HIDISC: A Hyperbolic Framework for Domain Generalization with Generalized Category Discovery

Vaibhav Rathore[1] **Divyam Gupta**[1] **Biplab Banerjee**[1]
Indian Institute of Technology Bombay[1]
{vaibhav.rathor.in, divs25.iitb, getbiplab}@gmail.com

Abstract

Generalized Category Discovery (GCD) aims to classify test-time samples into either seen categories—available during training—or novel ones, without relying on label supervision. Most existing GCD methods assume simultaneous access to labeled and unlabeled data during training and arising from the same domain, limiting applicability in open-world scenarios involving distribution shifts. Domain Generalization with GCD (DG-GCD) lifts this constraint by requiring models to generalize to unseen domains containing novel categories, without accessing target-domain data during training. The only prior DG-GCD method, DG^{2}CD-Net [1], relies on episodic training with multiple synthetic domains and task vector aggregation, incurring high computational cost and error accumulation. We propose HIDISC, a hyperbolic representation learning framework that achieves domain and category-level generalization without episodic simulation. To expose the model to minimal but diverse domain variations, we augment the source domain using GPT-guided diffusion, avoiding overfitting while maintaining efficiency. To structure the representation space, we introduce *Tangent CutMix*, a curvature-aware interpolation that synthesizes pseudo-novel samples in tangent space, preserving manifold consistency. A unified loss—combining penalized Busemann alignment, hybrid hyperbolic contrastive regularization, and adaptive outlier repulsion—facilitates compact, semantically structured embeddings. A learnable curvature parameter further adapts the geometry to dataset complexity. HIDISC achieves state-of-the-art results on PACS [2], Office-Home [3], and DomainNet [4], consistently outperforming the existing Euclidean and hyperbolic (DG)-GCD baselines.[1]

1 Introduction

Deep neural networks have achieved impressive success in visual recognition [5, 6], yet typically assume a shared domain and label space between training and test data. This assumption breaks down in real-world applications such as autonomous driving [7] and medical diagnostics [8], where both *domain shift* and *label shift* frequently co-occur. While semi/self-supervised learning [9, 10] reduces labeling demands, it still operates under closed-world constraints. Domain Adaptation (DA) and Domain Generalization (DG) [11, 12] address distribution shift but assume a fixed set of categories. Open-set DG [13, 14] allows test-time novelty but collapses all unknowns into a single rejection class, erasing semantic granularity.

Generalized Category Discovery (GCD) [15–17] seeks to identify both known and novel classes from unlabeled test data but requires joint access to labeled and unlabeled samples from the same domain. Cross-Domain GCD (CD-GCD) [18, 19] introduces domain shift but still assumes concurrent access to source–target domains during training. In contrast, **Domain Generalization with GCD (DG-**

[1]Code Link : https://vaibhavrathore1999.github.io/HiDISC/

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

GCD) [1] represents a more realistic setting: *the model is trained solely on labeled source data and must generalize to an unseen target domain containing both seen and novel categories.*

Addressing DG-GCD requires (i) learning domain-invariant features and (ii) discovering novel semantic structures without supervision. The only existing solution tailored for DG-GCD, DG^{2}CD-Net [1] approaches this via episodic training with synthetic domains and task aggregation, but suffers from high computational cost and cumulative approximation errors that limit generalization.

From a different perspective, existing GCD and DG-GCD methods typically rely on Euclidean or hyperspherical geometry [1, 18], which struggle to capture semantic hierarchies. Hyperbolic geometry [20, 21], with its negative curvature and exponential volume growth, offers a natural alternative for modeling inter-class structure (Fig. 1). While hyperbolic embeddings have shown benefits in GCD (HypCD) [22] and DG [23] recently, their use in DG-GCD, where both domain and label shifts co-occur, remains unexplored. This raises our central question:

> *Can hyperbolic geometry provide a unified foundation for solving DG-GCD, addressing both distribution shift and novel-class discovery?*

Our approach. We introduce HIDISC, the first hyperbolic geometry-aware framework for DG-GCD that learns semantically structured and domain-invariant embeddings in the Poincaré ball [20] without requiring target supervision. Unlike HypCD [22], which addresses standard GCD in single-domain settings, and DG^{2}CD-Net [1], which operates in Euclidean space and relies on episodic simulation, HIDISC provides a unified, non-episodic solution using the representational advantages of hyperbolic space.

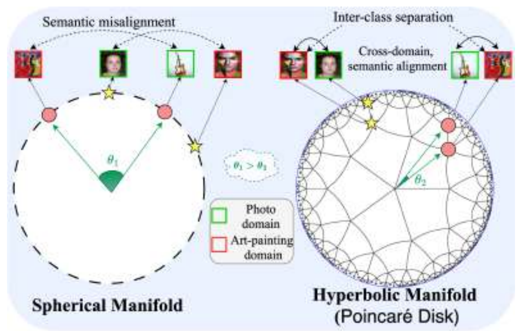

Figure 1: **Spherical** vs. **hyperbolic** (Poincaré) embeddings on PACS. Same-class samples from different domains (green/red) cluster more tightly in hyperbolic space, demonstrating improved class separation. Refer to **Sup. Mat.** for quantitative analysis.

Since the hyperbolic space offers substantial domain invariance and focuses on shared semantics [23] (Fig. 1, **Sup. Mat.**), we still chose to augment the training source domain with controlled stylistic variations and generate only 1–2 synthetic domains per image via a GPT-4o [24]-guided diffusion model, avoiding the computational overhead of task-based simulation in DG^{2}CD-Net. A novel *domain-diversity score* ranks these augmentations by measuring source divergence and intra-pair variability, enabling a principled and scalable domain diversification strategy.

To ensure the model does not overfit the known classes and encourage semantic diversity, we propose *Tangent CutMix*, a curvature-aware interpolation method that mixes labeled features in the tangent space of the Poincaré ball to create pseudo-open samples. Unlike Euclidean mixing used in SimGCD [25] or CMS [26], our method preserves hyperbolic consistency and generates geometrically valid pseudo-novel embeddings that support open-set regularization.

To structure the latent space, we introduce a unified loss that combines three novel components not jointly explored in prior (DG)-GCD literature: (i) a penalized Busemann loss that aligns seen-class features to fixed prototypes at the hyperbolic boundary while reserving interior space for unknowns; (ii) a hybrid hyperbolic contrastive loss balancing angular and geodesic similarities to enable fine-grained clustering across known and novel categories; and (iii) an adaptive outlier rejection loss that pushes synthetic cut-mix samples away from known-class regions, encouraging open-space generalization without relying on adversarial or domain-specific objectives. A learnable curvature parameter further adapts the geometry to dataset-specific complexity. Major contributions include:

- HIDISC, the first hyperbolic DG-GCD framework, jointly handles domain and category shift without target supervision or episodic simulation.
- A unified loss formulation integrating Busemann alignment, hybrid contrastive regularization, and an adaptive outlier repulsion.
- Tangent CutMix, the first open-set augmentation designed specifically for hyperbolic geometry.

– State-of-the-art results on PACS [2], Office-Home [3], and DomainNet [4], outperforming all the baselines consistently and reducing training FLOPs by over $96\times$ vs. DG^{2}CD-Net.

2 Related Works

Domain Generalization. DG aims to train models on labeled data from one or more source domains to generalize effectively to previously unseen target domains [12, 27]. DG variants include closed-set, open-set, single-source, and multi-source settings [28]. Methods like MixStyle [29] and StyleHallucination [30] enhance robustness via feature-level style perturbations, while meta-learning techniques [31, 32] simulate domain shifts episodically to improve adaptability. *Open-set DG methods address novel test-time classes [33, 34, 14, 35], but typically collapse all unknowns into a single "outlier" class, hindering fine-grained discovery needed in DG-GCD.*

Category Discovery. Category Discovery seeks to partition unlabeled data into known and novel categories. While Novel Category Discovery (NCD) [36] assumes complete disjointness between training and test classes, GCD allows overlap and requires identifying both seen and unseen categories during inference [15–17, 25, 37, 38, 26]. Most GCD approaches assume joint access to source and target domains during training, limiting real-world applicability. CD-GCD methods like CDAD-Net [18] and HiLo [28] reduce domain gaps via adversarial alignment or style normalization but still depend on concurrent domain access. To remove this constraint, DG-GCD [1] simulates domain shifts through text-driven image manipulation (e.g., InstructPix2Pix [39]) and aggregates task-specific knowledge via task vectors [40]. *However, these methods operate in Euclidean spaces, which struggle to encode the hierarchical and shared semantic structures crucial for robust domain and category generalization.*

Hyperbolic Embedding Spaces. Hyperbolic geometry, defined by negative curvature and exponential volume growth, is well-suited for modeling hierarchical and part-whole semantic structures [20, 21]. Hyperbolic embeddings have improved performance in classification [41–43], few-shot learning [44], segmentation [45], and action recognition [46], supported by hyperbolic variants of standard network components [47, 48, 21, 49]. Recent Busemann-based techniques [23, 50] anchor ideal prototypes on the Poincaré boundary for directional alignment. HypCD [22] successfully applies this to GCD, but assumes joint source–target access. Beyond these, hyperbolic methods have been applied across diverse tasks: Ge et al. [51] explore contrastive learning for hierarchical scene–object representation, Yue et al. [52] study metric learning with hard negatives, Liu et al. [53] extend contrastive learning to EEG, Sun & Ma [54] investigate recommendation, while others address hashing [55] and face anti-spoofing with hierarchical prototypes [56]. *To date, no work has explored hyperbolic representations for DG-GCD, which combines open-set discovery and domain shift without target supervision.*

3 Methodology

3.1 The DG-GCD Problem Definition

In DG-GCD, we are given a labeled source-domain dataset:

$$D_S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}, \quad x_i^s \in \mathcal{X}_s,\ y_i^s \in \mathcal{Y}_s,$$

where x_i^s represents source-domain inputs, and y_i^s denotes labels drawn from a set of *known* categories $\mathcal{Y}_s$. At test time, we encounter an unlabeled target-domain dataset:

$$D_T = \{x_j^t\}_{j=1}^{n_t}, \quad x_j^t \in \mathcal{X}_t,$$

where samples belong either to known categories ($\mathcal{Y}_t^{\text{old}} = \mathcal{Y}_s$) or previously unseen, *novel* categories ($\mathcal{Y}_t^{\text{new}}$), such that $\mathcal{Y}_t^{\text{new}} \cap \mathcal{Y}_s = \emptyset$. Crucially, the data distributions across domains differ significantly, i.e., $P(\mathcal{X}_s) \neq P(\mathcal{X}_t)$, and the target dataset D_T is inaccessible during training.

Our objective is to construct an embedding space using only D_S that generalizes across domains and categories, effectively clustering both known and novel-class samples from the unseen dataset D_T.

3.2 Rationale Behind Using Hyperbolic Space for DG-GCD

Semantic structures in visual data—such as hierarchies, taxonomies, and part–whole relationships—are inherently suited to spaces with exponential capacity. Hyperbolic space, characterized

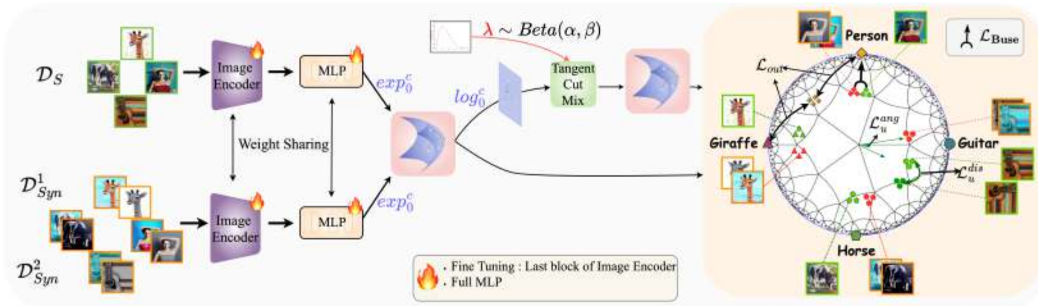

Figure 2: **Illustration of the HIDISC pipeline for DG-GCD in hyperbolic space**. The model is trained using labeled source data $\mathcal{D}_S$ (green borders) and 1–2 GPT-guided synthetic domains $\mathcal{D}^1_{\text{Syn}}$, $\mathcal{D}^2_{\text{Syn}}$ (orange borders) to simulate domain shift. Features from the shared encoder are projected to the Poincaré ball via $\exp_0^c$. To mimic novel categories, Tangent CutMix performs interpolation in the tangent space and maps the result z_{mix} back to hyperbolic space. The embedding space is structured via: (i) penalized Busemann loss $\mathcal{L}_{\text{Buse}}$ for aligning seen classes to boundary-fixed prototypes; (ii) hybrid contrastive loss $\mathcal{L}_u$ for clustering and separability; and (iii) adaptive outlier loss $\mathcal{L}_{\text{out}}$ to repel pseudo-novel points. Together, these shape a curvature-aware space for generalization and discovery.

by negative curvature and exponential volume growth, naturally encodes such structures, making it particularly beneficial for DG-GCD, where labeled classes typically reflect coarse semantic strata, while novel categories often reside in finer or more abstract regions.

In contrast to Euclidean or spherical embeddings [16, 57], which are constrained by polynomial growth, hyperbolic embeddings support both local compactness and global semantic separation. Moreover, hyperbolic geometry improves domain generalization by amplifying higher-level semantic distances and attenuating domain-specific low-level variations, thereby fostering robust, domain-invariant representations under substantial distributional shifts (more details provided in **Sup. Mat**).

Poincaré Ball Geometry. We adopt the Poincaré ball [20] as our hyperbolic model. For curvature $-c^2$, the n-dimensional ball is defined as:

$$\mathbb{D}_c^n = \left\{ \mathbf{a} \in \mathbb{R}^n \;\middle|\; c\|\mathbf{a}\|^2 < 1 \right\},$$

where $\|\cdot\|$ denotes the Euclidean norm. Additional geometric details are provided in the **Sup. Mat.**

3.3 Navigating through HIDISC for DG-GCD

We propose HIDISC (Fig. 2), a hyperbolic framework that jointly addresses the dual challenges of DG-GCD: domain-invariant representation learning and unsupervised semantic disentanglement. The synthesis-driven components of our model include: (i) *Synthetic Domain Augmentation*, which introduces a compact set of diverse, diffusion-generated domains to simulate realistic distribution shifts without relying on target access; and (ii) *Tangent CutMix*, a curvature-aware interpolation mechanism operating in the tangent space of the Poincaré ball, generating pseudo-novel samples while preserving manifold fidelity. Complementing these are three loss-driven modules: (iii) *Prototype Anchoring*, which aligns seen-class embeddings to fixed ideal prototypes on the Poincaré boundary, reserving central space for novel classes; (iv) *Adaptive Outlier Loss*, which ensures synthetic samples are repelled from known-class clusters, promoting open-space regularization; and (v) *Hybrid Hyperbolic Contrastive Loss*, which combines geodesic and angular similarity to improve local cohesion and global separability. Each component is described in detail in the subsequent sections.

3.3.1 Lightweight Synthetic Domain Augmentation

To simulate domain variability without relying on expensive episodic training (as in DG^{2}CD-Net [1]), we generate only one or two synthetic domains per experiment using a diffusion model guided by GPT-4o [24]-curated prompts (e.g., `underwater`, `night-time` variants of *class* instances) (see **Sup. Mat.** for qualitative visualizations). These synthetic domains serve as proxy distributions that expose the model to varied visual shifts and support generalization to unseen domains.

Unlike DG^{2}CD-Net, which depends on numerous episodic tasks and synthetic domain permutations, our strategy is lightweight and avoids both computational overhead and error propagation across episodes. Crucially, in hyperbolic space, even a small number of diverse augmentations can induce expansive representational changes due to the geometry's exponential capacity—effectively stretching the semantic space and encouraging separation between seen and unseen regions.

First, we introduce a *domain-diversity score* that ranks a given synthetic domain $\mathcal{D}_{\text{syn}}^{(s)}$ with respect to other synthetic domains $\{\mathcal{D}_{\text{syn}}^{(l)}\}_{l=1}^{\mathcal{M}}$ and the source domain $\mathcal{D}_S$ based on the notion of mutual diversity calculated using the Fréchet Inception Distance (FID) [58] for $\mathcal{M}$ synthesized domains:

$$\text{Score}(\mathcal{D}_{syn}^{(s)}) = \frac{1}{\mathcal{M}-1}\sum_{\substack{l=1,\\ l\neq s}}^{\mathcal{M}}\Big[\text{FID}(\mathcal{D}_S, \mathcal{D}_{\text{syn}}^{(s)}) + \text{FID}(\mathcal{D}_{\text{syn}}^{(s)}, \mathcal{D}_{\text{syn}}^{(l)})\Big]. \tag{1}$$

This scoring promotes both source-domain divergence and intra-pair complementarity. We select the top-scoring $1-2$ domains to augment $\mathcal{D}_S$ to obtain $\mathcal{D}_{\text{train}}$.

As shown in Fig. 3, excessive augmentation leads to overfitting on seen classes and degrades novel class discovery. *Notably, training solely on $\mathcal{D}_S$ yields competitive results, highlighting the inherent domain robustness of hyperbolic space, and using these augmentation provides marginal boosts.* Effects of redundant augmentations are mentioned in **Sup. Mat.**

3.3.2 Mapping Visual Features into Curvature-Aware Hyperbolic Geometry

With the augmented training set, we learn representations using a frozen DINO [59]-pretrained ViT [60] followed by a 3-layer MLP. The resulting Euclidean feature $\mathbf{z}^{\mathbb{E}} \in \mathbb{R}^d$ is projected into the Poincaré ball $\mathbb{D}_c^d$ via the exponential map:

$$\mathbf{z}_i^s = \exp_0^c(\mathbf{z}^{\mathbb{E}}) = \tanh(\sqrt{c}\|\mathbf{z}^{\mathbb{E}}\|)\cdot\frac{\mathbf{z}^{\mathbb{E}}}{\sqrt{c}\|\mathbf{z}^{\mathbb{E}}\|}, \tag{2}$$

where c is a learnable curvature parameter in our case to approximate the data complexity more effectively. Let $z_i := \mathbf{z}_i^s$ denote the hyperbolic feature. This projection facilitates the encoding of hierarchical semantics and ensures geometric consistency in the downstream tasks.

3.3.3 Tangent CutMix and Adaptive Outlier Loss

To hallucinate novel-category samples and regularize the open space in hyperbolic geometry, we introduce **Tangent CutMix** [61]—a curvature-aware variant of CutMix tailored for the Poincaré ball. Traditional CutMix interpolates feature representations in Euclidean space to synthesize outliers, which can violate the geometric constraints of hyperbolic space. In contrast, Tangent CutMix performs mixing in the tangent space at the origin, ensuring consistency with the underlying manifold structure.

Given two embeddings $z_i, z_j \in \mathbb{D}_c^d$ with different class labels, we:

(1) **Project to tangent space:** $v_i = \log_0^c(z_i),\ v_j = \log_0^c(z_j)$

(2) **Linear mix:** Compute $v_{\text{mix}}^{i,j} = \lambda v_i + (1-\lambda)v_j$, where $\lambda \sim \text{Beta}(1,1) = \text{Uniform}(0,1)$

(3) **Map back:** $z_{\text{mix}}^{i,j} = \exp_0^c(v_{\text{mix}})$

The resulting embedding z_{mix} represents a curvature-preserving interpolation of features with incompatible semantics, mimicking out-of-distribution behavior while remaining valid in the hyperbolic space. Furthermore, to prevent these synthetic features from collapsing into known class regions, we apply an adaptive outlier loss:

$$\mathcal{L}_{\text{out}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}(\mathcal{D}_{\text{train}})}\sum_{i,j,y_i\neq y_j}\max(0, \gamma - \min_{k\in\mathcal{Y}_s}\mathbb{D}_{\mathbb{H}}(z_{\text{mix}}^{i,j}, \mathbf{p}_k)), \tag{3}$$

where $\mathbb{D}_{\mathbb{H}}$ is the hyperbolic distance, and γ is a quantile-based adaptive margin over the distances from all class prototypes in $\{\mathbf{p}_k\}_{k=1}^{|\mathcal{Y}_s|}$. This encourages pseudo-novel embeddings to remain outside the regions occupied by seen classes, effectively reserving space for novel category discovery. For further analysis regarding adaptive margin and the generated CutMix samples, see **Sup. Mat.**

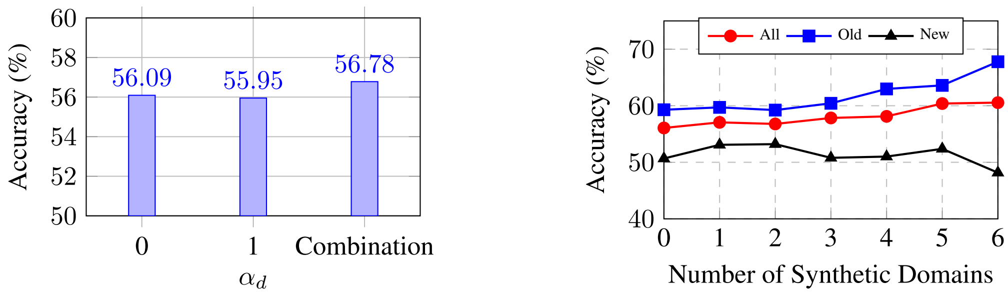

Figure 3: **(Left)** Effect of α_d in hybrid contrastive loss. A balanced combination of angular and geodesic components achieves the highest accuracy. **(Right)** Impact of synthetic domains on old and new category performance. While old-class accuracy increases due to augmented seen data, new-class performance slowly degrades with more synthetic domains, as they cause seen-class bias.

3.3.4 Prototype Anchoring with Penalized Busemann Loss

To enforce semantic structure among the known categories, we associate each class $k \in \mathcal{Y}_s$ with a fixed *ideal prototype* $\mathbf{p}_k \in \partial\mathbb{D}_c^d$ [23], placed uniformly on the boundary of the Poincaré ball. These prototypes serve as directional anchors and remain fixed throughout training, enabling compact clustering of seen-class features while leaving the interior volume of the ball available for unknown category discovery.

To align the features z_i with their respective class prototypes, we adopt a **penalized Busemann loss**:

$$\mathcal{L}_{\text{Buse}} = \log\left(\frac{\|z_i - \mathbf{p}_{y_i}\|^2}{1 - c\|z_i\|^2}\right) + \phi \log(1 - \|z_i\|^2), \tag{4}$$

where $\mathbf{p}_{y_i}$ is the prototype corresponding to the class label of z_i, and ϕ is a regularization coefficient. The first term guides directional alignment between features and their prototypes, preserving semantic proximity in the hyperbolic geometry. The second term penalizes embeddings that approach the boundary too aggressively, thereby maintaining stability during optimization and avoiding overconfidence.

3.3.5 Hybrid Hyperbolic Contrastive Loss

While the Busemann loss anchors known classes via directional alignment, it does not explicitly enforce local structure among unlabeled or novel samples. To address this, we incorporate a **hybrid hyperbolic contrastive loss** [22], designed to refine the latent space by encouraging consistency between augmented views and separating unrelated instances—even in the absence of explicit labels.

For each positive pair of embeddings z_i', z_i'', corresponding to different augmentations of the same input, we define the contrastive objective as:

$$\mathcal{L}_u = \frac{1}{|B|} \sum_{i \in B} -\log \frac{\exp(\delta(z_i'', z_i')/\tau)}{\sum_{j \neq i} \exp(\delta(z_i', z_j)/\tau)}, \tag{5}$$

where τ is a temperature hyperparameter and B is the batch of samples. We use a **hybrid similarity function** $\delta(.,.)$, which linearly combines distance-based and angle-based measures:

$$\delta(.,.) = \alpha_d \cdot \underbrace{[-\mathbb{D}_{\mathbb{H}}(.,.)]}_{L_u^{dis}} + (1-\alpha_d) \cdot \underbrace{\cos(.,.)}_{L_u^{ang}}, \tag{6}$$

$\cos(\cdot,\cdot)$ computes cosine similarity in the tangent space, thanks to the co-conformality of the Euclidean and Hyperbolic spaces. α_d is the balancing factor (for more details see **Sup. Mat.**).

This hybrid formulation leverages the metric structure of hyperbolic space to promote global semantic separation via geodesic distances, while retaining angular consistency within local neighborhoods. Fig. 3 shows the importance of the full δ over the individual distance metrics.

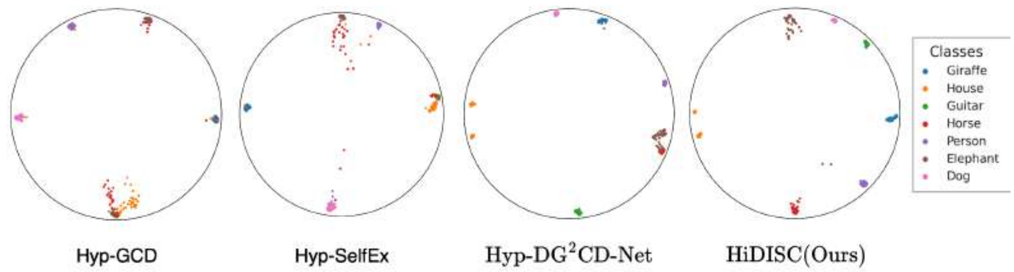

Figure 4: **Poincaré-disk UMAP [62] embeddings** of the target domain ("Photo") clusters, as produced by Hyp-GCD [22], Hyp-SelfEx [22], Hyp-DG^{2}CD-Net, and HIDISC(Ours) for the PACS dataset, with "Sketch" as the source. HIDISC produces a visually clean and compact embedding space, supported by **silhouette scores** [63] ($\in [-1.1]$, $\uparrow$), indicating improved cluster compactness and separation: (Hyp-GCD: **-0.52**, Hyp-SelfEx: **-0.42**, Hyp-DG^{2}CD-Net: **-0.29**, HIDISC: **-0.14**)

3.3.6 Training Objective

Our final *minimization* objective integrates all components:

$$\mathcal{L}_{\text{total}} = \lambda_1 \underbrace{\mathcal{L}_{\text{Buse}}}_{\text{Semantic alignment}} + \lambda_2 \underbrace{\mathcal{L}_{u}}_{\text{Contrastive regularization}} + \lambda_3 \underbrace{\mathcal{L}_{\text{out}}}_{\text{Outlier repulsion}}, \quad \text{where } \lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{7}$$

Test-time Protocol. After training, we extract hyperbolic features from target-domain samples and perform clustering using K-Means as in [1, 15]. See **Sup. Mat.** for detailed algorithm.

3.4 Theoretical Justification: Generalization in Euclidean vs. Hyperbolic Spaces

We analyze HIDISC under the lens of generalization theory in hyperbolic space. Given $\mathcal{D}_S$, $\mathcal{D}_T$, and synthetic augmentations $\{\mathcal{D}_{\text{syn}}^{(l)}\}_{l=1}^{\mathcal{M}}$, the goal is to minimize the expected target risk:

$$\mathcal{L}_T(f) = \mathbb{E}_{x\sim T}[\ell(f(x))], \tag{8}$$

Extending Rademacher-based analysis to the Poincaré ball $\mathbb{D}_c^d$ [21], we obtain:

$$\mathcal{L}_T(f) \leq \mathcal{L}_{S'}(f) + \Delta_{\mathbb{H}}(S', T) + \mathcal{R}_{\mathbb{H}}(\mathcal{H}) + \epsilon, \tag{9}$$

where: $\mathcal{L}_{S'}(f)$: empirical loss over the augmented training set S'; $\Delta_{\mathbb{H}}(S', T)$: hyperbolic discrepancy between augmented source and target; $\mathcal{R}_{\mathbb{H}}(\mathcal{H})$: Rademacher complexity of the hypothesis class $\mathcal{H}$; ϵ: a residual optimization error.

Compared to the Euclidean bound $\mathcal{L}_T^{\mathbb{E}}(f) \leq \mathcal{L}_{S'}(f) + \Delta_{\mathbb{E}}(S', T) + \mathcal{R}_{\mathbb{E}}(\mathcal{H}) + \epsilon$, the hyperbolic version benefits from the exponential volume and hierarchical structure of $\mathbb{D}_c^d$. This allows semantically distant concepts to be placed further apart with less distortion and curvature-driven compression around known classes—thereby making fewer, well-chosen augmentations sufficient to span the generalization space. As such, $\Delta_{\mathbb{H}}(S', T) < \Delta_{\mathbb{E}}(S', T)$ holds under the same augmentation budget, yielding a tighter bound (see **Sup. Mat.** for a formal proof).

Each loss in HIDISC contributes to improving specific terms: **(i)** The Busemann loss $\mathcal{L}_{\text{Buse}}$ aligns seen-class features to ideal prototypes at the boundary, stabilizing $\mathcal{L}_{S'}(f)$ via directional compactness; **(ii)** The hybrid contrastive loss $\mathcal{L}_u$ integrates angular and geodesic similarity to encourage semantically meaningful clusters and reduce model complexity $\mathcal{R}_{\mathbb{H}}(\mathcal{H})$; **(iii)** The outlier loss $\mathcal{L}_{\text{out}}$, applied on Tangent CutMix samples, helps partition the open space, reducing false positives on novel categories without explicit domain alignment; **(iv)** The curated synthetic domains $\{\mathcal{D}_{\text{syn}}^{(l)}\}$ enrich S', approximating T's support and reducing $\Delta_{\mathbb{H}}(S', T)$ in a geometry-consistent manner.

In **Sup. Mat.**, we show that FID-based estimates of $\Delta_{\mathbb{H}}$ yield minimal improvement over the inherent domain-independence of hyperbolic geometry. We also compare our loss terms in Euclidean and hyperbolic spaces, demonstrating that hyperbolic geometry better reduces the generalization gap.

4 Experimental Evaluations

Dataset Details. We evaluate our method on three standard DG-GCD benchmarks: PACS [2], Office-Home[3], and Domain-Net[4]. We follow the protocol of [1] for constructing known/novel class-splits and source-target domain pairs. The dataset details are provided in the **Sup. Mat**.

Table 1: **Comparison of clustering accuracy** (%) for known (Old), novel (New), and overall (All) categories across PACS, Office-Home, and DomainNet. It can be seen that HIDISC beats other synthetic domain augmentation based baselines using significantly less number of synthetic domains (from 6/9 to 2). (**Bold** : best , <u>underline</u> : second best).

Method	Venue	PACS			Office-Home			DomainNet			Avg.		
		All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	ICLR'21	41.98	50.91	33.16	26.17	29.13	21.62	25.35	26.48	22.41	31.17	35.51	25.73
GCD [15]	CVPR'22	52.28	62.20	38.39	52.71	54.19	50.29	27.41	27.88	26.13	44.13	48.09	38.27
SimGCD [25]	ICCV'23	34.55	38.64	30.51	36.32	49.48	13.55	2.84	2.16	3.75	24.57	30.09	15.94
CMS [26]	CVPR'24	28.95	28.13	36.80	10.02	9.66	10.53	2.33	2.40	2.17	13.77	13.40	16.50
SelfEx [64]	ECCV'24	71.82	73.37	71.55	50.18	48.59	52.16	24.78	24.99	24.21	48.93	48.98	49.31
CDAD-Net [18]	CVPR-W'24	69.15	69.40	68.83	53.69	<u>57.07</u>	47.32	24.12	23.99	24.35	48.99	50.15	46.83
GCD+ 6 Synth	CVPR'22	65.33	67.10	64.42	50.50	51.48	48.96	24.71	24.80	21.94	46.85	47.78	45.11
SimGCD+ 6 Synth	ICCV'23	39.76	43.76	35.97	35.57	48.58	12.89	2.71	1.99	4.14	26.01	31.44	17.67
CMS+ 6 Synth	CVPR'24	28.01	26.71	29.04	12.09	12.66	11.13	3.22	3.28	3.03	14.44	14.22	14.40
CDAD+ 6 Synth	CVPR-W'24	60.76	61.67	59.49	53.49	56.90	47.76	23.85	23.88	24.26	46.03	47.47	43.84
Hyp-GCD [22]	CVPR'25	65.33	67.11	64.42	50.13	49.36	48.08	22.88	23.74	25.89	46.12	46.74	46.13
Hyp-SelfEx [64]	ECCV'24	72.44	74.70	71.20	52.91	52.65	52.96	<u>29.30</u>	<u>30.45</u>	<u>26.37</u>	51.55	52.60	50.18
DG^{2}CD-Net [1] (9 Synth)	CVPR'25	73.30	<u>75.28</u>	72.56	<u>53.86</u>	53.37	**54.33**	29.01	30.38	25.46	<u>52.06</u>	<u>53.01</u>	<u>50.78</u>
Hyp-DG^{2}CD-Net †(9 Synth)	CVPR'25	<u>74.07</u>	74.40	<u>73.95</u>	49.40	50.29	48.03	22.31	21.52	24.29	48.59	48.74	48.76
HIDISC (Ours) (2 Synth)	–	**75.07**	**75.54**	**74.52**	**56.78**	**59.23**	<u>53.21</u>	**30.51**	**31.40**	**28.41**	**54.12**	**55.39**	**52.05**
Δ	–	+1.00	+0.26	+0.57	+2.92	+2.16	–1.12	+1.21	+0.95	+2.04	+2.06	+2.38	+1.27
CDAD-Net (DA) [UB]	CVPR-W'24	83.25	87.58	77.35	67.55	72.42	63.44	70.28	76.46	65.19	73.69	78.82	68.66

Evaluation Metrics. Following [15, 18, 1], we evaluate clustering using three metrics: **Old** (accuracy on known classes $\mathcal{Y}_t^{\text{old}}$), **New** (accuracy on novel classes $\mathcal{Y}_t^{\text{new}}$), and **All** (overall accuracy on $\mathcal{D}_T$). Hungarian matching is used to align predicted clusters with ground-truth labels. Scores are averaged over three runs and all source-target combinations. Further experimental details and hyper-parameter choices are mentioned in **Sup. Mat.**

4.1 Comparisons to the Literature

Table 1 compares our proposed HIDISC against state-of-the-art methods on the said datasets. Baselines are categorized into four groups: (i) **Euclidean source-only GCD methods**, including GCD [15], SimGCD [25], CMS [26], and SelfEx [64]; (ii) **Synthetic augmentation-based GCD methods**, such as SimGCD+Synthetic, CMS+Synthetic, and CDAD-Net+Synthetic [18], which incorporate domain-shifted images via diffusion-based generation; (iii) **Hyperbolic GCD methods**, including Hyp-GCD [22] and Hyp-SelfEx, which project features into hyperbolic space to improve clustering but do not generalize across domains. To ensure consistency with the DG-GCD setting, we retain only the components of these methods that rely on labeled data during training and omit terms involving unlabeled samples in all the above baselines, as recommended in [1]; and (iv) the **DG-GCD baseline** DG^{2}CD-Net [1], which simulates multiple domains using diffusion models and aggregates task-level knowledge via episodic training and task vectors. For a fairer comparison, we also implement a hyperbolic variant, Hyp-DG^{2}CD-Net, by replacing its embedding space with a Poincaré ball. As in [1], we report results for CDAD-Net [18] under joint access to source and target domains as an upper bound of our results.

Table 2: **Estimated number of clusters**. Correct estimates are in green, small errors in orange, and large deviations in red.

Method	PACS	Office-Home	DomainNet
Ground Truth	7	65	345
DG^{2}CD-Net	7	67	355
CDAD-Net (DG)	12	60	362
CDAD-Net (DA)	7	66	349
HIDISC (Ours)	7	66	351

Quantitatively, HIDISC achieves state-of-the-art performance across all metrics and datasets. It improves upon DG^{2}CD-Net by +2.06% in average overall clustering accuracy and by +1.27% on novel class discovery. On DomainNet—the most diverse and challenging benchmark—HIDISC outperforms the best previous method by +1.21%. UMAP visualizations (Fig. 4) show HIDISC forms a compact embedding space. These gains are achieved without target access and with over **96× lower training FLOPs** than [1] while using the same number of synthetic domains (see **Sup. Mat**). On the other hand, the performance of DG^{2}CD-Net degrades drastically as the number of synthetic domains is reduced (see **Sup. Mat.**)

Table 3: Impact of **loss components** of HIDISC on Office-Home

Config.	$\mathcal{L}_{\text{Buse}}$	$\mathcal{L}^{u}_{\text{hrep}}$	$\mathcal{L}_{\text{out}}$	Office-Home All	Old	New
Vanilla	✗	✗	✗	26.17	29.13	21.62
$\mathcal{L}_{\text{Buse}}$	✓	✗	✗	56.32	59.74	50.32
$\mathcal{L}^{u}_{\text{hrep}}$	✗	✓	✗	50.95	49.33	53.06
$\mathcal{L}_{\text{Buse}}+\mathcal{L}^{u}_{\text{hrep}}$	✓	✓	✗	56.29	60.36	50.41
$\mathcal{L}_{\text{Buse}}+\mathcal{L}_{\text{out}}$	✓	✗	✓	51.04	51.51	50.29
Full HIDISC	✓	✓	✓	**56.78**	**59.23**	**53.21**

Table 4: Performance metrics demonstrating the **influence of key model components** of HIDISC for Office-Home.

Model Variant	Office-Home All	Old	New
- With manual augmentations based $\mathcal{D}_{\text{syn}}$	50.80	51.75	49.15
- Without synthetic domain	56.07	59.29	50.67
- Fixed curvature (c=0.01, close to Euclidean)	56.23	58.65	52.68
- Fixed curvature (c=0.03)	55.67	57.39	52.69
- Cut-Mix (In Euclidean Space)	53.46	54.86	51.06
- Full HIDISC	**56.78**	**59.23**	**53.21**

Furthermore, Table 2 compares the **estimated number of clusters** inferred by each method against the ground truth on PACS, Office-Home, and DomainNet. HIDISC is found to approximate the cluster counts more precisely than the counterparts.

The **learnable curvature** converges to dataset-specific values: 0.041 for Office-Home, 0.059 for PACS, and 0.38 for DomainNet. The curvature evolution plots are mentioned in **Sup. Mat.**

4.2 Ablation Analysis

Impact of Loss Components and Key Model Components. Table 3 evaluates the contribution of each loss component in HIDISC on Office-Home. The vanilla model, trained without any loss terms, yields only 26.17% overall accuracy. Introducing the Busemann loss alone improves performance substantially to 56.32%, while the hybrid hyperbolic contrastive loss independently achieves 50.95%. Combining both leads to further gains, particularly in old-class accuracy. Incorporating the outlier repulsion term yields the best overall result, with 56.78% total accuracy, 59.23% on known classes, and 53.21% on novel classes.

Table 4 presents ablations on key architectural components. Manual augmentations achieve 50.80% accuracy. Replacing the learnable curvature with static values ($c = 0.01$ and 0.03) reduces accuracy considerably, as it fails to manage the data manifold effectively. Substituting Tangent CutMix with Euclidean CutMix lowers performance by 3.96%, confirming the benefits of curvature-consistent mixing. The complete HIDISC configuration consistently outperforms all variants, confirming the complementary benefits of its geometric and loss-driven design.

Ablation of Norm Radius and Slope in Hyperbolic Embedding. We study two key hyperparameters in our hyperbolic embedding setup: the ℓ_2 norm radius before exponential mapping and the slope ϕ in the penalized Busemann loss, both controlling embedding compactness and placement. As per Table 5, lower slopes (e.g., $\phi = 0.10$) favor seen-class accuracy but hurt generalization, while higher slopes (e.g., $\phi = 0.90$) improve novel-class performance by restricting dispersion. We choose $\phi = 0.75$ for balance. For the norm radius, Table 5 shows that 1.5 best balances alignment to boundary-anchored prototypes and generalization, whereas smaller values (e.g., 1.0) overfit $\mathcal{Y}_S$.

Table 5: **Ablation on hyperbolic embedding parameters on Office-Home.** (**Left**) Effect of slope coefficient ϕ in the penalized Busemann loss. Lower ϕ concentrates embeddings near the boundary, improving seen-class accuracy but reducing generalization. (**Right**) Effect of ℓ_2 radius constraint before exponential mapping. Radius = 1.5 yields the best trade-off between known and novel categories.

Slope ϕ	All	Old	New
0.10	58.84	65.77	47.07
0.75	**56.78**	**59.23**	**53.21**
0.90	57.76	62.82	49.18

Radius	All	Old	New
1.5	**56.78**	**59.23**	**53.21**
1.0	57.33	61.14	51.76
2.3	57.31	60.96	52.04

Choice of Hyperbolic Model: Poincaré vs. Lorentz Model

We adopt the Poincaré ball model, which we find performs more favorably than the Lorentz model [65] for DG-GCD on Office-Home in Table 6. Full theoretical details are in the **Sup. Mat.** The empirical comparison is below:

Hyperparameter Sensitivity for Loss Weights. We conducted an ablation study on the loss weights ($\lambda_1, \lambda_2, \lambda_3$) and found our chosen configuration achieves near-optimal performance, demonstrating robustness Table 7. More details are in the **Sup. Mat.**

Table 6: Comparison of Poincaré ball and Lorentz models on Office-Home.

Model	All	Old	New
Poincaré Ball [20]	**56.78**	**59.23**	**53.21**
Lorentz Model [65]	54.28	56.01	51.41

Table 7: Ablation study on loss term weights ($\lambda_1, \lambda_2, \lambda_3$) on OfficeHome.

	Loss Weights			Acc. (%)		
Config.	λ_1	λ_2	λ_3	All	Old	New
Config. 1	0.60	0.25	0.15	**56.78**	**59.23**	**53.21**
Config. 2	0.15	0.60	0.25	52.12	53.33	50.07
Config. 3	0.25	0.15	0.60	51.37	52.17	50.01

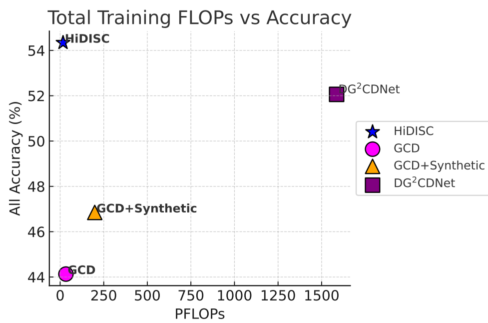

Figure 5: **Computational efficiency of HIDISC.** Hyp-Busemann requires only **16.53 PFLOPs** over 50 epochs with a batch size of 128×2, representing a $\sim 2\times$ reduction compared to GCD (33.06 PFLOPs), $\sim 12\times$ vs. GCD+Synthetic (198.36 PFLOPs), and nearly $\sim 96\times$ vs. DG^{2}CD-Net (1,586 PFLOPs). Despite this efficiency, Hyp-Busemann maintains superior accuracy without relying on episodic-training loops, simplifying the overall training pipeline.

5 Takeaways

We addressed the problem of DG-GCD, where novel categories emerge in unseen domains without target supervision. To this end, we proposed HIDISC, a hyperbolic representation learning framework that leverages penalized Busemann alignment, Tangent CutMix-based augmentation, and hybrid contrastive regularization to enable domain- and category-level generalization. Extensive experiments across PACS, Office-Home, and DomainNet show that HIDISC achieves state-of-the-art performance, particularly improving novel class discovery under domain shift. Our findings underscore the utility of hyperbolic geometry for scalable open-world recognition. **Future directions** include extending HIDISC to continual DG-GCD and integrating it with large-scale vision-language models.

Broader Impact and Limitations: While HIDISC advances open-world recognition under domain shift using geometry-aware learning, which is extremely practical, its reliance on synthetic augmentations guided by diffusion models may limit applicability in resource-constrained or safety-critical environments where generative artifacts could propagate bias.

Acknowledgments and Disclosure of Funding

We thank our colleague Shubranil B. for his assistance with the figures in this paper. We are also grateful to Adobe Research and the CMInDS department for providing the necessary resources and support.

References

[1] Vaibhav Rathore, Shubhranil B, Saikat Dutta, Sarthak Mehrotra, Zsolt Kira, and Biplab Banerjee. When domain generalization meets generalized category discovery: An adaptive task-arithmetic driven approach, 2025. URL `https://arxiv.org/abs/2503.14897`.

[2] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[3] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[4] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[5] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018(1):7068349, 2018.

[6] Junyi Chai, Hao Zeng, Anming Li, and Eric WT Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

[7] Tao et al. Sun. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *CVPR*, 2022.

[8] Jee Seok Yoon, Kwanseok Oh, Yooseung Shin, Maciej A Mazurowski, and Heung-Il Suk. Domain generalization for medical image analysis: A survey. *arXiv preprint arXiv:2310.08598*, 2023.

[9] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[10] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[11] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[12] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.

[13] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 153–168, 2018.

[14] Prathmesh Bele, Valay Bundele, Avigyan Bhattacharya, Ankit Jha, Gemma Roig, and Biplab Banerjee. Learning class and domain augmentations for single-source open-domain generalization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1816–1826, 2024.

[15] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7501, 2022.

[16] Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7579–7588, 2023.

[17] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Shahbaz Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3479–3488, 2023.

[18] Sai Bhargav Rongali, Sarthak Mehrotra, Ankit Jha, Shirsha Bose, Tanisha Gupta, Mainak Singha, Biplab Banerjee, et al. Cdad-net: Bridging domain gaps in generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2626, 2024.

[19] Shuo Wen and Maria Brbic. Cross-domain open-world discovery. *arXiv preprint arXiv:2406.11422*, 2024.

[20] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[21] Ines Chami, Aditya Wolf, Pierre Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Hyperbolic neural networks. In *NeurIPS*, pages 5345–5355, 2019.

[22] Yuanpei Liu, Zhenqi He, and Kai Han. Hyperbolic category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[23] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *Advances in neural information processing systems*, 34:103–115, 2021.

[24] OpenAI. Chatgpt, 2024. https://chat.openai.com/.

[25] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16590–16600, 2023.

[26] Sua Choi, Dahyun Kang, and Minsu Cho. Contrastive mean-shift learning for generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23094–23104, 2024.

[27] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and S Yu Philip. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022.

[28] Hongjun Wang, Sagar Vaze, and Kai Han. Hilo: A learning framework for generalized category discovery robust to domain shifts, 2024. URL `https://arxiv.org/abs/2408.04591`.

[29] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

[30] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation, 2022. URL `https://arxiv.org/abs/2204.02548`.

[31] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[32] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[33] Xiran Wang, Jian Zhang, Lei Qi, and Yinghuan Shi. Generalizable decision boundaries: Dualistic meta-learning for open set domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11564–11573, 2023.

[34] Ronghang Zhu and Sheng Li. Crossmatch: Cross-classifier consistency regularization for open-set single domain generalization. In *International conference on learning representations*, 2022.

[35] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9624–9633, 2021.

[36] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8401–8409, 2019.

[37] Florent Chiaroni, Jose Dolz, Ziko Imtiaz Masud, Amar Mitiche, and Ismail Ben Ayed. Parametric information maximization for generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1729–1739, 2023.

[38] Dahyun Kang, Piotr Koniusz, Minsu Cho, and Naila Murray. Distilling self-supervised vision transformers for weakly-supervised few-shot classification & segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19627–19638, 2023.

[39] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[40] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7409–7419, 2022.

[42] Yunhui Guo, Xudong Wang, Yubei Chen, and Stella X Yu. Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2022.

[43] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6418–6428, 2020.

[44] Zhi Gao, Yuwei Wu, Yunde Jia, and Mehrtash Harandi. Curvature generation in curved spaces for few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8691–8700, 2021.

[45] Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, and Serena Yeung. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2603–2612, 2021.

[46] Luca Franco, Paolo Mandica, Bharti Munjal, and Fabio Galasso. Hyperbolic self-paced learning for self-supervised skeleton-based action representations. *arXiv preprint arXiv:2303.06242*, 2023.

[47] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

[48] Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. *arXiv preprint arXiv:2303.15919*, 2023.

[49] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

[50] Martin Keller-Ressel. A theory of hyperbolic prototype learning. *arXiv preprint arXiv:2010.07744*, 2020.

[51] Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6840–6849, 2023.

[52] Yun Yue, Fangzhou Lin, Guanyi Mou, and Ziming Zhang. Understanding hyperbolic metric learning through hard negative sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1891–1903, 2024.

[53] Jiang Chang, Zhixin Zhang, Yuhua Qian, and Pan Lin. Multi-scale hyperbolic contrastive learning for cross-subject eeg emotion recognition. *IEEE Transactions on Affective Computing*, 2025.

[54] Shengyin Sun and Chen Ma. Hyperbolic contrastive learning with model-augmentation for knowledge-aware recommendation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 199–217. Springer, 2024.

[55] Rukai Wei, Yu Liu, Jingkuan Song, Yanzhao Xie, and Ke Zhou. Exploring hierarchical information in hyperbolic space for self-supervised image hashing. *IEEE Transactions on Image Processing*, 33:1768–1781, 2024.

[56] Chengyang Hu, Ke-Yue Zhang, Taiping Yao, Shouhong Ding, and Lizhuang Ma. Rethinking generalizable face anti-spoofing via hierarchical prototype-guided distribution refinement in hyperbolic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1032–1041, 2024.

[57] Jona Otholt, Christoph Meinel, and Haojin Yang. Guided cluster aggregation: A hierarchical approach to generalized category discovery. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2618–2627, January 2024.

[58] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, 1994.

[59] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[60] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[61] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[62] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[63] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[64] Sarah Rastegar, Mohammadreza Salehi, Yuki M Asano, Hazel Doughty, and Cees G M Snoek. Selex: Self-expertise in fine-grained generalized category discovery. In *European Conference on Computer Vision*, 2024.

[65] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pages 3779–3788. PMLR, 2018.

[66] Cédric Villani. *Optimal transport: old and new*. Springer, 2009.

[67] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[68] Kevin Verbeek and Subhash Suri. Metric embedding, hyperbolic space, and social networks. In *Proceedings of the 27th Annual Symposium on Computational Geometry*, pages 501–510. ACM, 2011.

[69] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[70] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state the contributions in the abstract and introduction sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the Limitations in 5

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper includes the full set of assumptions and a complete (and correct) proof in Section 3.4, with detailed extensions provided in the **Sup. Mat.**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: For details on reproducing the experimental results, please refer to section 4 and for further details refer **Sup. Mat.**

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Please refer to Section 4 for the datasets used , all are publicly available. Further code will be available on the project website.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Section 4 for experimental settings about the data splits , test details , training details(in Sup Mat) .

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper carefully reports the statistical significance details in Section 4, following standard practices in the literature.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are done on NVIDIA A100-SMX-80GB GPU's.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work in Section 5

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: This work poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all datasets and models utilized in our experiments.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This work does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The Gpt 4-o usage is described in Section 3.3

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

HIDISC: Supplementary Material

This supplementary document provides additional technical details, experimental results, and theoretical insights to complement the main paper. Specifically, it includes:

- **Quantitative and qualitative analyses** comparing Euclidean (spherical) and hyperbolic (Poincaré) embedding spaces, including inter-class separation and intra-class variability metrics (Sec. 6, Sec. 7).
- **Mathematical background** on Poincaré ball geometry and key hyperbolic operations used in our model (Sec. 8).
- **Details on synthetic data generation**, illustrating the lightweight augmentation pipeline and example images (Sec. 9, Sec. 9.1).
- **Comprehensive ablations** exploring the effects of synthetic domain count, adaptive outlier margin, embedding dimension, curvature learning, and domain combinations on model performance (Sec. 10.1–13.6, Sec. 13.4).
- **Theoretical justification** outlining generalization bounds and domain discrepancy advantages of hyperbolic embeddings over Euclidean embeddings (Sec. 11).
- **Extended comparisons** against state-of-the-art methods on three major benchmarks (PACS, Office-Home, DomainNet), with detailed accuracy tables and discussions (Sec. 13.10).
- **Additional visualization results** demonstrating embedding geometry and augmentation effects in hyperbolic space.

Together, these materials reinforce the core claims of our paper and provide the necessary context and evidence for reproducibility and deeper understanding.

6 Quantitative Comparison of Spherical vs. Hyperbolic Embeddings

To quantitatively support the qualitative illustration presented in Figure 1 of the main paper, we compare feature alignment in **Euclidean** (spherical) and **Hyperbolic** (Poincaré) embedding spaces across domains. Specifically, we evaluate the semantic consistency of same-class embeddings between two distinct domains—*Art Painting* and *Sketch*—from the **PACS** dataset.

We extract features using a frozen **DINO ViT-B16** [59] backbone and compute cross-domain similarity scores for each class as follows:

- **Euclidean Embedding**: Features are L2-normalized, and cosine similarity is computed via dot product between class-averaged embeddings from each domain.
- **Hyperbolic Embedding**: The same features are projected to the Poincaré ball via exponential mapping, and cosine similarity is computed in the tangent space, respecting hyperbolic geometry.

As shown in Table 8, same-class embeddings exhibit consistently high alignment in hyperbolic space (cosine similarity $\approx$ 0.98–0.99) across all classes, whereas their Euclidean counterparts show substantially lower similarity. This demonstrates that hyperbolic space facilitates superior cross-domain semantic consistency, even in the presence of strong domain shifts.

Class	**Euclidean Similarity**	**Hyperbolic Similarity**
Horse	0.1409	0.9812
Elephant	0.3350	0.9904
House	0.2418	0.9857
Dog	0.3091	0.9869
Guitar	0.3079	0.9833
Person	0.3459	0.9891
Giraffe	0.1439	0.9805

Table 8: Cosine similarity between class-wise average embeddings across *Art Painting* and *Sketch* domains in Euclidean and Hyperbolic spaces. Hyperbolic embeddings exhibit significantly improved semantic alignment.

These results validate our claim that the hyperbolic space enables improved inter-class separation and cross-domain semantic alignment, thereby providing an ideal geometric foundation for domain generalization and generalized category discovery (DG-GCD).

7 Inter-Class Separation and Intra-Class Variability

To further validate our theoretical motivation in lines 134–138 of the main paper, we evaluate both global semantic separation and local compactness in Euclidean vs. hyperbolic spaces.

We compute:

- **Inter-Class Distance (ICD)**: Mean cosine distance between class centroids.
- **Intra-Class Variability (ICV)**: Standard deviation of cosine similarity within each class across domains.

Metric	**Euclidean**	**Hyperbolic**
Inter-Class Distance ($\uparrow$)	0.45	**2.03**
Intra-Class Variability ($\downarrow$)	0.2964	**0.10**

Table 9: Comparison of inter-class separation and intra-class variability across embedding spaces. Hyperbolic space provides superior class separation and compactness.

These results show that hyperbolic embeddings support both global semantic separation and local compactness, as hypothesized. Higher inter-class distance amplifies semantic dissimilarity, while reduced intra-class variability reflects robustness to domain-specific low-level perturbations.

8 Poincaré Ball Geometry

In the main text (Sec. 2.1) we define the n-dimensional Poincaré ball of curvature $-c^2$ as

$$\mathbb{D}_c^n \;=\; \big\{\, x \in R^n : c\,\|x\|^2 < 1 \big\}, \qquad \|x\| = \sqrt{x^\top x}.$$

Below we collect the key operations used in our implementation.

Möbius Addition. For any $a, b \in \mathbb{D}_c^n$ the Möbius sum is

$$a \oplus_c b = \frac{(1 + 2\,c\,\langle a,b\rangle + c\,\|b\|^2)\,a \;+\; (1 - c\,\|a\|^2)\,b}{1 + 2\,c\,\langle a,b\rangle + c^2\,\|a\|^2\,\|b\|^2}\,.$$

Geodesic Distance. The Riemannian distance on $\mathbb{D}_c^n$ is

$$d_{\mathbb{D}_c}(a,b) = \frac{2}{\sqrt{c}}\,\tanh\!\big(\sqrt{c}\,\|{-a} \oplus_c b\|\big).$$

Exponential and Logarithmic Maps at the Origin. To move between Euclidean tangent vectors and the manifold:

$$\exp_0^c(u) = \tanh\!\big(\sqrt{c}\,\|u\|\big)\,\frac{u}{\sqrt{c}\,\|u\|}, \quad \log_0^c(x) = \tanh\!\big(\sqrt{c}\,\|x\|\big)\,\frac{x}{\sqrt{c}\,\|x\|},$$

for all $u \in R^n$ and $x \in \mathbb{D}_c^n$. These correspond to our ‘expmap0(u,c)‘ and its inverse in code.

Euclidean Limit. As $c \to 0$, hyperbolic operations recover their Euclidean counterparts:

$$\lim_{c\to 0} d_{\mathbb{D}_c}(a,b) = 2\,\|a-b\|, \quad \lim_{c\to 0} \exp_0^c(u) = u, \quad \lim_{c\to 0}(a \oplus_c b) = a + b.$$

This ensures full compatibility with our Euclidean baseline.

9 Technical details about Synthetic Data Generation

To implement the lightweight synthetic domain augmentation described in Sec. 3.3.1 of the main paper,following [1] for the image transformation tasks, we utilized the InstructPix2Pix pipeline from Hugging Face's Diffusers library, leveraging the `timbrooks/instruct- pix2pix` model, for image transformation tasks.To balance processing time with output quality, we set the `num_inference_steps` parameter to 10. The `image_guidance_scale` parameter was set to 1.0, e, ensuring the model preserved the essential structure of the input image while applying the desired transformations. Additionally, the `guidance_scale` was set to 7.5 to enhance the alignment with the transformation prompt. These settings enable the easy replication of our process while ensuring high-quality results.

9.1 Synthetic Domain Images

In Figure 6, we present a few examples of the synthetic images generated using the strategy outlined above for synthetic data generation.

On the Role of GPT-Guided Diffusion for Domain Augmentation

Although GPT-guided diffusion does not sample uniformly over the domain manifold, uniform coverage is neither tractable nor necessary for DG-GCD. Instead, HiDISC leverages task-aware, semantically structured augmentations that improve generalization:

- **Task-aware domain coverage.** Let $\mathcal{D} \subset \mathcal{X}$ denote the latent domain manifold. Uniform sampling $P(x) \sim \mathcal{U}(\mathcal{D})$ is intractable in high-dimensional image spaces. We instead construct a conditional distribution $P_G(x \mid t)$ using prompts t aligned with meaningful domain shifts, yielding diverse and relevant perturbations.
- **Empirical diminishing returns.** As shown in Fig. 3 (right), adding synthetic domains beyond 1–2 does not yield monotonic gains. Formally, the curvature of the loss $\mathcal{L}_{\text{gen}}$ satisfies $\frac{\partial^2 \mathcal{L}_{\text{gen}}}{\partial N^2} > 0$ for $N > 2$, indicating diminishing returns and potential overfitting.

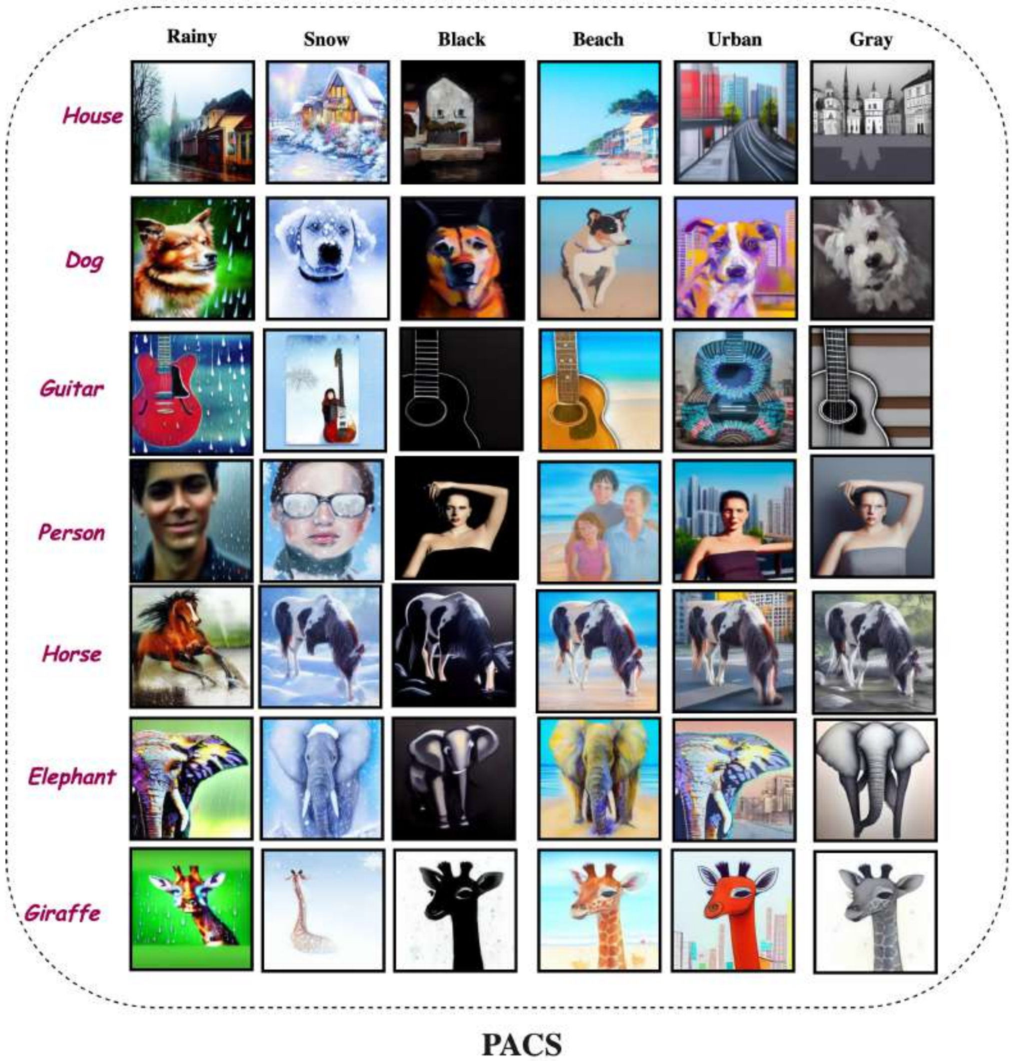

Figure 6: Examples of synthetic images generated for the PACS dataset, demonstrating various styles and domains across seven different categories.

- **Hyperbolic geometry enhances generalization.** Even without synthetic domains, HiDISC achieves 56.07% accuracy on Office-Home, surpassing Euclidean and augmented baselines. Theoretical analysis (Sec. 3.4) further shows that the generalization bound in hyperbolic space satisfies $\Delta_{\mathbb{H}} \leq \Delta_{\mathbb{E}}$, implying fewer samples are required to cover semantic variability.

In summary, GPT-guided augmentation supplies diverse but semantically structured shifts, while hyperbolic geometry ensures robust generalization without requiring exhaustive domain sampling.

10 Ablations

10.1 Number of Synthetic Domains

We quantify the effect of our lightweight synthetic domain augmentation (Sec. 3.3.1 in the main paper) by varying the number of GPT-4o–guided synthetic domains added per training image. Using the Office-Home benchmark, we train with $k \in \{0, \ldots, 6\}$ synthetic domains in addition to the original source data and report three metrics.

Figure 7 plots these metrics as a function of k. We observe:

1. *Monotonic gains* in **All** and **Old** accuracies up to $k = 6$, demonstrating that more diverse styles in hyperbolic space reinforce domain-invariant feature learning.
2. **New**–class accuracy peaks at $k = 2$ (53.21%) before declining, indicating that excessive augmentation may oversaturate the model and hinder novel category discrimination.

Overall, $k = 2$ supplies the best trade-off between known-class retention and novel-class discovery, and is used in our main experiments.

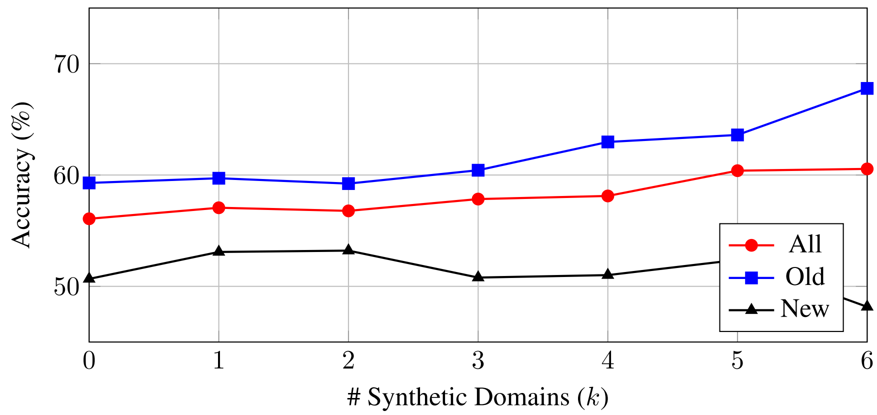

Figure 7: Ablation on the number of synthetic domains per image (k) for the Office-Home dataset. Adding up to six synthetic styles steadily improves overall (*All*) and seen-class (*Old*) accuracies, while novel-class (*New*) performance peaks at $k = 2$ before declining.

10.2 Adaptive Outlier Margin.

To set a robust margin for the adaptive outlier loss (Eq. 3 in the main paper) we compute the margin γ dynamically on the first training mini-batch as follows. Let

$$d_i = \min_{k \in \mathcal{Y}_s} d_{\mathbb{D}_c}(z_i, p_k) \quad \text{for each embedding } z_i$$

where p_k are the seen-class prototypes. We then define

$$\gamma = \text{Quantile}_{0.8}\big(\{d_i\}\big),$$

i.e. the 80th percentile of the per-sample minimum distances. Using this batch-wise adaptive margin ensures that (1) γ reflects the actual spread of the feature distribution, (2) outlier penalties remain neither too weak nor excessively strict, and (3) the model stays discriminative and resilient to noisy embeddings. In practice, γ is recomputed once at the start of training and held fixed thereafter.“‘

We study how the choice of the quantile α for setting the adaptive outlier margin γ (see Sec. 3.3.3) affects novel-class discovery on the PACS dataset. For each $\alpha \in \{0.05, 0.10, 0.20, 0.80, 0.90\}$, we compute γ as the α-quantile of per-sample minimum prototype distances on the first training batch, then hold γ fixed throughout training.

Table 10: New (%) on PACS for different quantile settings α.

Quantile α	0.05	0.10	0.20	0.80	0.90
New (%)	72.1	73.4	74.1	**74.52**	74.31

As Table 10 shows, moderate quantiles ($\alpha = 0.80$) yield the best novel-class discrimination .

10.3 Poincaré-Disk Visualizations of Tangent CutMix on PACS

For each domain, we plot the normalized feature prototypes of four seen classes (solid circles) alongside their Tangent CutMix augmentations (cross markers), all projected via UMAP from

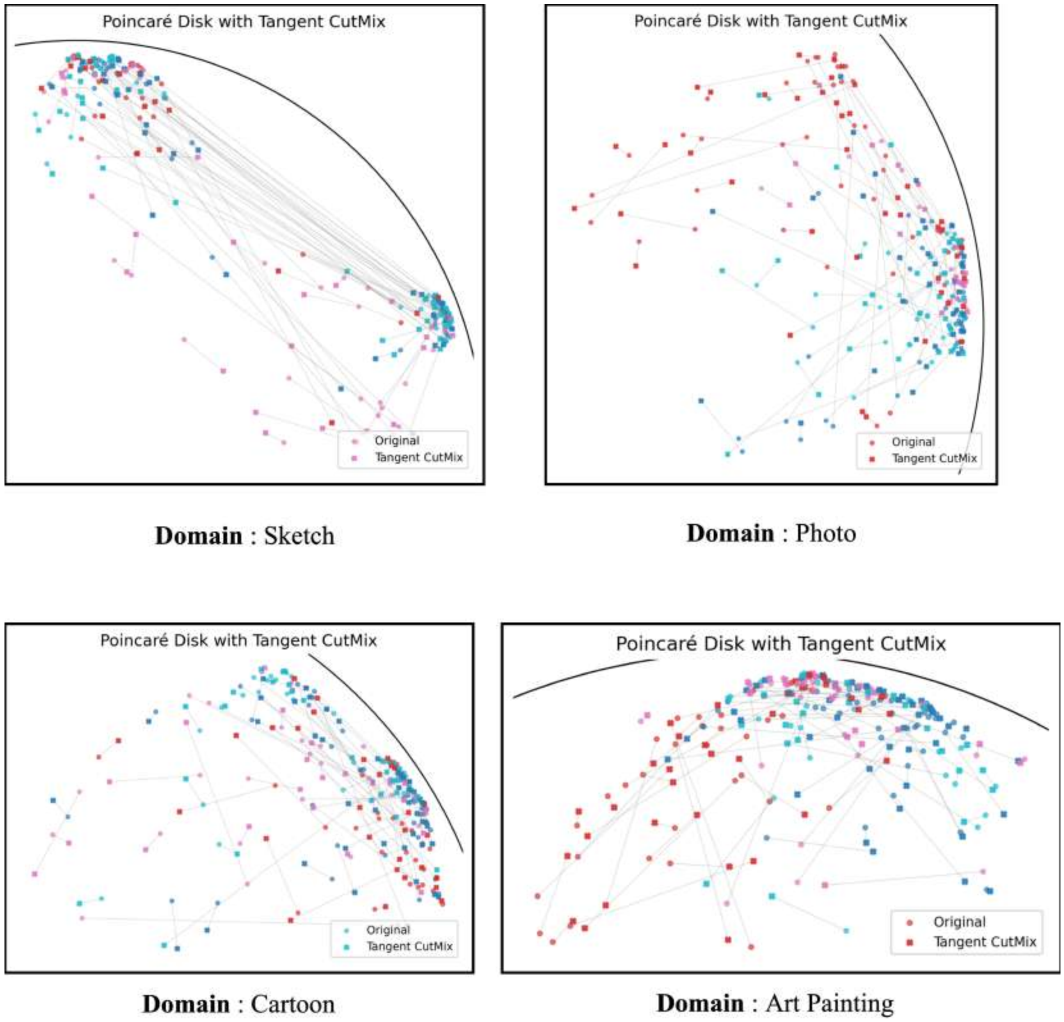

Figure 8: **UMAP-projected Poincaré-disk embeddings of original and Tangent CutMix samples on PACS.** We first embed features into the Poincaré ball and then apply UMAP to project onto 2D for visualization. Each subplot corresponds to one PACS domain (Sketch, Photo, Cartoon, Art Painting), showing four seen-class prototypes (solid circles) and their Tangent CutMix augmentations (cross markers). Grey lines link each synthetic point back to its original partners, highlighting curvature-aware mixing in the tangent space at the origin.

the hyperbolic Poincaré ball. Grey geodesic-consistent connectors illustrate the tangent-space interpolation at the origin. Across Sketch and Photo (top row), mixed samples fan out toward the periphery—regions of higher curvature—thereby expanding open space between clusters. Similarly, in Cartoon and Art Painting (bottom row), synthetic embeddings occupy under-populated areas while respecting manifold geometry. These UMAP-projected Poincaré visualizations demonstrate how Tangent CutMix generates out-of-distribution samples and reserves hyperbolic "reserve space" for novel-class discovery without collapsing into known regions.

10.4 Redundant Augmentations and Diminishing Returns

We call an augmentation "redundant" once it fails to introduce any novel variation in hyperbolic feature space beyond what our existing synthetic domains already cover. In practice we see that after adding one or two well-chosen domains , further domains lie in the same regions of the Poincaré ball (high pairwise FID Table 11) and thus bring negligible gains in novel-class discovery.

Table 11: Pairwise FID between synthetic domain pairs.

Idx	Domain D1	Domain D2	FID(D1,D2)
1	Rainy	Beach	123.33
2	Rainy	Black	133.74
3	Rainy	Gray	138.74
4	**Rainy**	**Urban**	**158.24**
5	Rainy	Snow	71.47
6	Beach	Black	118.41
7	Beach	Gray	93.15
8	Beach	Urban	120.73
9	Beach	Snow	89.21
10	Black	Gray	57.07
11	Black	Urban	123.01
12	Black	Snow	107.16
13	Gray	Urban	107.78
14	Gray	Snow	94.53
15	Urban	Snow	127.33

10.4.1 Effect of other domain combinations on performance

Table 12: Effect of domain-combination paths on Office-Home

Idx	All (%)	Old (%)	New (%)
1	57.34	60.96	51.20
2	58.34	61.99	52.04
3	57.22	61.25	50.38
4	**56.78**	**59.23**	**53.21**
5	57.71	62.21	50.06
6	57.55	62.12	49.79
7	58.93	63.14	51.85
8	58.08	63.93	48.37
9	56.42	59.41	51.31
10	58.62	63.27	50.79

Adaptive Balancing of Angular and Distance-based Losses. The parameter α_d to balance the contributions of distance-based and angle-based loss components during training. This parameter is scheduled to increase linearly with training progress and is defined as:

$$\alpha_d = \frac{e \cdot \alpha_d^{\max}}{50},$$

where e denotes the current training epoch, and $\alpha_d^{\max}$ is the maximum weight assigned to the angular loss. We set $\alpha_d^{\max} = 1.0$ for all three datasets. This scheduling encourages the model to initially emphasize structural separation through distance-based learning and gradually shift focus toward angular alignment, improving cluster compactness and generalization over time.

Table 13: Ablation on the maximum angular weight $\alpha_d^{\max}$ for the Office-Home dataset. We report overall accuracy (*All*), seen-class accuracy (*Old*), and novel-class accuracy (*New*).

$\alpha_d^{\max}$	All (%)	Old (%)	New (%)
0.25	56.43	58.10	53.95
0.50	55.23	56.31	53.61
0.75	56.05	58.81	52.02
1.00	**56.78**	**59.23**	**53.21**

10.5 Observation on Old-New class splits:

Table 14 illustrates the impact of varying base (old) and novel (new) class splits on the performance of HIDISC on the Office-Home dataset.

In Table 14, we observe that increasing the number of known classes (50–15 split vs. 40–25 split) leads to a small drop in Old-class accuracy ($\approx 1\%$). This can be explained by two interacting effects:

Splits (Old-New)	Office-Home		
	All	**Old**	**New**
40 - 25	56.78	59.23	53.21
30 - 35	55.31	58.87	52.47
25 - 40	54.76	57.95	52.84
55 - 10	57.85	59.63	48.46
50 - 15	57.13	58.23	53.59

Table 14: Sensitivity on different `Old-New` class splits.

- **Boundary crowding.** Adding more **Old** classes places additional prototypes closer to the Poincaré boundary, which increases overlap and slightly raises confusion among old categories. Interestingly, this higher density benefits discovery of **New** classes by pushing unfamiliar samples toward the interior, yielding the observed tradeoff (Old $\downarrow$, New $\uparrow$).
- **Per-class sample dilution.** With a fixed training budget, more classes imply fewer samples per class. This increases per-class variance and mildly reduces Old-class accuracy.

Overall, the effect is minor and consistent with our geometric intuition: higher boundary density aids novel-class separation at the cost of a small accuracy reduction on known classes.

11 Theoretical Justification: Generalization Bounds in Hyperbolic Space

11.1 Formal Justification: Why $\Delta_{\mathbb{H}} < \Delta_{\mathbb{E}}$ Can Hold

Let $\mathcal{D}'_S$ denote the augmented source domain and $\mathcal{D}_T$ the target domain. For a hypothesis class $\mathcal{H}_{\mathcal{G}}$ defined over geometry $\mathcal{G} \in \{\mathbb{E}, \mathbb{H}\}$, the domain discrepancy is:

$$\Delta_{\mathcal{G}}(\mathcal{D}'_S, \mathcal{D}_T) = \sup_{f \in \mathcal{H}_{\mathcal{G}}} \left| \mathbb{E}_{x \sim \mathcal{D}'_S}[f(x)] - \mathbb{E}_{x \sim \mathcal{D}_T}[f(x)] \right|.$$

Assume:

- (A1) The latent structure of $\mathcal{D}'_S \cup \mathcal{D}_T$ is approximately hierarchical (e.g., tree-like);
- (A2) The encoders $\phi_{\mathcal{E}}$, $\phi_{\mathcal{H}}$ are 1-Lipschitz under Euclidean and hyperbolic metrics respectively;
- (A3) $\mathcal{H}_{\mathcal{G}}$ is 1-Lipschitz with respect to $d_{\mathcal{G}}$.

Using Kantorovich–Rubinstein duality [66], we upper-bound the discrepancy via the Wasserstein-1 distance:

$$\Delta_{\mathcal{G}}(\mathcal{D}'_S, \mathcal{D}_T) \leq W_1^{\mathcal{G}}(\phi_{\mathcal{G}\#}\mathcal{D}'_S, \phi_{\mathcal{G}\#}\mathcal{D}_T),$$

where $\phi_{\mathcal{G}\#}\mathcal{D}$ is the pushforward distribution in $\mathcal{M}_{\mathcal{G}}$.

Why Hyperbolic Distance Tightens Discrepancy. In tree-like or hierarchical spaces, embeddings into hyperbolic geometry suffer significantly less distortion than Euclidean ones. Formally, Bourgain's theorem and subsequent results in geometric embedding theory show that the distortion for embedding n-node trees into:

- Euclidean space is $\Omega(\log n)$ [67];
- Hyperbolic space is $\mathcal{O}(1)$ [68].

Thus, for semantically structured data (as in class hierarchies or taxonomies), the Wasserstein-1 distance between distributions is tighter in hyperbolic space:

$$W_1^{\mathbb{H}}(\phi_{\mathbb{H}\#}\mathcal{D}'_S, \phi_{\mathbb{H}\#}\mathcal{D}_T) \leq W_1^{\mathbb{E}}(\phi_{\mathbb{E}\#}\mathcal{D}'_S, \phi_{\mathbb{E}\#}\mathcal{D}_T) + \mathcal{O}(\log n),$$

where n is the number of semantic entities.

Implication for DG-GCD. Since domain discrepancy $\Delta_{\mathbb{G}}$ directly impacts generalization bounds (cf. [69, 21]), this structural preservation implies:

$$\Delta_{\mathbb{H}}(\mathcal{D}'_S, \mathcal{D}_T) < \Delta_{\mathbb{E}}(\mathcal{D}'_S, \mathcal{D}_T),$$

leading to a tighter bound in the hyperbolic setting—provided semantic hierarchy and low distortion hold.

Hyperbolic geometry more naturally encodes semantic divergence with exponential volume growth, allowing fewer augmentations to approximate complex target distributions. This justifies why the same loss functions, when deployed in $\mathbb{D}_c^d$, yield a tighter generalization guarantee compared to their Euclidean counterparts.

On the Geometric Validity of Tangent CutMix

We provide a mathematical justification that Tangent CutMix preserves hyperbolic consistency under the Poincaré ball model. Let $\mathbb{D}_c^d = \{z \in \mathbb{R}^d : \|z\| < 1/\sqrt{c}\}$ denote the d-dimensional Poincaré ball with curvature $-c$. Given two embeddings $z_i, z_j \in \mathbb{D}_c^d$, we map them to the tangent space at the origin via the logarithmic map:

$$v_i = \log_0^c(z_i) = \frac{2}{\lambda_{z_i}} \tanh^{-1}(\sqrt{c}\|z_i\|) \cdot \frac{z_i}{\|z_i\|}, \quad \lambda_{z_i} = 1 - c\|z_i\|^2,$$

yielding $v_i, v_j \in \mathbb{R}^d$. Tangent CutMix then interpolates in this Euclidean tangent space:

$$v_{\text{mix}} = \lambda v_i + (1-\lambda) v_j, \quad \lambda \sim \text{Uniform}(0,1),$$

and projects back to the manifold using the exponential map:

$$z_{\text{mix}} = \exp_0^c(v_{\text{mix}}) = \tanh(\sqrt{c}\|v_{\text{mix}}\|) \cdot \frac{v_{\text{mix}}}{\sqrt{c}\|v_{\text{mix}}\|}.$$

Since $\tanh(x) < 1$ for all $x > 0$, we have $\|z_{\text{mix}}\| < 1/\sqrt{c}$, ensuring $z_{\text{mix}} \in \mathbb{D}_c^d$. Thus, Tangent CutMix guarantees valid hyperbolic embeddings.

To further ensure stability, we employ the penalized Busemann loss (Eq. 4), whose $\log(1-\|z\|^2)$ term discourages embeddings from approaching the boundary. Empirically, no violations of geometric constraints were observed. Replacing Tangent CutMix with Euclidean CutMix caused a significant performance drop (-3.96% on Office-Home; Table 4), underscoring the necessity of hyperbolic consistency for robust generalization.

On the Role of Hyperbolic Capacity in Data Augmentation

Hyperbolic geometry provides a natural explanation for why fewer augmentations are required compared to Euclidean spaces. In a d-dimensional Poincaré ball with curvature $-c$, the volume of a ball grows exponentially with radius, i.e.,

$$V_{\text{Hyp}}(r) \sim \exp(\sqrt{c}\, r),$$

in contrast to the polynomial growth $V_{\text{Euc}}(r) \sim r^d$ in Euclidean space. This exponential expansion increases representational capacity, enabling semantic clusters to be separated with less overlap even when data coverage is sparse. Consequently, a small number of augmentations suffices to populate the embedding space without significant distortion.

Empirically, we find that HiDISC achieves 56.07% accuracy on Office-Home even without any synthetic domains, outperforming several Euclidean and hyperbolic baselines that rely on 6–9 augmentations. Introducing only 1–2 GPT-guided augmentations improves performance further (56.78%), whereas excessive augmentation degrades novel-class accuracy due to overfitting (Fig. 3, right).

Finally, our generalization analysis (Sec. 3.4, Eq. 9) shows that the hyperbolic discrepancy term $\Delta_{\mathbb{H}}(S', T)$ yields a tighter Rademacher-based bound than its Euclidean counterpart $\Delta_{\mathbb{E}}(S', T)$ under the same augmentation budget. Together, these theoretical and empirical findings support the claim that hyperbolic geometry's exponential capacity reduces the reliance on heavy augmentation for domain generalization.

11.2 Euclidean vs. Hyperbolic Loss Ablation

Below we list the direct Euclidean analogues of our three hyperbolic loss components, obtained by replacing hyperbolic distances in the Poincaré ball with standard ℓ_2 norms in $\mathbb{R}^d$.

1. Euclidean Prototype-Softmax (Analogue of Busemann Loss)

$$\mathcal{L}_{\text{Euc-Buse}}(z, y) = -\log \frac{\exp\bigl(-\|z - \mu_y\|_2^2\bigr)}{\sum_{j=1}^{C} \exp\bigl(-\|z - \mu_j\|_2^2\bigr)} \quad , \quad \mu_j \;=\; \frac{1}{N_j} \sum_{i: y_i = j} z_i$$

where $\{\mu_j\}$ are the class prototypes in $\mathbb{R}^d$.

2. Euclidean Contrastive Loss (Analogue of Hyperbolic Contrastive Term)

$$\mathcal{L}_{\text{Euc-Con}} = -\frac{1}{|B|} \sum_{i \in B} \log \frac{\exp\bigl(\langle \hat{z}_i, \hat{z}_i^+ \rangle / \tau\bigr)}{\displaystyle\sum_{a \in B \setminus \{i\}} \exp\bigl(\langle \hat{z}_i, \hat{z}_a \rangle / \tau\bigr)}$$

where $\hat{z} = \frac{z}{\|z\|_2}$ denotes ℓ_2 normalization, B is the batch of views, and τ is the temperature.

3. Euclidean Outlier Repulsion (Analogue of Adaptive Outlier Loss)

$$\mathcal{L}_{\text{Euc-Out}} = \frac{1}{|B|} \sum_{i \in B} \max\Big(0,\ \gamma - \min_k \| z_i^{\text{mix}} - \mu_k \|_2\Big)$$

which pushes each synthetic mix-up embedding z_i^{mix} at least a margin γ away from all class prototypes in Euclidean space.

Table 15 shows that replacing hyperbolic distance with Euclidean norm widens the generalization gap: although both variants improve over a vanilla ViT baseline, only the hyperbolic formulation consistently boosts novel-class discovery (New) while raising overall (All) and old-class (Old) accuracy.

Table 15: Ablation of prototype-softmax loss in Euclidean vs. Hyperbolic space on Office-home (2 synthetic domains).

Manifold	All (%)	Old (%)	New (%)
Euclidean Manifold	49.46	52.08	45.66
Hyperbolic Manifold	**56.78**	**59.23**	**53.21**

12 Dataset details

Our experiments were carried out using three benchmark datasets: (i) **PACS** [2], (ii) **Office-Home** [3], and (iii) **DomainNet** [4].

For both the PACS and Office-Home datasets, we used each domain as the source, while treating all other domains as target domains. In the case of DomainNet, we selected a subset of source-target pairs, as outlined in Table 16, to ensure the model was evaluated across a variety of domain combinations.

Source	Targets
Sketch	Clipart, Painting, Infograph, Quickdraw, Real World
Painting	Clipart, Sketch, Infograph, Quickdraw, Real World
Clipart	Painting, Sketch, Infograph, Quickdraw, Real World

Table 16: Source-target configurations for DomainNet

13 Further Comparison with Literature

13.1 DG-GCD-Specific Adaptations for Baselines

As described in [1], we adapted several state-of-the-art methods for generalized category discovery (GCD) and domain generalization (DG) to the DG-GCD framework in a similar manner. This framework completely excludes access to target domain data during training, and for some methods, synthetic domain data is introduced to simulate domain shifts.

For the ViT-B/16 model pre-trained with DINO, we fine-tuned only the final block using source domain data, adhering to standard GCD protocols. We then evaluated the model on target domains without incorporating any target domain data during training. Similarly, for GCD, we fine-tuned the last block of the backbone using only source domain data and introduced a synthetic domain variant to better account for domain shifts.

We applied the same adaptation procedure to CMS (Contrastive Mean Shift) and SimGCD. For these methods, we fine-tuned the last block using only source domain data and created synthetic variants by incorporating synthetic domain data to evaluate their performance in handling domain shifts effectively.

For CDAD-Net, initially designed for cross-domain adaptation, we adapted it for the DG-GCD setting by training solely on source domain data, excluding target domain information. Additionally, we created synthetic variants to assess its performance on unseen domains.

For Hyp-GCD and Hyp-SelfEx, we followed the procedure outlined in [22]. As for the Hyp-DG^{2}CD-Net, we adapted the code directly from the authors, modifying the two loss functions in hyperbolic space following the strategy described in [22].

13.2 Implementation Details.

We build HIDISC in PyTorch and run all experiments on a single NVIDIA A100-SXM4-80GB GPU. Following [15, 1], we adopt a DINO-pretrained ViT-B/16 [60, 59], fine-tuning only its final transformer block and using the [CLS] token as our image representation.These features pass through a 3-layer MLP projection head g_ϕ to yield 32-dimensional embeddings. To simulate domain shift, we generate class-preserving synthetic domains using diffusion models guided by GPT-4o-curated prompts. These are merged with the labeled source domain D_S into a unified training set. We rank candidate domains using a composite FID-based score that balances source divergence and inter-domain diversity (Eq.1 of main paper).

During training, features are mapped into the Poincaré ball $\mathbb{D}_c^d$ using the exponential map, with a learnable curvature parameter c, initialized to 0.05. We clip embeddings to ensure numerical stability and normalize them using $\tanh$. Class prototypes are initialized randomly on the boundary and refined via a 1000-step SGD loop (learning rate 0.1, momentum 0.9). We employ a penalized Busemann loss for aligning features to fixed ideal prototypes and a hybrid hyperbolic contrastive loss [22] for instance-level alignment. The hybrid loss combines geodesic distance and cosine similarity in tangent space, with a weighting factor α_d that varies linearly over epochs.

Training is performed for 50 epochs using SGD (learning rate 0.01, momentum 0.9, weight decay 5×10^{-5}), with cosine annealing. The curvature c, projection head, final ViT block, and prototype embeddings are optimized jointly. For Office-Home, PACS, and Domain-Net, we set the Poincaré radius to 1.5, 1.0, and 2.3 respectively. We follow [1] for known/novel splits and use the K-estimation method from the same work to estimate the number of clusters K via Brent's algorithm, constrained to $[|\mathcal{Y}_s|, 1000]$.

Training Objective and Loss Function Combination: The final training objective is a weighted sum of three key loss components: the *Penalized Busemann Loss*, the *Hybrid contrastive Loss*, and the *Outlier Loss*. The total loss is formulated as follows.

$$\mathcal{L}_{\text{total}} = \lambda_1 \underbrace{\mathcal{L}_{\text{Buse}}}_{\text{Semantic alignment}} + \lambda_2 \underbrace{\mathcal{L}_u}_{\text{Contrastive regularization}} + \lambda_3 \underbrace{\mathcal{L}_{\text{out}}}_{\text{Outlier repulsion}}, \quad \text{where } \lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{10}$$

In our implementation, we set $\lambda_1 = 0.60$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.15$, which ensures a balanced contribution from each term. These hyperparameters are influenced by prior works [15, 1]. The *Penalized Busemann Loss* aligns the feature embeddings with ideal class prototypes in hyperbolic space, while the *Hybrid contrastive Loss* encourages discriminative features between positive and negative pairs across augmented views. The *Outlier Loss* with the adaptive margin penalizes embeddings that deviate significantly from the prototypes, improving generalization by limiting the influence of outliers. This combined loss function guides the model to learn robust, domain-invariant feature representations for both seen and unseen classes during training.

Algorithm 1 Training Procedure for DG-GCD with HIDISC

Input:

- Labeled source dataset $\mathcal{D}_S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with N known classes
- Synthetic domain augmentations $\{\mathcal{D}_{\text{syn}}^{(k)}\}_{k=1}^{M}$ to simulate domain shift
- Pretrained ViT backbone F_θ (frozen except last transformer block)
- Projection head G_ϕ mapping features to hyperbolic embeddings
- Fixed ideal prototypes $\{\mathbf{p}_j\}_{j=1}^{N}$ placed on the boundary $\partial\mathbb{D}_c^d$ of Poincaré ball
- Learnable curvature parameter c controlling hyperbolic geometry
- Hyperparameters: batch size B, epochs E, learning rates, penalization coefficient ϕ, number of views V, and loss weights $\lambda_1, \lambda_2, \lambda_3$ with $\lambda_1 + \lambda_2 + \lambda_3 = 1$

Output: Optimized parameters θ, ϕ, curvature c, and fixed prototypes $\{\mathbf{p}_j\}$

for $e = 1$ **to** E **do** ▷ Epoch loop
  Set F_θ, G_ϕ to training mode
  for each batch (X, Y) sampled from combined datasets $\mathcal{D}_S \cup \bigcup_k \mathcal{D}_{\text{syn}}^{(k)}$ **do** ▷ Batch loop
    Generate V augmented views per input in X
    Extract features: $Z^{\mathbb{E}} = F_\theta(X)$
    Project embeddings: $Z = G_\phi(Z^{\mathbb{E}})$
    Clip embeddings to radius r (dependent on dataset and curvature c)
    Generate mixed embeddings Z_{mix} by convex combination in tangent space at origin:

$$Z_{\text{mix}} = \exp_0^c\left(\lambda \log_0^c(Z_i) + (1-\lambda)\log_0^c(Z_j)\right), \quad \lambda \sim \text{Uniform}(0,1)$$

    Apply element-wise $\tanh$ to ensure embeddings lie strictly within Poincaré ball $\mathbb{D}_c^d$
    Construct batch prototypes $P_Y = \{\mathbf{p}_{y_i}\}_{i=1}^{B}$ repeated V times corresponding to labels Y
    Compute **penalized Busemann loss** [23]:

$$\mathcal{L}_{\text{Buse}} = \frac{1}{BV}\sum_{i=1}^{BV}\left[\log\left(\frac{\|z_i - \mathbf{p}_{y_i}\|^2}{1 - c\|z_i\|^2}\right) + \phi\log(1 - \|z_i\|^2)\right]$$

    Compute **hybrid hyperbolic contrastive loss** [22] on V views:

$$\mathcal{L}_u = \text{HybridContrastiveLoss}(Z, V, \alpha(e))$$

    ▷ $\alpha(e)$ is epoch-dependent weighting between geodesic and angular similarity
    if $e = 1$ and first batch **then**
      Compute adaptive margin δ as quantile over distances $d_{\mathbb{H}}(Z_{\text{mix}}, \mathbf{p}_j)$ to prototypes
    end if
    Compute **adaptive outlier loss**:

$$\mathcal{L}_{\text{out}} = \frac{1}{|Z_{\text{mix}}|}\sum_{z \in Z_{\text{mix}}} \text{ReLU}\big(\delta - \min_j d_{\mathbb{H}}(z, \mathbf{p}_j)\big)$$

    Aggregate total loss:

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{Buse}} + \lambda_2\mathcal{L}_u + \lambda_3\mathcal{L}_{\text{out}}$$

    Backpropagate $\mathcal{L}$ and update parameters θ, ϕ, c via optimizer
  end for
  Update learning rate scheduler
end for

13.3 Effect of Reduced Synthetic Domains on Model Performance

Table 17 presents a detailed comparison between HIDISC and DG^{2}CD-Net on the PACS dataset when using 2 versus 6 synthetic domains during training. While HIDISC maintains robust performance

Table 17: Performance comparison on the PACS dataset between HIDISC and DG^{2}CD-Net with 2 and 6 synthetic domains.

Model Variant	**All**	**Old**	**New**
HIDISC (2 Synthetic Domains)	75.07	75.54	74.52
DG^{2}CD-Net (2 Synthetic Domains)	66.86	69.11	63.75
HIDISC (6 Synthetic Domains)	74.00	75.64	71.95
DG^{2}CD-Net (6 Synthetic Domains)	73.30	75.28	72.56

with fewer synthetic domains, DG^{2}CD-Net experiences a marked degradation, confirming the claim in the main paper regarding DG^{2}CD-Net's sensitivity to synthetic domain count.

This underscores the efficiency and robustness of our lightweight synthetic domain augmentation strategy, which achieves strong generalization with fewer synthetic domain samples.

Why Naïve Hyperbolic Adaptations for DG^{2}CD-Net Fail

To better understand the design choices in HIDISC, we also implemented a hyperbolic variant of DG^{2}CD-Net (denoted Hyp-DG^{2}CD-Net). Interestingly, Hyp-DG^{2}CD-Net underperforms both the original DG^{2}CD-Net and our proposed HIDISC. We attribute this gap to several factors:

- **Task Vector Arithmetic Mismatch.** DG^{2}CD-Net aggregates task vectors via Euclidean averaging during episodic training. In hyperbolic space, vector addition is not associative or commutative, so retaining Euclidean arithmetic introduces geometric distortion.
- **Episodic Prototype Drift.** Injecting synthetic novel classes episodically without curvature-aware constraints causes embeddings to drift toward the boundary, reducing separability and increasing overconfidence.
- **Loss Function Misalignment.** Hyp-DG^{2}CD-Net reuses Euclidean-designed losses, which are not well-defined on manifolds. In contrast, HIDISC employs Busemann-aligned supervision, geodesic-aware contrastive regularization, and adaptive outlier repulsion to maintain consistency with curved geometry.
- **Empirical Results.** As shown in Table 1, Hyp-DG^{2}CD-Net performs consistently worse than both DG^{2}CD-Net and HIDISC across all benchmarks, underscoring the importance of end-to-end geometry-aware design rather than simple embedding substitution.

These findings emphasize that effective hyperbolic adaptation requires full-stack integration of geometry-aware components—embedding, loss design, and training protocol—rather than replacing only the representation space.

13.4 Learned Curvature Evolution

Our model learns the hyperbolic curvature c jointly during training, adapting it to dataset-specific geometry. Figure 9 illustrates the curvature evolution for three benchmark datasets: Office-Home, PACS, and DomainNet. These learned curvatures are consistent with the intuition that more complex or diverse datasets like DomainNet require a higher curvature, while relatively simpler domains such as Office-Home stabilize at lower curvature values.

13.5 Single-Source Domain Generalization

To isolate the effect of embedding geometry, we train two shallow MLPs from scratch—*Euclidean-Net* and *Hyperbolic-Net*—on a single source domain, and evaluate them on the three held-out domains of both the PACS and Office Home benchmarks. Both models share identical capacity (1.12 M trainable parameters), a unified training schedule (100 epochs, batch size 128, learning rate 1×10^{-3}), and are optimized using the cross-entropy loss.

Discussion. Across both benchmarks, Hyperbolic-Net substantially outperforms Euclidean-Net under the single-source regime. On PACS, hyperbolic embeddings yield a 73 % relative improvement (22.3 % vs. 12.9 %), while on Office Home they achieve a 215 % gain (6.17 % vs. 1.96 %). These

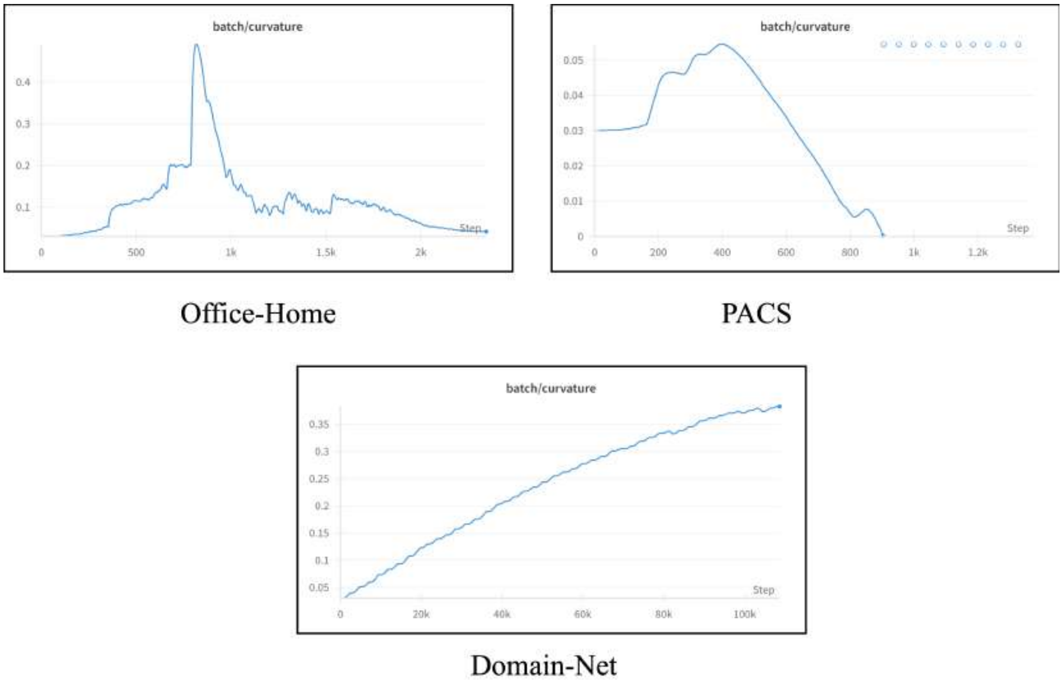

Figure 9: Learned curvature c evolution across training batches for Office-Home (left), PACS (center), and DomainNet (right).These plots were generated from training logs visualized using Weights & Biases (WandB) [70].

Table 18: Average test accuracy (%) when training on one source domain and testing on three held-out domains.

Dataset	EuclideanNet	HyperbolicNet	Relative Gain
PACS	12.9	22.3	+72.86%
Office Home	1.96	6.17	+214.79%
Domain Net	0.28	0.96	+ 242.85 %

consistent gains demonstrate that hyperbolic geometry inherently captures the relational structure necessary for generalization when only one domain is available.

13.6 Embedding Dimension

In order to investigate the effect of embedding dimensionality on model performance, we perform experiments on the Office-Home dataset by varying the embedding dimension between $\{32, 64, 128, 256, 512\}$. Figure 10 summarizes the results in three evaluation settings: overall accuracy (*All*), accuracy on seen classes (*Old*), and accuracy on novel classes (*New*).

We observe that lower-dimensional embeddings (*32, 64, 128*) consistently outperform larger dimensions (*256, 512*) across all metrics. In particular, a dimension of 64 achieves the highest precision in the seen class (64.00%), while a dimension of 32 provides the best performance in novel classes (53.23%). Increasing the dimension beyond 128 leads to a sharp degradation in performance, indicating that excessively large embeddings may overfit the seen classes while impairing generalization to novel categories. Based on these findings, we select a dimension of 32 as the default setting for subsequent experiments, achieving a favorable balance between the performance seen and the novel class performance.

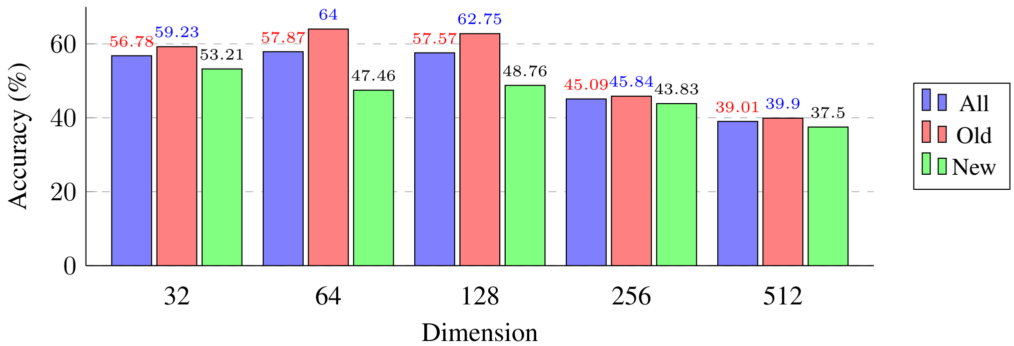

Figure 10: Ablation study on embedding dimension on the Office-Home dataset. Lower dimensions (32, 64, 128) perform better than larger ones (256, 512), suggesting that compact embeddings are beneficial for generalization.

Table 19: Impact of hyperbolic curvature c on performance.

Curvature c	**All (%)**	**Old (%)**	**New (%)**
0.01	56.23	58.65	52.68
0.03	55.67	57.39	52.69
0.05	56.01	58.19	52.78
0.10	56.27	58.52	52.97

13.7 For Non-learnable curvature

13.8 Prototype count analysis

We ablated the number of prototypes per class ($P \in \{1, 2, 4\}$) on Office-Home. Results are summarized below:

Prototypes / Class	**All**	**Old**	**New**
1	56.78	59.23	53.21
2	54.40	57.85	48.68
4	56.23	59.75	50.27

Table 20: Prototype count analysis

We find that performance peaks at $P = 1$, suggesting that one discriminative prototype per class is sufficient in the hyperbolic space. Larger values of P did not improve generalization and sometimes hurt it, possibly due to overfitting or prototype redundancy.

Ablation on Overconfidence Regularizers

To complement the main results, we evaluate alternative strategies for mitigating overconfident embeddings in hyperbolic space. Specifically, we compare our proposed **Penalized Busemann Loss** against entropy regularization, confidence penalty (KL to uniform), and logit margin penalty. Each variant replaces only the penalty term in the Busemann component while keeping all other settings identical (backbone, optimizer, loss weights, etc.), ensuring a controlled one-factor analysis.

Overall, confidence-based penalties yield stronger performance on “Old” categories but degrade novel discovery, while entropy maximization is least effective under domain shift. Our Penalized Busemann Loss achieves the most balanced results, underscoring the importance of geometry-aware regularization for stable and generalizable discovery in curved spaces.

Table 21: Ablation of overconfidence regularizers on Office-Home. Our Penalized Busemann Loss achieves the best overall trade-off between known and novel class performance.

Regularizer	**All**	**Old**	**New**
Entropy	51.63	57.31	41.86
Confidence Penalty	56.69	**61.15**	49.10
Logit Margin	54.49	56.54	51.02
Penalized Busemann (Ours)	**56.78**	59.23	**53.21**

Geometric Motivation and Theoretical Properties

1. Busemann-based Penalty Aligns with Hyperbolic Geometry. In the Poincaré ball, the *Busemann function* centered at prototype p is

$$B_p(z) \;=\; \log\frac{\|z-p\|^2}{1-c\,\|z\|^2}\,,$$

which measures signed distance to the horosphere through p. Penalizing $B_p(z)$

- aligns directly with the true geodesic bisector between classes,
- adapts automatically to curvature c via the $(1-c\|z\|^2)$ term,
- yields *uniform* radial repulsion—no directional bias in hyperbolic space.

2. Theoretical Properties.

- *Isometry-invariance:* Horosphere distances are invariant under hyperbolic isometries, preserving model symmetry.
- *Convexity in Busemann coordinates:* Under the change of variables $u = B_p(z)$, the penalty is convex, improving optimization stability.
- *Unified geometric form:*

$$\underbrace{\log\frac{\|z-p\|^2}{1-c\,\|z\|^2}}_{\text{Busemann}} + \underbrace{\phi\,\log(1-\|z\|^2)}_{\text{radial repulsion}} \quad\longrightarrow\quad \text{single, curvature-aware loss.}$$

Overall, unlike off-the-shelf entropy or KL penalties that operate in Euclidean logit-space, our Penalized Busemann Loss is *intrinsically hyperbolic*, principled by geometry, and thus yields superior control over overconfidence in the Poincaré embedding.

Choice of Hyperbolic Model: Poincaré vs. Lorentz

We adopt the Poincaré ball model rather than the Lorentz model due to both theoretical and empirical considerations. First, our Tangent CutMix strategy (Sec. 3.3.3) relies on linear interpolation via exponential/logarithmic maps, which have closed-form and numerically stable expressions in the Poincaré model. In contrast, Lorentz formulations require projection onto the hyperboloid with normalization constraints, complicating implementation. Second, the conformal property of the Poincaré ball (angle-preserving) ensures compatibility with cosine similarity in the tangent space, which is essential for the angular term in our hybrid contrastive loss (Eq. 6). Third, while we experimented with a Lorentzian variant, we observed unstable gradients and weaker convergence when optimizing Busemann-aligned and contrastive losses. Finally, the bounded geometry of the Poincaré ball allows for intuitive visualization and interpretability (e.g., Fig. 4), which is useful for cluster analysis in DG-GCD.

For completeness, we compare the two models empirically on Office-Home under the DG-GCD setting:

The Poincaré ball consistently outperforms the Lorentz model across all splits, supporting our choice as the more effective and stable geometry for HIDISC.

Model	All	Old	New
Poincaré Ball	**56.78**	**59.23**	**53.21**
Lorentz	54.28	56.01	51.41

Table 22: Comparison of Poincaré ball and Lorentz models in the DG-GCD setting on Office-Home.

Varying the Number of Source and Target Domains

To test robustness under different domain splits, we conducted an ablation where the number of source and target domains was systematically varied while keeping the total image budget fixed. For each configuration, the remaining domains were treated as target domains, and results were averaged across all possible target combinations.

Training Domains	Method	All ↑	Old ↑	New ↑
Art	DG²CD-Net	54.47	53.65	55.54
	HIDISC	54.33	54.75	53.53
Art + Clipart	DG²CD-Net	66.67	65.78	67.99
	HIDISC	67.34	70.78	62.02
Art + Clipart + Product	DG²CD-Net	66.83	66.89	66.75
	HIDISC	**69.86**	**79.17**	54.24

Table 23: Ablation varying the number of source and target domains (Office-Home). HIDISC consistently matches or exceeds DG²CD-Net across all scenarios.

Across all settings, HIDISC maintains competitive or superior performance relative to DG²CD-Net. Notably, with three source domains, HIDISC achieves the largest gain on Old classes, while still retaining strong performance on New categories. These results highlight that HIDISC generalizes robustly even when the number of target domains decreases, consistent with its source-only, domain-agnostic training paradigm.

13.9 Hyperparameter Sensitivity Analysis

We appreciate the concern about hyperparameter complexity. In fact, we conducted a comprehensive ablation over all key knobs—curvature c, slope ϕ, radius before mapping, contrastive weight α_d, and loss-term weights ($\lambda_1, \lambda_2, \lambda_3$) on the OfficeHome dataset. We found that only the penalized-Busemann slope and mapping radius materially affect results, with robust defaults ($\phi = 0.75$, radius = 1.5) generalizing across benchmarks. For the remaining loss weights, setting $(\lambda_1, \lambda_2, \lambda_3) = (0.60, 0.25, 0.15)$ balances our alignment, contrastive, and outlier terms and yields 56.78% overall accuracy (59.23% old, 53.21% new). Swapping to $(0.15, 0.60, 0.25)$ or $(0.25, 0.15, 0.60)$ only reduces overall accuracy to 52.12% and 51.37%, respectively—drops of $4\breve{}5$ points—demonstrating that our defaults achieve near-optimal performance without exhaustive tuning. Table 24 shows these results.

Table 24: Ablation study on the loss term weights ($\lambda_1, \lambda_2, \lambda_3$) on the OfficeHome dataset. Our default configuration (Config 1) provides the best balance and overall accuracy.

	Loss Weights			Accuracy (%)		
Configuration	λ_1	λ_2	λ_3	Avg. (All)	Avg. (Old)	Avg. (New)
Config 1 (ours)	0.60	0.25	0.15	**56.78**	**59.23**	**53.21**
Config 2	0.15	0.60	0.25	52.12	53.33	50.07
Config 3	0.25	0.15	0.60	51.37	52.17	50.01

13.10 Comprehensive comparative analyses of HIDISC across multiple datasets

This section presents a concise comparative analysis of HIDISC on PACS, Office-Home, and DomainNet in Table-25, 26 and 27 respectively. Each dataset challenges HIDISC with unique domain shifts, showcasing its adaptability and robustness. This evaluation aims to validate HIDISC's performance against established benchmarks, highlighting it's strengths and identifying opportunities for advancement in domain generalization.

Methods	Art Painting → Sketch			Art Painting → Cartoon			Art Painting → Photo			Photo → Art Painting			Photo → Cartoon			Photo → Sketch		
PACS	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	37.44	**50.73**	19.5	47.4	61.3	35.25	76.05	87.13	64.64	53.17	77.31	31.67	47.01	55.54	39.57	31.87	37.57	24.16
GCD [15]	32.02	41.53	19.12	46.78	60.35	28.57	79.16	99.45	48.73	74.73	80.26	67.31	57.53	60.46	53.6	46.23	**48.56**	43.08
SimGCD [25]	29.35	17.3	62.12	23.08	28.26	16.32	51.98	74.44	33.26	46.29	48.96	43.17	34.26	44.91	20.35	24.84	31.88	5.68
CDAD-Net [18]	46.02	45.95	46.21	51.71	53.43	49.46	99.04	99.21	98.9	76.61	76.97	76.19	56.78	56.67	56.93	**46.65**	46.15	48.01
GCD With Synthetic	45.78	36.71	58.01	54.84	73.47	38.57	82.6	66.29	99.39	79	86.84	72.02	53.56	67.93	41.01	44.18	47.78	39.32
CDAD-Net with Synthetic	43.09	42.53	44.6	49.45	59.31	36.58	99.16	99.21	99.12	65.38	62.83	68.36	42.92	41.97	44.15	41.51	43.79	35.32
Hyp-GCD [22]	45.78	36.71	58.01	54.84	**73.47**	38.57	82.6	66.29	99.39	79	86.84	72.02	53.56	67.93	41.01	44.18	47.78	39.32
Hyp-SelfEx [64]	45.03	36.57	56.45	62.33	71.28	54.52	98.37	98.17	98.57	88.21	91.38	85.38	59.26	**72.19**	47.96	39.24	37.24	41.94
Hyp-SimGCD	22.83	3.33	**75.88**	27.78	32.39	21.77	57.39	72.03	45.2	44.57	63.14	22.88	35.09	43.11	24.63	23.82	12.99	53.31
Hyp-DG²CD-Net [1]	46.79	43.75	51.26	**63.65**	61.78	**65.65**	**99.61**	99.6	**99.63**	89.87	93.25	85.79	57.36	60.41	54.1	46.58	43.32	51.35
DG²CD-Net	46.79	38.13	58.49	57.96	73.38	44.48	99.34	**99.7**	98.97	86.67	91.87	82.04	**62.97**	71.18	55.8	45.72	36.53	**58.13**
HIDISC(Ours)	**48.99**	43.43	57.16	62.29	59.79	64.95	99.19	99	99.48	**90.7**	**93.61**	**87.19**	61.84	60.04	**63.76**	46.56	43.92	50.44

Methods	Sketch → Art Painting			Sketch → Cartoon			Sketch → Photo			Cartoon → Art Painting			Cartoon → Sketch			Cartoon → Photo		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	23.93	26.53	21.61	40.61	58.92	24.62	33.29	33.88	32.69	38.09	47.36	29.82	33.57	35.67	30.74	41.38	39.08	43.74
GCD [15]	33.25	39.09	25.43	40.89	48.14	31.17	46.86	59.28	28.22	58.15	78.52	30.86	36	44.83	24.04	75.75	85.88	60.55
SimGCD [25]	21.19	31.91	8.67	23.17	36.77	5.4	34.22	27.46	40.8	38.38	42.07	34.07	34.84	33.94	37.31	53.05	45.85	59.06
CDAD-Net [18]	87.99	84.32	**92.28**	51.88	51.77	52.02	99.04	99.21	98.9	73.05	76.88	68.57	41.84	42.71	39.49	99.22	99.47	99.01
GCD With Synthetic	82.15	85.13	79.5	44.3	48.22	40.89	**99.49**	**99.76**	99.21	63.01	63.73	62.37	35.66	29.95	43.36	99.43	99.47	99.39
CDAD-Net with Synthetic	61.91	69.45	53.12	48.59	53.13	42.67	68.44	63.5	72.56	67.24	65.28	69.52	42.05	39.61	48.67	99.34	99.47	99.23
Hyp-GCD [22]	82.15	85.13	79.5	44.3	48.22	40.89	99.49	99.76	99.21	63.01	63.73	62.37	35.66	29.95	43.36	**99.43**	99.47	99.39
Hyp-SelfEx [64]	88.77	94.03	84.09	57.11	**73.52**	42.76	98.16	97.82	98.51	89.11	89.1	89.12	45.31	37.02	56.51	98.34	98.06	98.64
Hyp-SimGCD	22.82	22.77	22.87	32.26	45.48	15	27.68	7.79	44.25	23.62	35.79	9.4	30.11	32.22	24.38	33.9	23.76	42.35
Hyp-DG²CD-Net [1]	89.6	92.63	85.95	57.47	60.91	53.79	99.28	99.15	99.48	**93.26**	**95.89**	**90.1**	46.08	42.94	50.69	99.34	**99.5**	99.11
DG²CD-Net	88.75	93.52	84.49	56.76	72.14	43.33	99.13	98.7	**99.57**	90.77	93.37	88.46	**49.2**	43.18	**57.33**	95.57	91.62	**99.64**
HIDISC(Ours)	**92.36**	**94.59**	89.67	**65.7**	68.39	**62.83**	98.74	99.05	98.29	88.57	92.14	84.28	47.07	**53.12**	38.19	98.86	99.45	97.99

Table 25: Detailed comparison of our proposed HIDISC on DG-GCD with respect to referred literature for PACS Dataset.

Methods	Art → Clipart			Art → Product			Art → Real World			Clipart → Art			Clipart → Real World			Clipart → Product		
Office-Home	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	18.88	20.86	15.79	30.34	35.42	21.83	29.52	32.76	24.85	14.96	15.6	14.12	18.59	20.12	16.4	30.39	32.51	26.84
GCD [15]	31.65	32.11	30.93	63.18	64.35	61.22	63.85	66.56	59.96	51.96	52.7	51	62.62	65.29	58.79	60.59	67.13	49.61
SimGCD [25]	24.54	34.35	8.09	41.95	57.92	13.54	46.78	65.54	14.73	31.11	39.56	11.88	25.66	37.66	5.15	28.88	41.38	12.96
CDAD-Net [18]	30.95	33.65	26.43	64.99	68.04	59.32	67.5	**70.89**	61.72	53.36	56.05	47.23	64.7	69.4	55.25	**67.02**	68.8	**63.7**
GCD With Synthetic	29.86	31.04	28.02	57.92	63.12	49.19	59.47	59.59	59.29	53.3	52.84	53.89	61.46	58.27	66.06	63.84	64.04	63.51
CDAD-Net with Synthetic	**31.97**	**35.1**	26.71	65.39	68.94	62.51	**67.83**	70.87	62.64	53.51	56.65	46.37	66.97	**69.76**	62.2	61.4	65.55	57.4
Hyp-GCD [22]	28.99	29.1	29.28	67.3	66.38	64.99	63.79	63.14	62.05	45.88	43.92	40.44	60.27	58.07	54.36	63.33	62.02	60.04
Hyp-SelfEx [64]	30.42	28.84	32.89	64.26	67.7	58.5	64.53	60.91	69.73	49.8	49.18	50.61	63.78	59.62	69.76	64.23	66.35	60.67
Hyp-SimGCD [25]	23.34	34.69	4.31	38.5	54.96	7.84	46.37	70.58	5.03	21.82	29.42	4.55	25.98	37.79	5.83	29.93	41.55	8.28
Hyp-DG²CD-Net [1]	27.89	26.84	29.68	66.02	**70.03**	59.95	63.16	62.55	64.2	45.94	47.43	43.3	55.96	56.65	54.79	62.71	67.75	55.09
DG²CD-Net	31.51	31.96	30.81	**67.46**	68.73	**65.32**	64.45	60.25	**70.48**	50.76	48.76	53.36	64.77	60.58	**70.79**	65.34	67.48	61.76
HIDISC(Ours)	31.91	30.46	**34.18**	64.81	68.19	59.14	66.28	65.6	67.26	**58.65**	**60.68**	**56.02**	**67.82**	66.36	69.92	66.6	**70.19**	60.59

Methods	Product → Art			Product → Real World			Product → Clipart			Real World → Art			Real World → Product			Real World → Clipart		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	23.2	24.64	21.33	31.21	35.45	25.13	19.27	20.52	17.31	32.22	35.79	27.58	44.67	52.21	32.03	20.8	23.71	16.26
GCD [15]	50.27	48.18	52.99	65.07	63.09	67.91	29.08	29.22	28.87	54.26	54.05	54.55	69.04	72.76	62.79	31.04	34.93	24.97
SimGCD [25]	38.28	50.42	10.66	48.36	67.07	16.41	22.45	32.37	11.34	48.95	66.79	8.36	57.19	69.23	44.15	21.7	31.46	5.33
CDAD-Net [18]	50.1	52.43	44.67	66.47	**72.13**	56.81	31.36	34.6	25.94	54.68	58.07	46.96	61.39	64.79	55.06	31.78	**36.02**	24.69
GCD With Synthetic	49.18	46.54	52.61	63.4	59.67	**68.77**	28.43	27.72	29.55	51.71	61.55	38.91	61.14	65.34	54.1	26.38	28.11	23.68
CDAD-Net with Synthetic	54.12	57.67	46.04	66.97	70.2	61.46	32.34	**35.13**	28.68	53.72	56.89	46.5	56.47	62.33	45.62	31.19	33.67	27.02
Hyp-GCD [22]	45.43	44.46	42.73	61.63	59.64	56.3	27.15	28.35	30.39	45.82	45.28	44.33	64.34	63.74	**62.84**	27.66	28.25	29.25
Hyp-SelfEx [64]	51.97	48.53	**56.44**	66.06	65.07	67.49	29.01	28.93	29.13	55.43	53.52	**57.91**	65.3	73.33	51.83	30.12	29.83	30.57
Hyp-SimGCD [25]	33.29	45.26	6.07	40.77	60.68	6.75	16.62	23.46	5.15	46.39	63.59	7.24	44.88	66.32	4.96	20.98	30.97	4.24
Hyp-DG²CD-Net [1]	44.56	41.99	49.14	61.2	62.49	59.04	26.45	26.55	26.28	46.99	48.71	43.93	63.96	64.73	62.78	27.94	27.8	28.17
DG²CD-Net	52.45	50.51	54.98	67.87	69.88	64.97	30.71	30.05	31.75	52.31	49.42	56.07	67.37	71.65	60.19	31.28	31.13	**31.51**
HIDISC(Ours)	**60.12**	**65.52**	53.08	**68.95**	71.13	65.81	**32.63**	32.21	**33.3**	**61.54**	**71.14**	49.05	**69.22**	**73.41**	62.18	**32.79**	35.85	28.02

Table 26: Detailed comparison of our proposed HIDISC on DG-GCD with respect to referred literature for Office-Home Dataset

DomainNet															
Methods	**Sketch → Real**			**Sketch → Quickdraw**			**Sketch → Infograph**			**Sketch → Painting**			**Sketch → Clipart**		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	47.17	47.92	44.95	12.13	12.1	12.21	11.99	12.68	10.28	30.95	33.02	25.75	32.64	34.29	28.64
GCD [15]	51.13	51.88	48.92	**16.08**	15.65	**17.2**	12.6	12.57	12.68	35.25	35.96	33.46	31.22	30.85	32.1
SimGCD [25]	3.11	3.47	2.32	2.31	2.4	2.1	3.16	2.27	5.24	4.1	2.57	5.62	3.02	2.3	4.07
CDAD-Net [18]	48.21	47.7	49.77	12.27	11.52	14.24	12.07	12.69	11.34	35.47	36.39	32.86	18.63	17.52	20.39
GCD With Synthetic	53	51.71	47.64	13.71	13.79	13.99	12.24	11.99	11.37	35.43	34.12	30.83	22.49	22.2	21.49
CDAD-Net with Synthetic	47.11	46.09	49.4	12.75	13.1	14.05	12.52	13.04	11.92	35.87	36.73	33.35	18.99	17.68	21.07
Hyp-GCD [22]	47.08	49.78	**56.61**	12.88	13.37	14.66	11.04	11.32	11.95	31.33	31.84	33.1	17.87	18.85	21.23
Hyp-SelfEx [64]	52.34	53.02	50.35	12.91	12.76	13.32	12.27	12.42	11.89	33.82	34.49	32.14	19.63	19.11	20.87
Hyp-SimGCD [25]	0.54	0.64	0.25	0.29	0.4	0.001	1.52	0.001	3.3	1.19	0.001	4.54	1.07	0.76	1.55
Hyp-DG²CD-Net [1]	36.33	34.13	41.88	13.51	12.96	14.96	9.74	9.3	10.73	24.22	23.33	26.41	14.17	14.17	14.16
DG²CD-Net	53.67	**55.48**	48.35	15.9	**16**	15.63	14.63	15.66	12.06	37.44	39.53	32.19	30.47	32.89	24.58
HIDISC(Ours)	**55.27**	54.89	56.24	14.61	13.83	16.68	**15.35**	**16.3**	**13.2**	**39.8**	**41.15**	**36.49**	**35.74**	**38.61**	**28.8**

Methods	**Clipart → Infograph**			**Clipart → Quickdraw**			**Clipart → Sketch**			**Clipart → Real**			**Clipart → Painting**		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	12.18	12.64	11.03	12.13	12.1	12.21	24.76	26.24	21.27	44.14	45.43	40.34	26.76	28.7	21.91
GCD [15]	14.03	14.64	12.49	14.94	14.67	15.65	25.33	27.68	19.78	53.23	55.48	46.62	34.82	36.82	29.83
SimGCD [25]	2.03	0.4	3.94	0.5	0.3	1	1	0.02	3.842	1.64	1.07	2.42	2.07	2.05	2.13
CDAD-Net [18]	12.79	12.96	12.87	12.06	11.59	12.78	19	19.17	18.76	47.06	44.62	49.2	34.45	36.02	32.85
GCD With Synthetic	11.46	12.03	10.04	12.68	12.57	12.95	18.74	20.54	14.47	50.11	52.26	43.79	32.67	34.91	27.06
CDAD-Net with Synthetic	13	13.37	12.56	12.07	11.76	12.89	17.46	18.03	16.67	48.25	47.51	49.6	33.23	32.79	34.2
Hyp-GCD [22]	11.52	11.6	11.79	12.98	13.48	14.79	15.33	15.84	17.11	44.82	46.52	50.82	29.4	29.92	31.21
Hyp-SelfEx [64]	11.6	11.7	11.35	12.89	12.61	13.64	17.1	17.92	15.16	50.55	51.11	48.92	33.46	34.31	31.34
Hyp-SimGCD [25]	1.57	0.03	3.38	0.29	0.4	0.001	1.06	0.044	2.46	0.73	0.77	0.61	1.24	0.214	4.15
Hyp-DG²CD-Net [1]	11.95	11.81	12.27	14.1	13.43	15.88	16.21	15.76	17.32	48.04	45.92	53.42	31.28	30.73	32.65
DG²CD-Net	**15.81**	**17.09**	12.63	**14.53**	14.14	15.58	26.86	29.49	20.64	54.54	56.03	50.17	36.81	38.87	31.67
HIDISC(Ours)	15.7	16.34	**14.25**	14.35	13.83	15.72	26.6	27.56	24.24	55.04	54.49	56.42	39.08	40.13	36.5

Methods	**Painting → Infograph**			**Painting → Quickdraw**			**Painting → Sketch**			**Painting → Real**			**Painting → Clipart**		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
ViT [60]	12.2	13.1	9.94	12.13	12.1	12.21	23	24.78	18.79	51.53	54.16	43.8	26.57	28.08	22.92
GCD [15]	12.87	12.67	13.37	10.74	10.56	11.21	21.49	22.26	19.68	52.12	51.86	52.86	25.32	24.79	26.6
SimGCD [25]	3.2	2.6	3.8	3.5	2.32	4.65	4.23	3.56	4.86	4.2	3.52	5	4.49	3.6	5.23
CDAD-Net [18]	11.65	12.49	10.66	11.98	11.2	12.44	17.11	17.68	16.32	49.04	48.63	50.27	20.06	19.74	20.57
GCD With Synthetic	10.86	10.56	9.84	11.81	11.8	11.77	17.26	16.25	13.83	49.1	47.3	42.04	19.3	19.45	18.04
CDAD-Net with Synthetic	11.53	12.32	10.59	11.86	10.71	12.32	17.29	18.45	15.7	48.4	50.23	49.7	17.44	15.92	19.86
Hyp-GCD [22]	12.12	12.38	12.96	12.32	13.05	14.97	16.78	17.54	19.42	48.4	50.39	55.41	19.34	20.23	22.39
Hyp-SelfEx [64]	12.43	12.24	12.89	12.98	12.76	13.55	18.28	18.72	17.22	51	51.59	49.27	20.42	20.11	21.18
Hyp-SimGCD [25]	1.52	0.002	3.3	0.35	0.43	0.17	1.034	0.002	2.46	0.48	0.63	0.05	0.96	0.001	2.5
Hyp-DG²CD-Net [1]	12.94	12.65	13.59	14.14	13.45	15.96	17.54	16.84	19.25	50.59	49.07	54.45	19.95	19.31	21.49
DG²CD-Net	15.71	16.72	13.22	12.9	12.66	13.53	23.14	25.23	18.19	55.07	56.97	49.5	27.6	29.07	24.03
HIDISC(Ours)	15.98	17.15	13.36	13.03	12.62	14.1	25.78	27.37	21.9	57.1	58.76	52.9	34.29	37.99	25.34

Table 27: Detailed comparison of our proposed HIDISC on DG-GCD with respect to referred literature for DomainNet Dataset

13.11 Extended Tables for additional Baselines

Table 28: **Extended clustering accuracy comparison** (%) including Hyp-SimGCD, HiDISC, and Upper-Bound (UB) on PACS, Office-Home, and DomainNet. (**Bold**: best, underline: second best).

Method	Venue	PACS			Office-Home			DomainNet			Avg.		
		All	Old	New	All	Old	New	All	Old	New	All	Old	New
Hyp-SelfEx [64]	ECCV'24	72.44	74.70	71.20	52.91	52.65	52.96	29.30	30.45	26.37	51.55	52.60	50.18
Hyp-SimGCD [25]	ICCV'23	31.82	32.90	33.49	32.41	46.61	5.85	0.92	0.29	1.91	21.72	26.6	13.75
HiDISC (Ours) (2 Synth)	–	**75.07**	**75.54**	**74.52**	**56.78**	**59.23**	**53.21**	**30.51**	**31.40**	**28.41**	**54.12**	**55.39**	**52.05**
CDAD-Net (Upper-Bound)[18]	–	83.25	87.58	77.35	67.55	72.42	63.44	70.28	76.46	65.19	73.69	78.82	68.66

Comparison with Hyperbolic Prototype Methods: HPDR [56] address domain generalization for face anti-spoofing by leveraging hyperbolic prototypes to improve robustness across visual domains. While effective in this binary closed-set setting, their method (HPDR) does not extend to the more challenging DG-GCD problem, which requires discovering novel categories in unseen domains without target supervision. To clarify this distinction, we re-implemented their embedding learning component within our framework and evaluated it under the DG-GCD protocol. As shown in Table 29, HPDR performs significantly worse than our proposed HiDISC on the Office-Home benchmark, highlighting the importance of discovery-oriented objectives and domain-aware augmentations.

Method	All ↑	Old ↑	New ↑
HPDR [56]	15.05	16.55	12.58
HiDISC	56.78	59.23	53.21

Table 29: Comparison of Hu et al.'s method (HPDR) and HiDISC under the DG-GCD setting on Office-Home.

The performance gap can be attributed to three factors: (i) HPDR is designed for binary closed-set classification and lacks mechanisms to create prototypes for unseen classes; (ii) it optimizes closed-set objectives without clustering or discovery losses, yielding near-random novel-class assignments; and (iii) it does not incorporate domain-generalization strategies. In contrast, HiDISC couples prototype anchoring with domain-generalization regularizers and discovery losses, enabling robust alignment and clustering across unseen domains and categories.