
Supplementary Material for HiDISC : A Hyperbolic Framework for Domain Generalization with Generalized Category Discovery

Vaibhav Rathore¹ Divyam Gupta¹ Biplab Banerjee¹
Indian Institute of Technology Bombay¹
{vaibhav.rathor.in, divs25.iitb, getbiplab}@gmail.com

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. 1, Sec. 2).
- **Mathematical background** on Poincaré ball geometry and key hyperbolic operations used in our model (Sec. 3).
- **Details on synthetic data generation**, illustrating the lightweight augmentation pipeline and example images (Sec. 4, Sec. 4.1).
- **Comprehensive ablations** exploring the effects of synthetic domain count, adaptive outlier margin, embedding dimension, curvature learning, and domain combinations on model performance (Sec. 5.1–8.6, Sec. 8.4).
- **Theoretical justification** outlining generalization bounds and domain discrepancy advantages of hyperbolic embeddings over Euclidean embeddings (Sec. 6).
- **Extended comparisons** against state-of-the-art methods on three major benchmarks (PACS, Office-Home, DomainNet), with detailed accuracy tables and discussions (Sec. 8.10).
- **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.

1 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** [1] 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 1, same-class embeddings exhibit consistently high alignment in hyperbolic space (cosine similarity ≈ 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 1: 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).

2 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 2: 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.

3 Poincaré Ball Geometry

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

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

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

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

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

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

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

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

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

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

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

This ensures full compatibility with our Euclidean baseline.

4 Technical details about Synthetic Data Generation

To implement the lightweight synthetic domain augmentation described in Sec. 3.3.1 of the main paper, following [2] 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, 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.

4.1 Synthetic Domain Images

In Figure 1, 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 $P_G(x | t)$ 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}_{gen} satisfies $\frac{\partial^2 \mathcal{L}_{\text{gen}}}{\partial N^2} > 0$ for $N > 2$, indicating diminishing returns and potential overfitting.

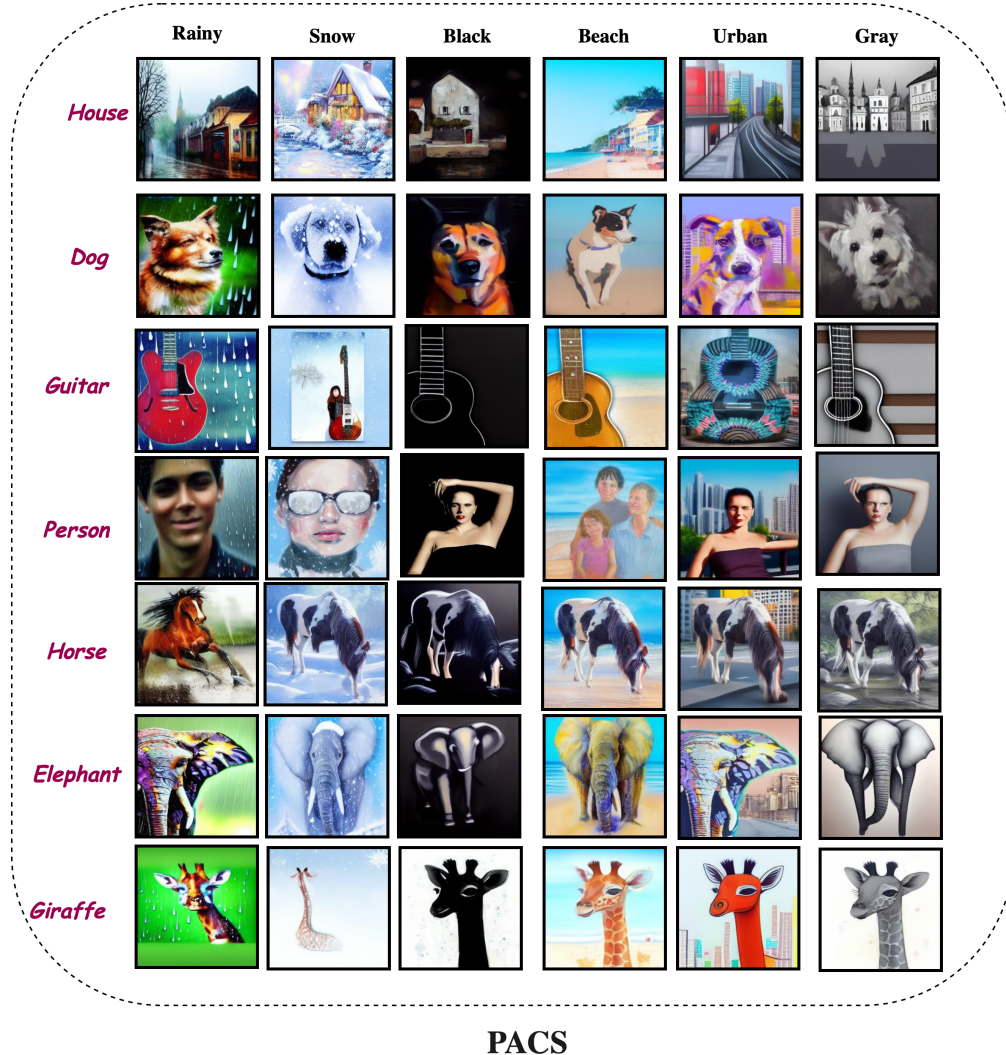


Figure 1: Examples of synthetic images generated for the PACS dataset, demonstrating various styles and domains across seven different categories.

- **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.

5 Ablations

5.1 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 $k \in \{0, \dots, 6\}$ synthetic domains in addition to the original source data and report three metrics.

Figure 2 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.

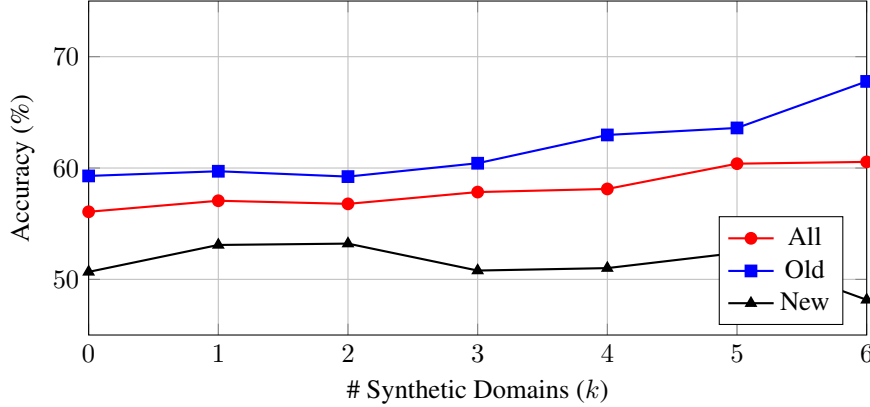


Figure 2: Ablation on the number of synthetic domains per image (k) for the Office-Home dataset. Adding up to six synthetic styles steadily improves overall (*All*) and seen-class (*Old*) accuracies, while novel-class (*New*) performance peaks at $k = 2$ before declining.

5.2 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

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

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 3: 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 3 shows, moderate quantiles ($\alpha = 0.80$) yield the best novel-class discrimination .

5.3 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

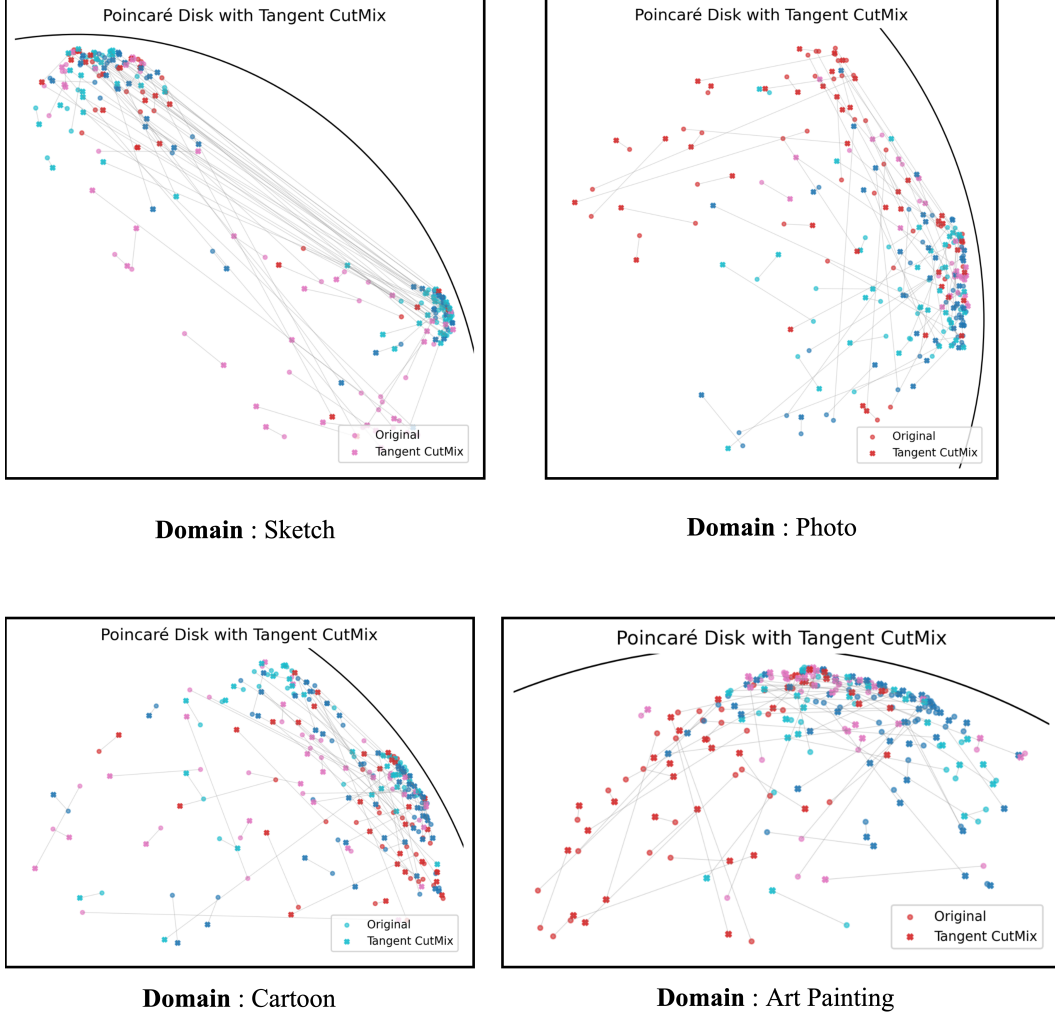


Figure 3: **UMAP-projected Poincaré-disk embeddings of original and Tangent CutMix samples on PACS.** We first embed features into the Poincaré ball and then apply UMAP to project onto 2D for visualization. Each subplot corresponds to one PACS domain (Sketch, Photo, Cartoon, Art Painting), showing four 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.

5.4 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 4) and thus bring negligible gains in novel-class discovery.

Table 4: 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

5.4.1 Effect of other domain combinations on performance

Table 5: 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 α_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 6: Ablation on the maximum angular weight α_d^{\max} for the Office-Home dataset. We report overall accuracy (*All*), seen-class accuracy (*Old*), and novel-class accuracy (*New*).

α_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

5.5 Observation on Old-New class splits:

Table 7 illustrates the impact of varying base (old) and novel (new) class splits on the performance of HiDISC on the Office-Home dataset.

In Table 7, 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 7: 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 ↓, New ↑).
- **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.

6 Theoretical Justification: Generalization Bounds in Hyperbolic Space

6.1 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:

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

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 [3], 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 $\Omega(\log n)$ [4];
- Hyperbolic space is $\mathcal{O}(1)$ [5].

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. [6, 7]), 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.

6.2 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)

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

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

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

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

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)

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

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

Table 8 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 8: 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

7 Dataset details

Our experiments were carried out using three benchmark datasets: (i) **PACS** [8], (ii) **Office-Home** [9], and (iii) **DomainNet** [10].

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 9, 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 9: Source-target configurations for DomainNet

8 Further Comparison with Literature

8.1 DG-GCD-Specific Adaptations for Baselines

As described in [2], 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 [11]. As for the Hyp-DG²CD-Net, we adapted the code directly from the authors, modifying the two loss functions in hyperbolic space following the strategy described in [11].

8.2 Implementation Details.

We build HiDISC in PyTorch and run all experiments on a single NVIDIA A100-SXM4-80GB GPU. Following [12, 2], we adopt a DINO-pretrained ViT-B/16 [13, 1], 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 [11] 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 [2] 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.

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

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 [12, 2]. 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 $\{\mathbf{p}_j\}$

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

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

- 9: Apply element-wise tanh to ensure embeddings lie strictly within Poincaré ball \mathbb{D}_c^d
- 10: Construct batch prototypes $P_Y = \{\mathbf{p}_{y_i}\}_{i=1}^B$ repeated V times corresponding to labels Y
- 11: Compute **penalized Busemann loss** [14]:

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

- 12: Compute **hybrid hyperbolic contrastive loss** [11] on V views:

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

▷ $\alpha(e)$ is epoch-dependent weighting between geodesic and angular similarity

- 13: **if** $e = 1$ and first batch **then**
- 14: Compute adaptive margin δ as quantile over distances $d_{\mathbb{H}}(Z_{\text{mix}}, \mathbf{p}_j)$ to prototypes
- 15: **end if**
- 16: Compute **adaptive outlier loss**:

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

- 17: Aggregate total loss:

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

- 18: Backpropagate \mathcal{L} and update parameters θ, ϕ, c via optimizer
 - 19: **end for**
 - 20: Update learning rate scheduler
 - 21: **end for**
-

8.3 Effect of Reduced Synthetic Domains on Model Performance

Table 10 presents a detailed comparison between HiDISC and DG²CD-Net on the PACS dataset when using 2 versus 6 synthetic domains during training. While HiDISC maintains robust performance

Table 10: Performance comparison on the PACS dataset between HiDISC and DG²CD-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, DG²CD-Net experiences a marked degradation, confirming the claim in the main paper regarding DG²CD-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 DG²CD-Net Fail

To better understand the design choices in HiDISC, we also implemented a hyperbolic variant of DG²CD-Net (denoted Hyp-DG²CD-Net). Interestingly, Hyp-DG²CD-Net underperforms both the original DG²CD-Net and our proposed HiDISC. We attribute this gap to several factors:

- **Task Vector Arithmetic Mismatch.** DG²CD-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-DG²CD-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-DG²CD-Net performs consistently worse than both DG²CD-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.

8.4 Learned Curvature Evolution

Our model learns the hyperbolic curvature c jointly during training, adapting it to dataset-specific geometry. Figure 4 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.

8.5 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

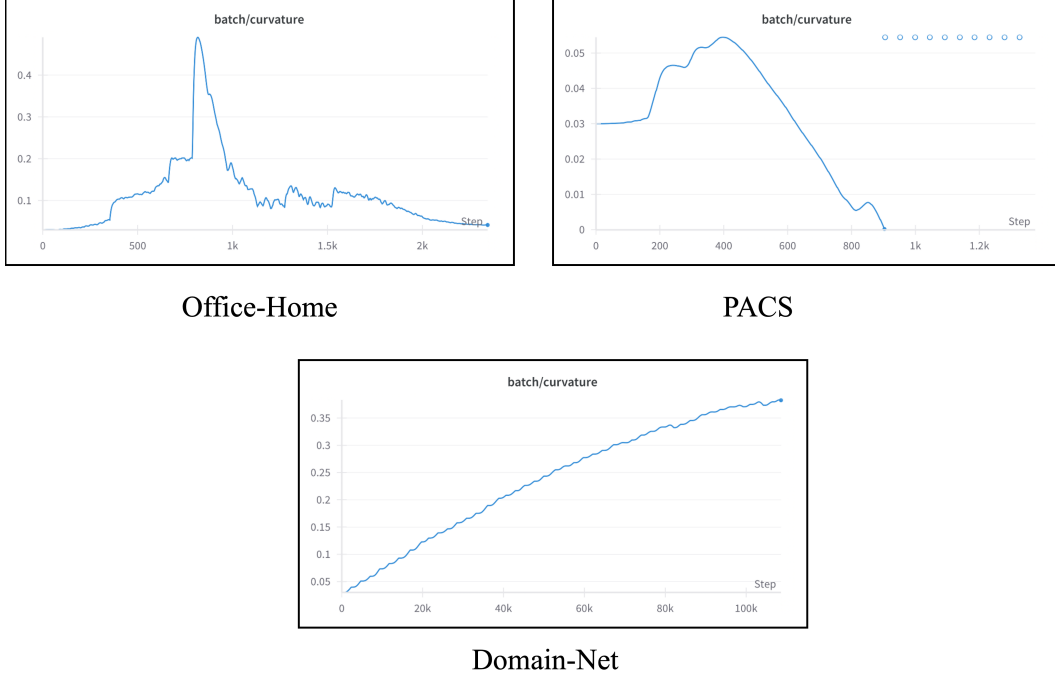


Figure 4: Learned curvature c evolution across training batches for Office-Home (left), PACS (center), and DomainNet (right). These plots were generated from training logs visualized using Weights & Biases (WandB) [15].

Table 11: 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.

8.6 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 5 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.

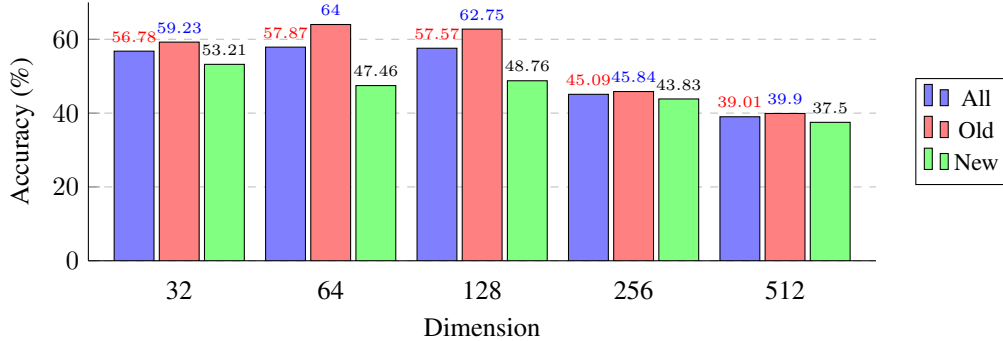


Figure 5: Ablation study on embedding dimension on the Office-Home dataset. Lower dimensions (32, 64, 128) perform better than larger ones (256, 512), suggesting that compact embeddings are beneficial for generalization.

Table 12: 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

8.7 For Non-learnable curvature

8.8 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 13: 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 14: 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:*

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

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. ??) 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 15: 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 \uparrow	Old \uparrow	New \uparrow
Art	DG ² CD-Net	54.47	53.65	55.54
	HiDISC	54.33	54.75	53.53
Art + Clipart	DG ² CD-Net	66.67	65.78	67.99
	HiDISC	67.34	70.78	62.02
Art + Clipart + Product	DG ² CD-Net	66.83	66.89	66.75
	HiDISC	69.86	79.17	54.24

Table 16: 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.

8.9 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 4~5 points—demonstrating that our defaults achieve near-optimal performance without exhaustive tuning. Table 17 shows these results.

Table 17: 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.

Configuration	Loss Weights			Accuracy (%)		
	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

8.10 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-18, 19 and 20 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.

PACS																		
Methods	Art Painting → Sketch			Art Painting → Cartoon			Art Painting → Photo			Photo → Art Painting			Photo → Cartoon			Photo → Sketch		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
VIT [13]	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 [12]	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 [16]	29.35	17.3	<u>62.12</u>	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 [17]	46.02	<u>45.95</u>	46.21	51.71	53.43	49.46	99.04	<u>99.21</u>	98.9	76.61	76.97	76.19	56.78	56.67	<u>56.93</u>	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 [11]	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	<u>47.78</u>	39.32
Hyp-SelfEx [18]	45.03	36.57	56.45	<u>62.33</u>	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	<u>53.31</u>
Hyp-DG ² CD-Net [2]	<u>46.79</u>	43.75	51.26	63.65	61.78	65.65	99.61	<u>99.6</u>	99.63	<u>89.87</u>	<u>93.25</u>	85.79	57.36	60.41	54.1	<u>46.58</u>	43.32	51.35
DG ² CD-Net	<u>46.79</u>	38.13	58.49	<u>57.96</u>	<u>73.38</u>	44.48	<u>99.34</u>	99.7	98.97	86.67	91.87	82.04	62.97	<u>71.18</u>	55.8	45.72	36.53	58.13
HiDISC(Ours)	48.99	43.43	57.16	62.29	59.79	<u>64.95</u>	99.19	99	<u>99.48</u>	90.7	93.61	87.19	<u>61.84</u>	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 [13]	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 [12]	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 [16]	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 [17]	87.99	84.32	92.28	51.88	51.77	52.02	99.04	<u>99.21</u>	98.9	73.05	76.88	68.57	41.84	42.71	39.49	99.22	<u>99.47</u>	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	<u>99.47</u>	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	<u>99.47</u>	99.23
Hyp-GCD [11]	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	<u>99.47</u>	99.39
Hyp-SelfEx [18]	88.77	<u>94.03</u>	84.09	57.11	73.52	42.76	98.16	97.82	98.51	89.11	89.1	<u>89.12</u>	45.31	37.02	<u>56.51</u>	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-DG ² CD-Net [2]	<u>89.6</u>	92.63	85.95	<u>57.47</u>	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
DG ² CD-Net	88.75	93.52	84.49	56.76	<u>72.14</u>	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	<u>89.67</u>	65.7	68.39	62.83	98.74	99.05	98.29	88.57	92.14	84.28	<u>47.07</u>	53.12	38.19	98.86	99.45	97.99

Table 18: Detailed comparison of our proposed HiDISC on DG-GCD with respect to referred literature for PACS Dataset.

Office-Home																		
Methods	Art → Clipart			Art → Product			Art → Real World			Clipart → Art			Clipart → Real World			Clipart → Product		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
VIT [13]	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 [12]	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 [16]	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 [17]	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	<u>69.4</u>	55.25	67.02	<u>68.8</u>	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	<u>53.89</u>	61.46	58.27	66.06	63.84	64.04	<u>63.51</u>
CDAD-Net with Synthetic	31.97	35.1	26.71	65.39	<u>68.94</u>	62.51	67.83	<u>70.87</u>	62.64	<u>53.51</u>	<u>56.65</u>	46.37	<u>66.97</u>	69.76	62.2	61.4	65.55	57.4
Hyp-GCD [11]	28.99	29.1	29.28	<u>67.3</u>	66.38	<u>64.99</u>	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 [18]	30.42	28.84	<u>32.89</u>	64.26	67.7	58.5	64.53	60.91	<u>69.73</u>	49.8	49.18	50.61	63.78	59.62	69.76	64.23	66.35	60.67
Hyp-SimGCD [16]	23.34	<u>34.69</u>	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-DG ² CD-Net [2]	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
DG ² CD-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)	<u>31.91</u>	30.46	34.18	64.81	68.19	59.14	<u>66.28</u>	65.6	67.26	58.65	60.68	56.02	67.82	66.36	<u>69.92</u>	<u>66.6</u>	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 [13]	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 [12]	50.27	48.18	52.99	65.07	63.09	<u>67.91</u>	29.08	29.22	28.87	54.26	54.05	54.55	69.04	72.76	<u>62.79</u>	31.04	34.93	24.97
SimGCD [16]	38.28	50.42	10.66	48.36	67.07	16.41	22.45	32.37	11.34	48.95	<u>66.79</u>	8.36	57.19	69.23	44.15	21.7	31.46	5.33
CDAD-Net [17]	50.1	52.43	44.67	66.47	72.13	56.81	31.36	<u>34.6</u>	25.94	54.68	58.07	46.96	61.39	64.79	55.06	<u>31.78</u>	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	<u>54.12</u>	<u>57.67</u>	46.04	66.97	70.2	61.46	<u>32.34</u>	35.13	28.68	53.72	56.89	46.5	56.47	62.33	45.62	31.19	33.67	27.02
Hyp-GCD [11]	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 [18]	51.97	48.53	56.44	66.06	65.07	67.49	29.01	28.93	29.13	<u>55.43</u>	<u>53.52</u>	57.91	65.3	<u>73.33</u>	51.83	30.12	29.83	<u>30.57</u>
Hyp-SimGCD [16]	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-DG ² CD-Net [2]	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
DG ² CD-Net	52.45	50.51	54.98	<u>67.87</u>	<u>69.88</u>	64.97	30.71	30.05	<u>31.75</u>	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	<u>71.13</u>	65.81	32.63	32.21	33.3	61.54	71.14	49.05	69.22	73.41	62.18	32.79	<u>35.85</u>	28.02

Table 19: 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 [13]	47.17	47.92	44.95	12.13	12.1	12.21	11.99	12.68	10.28	30.95	33.02	25.75	<u>32.64</u>	<u>34.29</u>	<u>28.64</u>
GCD [12]	51.13	51.88	48.92	16.08	<u>15.65</u>	17.2	12.6	12.57	<u>12.68</u>	35.25	35.96	<u>33.46</u>	31.22	30.85	32.1
SimGCD [16]	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 [17]	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 [11]	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 [18]	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 [16]	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-DG ² CD-Net [2]	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
DG ² CD-Net	<u>53.67</u>	55.48	48.35	<u>15.9</u>	16	15.63	<u>14.63</u>	<u>15.66</u>	12.06	<u>37.44</u>	<u>39.53</u>	32.19	30.47	32.89	24.58
HiDISC(Ours)	55.27	<u>54.89</u>	<u>56.24</u>	14.61	13.83	<u>16.68</u>	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 [13]	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 [12]	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 [16]	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 [17]	12.79	12.96	<u>12.87</u>	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 [11]	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 [18]	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 [16]	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-DG ² CD-Net [2]	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
DG ² CD-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)	<u>15.7</u>	<u>16.34</u>	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 [13]	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 [12]	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 [16]	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 [17]	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 [11]	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 [18]	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 [16]	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-DG ² CD-Net [2]	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
DG ² CD-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 20: Detailed comparison of our proposed HiDISC on DG-GCD with respect to referred literature for DomainNet Dataset

8.11 Extended Tables for additional Baselines

Table 21: **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 [18]	ECCV'24	<u>72.44</u>	<u>74.70</u>	<u>71.20</u>	<u>52.91</u>	<u>52.65</u>	<u>52.96</u>	<u>29.30</u>	<u>30.45</u>	<u>26.37</u>	<u>51.55</u>	<u>52.60</u>	<u>50.18</u>
Hyp-SimGCD [16]	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)[17]	–	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 [19] 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 22, 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 [19]	15.05	16.55	12.58
HiDISC	56.78	59.23	53.21

Table 22: 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.

References

- [1] 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.
- [2] 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>.
- [3] Cédric Villani. *Optimal transport: old and new*. Springer, 2009.
- [4] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
- [5] 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.
- [6] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [7] Ines Chami, Aditya Wolf, Pierre Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Hyperbolic neural networks. In *NeurIPS*, pages 5345–5355, 2019.
- [8] 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.
- [9] Hemant 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busmann learning with ideal prototypes. *Advances in neural information processing systems*, 34:103–115, 2021.
- [15] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL <https://www.wandb.com/>. Software available from wandb.com.
- [16] 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.
- [17] 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.

- [18] 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.
- [19] 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.