BEYOND CATEGORY ASSIGNMENTS: LEARNING SAMPLE-PROTOTYPE DYNAMICS FOR GENERALIZED CATEGORY DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalized Category Discovery (GCD) aims to recognize known categories while discovering novel categories from partially labeled data. Existing methods often exploit predicted sample-category relations as pseudo-label supervision for representation learning. However, such supervision does not fully exploit the intermediate geometric structure between samples and candidate prototypes, and relies on uncertain assignments that may reinforce early errors. We introduce a continuous perspective on GCD representation learning by expanding categorical assignments into sample-prototype transport, providing supervision on how representations evolve toward candidate prototypes. Based on this perspective, we propose **DynGCD**, a plug-and-play framework that models continuous sample-prototype transport through prototype-conditioned flow matching while jointly learning uncertain sample-prototype couplings. The couplings preserve multiple plausible category relations and determine which candidate transports contribute to flow learning, while the learned transport consistency and evolving representations in turn refine the couplings. This mutual refinement allows category relations and transport dynamics to progressively co-evolve during training, reducing reliance on fixed categorical pseudo-targets. DynGCD requires only sample representations and category prototypes during training, making it readily applicable to existing GCD frameworks while preserving their native inference procedures. Experiments on generic and fine-grained benchmarks across *parametric*, *non-parametric*, and *vision–language* GCD paradigms demonstrate consistent improvements, with particularly strong gains on novel categories.

1 INTRODUCTION

Generalized Category Discovery (GCD) aims to recognize known classes and discover novel categories from a partially labeled dataset, where the unlabeled subset contains samples from seen and unseen categories (Vaze et al., 2022a). The key challenge arises from asymmetric supervision (Liu & Han, 2025; He et al., 2025b) between known and unknown categories. Known categories are supported by labeled examples that anchor their semantic identities and guide representation learning, whereas novel categories have no category labels, so both the category assignments and representations should be inferred during training. This asymmetry creates a fundamental difficulty: discovering novel categories relies on sufficiently discriminative representations, yet learning such representations in turn requires informative cues about the structure of these unknown categories.

Existing GCD methods can be characterized by how they produce category predictions. *Non-parametric* methods infer category assignments by clustering the representation space through similarity comparisons (Vaze et al., 2022a; Rastegar et al., 2024), whereas *parametric* methods learn an explicit classifier, typically prototype-based, to predict category indices (Wen et al., 2023; Wang et al., 2024; Xu et al., 2025). More recently, *vision–language* approaches additionally incorporate semantic cues from language (Ouldnoughi et al., 2023; Wang et al., 2025; Yang et al., 2025). Despite their different prediction mechanisms, they typically learn representations from partially labeled data through a combination of supervised and self-supervised objectives. Many methods (Rastegar et al., 2024; Wen et al., 2023) further exploit predicted sample-category relations, typically represented by category assignments over classifier prototypes or cluster centers, to provide additional supervision

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

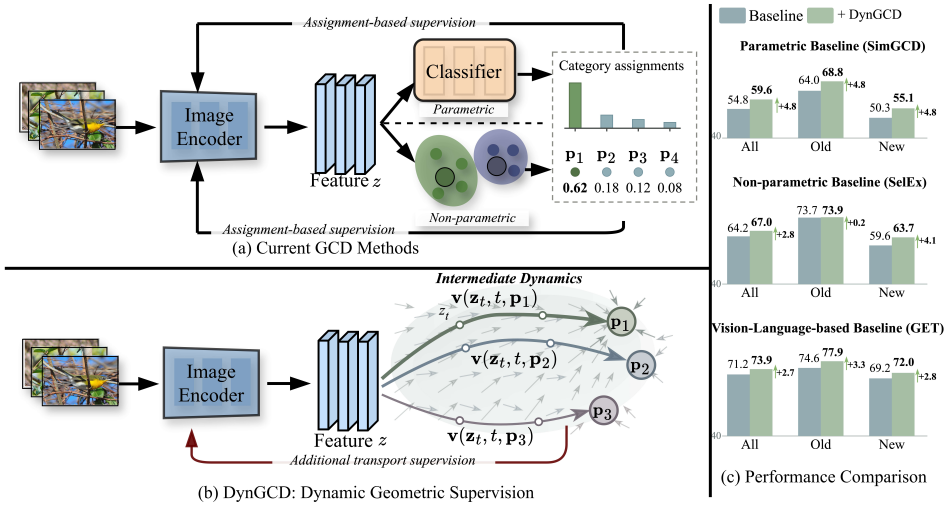


Figure 1: (a) Existing GCD learners use sample-category assignments to supervise representation learning. (b) **DynGCD** acts as an **add-on** to the host learner, exploiting sample–prototype relations as additional **geometric supervision** through prototype-conditioned flow matching. (c) DynGCD consistently improves parametric, non-parametric, and vision–language-based GCD baselines.

for representation learning (*cf.*, Fig. 1a). However, such assignment-based supervision may overlook two aspects of sample-category relations. **1) Geometric Underutilization.** Hard or soft category assignments specify which prototypes are plausible, but do not explicitly exploit the intermediate geometric structure between a sample and its candidate prototypes for representation learning. **2) Uncertain Supervision.** For unlabeled samples, these assignments are themselves model predictions. Directly reusing them as training targets may therefore reinforce early mistakes, creating a self-confirming loop of erroneous assignments.

These limitations motivate us to go beyond assignment-based supervision by modeling sample-category relations as a continuous process. A category prediction specifies a category prototype that the sample representation should be closer to, and we further model the continuous evolution of the representation toward the prototype. Flow matching (Lipman et al., 2023) naturally provides a principled formulation for modeling such continuous dynamics by learning a velocity field along the sample-prototype trajectory. Yet directly constructing such transport from predicted assignments would inherit their uncertainty: for an unlabeled sample, its category identity remains unknown, and the sample-category coupling is inherently ambiguous. Simply selecting the currently most likely category would reduce this uncertainty to a hard pseudo-target, leading to erroneous early assignments. The key challenge is therefore to jointly infer uncertain sample-category couplings and learn the geometric transport they support.

We address this challenge with **DynGCD**, a plug-and-play framework that jointly models uncertain sample-category couplings and continuous sample–prototype transport. To better exploit the geometry of sample-category relations, DynGCD uses category-prototype-conditioned flow matching to expand assignment-based supervision into continuous transport supervision. For uncertain supervision, DynGCD maintains soft couplings over multiple plausible prototypes, estimated from both sample-prototype similarity and flow-transport consistency. The couplings further determine which candidate trajectories contribute to flow learning and how strongly they are weighted. This mutual refinement allows sample-category couplings and geometric transport to progressively co-evolve throughout training, with updated flow transport consistency continuously refining the couplings. As both components operate only on sample representations and category prototypes, DynGCD can be readily integrated into GCD methods with different prediction mechanisms while preserving their native inference procedures. We instantiate DynGCD on SimGCD (Wen et al., 2023), SelEx (Rastegar et al., 2024), and GET (Wang et al., 2025), covering *parametric*, *non-parametric*, and *vision–language* GCD paradigms. As shown in Fig. 1(c), DynGCD consistently improves performance across all three paradigms, with particularly pronounced gains on novel categories.

Our contributions are summarized as follows: **(i)** We introduce a continuous perspective on GCD representation learning by extending categorical assignments to sample–prototype transport, providing auxiliary supervision beyond category correspondence alone. **(ii)** We propose **DynGCD**, a

plug-and-play framework that jointly models uncertain sample–prototype couplings and prototype-conditioned flows, enabling mutual refinement between category relations and transport dynamics. **(iii)** We demonstrate consistent improvements across *parametric*, *non-parametric*, and *vision–language* GCD methods on both generic and fine-grained benchmarks, while preserving their native inference procedures.

2 Related Work

Generalized Category Discovery (GCD). GCD transfers knowledge from labeled examples to recognize known and novel categories in unlabeled data (Vaze et al., 2022a; He et al., 2025b; Tang et al., 2025b). Existing methods can be discussed from three perspectives. *Parametric methods* learn explicit classifiers over both category sets, using self-distillation and distribution-guided debiasing to improve unlabeled predictions (Wen et al., 2023; Liu & Han, 2025). *Non-parametric methods* assign categories by clustering learned representations, with contrastive learning and hierarchical self-supervision improving their discriminability (Vaze et al., 2022a; Rastegar et al., 2023; 2024). *Vision–language methods*, orthogonal to this prediction-based distinction, exploit complementary textual cues through cross-modal embeddings (Zheng et al., 2024; Wang et al., 2025). Across these paradigms, recent representation-level advances exploit visual primitives and compositional structure (Tang et al., 2025a; 2026c), or improve feature geometry and local token information (Liu et al., 2025c; Tang et al., 2026a; Yang et al., 2026). Further work explores hypothesis–verification reasoning and biomedical discovery (Zhang et al., 2026; Tang et al., 2026b). Complementary to these methods, DynGCD explores supervision beyond category assignments by modeling the intermediate dynamics between samples and candidate prototypes through prototype-conditioned transport.

Flow Matching (FM). FM learns a time-dependent velocity field along a prescribed probability path through direct velocity regression (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023). Initially developed for generative modeling (Esser et al., 2024), FM has recently been extended to semantic representation spaces and discriminative tasks, including cross-modal representation learning (Liu et al., 2025b; He et al., 2025a), depth estimation and segmentation (Gui et al., 2025; Bogensperger et al., 2025), few-shot recognition (Jiang et al., 2026; Li et al., 2026; Chen et al., 2026), compositional zero-shot learning (He et al., 2026b), and composed image retrieval (He et al., 2026a). These works demonstrate that FM can effectively model task-specific transport beyond data generation. We jointly refine uncertain sample–prototype couplings and prototype-conditioned flows, allowing category discovery and transport learning to co-evolve while providing geometric supervision beyond category assignments.

3 Preliminaries

Problem Formulation. GCD aims to classify unlabeled samples from known categories while simultaneously discovering those from novel categories (Vaze et al., 2022a). Let $\mathcal{D}_l = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^{N_l} \subset \mathcal{X} \times \mathcal{Y}_l$ denote the labeled dataset, and let $\mathcal{D}_u = \{(\mathbf{x}_i^u, y_i^u)\}_{i=1}^{N_u} \subset \mathcal{X} \times \mathcal{Y}_u$ denote the unlabeled dataset, where y_i^u is unobserved during training. The unlabeled set contains samples from both known and novel categories, such that $\mathcal{Y}_l \subset \mathcal{Y}_u$. We denote the number of labeled categories by $M = |\mathcal{Y}_l|$ and assume that the total number of categories $K = |\mathcal{Y}_u|$ is known, following prior GCD works (Wen et al., 2023; Rastegar et al., 2024; Wang et al., 2025).

3.1 Flow Matching

Flow matching learns a time-dependent velocity field that transports samples between two distributions by regressing the velocity of a prescribed probability path (Lipman et al., 2023). Let $(\mathbf{x}_0, \mathbf{x}_1) \sim \gamma$ denote a source–target pair drawn from a coupling γ between p_0 and p_1. For the straight probability path

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1\,, \qquad t \in [0,1]\,, \tag{1}$$

the target velocity is $\mathbf{u}_t = \mathbf{x}_1 - \mathbf{x}_0$. A velocity field v_ψ is trained with

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1),t}\left[\|v_\psi(\mathbf{x}_t,t) - \mathbf{u}_t\|_2^2\right]. \tag{2}$$

When the transport depends on auxiliary information $\mathbf{c}$, conditional flow matching replaces the field with $v_\psi(\mathbf{x}_t, t; \mathbf{c})$. Thus, applying conditional flow matching requires specifying not only the source and target spaces, but also the conditioning information.

3.2 Unified Interface for GCD Learners

Despite their different prediction mechanisms, existing GCD learners expose a common set of quantities that can be connected to conditional flow matching. Given an image $\mathbf{x}_i$, a host GCD learner f_θ produces the normalized representation $\mathbf{z}_i = \frac{f_\theta(\mathbf{x}_i)}{\|f_\theta(\mathbf{x}_i)\|_2} \in \mathbb{S}^{d-1}$.

The host further maintains K normalized category prototypes $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]^\top \in \mathbb{R}^{K\times d}$, which may correspond to the weights of a prototype-based parametric classifier or category centers from clustering. Together with its native objective $\mathcal{L}_{\text{GCD}}$, these quantities define the host interface

$$\big(\mathbf{z}_i, \mathbf{P}, \mathcal{L}_{\text{GCD}}\big). \tag{3}$$

This abstraction covers the three different paradigms in GCD. In SimGCD (Wen et al., 2023), the normalized classifier weights form $\mathbf{P}$. In SelEx (Rastegar et al., 2024), $\mathbf{P}$ contains the finest-level centers obtained by semi-supervised K-means. GET (Wang et al., 2025) provides category-level visual prototypes derived from its vision–branch. Their internal losses and native prediction rules remain unchanged as part of the respective host learners. Under this interface, $\mathbf{z}_i$ provides a source representation and $\mathbf{P}$ defines the candidate target prototypes. However, for an unlabeled sample, the correct prototype is unknown. DynGCD therefore estimates a soft sample–prototype coupling and uses it to learn a prototype-conditioned flow jointly with the host representation.

4 DynGCD

Overview. DynGCD augments a host GCD learner with geometric supervision derived from sample-prototype transport. Given the image feature $\mathbf{z}_i$, category prototypes $\{\mathbf{p}_c\}_{c=1}^K$, and the original objective $\mathcal{L}_{\text{GCD}}$, we first formulate prototype-conditioned flow matching in §4.1, modeling each sample-prototype relation as a class-directed transport process. Building on this formulation, DynGCD proceeds in two steps (*cf.*, Fig. 2). First, in §4.2, we estimate a soft coupling between unlabeled samples and candidate prototypes, preserving multiple plausible correspondences rather than committing to a single pseudo-target. Second, in §4.3, we select reliable sample-prototype pairs according to their coupling confidence and use them to supervise the prototype-conditioned velocity field. The resulting flow objective is optimized jointly with $\mathcal{L}_{\text{GCD}}$, providing complementary geometric supervision to the host representation space.

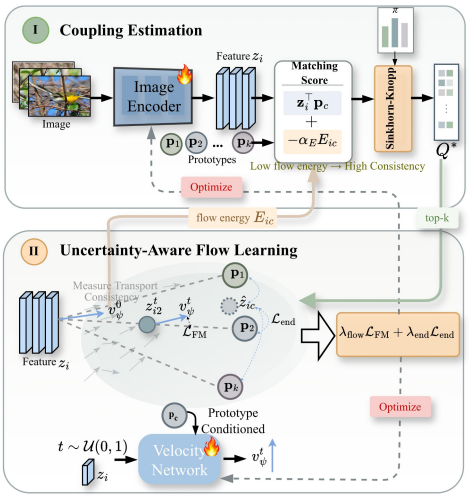


Figure 2: **Overview of DynGCD.**

4.1 Prototype-Conditioned Flow Matching

We first model the geometry of a sample-prototype relation through a conditional velocity field. Given an image feature $\mathbf{z}_i$ and a candidate prototype $\mathbf{p}_c$, we formulate their relation as a transport process from the sample representation to the prototype. Specifically, for sample i, candidate class c, and time $t \in [0, 1]$, following Lipman et al. (2023), we define:

$$\mathbf{z}_{ic}^t = (1-t)\mathbf{z}_i + t\mathbf{p}_c\,, \qquad \mathbf{u}_{ic} = \frac{\mathrm{d}\mathbf{z}_{ic}^t}{\mathrm{d}t} = \mathbf{p}_c - \mathbf{z}_i\,. \tag{4}$$

Here, $\mathbf{z}_{ic}^t$ is an intermediate state along the transport path, and $\mathbf{u}_{ic}$ is the target velocity toward $\mathbf{p}_c$. Let ψ denote the parameters of the velocity network. We learn a prototype-conditioned field

Algorithm 1 End-to-end Training Algorithm for DynGCD.

Input: Labeled set $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_l}$; unlabeled set $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=1}^{N_u}$; host GCD learner with parameters θ; velocity field v_ψ; epochs $E_{\max}$ and iterations $I_{\max}$.

for $e = 1$ **to** $E_{\max}$ **do**
  Extract unlabeled representations $\{\mathbf{z}_i\}_{i=1}^{N_u}$ and prototypes $\mathbf{P}$
  Compute E_{ic} and s_{ic} // Eqs. (5), (6)
  Compute $\boldsymbol{\pi}$ and Sinkhorn coupling $\mathbf{Q}^\star$ // Eqs. (7), (8)
  for iter $= 1$ **to** $I_{\max}$ **do**
    Sample labeled batch $\mathcal{B}_l$ and unlabeled batch $\mathcal{B}_u$
    Compute host objective $\mathcal{L}_{\text{GCD}}$
    Set $\mathbf{q}_i$ to the one-hot ground-truth distribution or cached $\mathbf{Q}_i^\star$
    Compute r_i, $\mathcal{T}_i$, and ω_{ic} // Eqs. (9), (10)
    Sample $t \sim \mathcal{U}(0,1)$ and construct $\mathbf{z}_{ic}^t$ // Eq. (4)
    Compute $\mathcal{L}_{\text{FM}}$ and $\mathcal{L}_{\text{end}}$ // Eqs. (11), (13)
    Compute joint objective $\mathcal{L}$ // Eq. (14)
    Update θ via $\nabla_\theta \mathcal{L}$ and ψ via $\nabla_\psi \mathcal{L}$
  end
end

Output: Optimized host parameters θ; discard v_ψ at inference.

$v_\psi(\mathbf{z}, t; \mathbf{p}_c) \in \mathbb{R}^d$, where $\mathbf{z} \in \mathbb{R}^d$ denotes a latent feature. Evaluated at $\mathbf{z}_{ic}^t$, the field predicts the transport direction associated with prototype $\mathbf{p}_c$. Prototype conditioning allows an unlabeled sample to retain multiple plausible sample-prototype relations, whose contributions are later determined by the estimated coupling. In this way, DynGCD represents sample-prototype relations as class-directed transport geometry that can be further exploited for representation learning.

4.2 Latent Sample-Prototype Coupling

Prototype conditioning allows each unlabeled sample to form transport relations with multiple candidate prototypes, but not all such relations are equally plausible. We therefore estimate a soft coupling over candidate prototypes to emphasize plausible correspondences while retaining assignment uncertainty. Periodically during training, we extract the current representations of all N_u unlabeled samples and evaluate each sample-prototype pair. In addition to prototype affinity, we use the learned velocity field to assess transport consistency: a plausible prototype should induce a transport direction that agrees with the learned field. We define the flow energy as

$$E_{ic} = \left\| v_\psi\left(\mathbf{z}_{ic}^0, 0; \mathbf{p}_c\right) - \mathbf{u}_{ic} \right\|_2^2 . \tag{5}$$

A smaller E_{ic} indicates that the learned velocity field is more consistent with the transport direction from $\mathbf{z}_i$ to $\mathbf{p}_c$. We combine the flow energy with prototype affinity into a compatibility score:

$$s_{ic} = \mathbf{z}_i^\top \mathbf{p}_c - \alpha_E E_{ic} , \tag{6}$$

where α_E controls the contribution of flow energy.

Normalizing s_{ic} independently for each sample may concentrate assignments on only a few prototypes. We therefore estimate a prototype-mass prior from the current representation space and interpolate it with a uniform prior to mitigate unreliable early estimates:

$$\boldsymbol{\pi} = (1-\rho)\frac{\mathbf{1}_K}{K} + \rho\bar{\boldsymbol{\pi}} , \qquad \bar{\pi}_c = \frac{1}{N_u}\sum_{i=1}^{N_u} \operatorname{softmax}\left(\mathbf{z}_i^\top \mathbf{P}^\top\right)_c , \tag{7}$$

where ρ controls the empirical prior strength.

Given the compatibility scores and the prototype-mass prior, we estimate a globally regularized soft coupling $\mathbf{Q} \in \mathbb{R}_+^{N_u \times K}$, where Q_{ic} represents the mass assigned from sample i to prototype c. Specifically, we solve the entropically regularized optimal transport problem (Cuturi, 2013):

$$\begin{aligned} \mathbf{Q}^\star = \ & \arg\max_{\mathbf{Q}\geq 0} \sum_{i=1}^{N_u}\sum_{c=1}^{K} Q_{ic} s_{ic} + \tau_{\text{sk}} \mathcal{H}(\mathbf{Q}) \\ \text{s.t.} \quad & \mathbf{Q}\mathbf{1}_K = \mathbf{1}_{N_u} , \qquad \mathbf{Q}^\top \mathbf{1}_{N_u} = N_u \boldsymbol{\pi} , \end{aligned} \tag{8}$$

where $\mathcal{H}(\mathbf{Q}) = -\sum_{i,c} Q_{ic} \log Q_{ic}$ and $\tau_{\text{sk}} > 0$ is the entropy regularization strength. The row constraint produces a distribution over candidate prototypes for each sample, while the column constraint regulates the aggregate prototype usage according to $\boldsymbol{\pi}$. We solve Eq. (8) with Sinkhorn–Knopp iterations, cache $\mathbf{Q}^\star$ by sample index, and update it periodically during training.

4.3 Uncertainty-Aware Flow Learning

The soft coupling above captures both plausible prototype correspondences and their uncertainty. Rather than collapsing it into a single pseudo-label, we directly use this uncertainty to determine the contribution of each sample-prototype relation to flow learning. For each training sample i, let $\mathbf{q}_i$ denote its coupling distribution. For labeled samples, $\mathbf{q}_i$ is the one-hot ground-truth distribution, and for unlabeled samples, it is the corresponding cached row of $\mathbf{Q}^\star$. We quantify sample-level certainty by the normalized entropy:

$$r_i = 1 - \frac{\mathcal{H}(\mathbf{q}_i)}{\log K}, \qquad \mathcal{H}(\mathbf{q}_i) = -\sum_{c=1}^{K} q_{ic} \log q_{ic}. \tag{9}$$

A concentrated coupling yields a larger r_i, while an ambiguous one receives a smaller weight. For labeled samples, the one-hot distribution naturally gives $r_i = 1$.

To preserve multiple plausible correspondences while avoiding supervision from unlikely ones, we restrict each coupling distribution to its top-k candidates. Let $\mathcal{T}_i$ denote their indices and define

$$\omega_{ic} = \frac{r_i \widetilde{q}_{ic}}{\sum_{\ell \in \mathcal{B}} \sum_{j \in \mathcal{T}_\ell} r_\ell \widetilde{q}_{\ell j}}, \qquad \widetilde{q}_{ic} = \frac{q_{ic}}{\sum_{j \in \mathcal{T}_i} q_{ij}}, \tag{10}$$

where $\mathcal{B}$ denotes the set of mini-batch samples. The resulting weight ω_{ic} jointly reflects the certainty of sample i and the relative plausibility of prototype c. Thus, uncertain samples contribute less overall, while multiple plausible prototypes can still provide supervision without being prematurely collapsed into a fixed pseudo-label.

For each retained sample-prototype pair (i, c), we independently sample $t \sim \mathcal{U}(0, 1)$ and optimize the uncertainty-weighted flow-matching objective:

$$\mathcal{L}_{\text{FM}} = \sum_{i \in \mathcal{B}} \sum_{c \in \mathcal{T}_i} \omega_{ic} \left\| v_\psi\big(\mathbf{z}_{ic}^t, t; \mathbf{p}_c\big) - \mathbf{u}_{ic} \right\|_2^2. \tag{11}$$

The prototypes are treated as stop-gradient targets, while the velocity network and representation encoder are optimized jointly. In this way, the velocity field provides additional geometric supervision to the representation space through reliable sample-prototype transport relations.

Beyond matching the transport velocity, we further constrain the predicted endpoint to be category-discriminative. From an intermediate state $\mathbf{z}_{ic}^t$, we obtain a one-step estimate of the destination,

$$\widehat{\mathbf{z}}_{ic} = \text{Norm}\big(\mathbf{z}_{ic}^t + (1 - t) v_\psi\big(\mathbf{z}_{ic}^t, t; \mathbf{p}_c\big)\big). \tag{12}$$

While velocity matching supervises the transport direction, it does not explicitly enforce the relative discriminability of the resulting endpoint. We therefore encourage $\widehat{\mathbf{z}}_{ic}$ to favor its candidate prototype over the others:

$$\mathcal{L}_{\text{end}} = -\sum_{i \in \mathcal{B}} \sum_{c \in \mathcal{T}_i} \omega_{ic} \log \frac{\exp\big(\widehat{\mathbf{z}}_{ic}^\top \mathbf{p}_c / \tau_e\big)}{\sum_{j=1}^{K} \exp\big(\widehat{\mathbf{z}}_{ic}^\top \mathbf{p}_j / \tau_e\big)}, \tag{13}$$

where τ_e is the endpoint-classification temperature. This objective complements velocity matching by encouraging the predicted transport endpoint to lie in a category-discriminative region.

Finally, DynGCD is optimized jointly with the host objective:

$$\mathcal{L} = \mathcal{L}_{\text{GCD}} + \lambda_{\text{flow}} \mathcal{L}_{\text{FM}} + \lambda_{\text{end}} \mathcal{L}_{\text{end}}, \tag{14}$$

where λ_{flow} and λ_{end} control the two auxiliary losses. During training, the cached coupling and current prototypes serve as fixed targets within each minibatch, while the representation encoder and velocity network are updated jointly. The updated representations and velocity field are then used in the next coupling update. At inference, the flow module is removed and the host learner keeps its original inference pipeline. The training procedure of DynGCD is summarized in Alg. 1.

Table 1: Results on fine-grained datasets (CUB (Wah et al., 2011), Stanford-Cars (Krause et al., 2013), FGVC-Aircraft (Maji et al., 2013)), reported for *All*/*Old*/*New* categories.

	Method	CUB			Stanford-Cars			FGVC-Aircraft			Average		
		All	Old	New	All	Old	New	All	Old	New	All	Old	New
DINO	GCD (Vaze et al., 2022a)[CVPR2022]	51.3	56.6	48.7	39.0	57.6	29.9	45.0	41.1	46.9	45.1	51.8	41.8
	PromptCAL (Zhang et al., 2023)[CVPR2023]	62.9	64.4	62.1	50.2	70.1	40.6	52.2	52.2	52.3	55.1	62.2	51.7
	GPC (Zhao et al., 2023)[ICCV2023]	52.0	55.5	47.5	38.2	58.9	27.4	43.3	40.7	44.8	44.5	51.7	39.9
	μGCD (Vaze et al., 2023)[NeurIPS2023]	65.7	68.0	64.6	56.5	68.1	50.9	53.8	55.4	53.0	58.7	63.8	56.2
	InfoSieve (Rastegar et al., 2023)[NeurIPS2023]	69.4	77.9	65.2	55.7	74.8	46.4	56.3	63.7	52.5	60.5	72.1	54.7
	SPTNet (Wang et al., 2024)[ICLR2024]	65.8	68.8	65.1	59.0	79.2	49.3	59.3	61.8	58.1	61.4	69.9	57.5
	FlipClass (Lin et al., 2024)[NeurIPS2024]	71.3	71.3	71.3	63.1	81.7	53.8	59.3	66.9	55.4	64.6	73.3	60.2
	DebGCD (Liu & Han, 2025)[ICLR2025]	66.3	71.8	63.5	65.3	81.6	57.4	61.7	63.9	60.6	64.4	72.4	60.5
	MOS (Peng et al., 2025)[CVPR2025]	69.6	72.3	68.2	64.6	80.9	56.7	61.1	66.9	58.2	65.1	73.4	61.0
	APL (Dai et al., 2025)[CVPR2025]	68.5	73.1	66.2	62.3	80.7	53.4	60.9	63.5	59.6	63.9	72.4	59.7
	RLCD (Liu et al., 2025a)[ICML2025]	70.0	79.1	65.4	64.9	79.3	58.0	60.6	62.2	59.8	65.2	73.5	61.1
	AllGCD (Cao et al., 2025)[ICCV2025]	68.4	75.1	65.1	60.5	77.0	52.5	57.4	62.4	54.9	62.1	71.5	57.5
	AF (Xu et al., 2025)[ICCV2025]	69.0	74.3	66.3	**67.0**	80.7	**60.4**	59.4	68.1	55.0	65.1	74.4	60.6
	SEAL (He et al., 2025c)[NeurIPS2025]	66.2	72.1	63.2	65.3	79.3	58.5	62.0	65.3	60.4	64.5	72.2	60.7
	†TAR (Yang et al., 2026)[CVPR2026]	70.6	74.8	68.5	65.7	**82.4**	57.5	62.0	**69.1**	58.7	66.1	75.4	61.6
	CPF-GCD (Dai et al., 2026)[ICML2026]	**79.8**	80.5	**79.4**	58.9	79.0	49.2	61.8	69.0	58.1	66.8	**76.2**	62.2
	†RCA (Yu et al., 2026)[ICML2026]	71.2	76.4	68.6	65.0	79.2	58.2	60.1	67.1	56.6	65.4	74.2	61.1
	SimGCD* (Wen et al., 2023)[ICCV2023]	60.9	66.5	58.1	51.6	67.8	43.8	52.0	57.8	49.0	54.8	64.0	50.3
	+ DynGCD	63.0	67.7	60.7	59.0	76.2	50.6	56.8	62.6	53.9	59.6	68.8	55.1
	SelEx* (Rastegar et al., 2024)[ECCV2024]	77.5	79.8	76.5	56.3	76.2	46.7	58.7	65.0	55.5	64.2	73.7	59.6
	+ DynGCD	79.5	**80.7**	78.9	58.0	74.3	50.1	**63.6**	66.8	**62.0**	**67.0**	73.9	**63.7**
	Avg. Δ	**+2.1**	**+1.0**	**+2.5**	**+4.6**	**+3.2**	**+5.1**	**+4.8**	**+3.3**	**+5.7**	**+3.8**	**+2.5**	**+4.4**
DINOv2	GCD (Vaze et al., 2022a)[CVPR2022]	71.9	71.2	72.3	65.7	67.8	64.7	55.4	47.9	59.2	64.3	62.3	65.4
	SPTNet (Wang et al., 2024)[ICLR2024]	76.3	79.5	74.6	72.3	82.0	65.7	68.0	75.2	60.8	72.2	78.9	67.0
	DebGCD (Liu & Han, 2025)[ICLR2025]	77.5	80.8	75.8	75.4	87.7	69.5	71.9	76.0	69.8	74.9	81.5	71.7
	APL (Dai et al., 2025)[CVPR2025]	75.1	79.1	73.2	73.4	87.6	66.7	68.8	74.1	66.6	72.4	80.3	68.8
	RLCD (Liu et al., 2025a)[ICML2025]	78.7	79.5	78.3	79.5	91.8	73.5	72.6	77.3	70.3	76.9	82.9	74.0
	AllGCD (Cao et al., 2025)[ICCV2025]	78.4	82.8	76.2	76.2	88.3	70.4	–	–	–	–	–	–
	SEAL (He et al., 2025c)[NeurIPS2025]	76.7	78.3	75.9	77.7	88.7	72.4	74.6	73.2	75.3	76.3	80.1	74.5
	CURE (Bian et al., 2026)[ICML2026]	81.3	80.4	81.7	79.9	88.1	76.0	–	–	–	–	–	–
	†FourEx (Rastegar et al., 2026)[ECCV2026]	**87.8**	**86.3**	88.6	80.1	**94.3**	73.2	81.5	**84.7**	80.0	83.1	**88.4**	80.6
	SimGCD* (Wen et al., 2023)[ICCV2023]	73.5	77.1	71.7	70.9	84.6	64.3	67.0	65.0	68.0	70.5	75.6	68.0
	+ DynGCD	76.3	80.5	74.1	80.0	89.7	75.3	69.6	67.4	70.7	75.3	79.2	73.4
	SelEx* (Rastegar et al., 2024)[ECCV2024]	86.8	84.7	87.8	**81.7**	92.1	76.7	82.0	83.0	81.5	83.5	86.6	82.0
	+ DynGCD	87.0	82.5	**89.3**	81.5	91.0	**77.0**	**82.5**	81.7	**82.9**	**83.7**	85.1	**83.1**
	Avg. Δ	**+1.5**	**+0.6**	**+2.0**	**+4.4**	**+2.0**	**+5.7**	**+1.6**	**+0.6**	**+2.0**	**+2.5**	**+1.0**	**+3.3**
CLIP	GCD (Vaze et al., 2022a)[CVPR2022]	57.6	65.2	53.8	65.1	75.9	59.8	45.3	44.4	45.8	56.0	61.8	53.1
	SimGCD (Wen et al., 2023)[ICCV2023]	71.7	76.5	69.4	70.0	83.4	63.5	54.3	58.4	52.2	65.3	72.8	61.7
	CLIP-GCD (Ouldnoughi et al., 2023)[arXiv2023]	62.8	77.1	55.7	70.6	88.2	62.2	50.0	56.6	46.5	61.1	74.0	54.8
	TextGCD‡ (Zheng et al., 2024)[ECCV2024]	76.6	**80.6**	74.7	**86.9**	87.4	**86.7**	–	–	–	–	–	–
	CPT (Yang et al., 2025)[IJCV2025]	70.1	73.5	68.4	74.2	84.3	69.3	–	–	–	–	–	–
	†TAR (Yang et al., 2026)[CVPR2026]	**78.4**	80.2	**77.5**	80.4	**91.1**	75.2	60.3	**64.8**	58.5	73.0	**78.7**	70.4
	GET* (Wang et al., 2025)[CVPR2025]	74.6	75.7	74.0	79.1	86.3	74.7	59.8	61.7	58.8	71.2	74.6	69.2
	+ DynGCD	77.0	78.7	76.2	82.0	90.5	77.8	**62.8**	64.5	**61.9**	**73.9**	77.9	**72.0**
	Δ	**+2.4**	**+3.0**	**+2.2**	**+2.9**	**+4.2**	**+3.1**	**+3.0**	**+2.8**	**+3.1**	**+2.7**	**+3.3**	**+2.8**

* denotes results reproduced by us. † denotes results based on the best baseline. ‡ TextGCD additionally uses an auxiliary CLIP ViT-H and GPT-3-derived textual attributes.

5 Experiments

5.1 Setups and Implementations

Datasets. We evaluate our method on both generic and fine-grained benchmarks, including CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), ImageNet-100 (Deng et al., 2009), and the Semantic Shift Benchmark (SSB) (Vaze et al., 2022b), which contains CUB (Wah et al., 2011), Stanford-Cars (Krause et al., 2013), and FGVC-Aircraft (Maji et al., 2013).

Evaluation Metrics. Following prior GCD work (Vaze et al., 2022a), we use clustering accuracy (*ACC*) as the primary evaluation metric. Given the ground-truth labels y_i and predicted cluster assignments $\hat{y}_i$ for samples in $\mathcal{D}_u$, *ACC* is computed as $\frac{1}{|\mathcal{D}_u|}\sum_{i=1}^{|\mathcal{D}_u|}\mathbb{1}\big(y_i = h(\hat{y}_i)\big)$ where h denotes the optimal permutation that matches predicted clusters to ground-truth categories. We report *ACC* separately for all classes (*All*), known classes (*Old*), and novel classes (*New*).

Implementation Details. We instantiate DynGCD on SimGCD (Wen et al., 2023), SelEx (Rastegar et al., 2024), and GET (Wang et al., 2025), representing parametric, non-parametric, and vision–

Table 2: Results on generic datasets (CIFAR-10/100 (Krizhevsky, 2009) and ImageNet-100 (Deng et al., 2009)), reported for *All*/*Old*/*New* categories.

	Method	CIFAR-10			CIFAR-100			ImageNet-100			Average		
		All	Old	New	All	Old	New	All	Old	New	All	Old	New
DINO	GCD (Vaze et al., 2022a)[CVPR2022]	91.5	97.9	88.2	73.0	76.2	66.5	74.1	89.8	66.3	79.5	88.0	73.7
	PromptCAL (Zhang et al., 2023)[CVPR2023]	**97.9**	96.6	98.5	81.2	84.2	75.3	83.1	92.7	78.3	87.4	91.2	84.0
	GPC (Zhao et al., 2023)[ICCV2023]	90.6	97.6	87.0	75.4	84.6	60.1	75.3	93.4	66.7	80.4	**91.9**	71.3
	InfoSieve (Rastegar et al., 2023)[NeurIPS2023]	94.8	97.7	93.4	78.3	82.2	70.5	80.5	93.8	73.8	84.5	91.2	79.2
	SPTNet (Wang et al., 2024)[ICLR2024]	97.3	95.0	98.6	81.3	84.3	75.6	85.4	93.2	81.4	88.0	90.8	85.2
	DebGCD (Liu & Han, 2025)[ICLR2025]	97.2	94.8	98.4	83.0	84.6	79.9	85.9	94.3	81.6	88.7	91.2	86.6
	RLCD (Liu et al., 2025a)[ICML2025]	97.4	96.4	97.9	**83.4**	84.2	81.9	**86.9**	94.2	**83.2**	**89.2**	91.6	**87.7**
	AllGCD (Cao et al., 2025)[ICCV2025]	-	-	-	82.3	82.3	82.2	86.5	**94.7**	82.3	-	-	-
	AF (Xu et al., 2025)[ICCV2025]	97.8	95.9	**98.8**	82.2	**85.0**	76.5	85.4	94.6	80.8	88.5	91.8	85.4
	SEAL (He et al., 2025c)[NeurIPS2025]	97.2	94.7	98.4	82.1	81.7	**83.0**	84.6	90.9	81.3	88.0	89.1	87.6
	SimGCD* (Wen et al., 2023)[ICCV2023]	97.1	95.1	98.1	80.0	80.7	78.7	83.5	93.9	78.3	86.9	89.9	85.0
	+ DynGCD	97.2	94.8	98.3	80.6	81.1	79.5	85.8	92.4	82.5	87.9	89.4	86.8
	SelEx* (Rastegar et al., 2024)[ECCV2024]	95.9	**98.1**	94.8	80.8	83.2	75.9	84.6	93.8	80.0	87.1	91.7	83.6
	+ DynGCD	96.5	97.5	95.5	80.9	83.3	76.2	85.3	94.3	80.8	87.6	91.7	84.2
	Avg. Δ	**+0.4**	**-0.4**	**+0.4**	**+0.4**	**+0.2**	**+0.6**	**+1.5**	**-0.5**	**+2.5**	**+0.7**	**-0.2**	**+1.2**
DINOv2	GCD (Vaze et al., 2022a)[CVPR2022]	97.8	99.0	97.1	79.6	84.5	69.9	78.5	89.5	73.0	85.3	91.0	80.0
	SPTNet (Wang et al., 2024)[ICLR2024]	98.9	99.1	98.8	89.0	91.5	79.2	90.1	96.1	87.1	92.7	95.6	88.4
	DebGCD (Liu & Han, 2025)[ICLR2025]	98.9	97.5	99.6	90.1	90.9	88.6	**93.2**	**97.0**	**91.2**	**94.1**	95.1	93.1
	RLCD (Liu et al., 2025a)[ICML2025]	**99.0**	98.9	99.1	**91.2**	91.2	**91.2**	92.1	96.2	90.0	**94.1**	95.4	93.4
	SEAL (He et al., 2025c)[NeurIPS2025]	98.9	98.1	99.3	89.8	90.4	89.5	91.3	93.3	90.3	93.3	93.9	93.0
	CURE (Bian et al., 2026)[ICML2026]	98.8	98.8	98.8	89.1	90.1	87.1	91.2	96.7	88.4	93.0	95.2	91.4
	SimGCD* (Wen et al., 2023)[ICCV2023]	98.5	96.1	**99.7**	88.0	87.9	88.0	90.2	96.6	87.0	92.2	93.5	91.6
	+ DynGCD	98.8	97.1	**99.7**	89.6	89.0	90.9	92.2	95.6	90.5	93.5	93.9	**93.7**
	SelEx* (Rastegar et al., 2024)[ECCV2024]	98.5	98.7	98.4	88.6	91.2	83.4	90.8	96.3	85.2	92.6	95.4	89.0
	+ DynGCD	98.7	**99.2**	98.5	90.7	**92.0**	88.1	92.8	96.9	90.8	**94.1**	**96.0**	92.5
	Avg. Δ	**+0.2**	**+0.8**	**+0.0**	**+1.8**	**+1.0**	**+3.8**	**+2.0**	**-0.2**	**+4.6**	**+1.4**	**+0.5**	**+2.8**
CLIP	GCD (Vaze et al., 2022a)[CVPR2022]	94.0	97.3	92.3	74.8	79.8	64.6	75.8	87.3	70.0	81.5	88.1	75.6
	SimGCD (Wen et al., 2023)[ICCV2023]	97.0	94.2	98.4	81.1	85.0	73.3	90.8	95.5	88.5	89.6	91.6	86.7
	CLIP-GCD (Ouldnoughi et al., 2023)[arXiv2023]	96.6	97.2	96.4	85.2	85.0	**85.6**	84.0	95.5	78.2	88.6	**92.6**	86.7
	TextGCD‡ (Zheng et al., 2024)[ECCV2024]	**98.2**	**98.0**	98.6	**85.7**	86.3	84.6	88.0	92.4	85.2	**90.6**	92.2	**89.5**
	CPT (Yang et al., 2025)[IJCV2025]	96.3	96.2	96.4	81.3	81.3	81.2	89.2	95.2	86.1	88.9	90.9	87.9
	GET* (Wang et al., 2025)[CVPR2025]	97.2	94.7	98.4	80.9	85.4	71.8	90.2	95.5	87.5	89.4	91.9	85.9
	+ DynGCD	97.5	95.0	**98.8**	82.2	**86.5**	73.4	**92.0**	**96.1**	**90.0**	**90.6**	92.5	87.4
	Δ	**+0.3**	**+0.3**	**+0.4**	**+1.3**	**+1.1**	**+1.6**	**+1.8**	**+0.6**	**+2.5**	**+1.1**	**+0.7**	**+1.5**

* denotes results reproduced by us. ‡ TextGCD additionally uses an auxiliary CLIP ViT-H and GPT-3-derived textual attributes.

language GCD paradigms, respectively. We evaluate DynGCD across three backbone families, including DINO (Caron et al., 2021), DINOv2 (Oquab et al., 2023), and CLIP (Radford et al., 2021). For each host learner, we retain its original training configuration, including the training schedule, optimizer, and other hyperparameters. Additional implementation details and DynGCD-specific hyperparameters are provided in the Appendix A1.

5.2 Quantitative Comparison

Tabs. 1 and 2 summarize the results on three generic datasets (CIFAR-10, CIFAR-100, and ImageNet-100) and three fine-grained datasets (CUB, Stanford-Cars, and FGVC-Aircraft) using DINO (Caron et al., 2021), DINOv2 (Oquab et al., 2023) and CLIP (Radford et al., 2021).

Fine-Grained Datasets. DynGCD consistently improves the host learners across different backbones and GCD paradigms on fine-grained benchmarks. With DINO, DynGCD improves the average *All*/*Old*/*New* accuracy of SimGCD (Wen et al., 2023) and SelEx (Rastegar et al., 2024) by $3.8/2.5/4.4$ points, while the corresponding gains with DINOv2 are $2.5/1.0/3.3$ points. The improvements are thus more pronounced on *New* categories, especially on challenging fine-grained datasets: the average New accuracy gains reach 5.1 and 5.7 points on SCars and Aircraft with DINO, and 5.7 points on SCars with DINOv2. DynGCD also improves the vision–language-based GET (Wang et al., 2025) by 2.7 points on average, demonstrating that the proposed supervision complements language-enhanced category prediction. Overall, the greater improvements on new categories are consistent with our motivation: beyond category assignments, modeling sample-prototype dynamics provides additional geometric cues for unlabeled samples with uncertain relations.

Generic Datasets. DynGCD also yields consistent improvements on coarse-grained benchmarks, showing that its effectiveness is not limited to fine-grained recognition. Averaged over SimGCD and SelEx, DynGCD improves the *All* accuracy by 0.7 points with DINO and 1.4 points with DINOv2,

Table 3: Ablation study of Conditional Flow Matching, Endpoint Loss, and Latent Coupling using SimGCD (Wen et al., 2023) with DINO (Caron et al., 2021) pretrained weights on the SSB (Vaze et al., 2022b) benchmark. The baseline corresponds to our reproduced SimGCD results.

Variant	Conditional Flow Matching	Endpoint Loss	Latent Coupling	CUB All	CUB Old	CUB New	Stanford-Cars All	Stanford-Cars Old	Stanford-Cars New	FGVC-Aircraft All	FGVC-Aircraft Old	FGVC-Aircraft New
Baseline	✗	✗	✗	60.9	66.5	58.1	51.6	67.8	43.8	52.0	57.8	49.0
(1)	✓	✗	✗	61.3	**67.9**	58.0	57.0	75.9	47.9	53.3	**62.9**	48.5
(2)	✓	✓	✗	61.5	67.4	58.6	57.2	74.7	48.8	53.6	59.9	52.4
(3)	✓	✗	✓	60.8	66.6	57.9	54.0	74.4	44.7	53.8	58.6	51.4
DynGCD	✓	✓	✓	**63.0**	67.7	**60.7**	**59.0**	**76.2**	**50.6**	**56.8**	62.6	**53.9**

with larger gains of 1.2 and 2.8 points on *New* categories, respectively. The benefit is particularly evident on ImageNet-100, where the average New-category accuracy increases by 2.5 points with DINO and 4.6 points with DINOv2. The same trend is observed for the vision–language-based GET, with gains of 1.1 points overall and 1.5 points on novel categories. Meanwhile, *Old* accuracy is largely maintained, with only minor fluctuations under some settings, suggesting that the improvements are primarily associated with better discovery of unlabeled categories rather than stronger fitting to labeled ones. Together with the fine-grained results, these consistent gains across datasets, backbones, and prediction paradigms support the generality of learning sample-prototype dynamics as complementary supervision for GCD.

5.3 Diagnostic Study

Component Analysis. We ablate Conditional Flow Matching (CFM), Endpoint Loss, and Latent Coupling on the SSB benchmark in Tab. 3. **First**, CFM alone consistently improves the baseline, validating our central motivation that sample-prototype relations provide useful supervision beyond category assignments through their intermediate transport dynamics. **Second**, adding the Endpoint Loss generally brings further improvements, especially on *New* categories, showing that encouraging discriminative transport endpoints complements the learned geometric dynamics. **Third**, adding Latent Coupling to CFM alone yields mixed changes, while its benefit becomes evident when combined with the other components, suggesting that uncertainty-aware sample-prototype coupling is most effective when supported by well-formed transport supervision. **Finally**, the full model achieves the best overall accuracy and the highest *New* accuracy throughout, demonstrating the complementarity of geometric transport, endpoint discrimination, and uncertain coupling.

Flow Analysis. To verify that the performance gains do not simply arise from an additional trainable network or improved sample-prototype coupling, we introduce two controls based on SimGCD on SSB Benchmarks. The capacity-matched regressor uses the same architecture as the flow network, but removes time conditioning and directly predicts sample-to-prototype displacements. The assignment-only variant uses the flow-derived coupling as soft pseudo-labels while preventing the flow losses from updating the backbone. DynGCD consistently outperforms both controls, demonstrating the benefit of learning dynamic sample-prototype transport as representation supervision. Detailed results are provided in Appendix A2.

Table 4: Flow analysis.

Method	Avg. All	Avg. Old	Avg. New
SimGCD	54.8	64.0	50.3
+Regressor	55.8	67.0	50.4
+Assignment	48.9	63.5	41.8
+DynGCD	**59.6**	**68.8**	**55.1**

6 Conclusion

We present **DynGCD**, a plug-and-play framework that goes beyond assignment-based supervision by modeling sample-prototype relations as continuous transport. DynGCD uses prototype-conditioned flow matching to exploit intermediate geometric structure, while maintaining soft couplings over multiple plausible categories to account for uncertain supervision. The resulting coupling–flow interaction enables mutual refinement between sample–category relations and transport dynamics throughout training. DynGCD can be readily integrated into parametric, non-parametric, and vision–language GCD methods while preserving their native inference procedures. Extensive experiments on generic and fine-grained benchmarks demonstrate consistent improvements across all three paradigms, with particularly strong gains on novel categories.

AI Use Statement. Generative AI tools were used to polish the manuscript and improve the clarity and readability of the writing. Their use was strictly limited to linguistic refinement and did not contribute to research ideation, data collection or generation, mathematical derivations or proofs, experimental design or implementation, data analysis, or the formulation of scientific claims and conclusions. All AI-assisted edits were carefully reviewed and, where necessary, revised by the authors to ensure that they faithfully preserved the intended scientific content. The authors take full responsibility for the accuracy, integrity, and entirety of the final manuscript, including all methods, results, analyses, and conclusions.

REFERENCES

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.

Yuwei Bian, Shidong Wang, and Haofeng Zhang. CURE: Consistency-under-unified semantic regularization for generalized category discovery. In *ICML*, 2026.

Lea Bogensperger, Dominik Narnhofer, Alexander Falk, Konrad Schindler, and Thomas Pock. FlowSDF: Flow matching for medical image segmentation using distance transforms. *IJCV*, 2025.

Xinzi Cao, Ke Chen, Feidiao Yang, Xiawu Zheng, Yonghong Tian, and Yutong Lu. AllGCD: Leveraging all unlabeled data for generalized category discovery. In *ICCV*, 2025.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Hongxu Chen, Yanghao Wang, Bowei Zhu, Hongxiang Li, Zhen Wang, Ziqi Jiang, Lin Li, Rui Liu, and Long Chen. Direct product flow matching: Decoupling radial and angular dynamics for few-shot adaptation. *arXiv preprint arXiv:2605.05054*, 2026.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

Boyang Dai, Chaoqi Chen, and Yizhou Yu. Identifying latent concepts and structures for generalized category discovery. In *ICML*, 2026.

Qiyuan Dai, Hanzhuo Huang, Yu Wu, and Sibei Yang. Adaptive part learning for fine-grained generalized category discovery: A plug-and-play enhancement. In *CVPR*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. DepthFM: Fast generative monocular depth estimation with flow matching. In *AAAI*, 2025.

Ju He, Qihang Yu, Qihao Liu, and Liang-Chieh Chen. FlowTok: Flowing seamlessly across text and image tokens. In *ICCV*, 2025a.

Zhenqi He, Yuanpei Liu, and Kai Han. Category discovery: An open-world perspective. *arXiv preprint arXiv:2509.22542*, 2025b.

Zhenqi He, Yuanpei Liu, and Kai Han. SEAL: Semantic-aware hierarchical learning for generalized category discovery. In *NeurIPS*, 2025c.

Zhenqi He, Ziqi Jiang, Yuanpei Liu, Yanghao Wang, Teng Wang, and Long Chen. FlowCIR: Semantic transport via flow matching for zero-shot composed image retrieval. In *ECCV*, 2026a.

Zhenqi He, Lin Li, and Long Chen. FlowComposer: Composable flows for compositional zero-shot learning. In *CVPR*, 2026b.

Ziqi Jiang, Yanghao Wang, and Long Chen. Exploring cross-modal flows for few-shot learning. In *ICLR*, 2026.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lin Li, Ziqi Jiang, Gefan Ye, Zhenqi He, Jiahui Li, Jun Xiao, Kwang-Ting Cheng, and Long Chen. Path-decoupled hyperbolic flow matching for few-shot adaptation. In *ICML*, 2026.

Haonan Lin, Wenbin An, Jiahao Wang, Yan Chen, Feng Tian, Mengmeng Wang, Guang Dai, Qianying Wang, and Jingdong Wang. Flipped classroom: Aligning teacher attention with student in generalized category discovery. In *NeurIPS*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Duo Liu, Zhiquan Tan, Linglan Zhao, Zhongqiang Zhang, Xiangzhong Fang, and Weiran Huang. Generalized category discovery via reciprocal learning and class-wise distribution regularization. In *ICML*, 2025a.

Qihao Liu, Xi Yin, Alan Yuille, Andrew Brown, and Mannat Singh. Flowing from words to pixels: A noise-free framework for cross-modality evolution. In *CVPR*, 2025b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Yuanpei Liu and Kai Han. DebGCD: Debiased learning with distribution guidance for generalized category discovery. In *ICLR*, 2025.

Yuanpei Liu, Zhenqi He, and Kai Han. Hyperbolic category discovery. In *CVPR*, 2025c.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Rabah Ouldnoughi, Chia-Wen Kuo, and Zsolt Kira. CLIP-GCD: Simple language guided generalized category discovery. *arXiv preprint arXiv:2305.10420*, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

Zhengyuan Peng, Jinpeng Ma, Zhimin Sun, Ran Yi, Haichuan Song, Xin Tan, and Lizhuang Ma. MOS: Modeling object-scene associations in generalized category discovery. In *CVPR*, 2025.

Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *CVPR*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Sarah Rastegar, Hazel Doughty, and Cees G. M. Snoek. Learn to categorize or categorize to learn? self-coding for generalized category discovery. In *NeurIPS*, 2023.

Sarah Rastegar, Mohammadreza Salehi, Yuki M. Asano, Hazel Doughty, and Cees G. M. Snoek. SelEx: Self-expertise in fine-grained generalized category discovery. In *ECCV*, 2024.

Sarah Rastegar, Mina Ghadimi Atigh, Pascal Mettes, Yuki M. Asano, and Cees G. M. Snoek. Fourier self-supervision for fine-grained generalized category discovery. In *ECCV*, 2026.

Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

Luyao Tang, Kunze Huang, Chaoqi Chen, Yuxuan Yuan, Chenxin Li, Xiaotong Tu, Xinghao Ding, and Yue Huang. Dissecting generalized category discovery: Multiplex consensus under self-deconstruction. In *ICCV*, 2025a.

Luyao Tang, Yuxuan Yuan, Chaoqi Chen, Zeyu Zhang, Yue Huang, and Kun Zhang. Ocrt: Boosting foundation models in the open world with object-concept-relation triad. In *CVPR*, 2025b.

Luyao Tang, Kunze Huang, Chaoqi Chen, and Cheng Chen. Bures-isotropy alignment: Manifold learning of generalized category discovery. In *ICLR*, 2026a.

Luyao Tang, Yingkai Yang, Hanqi Chen, Jiewei Zheng, Chaoqi Chen, and Cheng Chen. Generalized biomedicine discovery. In *ECCV*, 2026b.

Luyao Tang, Jiewei Zheng, Kunze Huang, Chaoqi Chen, Yue Huang, and Cheng Chen. CoGe-GCD: Reframing generalized category discovery with compositional generalization. In *ICML*, 2026c.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022a.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. The semantic shift benchmark. In *International Conference on Machine Learning Workshop*, 2022b.

Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. No representation rules them all in category discovery. In *NeurIPS*, 2023.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Enguang Wang, Zhimao Peng, Zhengyuan Xie, Fei Yang, Xialei Liu, and Ming-Ming Cheng. GET: Unlocking the multi-modal potential of CLIP for generalized category discovery. In *CVPR*, 2025.

Hongjun Wang, Sagar Vaze, and Kai Han. SPTNet: An efficient alternative framework for generalized category discovery with spatial prompt tuning. In *ICLR*, 2024.

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *ICCV*, 2023.

Qiyu Xu, Zhanxuan Hu, Yu Duan, Ercheng Pei, and Yonghang Tai. A hidden stumbling block in generalized category discovery: Distracted attention. In *ICCV*, 2025.

Muli Yang, Jie Yin, Yanan Gu, Cheng Deng, Hanwang Zhang, and Hongyuan Zhu. Consistent prompt tuning for generalized category discovery. *IJCV*, 2025.

Xingyu Yang, Yu Zhang, Siya Mi, and Xiu-Shen Wei. TAR: Token-aware refinement for fine-grained generalized category discovery. In *CVPR*, 2026.

Jiawei Yu, Zijian Gao, Tianjiao Wan, Xuan Liu, Cheng Yang, and Kele Xu. Reliable confidence alignment for generalized category discovery. In *ICML*, 2026.

Baoqiang Zhang, Kunze Huang, Luyao Tang, and Xiaotong Tu. HVGCD: Rethinking generalized category discovery through hypothesis–verification. In *ECCV*, 2026.

Sheng Zhang, Salman H. Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Shahbaz Khan. PromptCAL: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *CVPR*, 2023.

Bingchen Zhao, Xin Wen, and Kai Han. Learning semi-supervised gaussian mixture models for generalized category discovery. In *ICCV*, 2023.

Haiyang Zheng, Nan Pu, Wenjing Li, Nicu Sebe, and Zhun Zhong. Textual knowledge matters: Cross-modality co-teaching for generalized visual class discovery. In *ECCV*, 2024.

APPENDIX

A1 Reproducibility and Implementation Details **14**

A1.1 Dataset Statistics 14

A1.2 Evaluation Metrics 14

A1.3 Additional Implementation Details 14

A2 More Experimental Results **15**

A2.1 Additional Quantitative Results 15

A2.2 Robustness to Unknown Category Number 15

A2.3 Additional Ablation Studies 15

A2.4 Analysis on Flow Energy 17

A2.5 Stability analysis 17

A3 Computational Analysis **18**

A3.1 Training Time Comparison 18

A3.2 Flow Model Size Analysis 18

A1 Reproducibility and Implementation Details

A1.1 Dataset Statistics

For each dataset, we adhere to the data splitting scheme described in (Vaze et al., 2022a). In this scheme, 50% of the classes are sampled as *Old*, with the exception of CIFAR-100, which samples 80% of the classes. Following this, 50% of the images from known classes are used to create the labeled dataset $\mathbf{D}_l$, while the remaining images are allocated to the unlabeled dataset $\mathbf{D}_u$. The statistics for all the datasets utilized in this work are summarized in Tab. A1.

Table A1: Overview of the datasets used in the evaluation, including the classes in the labeled and unlabeled sets ($M = |\mathbf{Y}_l|$, $K = |\mathbf{Y}_l \cup \mathbf{Y}_u|$) and counts of images ($|\mathbf{D}_l|$, $|\mathbf{D}_u|$). 'FG' denotes fine-grained.

Dataset	FG	$\lvert\mathbf{D}_l\rvert$	M	$\lvert\mathbf{D}_u\rvert$	K
CIFAR-10 (Krizhevsky, 2009)	✗	12.5K	5	37.5K	10
CIFAR-100 (Krizhevsky, 2009)	✗	20.0K	80	30.0K	100
ImageNet-100 (Deng et al., 2009)	✗	31.9K	50	95.3K	100
CUB (Wah et al., 2011)	✓	1.5K	100	4.5K	200
Stanford-Cars (Krause et al., 2013)	✓	2.0K	98	6.1K	196
FGVC-Aircraft (Maji et al., 2013)	✓	1.7K	50	5.0K	100
Herbarium19 (Tan et al., 2019)	✓	8.9K	341	25.4K	683
Oxford-Pet (Parkhi et al., 2012)	✓	0.9K	19	2.7K	37

A1.2 Evaluation Metrics

We evaluate performance using clustering accuracy (*ACC*), following the standard GCD protocol (Vaze et al., 2022a). Given ground-truth labels y_i and predicted cluster assignments $\hat{y}_i$ for samples in the unlabeled set $\mathbf{D}_u$, *ACC* is defined as $ACC = \frac{1}{|\mathbf{D}_u|}\sum_{i=1}^{|\mathbf{D}_u|} \mathbb{1}\big(y_i = h(\hat{y}_i)\big)$, where h is the optimal one-to-one mapping (permutation) between predicted clusters and ground-truth categories, obtained via the Hungarian algorithm. Following prior work, we report *ACC* over all unlabeled samples (*All*), and also separately on the subsets belonging to known classes (*Old*) and novel classes (*New*).

A1.3 Additional Implementation Details

Unless otherwise specified, we retain the top $k = 5$ candidate prototypes for each unlabeled sample and set the flow-energy coefficient to $\alpha_E = 0.25$. For latent coupling, we use a prior interpolation coefficient of $\rho = 0.5$. We set the Sinkhorn temperature to $\tau_{\text{sk}} = 0.05$, perform 10 Sinkhorn–Knopp iterations, and update the coupling cache once per epoch. For uncertainty-aware flow learning, the endpoint-classification temperature is set to $\tau_e = 0.1$. The loss weights λ_{flow} and λ_{end} are selected according to validation-set performance. The prototypes and cached coupling are treated as stop-gradient targets, whereas the sample representation is not detached when constructing either the intermediate state or the target velocity. Consequently, $\mathcal{L}_{\text{FM}}$ updates the representation encoder through both z_{ic}^t and $u_{ic} = p_c - z_i$, while jointly optimizing the velocity network. All remaining data augmentations, optimization settings, and baseline-specific hyperparameters follow those of the corresponding host GCD learner. All experiments are conducted on a single NVIDIA RTX3090 24GB GPU. For the flow matching network, we implement it as a residual MLP following Jiang et al. (2026); He et al. (2026a). For d-dimensional features, we concatenate the intermediate state $\mathbf{z}_{ic}^t$ with its candidate prototype $\mathbf{p}_c$, project the resulting $2d$-dimensional vector to width d, and pass it through 12 residual blocks. A sinusoidal time embedding modulates each block through adaptive layer normalization and residual gating; a final conditioned projection predicts a d-dimensional velocity. The residual modulation layers and final output projection are zero-initialized.

Table A2: Results on the additional fine-grained datasets Oxford-Pet (Parkhi et al., 2012) and Herbarium19 (Tan et al., 2019), reported for *All*/*Old*/*New* categories using DINO (Caron et al., 2021) pretrained weights. * denotes results reproduced by us.

Method	Oxford-Pet			Herbarium19		
	All	Old	New	All	Old	New
GCD (Vaze et al., 2022a)[CVPR2022]	80.2	85.1	77.6	35.4	51.0	27.0
DCCL (Pu et al., 2023)[CVPR2023]	88.1	88.2	88.0	–	–	–
μGCD (Vaze et al., 2023)[NeurIPS2023]	–	–	–	45.8	61.9	37.2
InfoSieve (Rastegar et al., 2023)[NeurIPS2023]	90.7	95.2	88.4	40.3	59.0	30.2
DebGCD (Liu & Han, 2025)[ICLR2025]	93.0	86.4	96.5	44.7	59.4	36.8
HypCD (Liu et al., 2025c)[CVPR2025]	92.7	91.5	93.3	40.5	49.0	36.0
SEAL (He et al., 2025c)[NeurIPS2025]	92.9	88.9	95.0	46.9	45.8	48.2
SimGCD* (Wen et al., 2023)[ICCV2023]	91.9	90.1	92.9	44.2	57.6	37.0
+ DynGCD	92.5	90.8	93.4	45.8	58.5	39.0
SelEx* (Rastegar et al., 2024)[ECCV2024]	92.5	93.1	92.2	39.8	55.1	31.6
+ DynGCD	93.2	93.4	93.2	43.0	56.0	36.0
Avg. Δ	**+0.7**	**+0.5**	**+0.8**	**+2.4**	**+0.9**	**+3.2**

A2 More Experimental Results

A2.1 Additional Quantitative Results

To further evaluate the generality of DynGCD, we conduct experiments on two additional fine-grained datasets: Oxford-Pet (Parkhi et al., 2012) and Herbarium19 (Tan et al., 2019). Oxford-Pet contains diverse cat and dog breeds with subtle inter-class differences, whereas Herbarium19 is a long-tailed botanical dataset characterized by fine-grained categories and substantial intra-class variation. Tab. A2 summarizes the results obtained by integrating DynGCD with SimGCD (Wen et al., 2023) and SelEx (Rastegar et al., 2024) using DINO (Caron et al., 2021) pretrained weights.

Table A3: Results with estimated category numbers on Stanford-Cars (Krause et al., 2013) and CIFAR-100 (Krizhevsky, 2009). All methods use DINO (Caron et al., 2021) pretrained weights and the category-number estimates from GCD (Vaze et al., 2022a).

Method	Stanford-Cars			CIFAR-100		
	All	Old	New	All	Old	New
GCD (Vaze et al., 2022a)[CVPR2022]	35.0	56.0	24.8	73.0	76.2	66.5
μGCD (Vaze et al., 2023)[NeurIPS2023]	56.3	66.8	51.1	–	–	–
SimGCD (Wen et al., 2023)[ICCV2023]	48.2	64.5	40.3	80.2	81.4	77.8
+ DynGCD	56.6	72.1	49.1	81.1	82.0	79.3

A2.2 Robustness to Unknown Category Number

Following standard GCD evaluation protocols (Vaze et al., 2022a; Wen et al., 2023; Wang et al., 2024; Rastegar et al., 2024; Liu & Han, 2025; Liu et al., 2025c; He et al., 2025c), our main experiments assume that the total number of categories K is known. We further evaluate the robustness of DynGCD when K is unknown and must be estimated. Following GCD (Vaze et al., 2022a), we adopt its category-number estimates for Stanford-Cars and CIFAR-100, whose ground-truth category numbers are 196 and 100, respectively. Tab. A3 reports the results under these estimated category numbers. In this setting, DynGCD uses the same category-number estimates as its host learner without modifying either the estimation procedure or the host model's inference rule.

A2.3 Additional Ablation Studies

Hyperparameter Sensitivity. Following common practice (Liu & Han, 2025; Wen et al., 2023), we perform hyperparameter selection on a held-out validation split. Specifically, we examine the sensitivity to λ_{flow}, λ_{end}, and α_E on FGVC-Aircraft (Maji et al., 2013) and Stanford-Cars (Krause et al., 2013), as shown in Tab. A4. For each hyperparameter, the final value is selected solely according to the *Old* accuracy on the held-out validation set, without using the information of unseen categories for model selection. Across the tested ranges, the different settings lead to moderate

Table A4: Hyperparameter ablation on FGVC-Aircraft and Stanford-Cars for λ_{flow}, λ_{end}, and α_E.

Param.	FGVC-Aircraft						Stanford-Cars					
	Unlabeled Set			Validation Set			Unlabeled Set			Validation Set		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
$\lambda_{\text{flow}} = 0.5$	54.4	61.3	50.9	56.4	63.0	49.8	58.3	**77.5**	49.0	61.8	82.1	42.3
$\lambda_{\text{flow}} = 1.0$	**56.8**	**62.6**	**53.9**	**57.8**	**64.2**	**51.4**	**59.0**	76.2	**50.6**	**63.5**	**82.8**	**44.9**
$\lambda_{\text{flow}} = 1.5$	52.9	59.2	49.8	55.9	61.8	49.9	56.4	74.2	47.8	60.4	79.4	42.0
$\lambda_{\text{end}} = 0.1$	53.7	60.9	50.0	56.0	62.8	49.2	56.0	74.5	47.1	60.7	80.2	41.8
$\lambda_{\text{end}} = 0.2$	**56.8**	**62.6**	53.9	**57.8**	**64.2**	51.4	59.0	76.2	**50.6**	**63.5**	**82.8**	**44.9**
$\lambda_{\text{end}} = 0.3$	56.4	59.5	**54.8**	57.6	63.4	51.7	59.0	77.1	50.3	62.9	82.0	44.4
$\lambda_{\text{end}} = 0.5$	56.7	61.4	54.3	57.5	62.9	**52.1**	**59.3**	**77.7**	50.4	63.0	82.5	44.2
$\alpha_E = 0.1$	56.1	61.3	53.5	57.3	63.5	51.0	58.4	**77.3**	49.3	62.4	81.9	43.5
$\alpha_E = 0.25$	**56.8**	**62.6**	**53.9**	**57.8**	**64.2**	**51.4**	**59.0**	76.2	**50.6**	**63.5**	**82.8**	**44.9**
$\alpha_E = 0.5$	55.1	60.8	52.3	57.0	62.6	51.3	58.6	75.8	50.3	62.3	81.2	44.0

performance variations, while the configurations favored by validation *Old* accuracy generally also yield competitive performance on the unlabeled set. This supports the use of the held-out labeled split as a practical criterion for hyperparameter selection.

Flow vs. Regressor. To determine whether the improvement comes from learning time-dependent sample-prototype dynamics rather than merely introducing an auxiliary network, we replace the prototype-conditioned velocity field with a static displacement regressor. The regressor uses exactly the same backbone, depth, hidden width, and number of trainable parameters as the flow network, takes $(\mathbf{z}_i, \mathbf{p}_c)$ as input, and directly predicts the displacement $\mathbf{p}_c - \mathbf{z}_i$. We retain the same coupling pipeline, uncertainty weighting, top-k candidate selection, optimization schedule, and endpoint objective, while replacing flow energy with the corresponding regression residual. The time input and randomly sampled intermediate state z_{ic}^t are removed. This capacity-controlled comparison isolates the contribution of modeling a time-conditioned transport field from that of additional model capacity or direct displacement regression. The consistent improvements over the capacity-matched regressor, particularly on novel categories, demonstrate that DynGCD benefits from modeling time-dependent sample-prototype transport dynamics rather than merely introducing auxiliary model capacity or directly regressing prototype displacements.

Flow vs. Direct Prototype Attraction. We further test whether the learned flow can be reduced to a conventional prototype compactness objective. In this variant, we remove the velocity network and its flow-energy term entirely. We construct the balanced soft coupling from the prototype-affinity scores, retain the same uncertainty weighting, and top-k candidate selection, and directly minimize $\sum_{i,c} \omega_{ic} \|\mathbf{z}_i - \mathbf{p}_c\|_2^2$. This parameter-free control tests whether explicitly learning sample-prototype transport provides supervision beyond directly pulling each representation toward its candidate prototypes. The clear gains over direct prototype attraction, especially on novel categories, demonstrate that explicitly learning sample-prototype transport provides more effective geometric supervision than simply pulling representations toward candidate prototypes.

Flow vs. Assignment-Only Supervision. To disentangle whether the gains of DynGCD arise from obtaining better sample–prototype assignments or from using the corresponding relations as transport supervision, we introduce an assignment-only control. We retain the complete coupling-estimation pipeline: the prototype-conditioned flow is still trained, its energy contributes to the compatibility scores, and the resulting scores are normalized into the same Sinkhorn coupling $\mathbf{Q}^\star$. However, the representations supplied to the flow branch are stop-gradient copies, so $\mathcal{L}_{\text{FM}}$ and $\mathcal{L}_{\text{end}}$ update only the velocity network and provide no gradient to the representation encoder. After applying the same uncertainty weighting, and top-k selection, we treat $\mathbf{Q}^\star$ as a fixed soft pseudo-label and optimize

$$\mathcal{L}_{\text{assign}} = -\sum_{i \in \mathcal{B}} \sum_{c \in \mathcal{T}_i} \omega_{ic} \log \frac{\exp(\ell_{ic})}{\sum_{k=1}^{K} \exp(\ell_{ik})}, \tag{A1}$$

where $\mathcal{B}$ denotes the mini-batch samples, $\mathcal{T}_i$ is the top-k candidate set, ω_{ic} is the uncertainty-aware coupling weight derived from $\mathbf{Q}^\star$, and ℓ_{ic} is the host classifier logit for category c. Consequently, the encoder is optimized by $\mathcal{L}_{\text{GCD}} + \lambda_{\text{assign}} \mathcal{L}_{\text{assign}}$, while the flow network serves only to refine

Table A5: Comparison of prototype-conditioned flow matching with a capacity-matched static regressor, direct prototype attraction, and flow-derived assignment-only supervision, implemented on SimGCD (Wen et al., 2023)+DynGCD with a DINO (Caron et al., 2021) ViT-B/16 backbone. The full formulation achieves the highest average All and New accuracy.

Variant	CUB			Stanford-Cars			FGVC-Aircraft			Average		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
Regressor	58.5	64.0	55.8	56.4	74.7	47.6	52.6	62.4	47.7	55.8	67.0	50.4
Direct Attraction	54.1	71.4	45.5	52.4	78.7	39.8	54.6	65.1	49.3	53.7	71.7	44.9
Assignment-Only	53.9	65.1	48.4	46.4	72.8	33.7	46.3	52.6	43.2	48.9	63.5	41.8
DynGCD	63.0	67.7	60.7	59.0	76.2	50.6	56.8	62.6	53.9	59.6	68.8	55.1

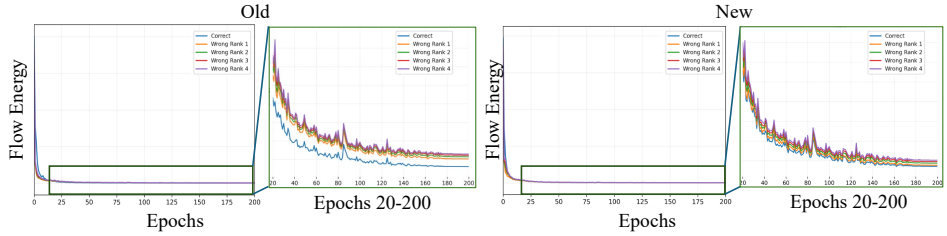


Figure A1: Flow energy analysis of SimGCD (Wen et al., 2023)+DynGCD on FGVC-Aircraft with a DINO (Caron et al., 2021) ViT-B/16 backbone. The average energy of the correct prototype and top-ranked incorrect prototypes is reported over training for old and new categories. Lower energy indicates stronger agreement with the learned sample-prototype transport.

the coupling. Comparing this control with the full DynGCD isolates the benefit of transport-based representation supervision beyond any improvement attributable to flow-derived assignments alone.

A2.4 Analysis on Flow Energy

We analyze whether flow energy provides a meaningful measure of sample-prototype compatibility. At each epoch, we compute flow energy on the unlabeled training set using evaluation transforms and ℓ_2-normalized representations. Following Eq. (5), for each sample-prototype pair we compute

$$E(z, p) = \|v_\psi(z, 0; p) - (p - z)\|_2^2 , \tag{A2}$$

which measures the discrepancy between the learned velocity and the direct transport direction toward the prototype. For analysis only, the correct prototype is identified by Hungarian matching between classifier predictions and ground-truth labels; the remaining prototypes are ranked by classifier logits, and we report the average energy of the correct prototype together with the top-ranked incorrect candidates. Both matching and ranking are recomputed at every epoch, so each curve represents a candidate rank rather than a fixed prototype identity.

As shown in Fig. A1, flow energy decreases rapidly as the velocity field is optimized. More importantly, the correct prototype consistently exhibits lower energy than incorrect candidates, while increasingly mismatched candidates generally yield higher energy. This separation is particularly clear for old categories and remains evident for novel categories despite the absence of labeled category anchors. Although the absolute energies gradually converge later in training, their relative ordering is largely preserved. These results suggest that the learned velocity field captures meaningful sample-prototype dynamics: compatible prototypes induce transport directions that better agree with the learned flow, whereas incompatible prototypes produce larger velocity residuals. Flow energy therefore provides complementary geometric evidence for estimating uncertain sample-prototype couplings beyond category assignments.

A2.5 Stability Analysis

The main tables report the results under the default random seed, whereas Tab. A6 additionally evaluates three independent runs to assess training stability. Tab. A6 reports the mean and standard

Table A6: Stability analysis over three random seeds. Results are reported as mean ± standard deviation (%).

Method	CUB			Stanford-Cars			FGVC-Aircraft		
	All	Old	New	All	Old	New	All	Old	New
SimGCD	61.17 ± 0.64	65.80 ± 1.48	58.86 ± 1.78	53.95 ± 2.33	72.61 ± 4.16	44.94 ± 2.13	52.62 ± 1.56	57.73 ± 2.10	50.08 ± 1.43
+ DynGCD	$\mathbf{62.60 \pm 0.52}$	$\mathbf{66.60 \pm 1.22}$	$\mathbf{60.61 \pm 1.48}$	$\mathbf{59.39 \pm 1.96}$	$\mathbf{77.61 \pm 3.55}$	$\mathbf{50.59 \pm 1.82}$	$\mathbf{58.55 \pm 1.31}$	$\mathbf{63.93 \pm 1.78}$	$\mathbf{55.88 \pm 1.21}$

deviation of clustering accuracy (%) for All, Old, and New classes. Across all three datasets and class splits, adding DynGCD increases mean accuracy and reduces the standard deviation, indicating more stable performance across random seeds.

A3 Computational Analysis

A3.1 Training Time Comparison

Table A7: Training time comparison on fine-grained benchmarks with DINO (Caron et al., 2021) pretrained weights. Time is recorded in Hours (h).

Method	CUB	SCars	Aircraft
SimGCD (Wen et al., 2023)	3.5 h	4.7 h	4.1 h
+ DynGCD	3.9 h	5.3 h	4.5 h

Tab. A7 compares the training cost of SimGCD and DynGCD on three fine-grained benchmarks. Adding DynGCD incurs only a modest overhead of approximately 0.4–0.6 hours, corresponding to a 10–13% increase over the original SimGCD training time. Considering that DynGCD is introduced only during training and leaves the inference procedure unchanged, the additional computational cost remains limited.

A3.2 Flow Model Size Analysis

Tab. A8 reports the parameter count and computational cost of the conditional flow network. With a feature dimension of 768 and 12 residual blocks, the network contains 39.194 million parameters and requires 0.07825 GFLOPs for a single forward evaluation of one feature–prototype pair. We count linear-layer matrix multiplications using two FLOPs per multiply–accumulate operation, excluding normalization, activation, and other elementwise operations. These measurements cover only the flow network and exclude the backbone and classification head.

Table A8: Model size and computational cost of the conditional flow network with feature dimension 768 and 12 residual blocks.

Metric	Value
Parameters (M)	39.194
GFLOPs per feature–prototype pair	0.07825