$\begin{align*}
\Omega(K, C+1) &= \left(\binom{C+1}{K} - 1\right) + \sum_{k=0}^{K} \binom{C+1}{k} + \sum_{k=0}^{K-1} \binom{C}{k} + \sum_{k=0}^{K} \binom{C}{k} \\
&= \left(\binom{C+1}{K} - 1\right) + \sum_{k=0}^{K} \binom{C+1}{k} + \sum_{k=0}^{K} \binom{C}{k} + \sum_{k=0}^{K} \binom{C}{k} \\
&= \left(\binom{C}{K} + \binom{C-1}{K-1}\right) + \sum_{k=0}^{C} \binom{C}{k} + \sum_{k=0}^{C} \binom{C}{k} + \cdots + \left(\binom{C}{K+1} + \binom{C}{K}\right)
\end{align*}$$
Note that \( \binom{m}{n} = \binom{m-1}{n} + \binom{m-1}{n-1} \), then we have:
\[
\Omega(K, C + 1) = \left( \frac{\Omega(K, C)}{K} \right) + \left( \frac{\Omega(K, C + 1)}{K} \right) + \cdots + \left( \frac{\Omega(K, C)}{K} \right)
\]
\[
= \frac{\Omega(K, C)}{K} + \sum_{k=0}^{K} \frac{\Omega(K, C + 1)}{K} \tag{17}
\]
which satisfies the proposition when \( C = C + 1 \).

Finally, according to mathematical induction, for any \( K \geq 2, C \geq 1, K, C \in \mathbb{N}_+ \), the upper bound of distinguishable hypersphere number in \( K \)-dimensional metric space is obtained that:
\[
\hat{\Omega}(K, C) = \sup_{\theta} \Omega(K, C) = \left( \frac{\Omega(K, C)}{K} \right) + \sum_{k=0}^{K} \left( \frac{\Omega(K, C + 1)}{K} \right) \tag{18}
\]
When \( C = K + 1 \), we can see that the ideal distinguishable hyperspace number \( \hat{\Omega}(K, C) \) is equal to the upper bound \( \hat{\Omega}(K, C) \) because the hash bit length is large enough to enumerate all possible situations, as Eq. 19 illustrates.
\[
\hat{\Omega}(K, C) = \left( \frac{\Omega(K + 1)}{K} \right) + \sum_{k=0}^{K} \left( \frac{\Omega(K + 1)}{K} \right)
\]
\[
= \left( \frac{\Omega(K + 1)}{K} \right) + \left( \frac{\Omega(K + 1)}{K} \right) + \cdots + \left( \frac{\Omega(K + 1)}{K} \right)
\]
\[
= \sum_{k=0}^{K} \left( \frac{\Omega(K + 1)}{K} \right) = 2^{K + 1} = \Omega^*(K, C) \tag{19}
\]
Then, we will prove that when \( C > K + 1, \hat{\Omega}(K, C) < \Omega^*(K, C) \) by mathematical induction below.

• a). Considering the initial situation that for any \( K \geq 2, K \in \mathbb{N}_+ \), when \( C = K + 2 \), according to Eq. 10, we have:
\[
\hat{\Omega}(K, C) = \Omega(K, K + 2)
\]
\[
= \Omega(K, K + 1) + \Omega(K - 1, K + 1)
\]
\[
= \Omega(K, K + 1) + \Omega(K - 1, K) + \Omega(K - 2, K) \tag{20}
\]
As Eq. 13 and Eq. 19 illustrate, we have:
\[
\hat{\Omega}(K, C) = 2^{K+1} + 2^{K-2} + \cdots + 2^4 + 2^4 - 4 + 2
\]
\[
= 2^{K+2} - 2
\]
\[
= 2^{K+2} - 2
\]
\[
< 2^{K+2} = \Omega^*(K, C),
\]
which satisfies the proposition when \( C = K + 2 \).

• b). For any \( K \geq 2, K \in \mathbb{N}_+ \), and a specific \( I \geq 2, I \in \mathbb{N}_+ \), let \( C = K + I \), we assume the inequation \( \hat{\Omega}(K, C) < \Omega^*(K, C) \) satisfies:
\[
\hat{\Omega}(K, C) = \hat{\Omega}(K, K + I) < \Omega^*(K, K + I) = 2^{K+I}, \tag{22}
\]
• c). Then, when \( C = K + I + 1 \), we have:
\[
\hat{\Omega}(K, C) = \hat{\Omega}(K, K + I + 1)
\]
\[
= \hat{\Omega}(K, K + I) + \Omega(K - 1, K + I)
\]
\[
= \hat{\Omega}(K, K + I) + \Omega(K - 1, K + I) + \Omega(K - 2, K + I - 1)
\]
\[
= \cdots \tag{23}
\]
\[
\Omega(K, K + I) + \Omega(K - 1, K + I) + \cdots + \Omega(3, 3 + I)
\]
\[
+ \Omega(2, 3 + I)
\]
\[
< 2^{K+I} + 2^{K+I-1} + \cdots + 2^{2+I} + (3 + I)^2 - (3 + I) + 2
\]
\[
= 2^{K+I} - 2^{2+I} + 3 + 8 \tag{24}
\]
Note that \( 2^{3+I} > I^2 + 2I + 8 \) when \( I \geq 2 \), then we have:
\[
\hat{\Omega}(K, C) < 2^{K+I+1} = \Omega^*(K, C),
\]
which satisfies the proposition when \( C = K + I + 1 \).

Finally, according to mathematical induction, \( \hat{\Omega}(K, C) < \Omega^*(K, C) \) when \( C > K + 1 \).

\[\square\]

Theorem A.3. When the proxies have converged to fixed positions (i.e., the angles of proxy pairs are constant), then the best position of 2-label samples is the middle of the 2 positive proxies. The n-label scenarios can be deduced in a similar fashion.

Proof. Suppose the feature vector \( \sigma_x \) contains 2 labels \( y_1, y_2 \), with corresponding proxies \( p_1, p_2 \). Let \( \theta_1 = (\sigma_x, p_1), \theta_2 = (\sigma_x, p_2), \theta_3 = (p_1, p_2), \) where \( \theta_1, \theta_2, \theta_3 \in [0, \pi] \).

Note that the effect of negative proxies is negligible when about convergence, because they are away from \( \sigma_x \), so we only consider the gradient from \( p_1, p_2 \).

Then we have \( L_+ = -(\cos \theta_1 + \cos \theta_2) \), and the objective function is \( \text{arg max}(\cos \theta_1 + \cos \theta_2) \) accordingly.

If \( \sigma_x \) is non-coplanar with \( p_1, p_2 \), we have the projection \( \sigma_x' \) in the plane defined by \( p_1, p_2 \), and denote the corresponding angles with \( p_1, p_2 \) as \( \theta_1', \theta_2' \), such that \( \theta_1' < \theta_1, \theta_2' < \theta_2 \Rightarrow \cos \theta_1' + \cos \theta_2' > \cos \theta_1 + \cos \theta_2 \).

Hence, the optimal objective function is satisfied when \( \sigma_x, p_1, p_2 \) are coplanar such that \( \theta_1' = \theta_1, \theta_2' = \theta_2 \), and ensures \( \theta_3 = \theta_1 + \theta_2 \). Then we have:
\[
\text{arg min } L_+ = \text{arg max}((1 + \cos \theta_3) \cos \theta_1 + \sin \theta_3 \sqrt{1 - \cos^2 \theta_1})
\]
\[\tag{25}\]
To obtain the extreme point of \( L_+ \), let:
\[
\frac{\partial L_+}{\partial \cos \theta_1} = 1 + \cos \theta_1 - \sin \theta_3 \frac{\cos \theta_1}{\sqrt{1 - \cos^2 \theta_1}} = 0
\]
\[\tag{26}\]
Note that \( 1 + \cos \theta_3 \geq 0, \sin \theta_3 \geq 0, \sqrt{1 - \cos^2 \theta_1} \geq 0, \) we can get \( \cos \theta_1 \geq 0 \). Thus \( \theta_1 \in [0, \frac{\pi}{2}] \). Then, it is easy to obtain that:
\[
\frac{1 - \cos^2 \theta_3}{2 + 2 \cos \theta_3} = 1 - \cos^2 \theta_1
\]
\[
\Rightarrow \cos^2 \theta_1 = 1 - \frac{1 - \cos^2 \theta_3}{2 + 2 \cos \theta_3} \tag{27}
\]
\[
= \cos^2 \theta_3 \frac{2}{2}
\]

Considering the domain of \( \theta_1, \theta_2, \theta_3 \), we have \( \cos \theta_1 = \cos \frac{\theta_1}{2} \Rightarrow \theta_1 = \theta_2 = \frac{\theta_3}{2}, i.e., \sigma_x \) will be embedded into the middle of the two proxies.

Similarly, we can extend the conclusion into n-label scenarios where the optimal solution is satisfied when \( \sigma_x' \) is in the middle of the n-proxies, as we claimed in the main paper.

\[\square\]