Make Prototypes Perform Again: Prior-Prototypes Based Feature learning Framework for Few-Shot Hashing

Yi Lu[1], Shu Li[1], Hualong Dong[2], Shuxiang Hou[1], Yurong Qian[1,3*]

[1]School of Computer Science and Technology, Xinjiang University
[2]School of Software, Xinjiang University
[3]Joint International Research Laboratory of Silk Road Multilingual Cognitive Computing

Email:
qyr@xju.edu.cn
{outman, lsmiao, dhl, hsx}@stu.xju.edu.cn

Index Terms

Few-Shot Learning, Hashing Learning, Prototype, Diffusion Mechanism, Self-consistency

Introduction

- Deep hashing methods typically rely on high-quality feature embeddings to generate compact hash codes that preserve semantic information.
- However, supervised learning restricts the feature extractor to the training data's prior distribution, causing performance degradation on unseen categories. Combined with privacy concerns and annotation costs, this label scarcity further limits the scalability of deep hashing models in open-world scenarios.
- Few-shot learning (FSL, As shown in Fig. 1(1)(2)) has emerged as a widely adopted paradigm to address these limitations, enabling model adaptation with limited supervision.

How can existing deep models be effectively leveraged to generate high-quality feature embeddings, ensuring the production of robust and efficient hash codes under limited supervision?

💡 This work propose a novel **Prior-Prototypes (PP)** based framework for Few-Shot Hashing Learning (As shown in Fig. 1(3)).

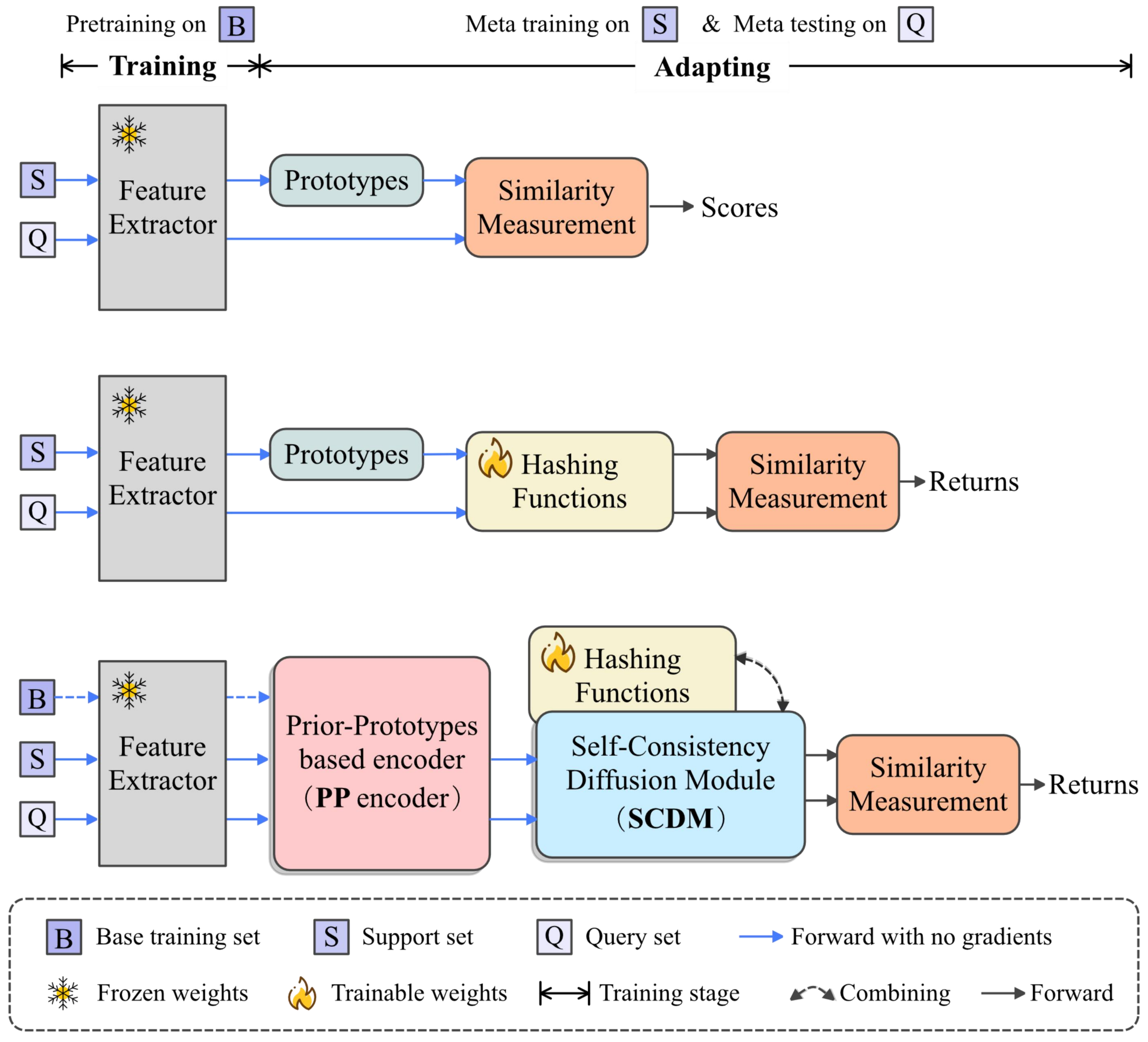

Fig 1 Comparison of FSL, FSHL, and the Proposed PP Framework

Methodology

The **PP framework** consists of two key components:

- **Prior-Prototypes based encoder**: Inspired by the concept of genus and differentia, we design the PP encoder to characterize unknown categories by measuring their metric differences from known ones. It constructs Prior-Prototypes using the base training set and derives PP embeddings for novel samples by comparing them against these prototypes.
- **Self-Consistency Diffusion module**: By incorporating reconstruction constraints, the Self-Consistency Diffusion Module (SCDM) enhances the quality of hash codes by increasing their sensitivity to fine-grained feature differences. Additionally, the diffusion mechanism promotes sample collisions, facilitating information exchange and further refining the feature encoding process.

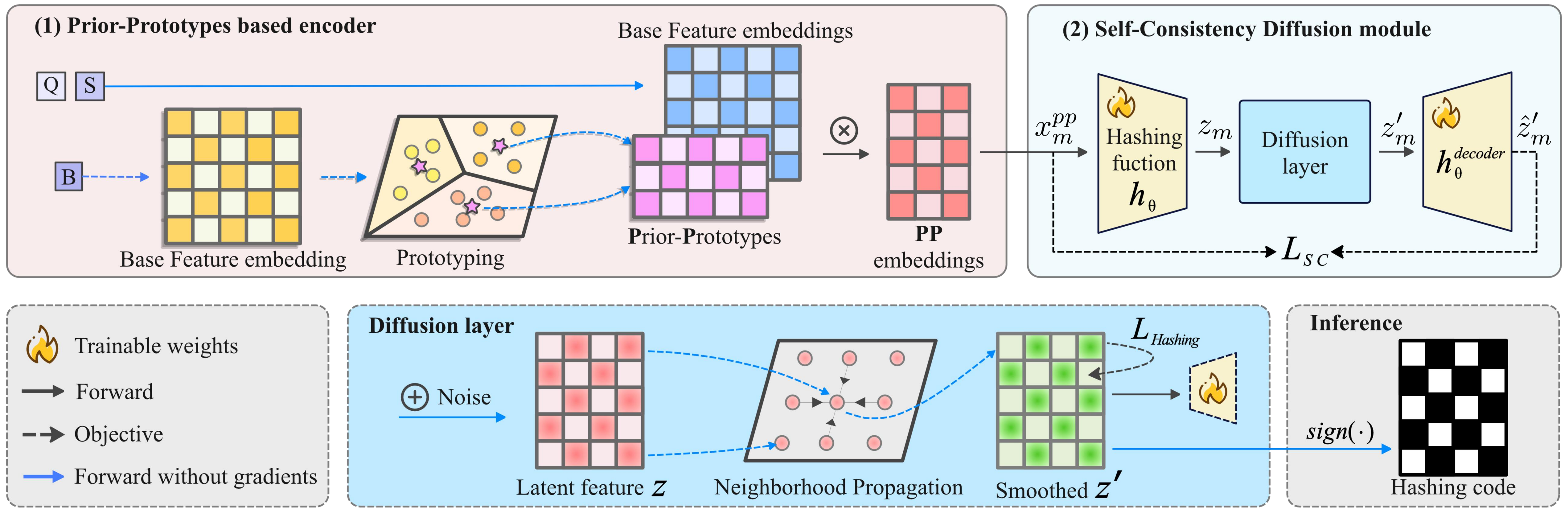

Fig 2. Overview of the PP framework.

(1) Prior-Prototypes based encoder:

a. Freeze the feature extractor f_θ^* and compute Prior-Prototypes for each base class C in train set, serving as the genus.

$$\mathbf{p}_c = \frac{1}{|\mathcal{C}_c|}\sum_{x_n\in\mathcal{C}_c} f_\theta^*(x_n), \quad c\in\mathcal{Y}^{\text{base}}$$

b. For a novel sample x_m, extract its base feature via $f_\theta^*(x_m)$ and compute distances d_m^c to each $\mathbf{p}_c$, capturing the differentia:

$$d_m^c = \|f_\theta^*(x_m) - \mathbf{p}_c\|_2$$

c. Define PP embedding x_m^{pp} as a similarity-based embedding following Aristotle's genus-and-differentia principle.

$$x_m^{pp} = [d_m^1, d_m^2, \ldots, d_m^{N_C}] \in \mathcal{E}^{pp}_{\mathcal{D}\text{adapt}}$$

(2) Self-Consistency Diffusion module:

a. Apply diffusion to the hash head output via:

$$z'_m = z_m - \alpha\cdot\mathbf{L}\cdot z_m$$

where z_m is the latent representation from the hashing head, and $\mathbf{L}$ is the Laplacian matrix built from sample-wise distances.

b. A lightweight mirrored decoder $h_\theta^{\text{decoder}}$ is used to reconstruct the PP embedding from the diffused feature , providing a self-consistency constraint that guides the hash function to retain semantic structure:

$$\mathcal{L}_{\text{SC}} = \|\hat{z}'_m - x_m^{pp}\|_2$$

Results

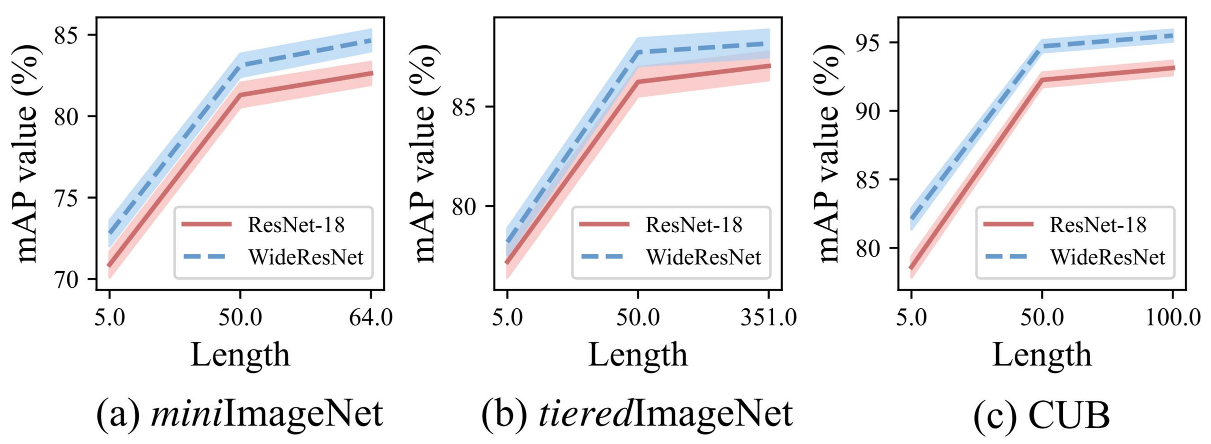

Fig 3. mAP (%) of PP embeddings with different numbers of PPs and 95% confidence intervals. The performance generally improves with more PPs, indicating a positive but nonlinear correlation. Using only 50 PPs already achieves strong results.

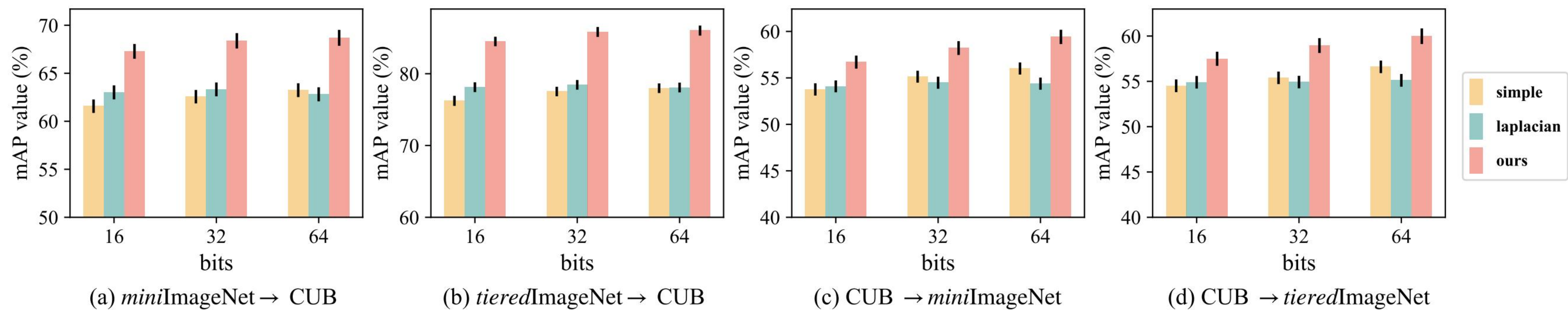

Fig 4. mAP (%) across datasets with 95% confidence intervals (ResNet-18 as feature extractor).

Tab 1. Step-wise inference time cost on *tiered*ImageNet with 351 PPs, 512-dimensional feature representations, 64-bit hash codes, and 6 diffusion layers.

No.	Inference Step	Runtime (ms)
①	Feature Extraction (ResNet-18)	5.095
②	Hashing Head	0.049
③	PP Encoder	0.056
④	SCDM	0.097

Fig 5. QR code to the GitHub repository with additional implementation details and the full paper.