COMPASS: RADIALLY DOMINANT HYPERBOLIC TRANSPORT FOR GRAPH DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Leading solutions in graph domain adaptation treat domain shift as distribution alignment, learning domain-invariant representations by reducing the discrepancy between two graphs. Such methods transfer well across visual scenes, where CNN features decouple appearance from semantic structure. In a graph, however, a node's adjacency, neighborhood scale and branch membership are deeply entangled, so learning domain-invariant features by directly aligning distributions is often impractical. To address this, through an in-depth analysis of how nodes behave during transfer, we present **Compass**, which recasts graph domain adaptation as the fine-grained movement of individual nodes, termed *transport*, driven by a position-adaptive hyperbolic velocity field. Specifically, the transport has two properties. The first governs *how far* each node moves: hubs, rescaled strongly by the shift, and peripheral nodes, barely touched, each receive a correction matched to their structural position. The second governs *which way* it moves: along the radial direction that carries the domain gap and away from the angular direction that carries class, so trajectories of different classes stay apart, transported labels remain valid without target labels, and a few integration steps suffice. **Compass** reveals that graph adaptation has so far neglected the direction of movement, and the constraint it introduces is flexible, orthogonal to current top solutions and plug-and-play, bringing consistent gains on mainstream benchmarks.

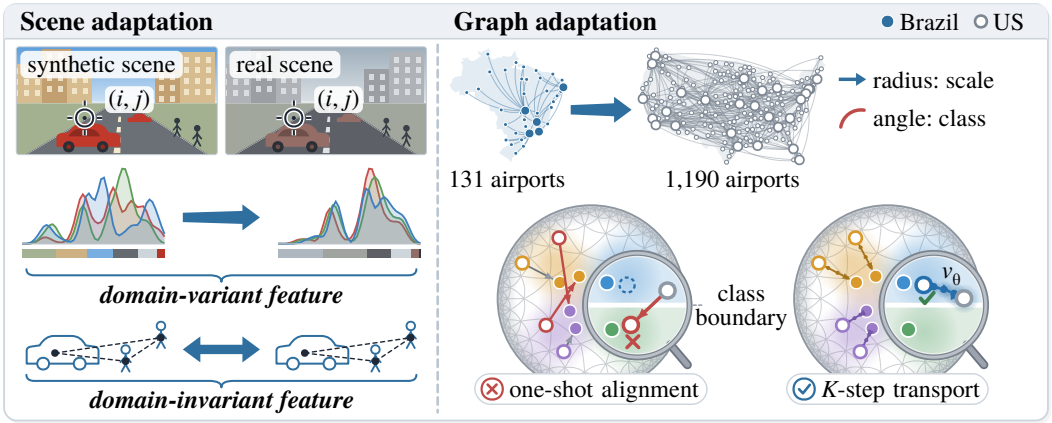

Figure 1: **Left, scene adaptation.** In CNN features the domain-variant feature (appearance) and the domain-invariant feature (structure) are decoupled, so a direct alignment of the two domains works well. **Right, graph adaptation.** In Euclidean GNN features the domain-variant scale of a network (Brazil, 131 airports; US, 1,190) and the domain-invariant class are entangled, and the two graphs share no node, so a one-shot alignment pushes nodes near a class boundary into the neighboring class. In hyperbolic space scale becomes the radius and class the angle, and a K-step transport moves each node along its own direction without leaving its class.

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.

1 INTRODUCTION

Since labels are costly while graphs are plentiful, transferring a model across graphs is a fundamental requirement in graph learning, and the crux of it lies in handling *domain shift*. A graph describes its nodes in two ways, and representative shifts follow each: ❶ *attribute* shift, in the content a node carries, and ❷ *structural* shift, in the connections a node forms. Unsupervised graph domain adaptation (GDA) aims to learn from a labeled source graph so as to perform well on an unlabeled target graph, in a setting where both may occur at once.

GDA is customarily cast as a problem of *distribution alignment*: one learns domain-invariant representations by reducing the discrepancy between the two graphs, through adversarial discriminators (Wu et al., 2020; Shen et al., 2020; Zhang et al., 2021), discrepancy penalties (You et al., 2023; Dai et al., 2023; Yang et al., 2025), class-conditional edge reweighting (Liu et al., 2023; 2024c), or the construction of intermediate graphs that project both domains into a shared semantic space (Kumar et al., 2020; He et al., 2024; Zhuang et al., 2024; Lei et al., 2026; Chen et al., 2026b). Methods of this kind have worked well for domain adaptation in vision (Fig. 1, left). The structure of a car itself, its body, its mirrors and its tires, and the spatial structure between the car and the pedestrians beside it are *domain-invariant features* that barely change across domains; what actually changes is appearance, the illumination, the color and the texture, which are *domain-variant features*. The success of one-shot alignment in vision rests precisely on these two being decoupled across the channels of a representation and therefore easy to separate, a separation that style transfer (Huang & Belongie, 2017) and Fourier alignment (Yang & Soatto, 2020) exploit directly. Graphs are different. The domain-invariant feature vision pursues is the scene structure, whereas on a graph a node's adjacency, the scale of its neighborhood, meaning how many neighbors it has, how densely they are wired and how deep it sits, and its branch membership, meaning which part of the hierarchy it belongs to and which nodes belong there with it, are intricate and deeply entangled, so that learning a domain-invariant feature through a single alignment is often impractical.

Specifically, graphs collected from different sources are shaped by different, domain-specific growth mechanisms, such as preferential attachment in transport networks or cumulative citation in scholarly graphs, and even graphs of the same type may differ markedly in size, density and degree distribution. Consider the transfer from the Brazilian to the American airport network (Fig. 1, right), in which each airport is labeled by its level of activity. Two obstacles make a single alignment ineffective. **(1) The domain-variant and domain-invariant features are entangled.** The American network is an order of magnitude larger (1,190 airports versus 131), and this growth rescales node degree non-uniformly, strongly for hubs and marginally for peripheral airports; this scale is the domain-variant feature, whereas the connectivity pattern that places a node in the network hierarchy carries its class and is the domain-invariant feature. Since message passing aggregates over neighbors, a GNN encodes both in the same Euclidean coordinates, so any correction of scale also perturbs class, and one-shot alignment displaces nodes near class boundaries into adjacent classes (Fig. 1, one-shot alignment). **(2) There is no reference frame for alignment.** The pixel at (i, j) refers to the same location in a synthetic and a real scene, so its correction can be supervised pixel by pixel. The two airport networks share no node, so alignment can only be imposed on distributions, and matching distributions does not determine where any individual node lands. A global rescaling, a class-block reweighting (Liu et al., 2023; 2024c) or a distributional penalty may therefore bring the two networks closer on average while leaving individual nodes misplaced, and class-conditional alignment, which would avoid this, requires target labels that are unavailable in the unsupervised setting.

Given the above, successful GDA calls for a shift from coarse-grained distribution alignment to fine-grained movement of individual nodes, which we term *transport*, carried out without disturbing their class. Such a transport must satisfy two properties. **First**, it must be flexible at the level of each node, so that the magnitude of the correction can vary with a node's structural position. **Second**, its direction must be constrained, advancing only along the directions that carry the domain discrepancy while suppressing those that carry class information. Unlike Euclidean space, a hyperbolic representation reflects structural depth chiefly in the radial coordinate and class semantics chiefly in the angular one (Nickel & Kiela, 2017; Chami et al., 2019). The non-uniform structural scaling between domains then becomes a position-dependent radial displacement, while the class-bearing angle can be preserved throughout the transport. We do not assume this factorization but measure it: on every benchmark we decompose the source-to-target displacement of node representations into radial and angular parts and find that 81% of it is radial, and we report the

δ-hyperbolicity of each graph together with the correlation between hyperbolic radius and structural depth. This finding has so far only been asserted by citation in prior work and, to our knowledge, never tested on adaptation data (Section 4.2).

Continuous-time transport supplies the operational object such a constraint requires. An alignment fixes only a start and an end, whereas a continuous path defines a velocity at every node and every instant, so that radial motion can be advanced and angular motion explicitly suppressed. Continuity, on this reading, is not a claim about how graphs evolve in the world; it is what makes direction a controllable quantity. Methods that insert intermediate graphs move partway toward this (He et al., 2024), yet they subdivide the *magnitude* of the movement and not its direction: every hop stays isotropic and must relabel a target it has never seen, and that error compounds at a rate shown to be tight (Kumar et al., 2020).

Building on this, we present **`Compass`**, which learns a position-adaptive hyperbolic velocity field that corrects structural shift while protecting class-discriminative information, using neither target labels, pseudo-labels nor an estimate of the target class proportions. Three observations follow. **First**, since the discrepancy is radial the transport should be too: we learn a continuous-time velocity field on a manifold of learned curvature and penalize its angular part, so that the displacement of each node is read from its own position. This is a *geometric* form of class-conditional regularization, and unlike its statistical counterparts (Liu et al., 2023; 2024c; Wang et al., 2024), holding a node's angle fixed requires no knowledge of its class, which is why it needs no target label. **Second**, prior work names *path entanglement* as the hazard of transport (Li et al., 2026) but never measures it; we define it as the expected geodesic proximity between class-conditional trajectories, show that it predicts the degradation of Macro-F1 under shift, and make it the quantity our objective bounds. **Third**, a transport confined to one coordinate has a low *effective* dimension, and the step count a sampler requires scales with intrinsic rather than ambient dimension (Li & Yan, 2024; Potaptchik et al., 2025); **`Compass`** reaches its accuracy plateau at 8 function evaluations where a Euclidean transport of matched quality requires 64.

Through this analysis of where the domain gap resides, our work exhibits several compelling facets. **First**, it reveals an axis of graph shift that existing GDA methods leave unconstrained. These methods differ in how they match two distributions, yet all of them let a representation move in any direction; we show by measurement that the direction of movement is what separates a structural shift from a change of class. **Second**, the resulting constraint is orthogonal to existing methods and plug-and-play. Added to DiffGDA and ADAlign without changing their alignment objectives, the angular penalty and head improve both on every benchmark, most on the purely structural Airport benchmark and on the tail classes of MAG, the same pattern that **`Compass`** itself shows. **Third**, accuracy and efficiency follow from one constraint rather than being traded against each other: **`Compass`** matches or exceeds the strongest published methods on all six benchmarks, with its largest margin, 2.2 points of Macro-F1, on the tail classes of MAG, while reducing end-to-end adaptation time by $2.1\times$ against the closest continuous-transport competitor.

2 Related Work

Graph domain adaptation. Most node-level GDA methods learn domain-invariant representations by one-shot distribution alignment, through adversarial discriminators (Zhang et al., 2019; Wu et al., 2020; Shen et al., 2020; Zhang et al., 2021), spectral or MMD-type discrepancies (You et al., 2023; Dai et al., 2023; Yang et al., 2025; Chen et al., 2026a), or class-conditional edge reweighting that corrects conditional structure shift (Liu et al., 2023; 2024c). A second line bridges the gap gradually, through intermediate graphs (Kumar et al., 2020; He et al., 2024; Lei et al., 2026) or, in DiffGDA (Chen et al., 2026b), through a continuous SDE with density-ratio guidance. All of these methods operate in Euclidean space and constrain only where representations arrive; **`Compass`** constrains the direction along which each node travels. Appendix J reviews further lines, including graph-level, open-set and class-imbalanced GDA.

Class-conditional alignment. Aligning marginals while label proportions differ provably increases the joint error (Zhao et al., 2019), and on graphs node-level invariant learning fails for the same reason (Wang et al., 2024). Existing remedies are statistical, relying on estimated label ratios (Liu et al., 2024c; Lipton et al., 2018), inferred environments (Li et al., 2022; Gui et al., 2023) or pseudo-labels

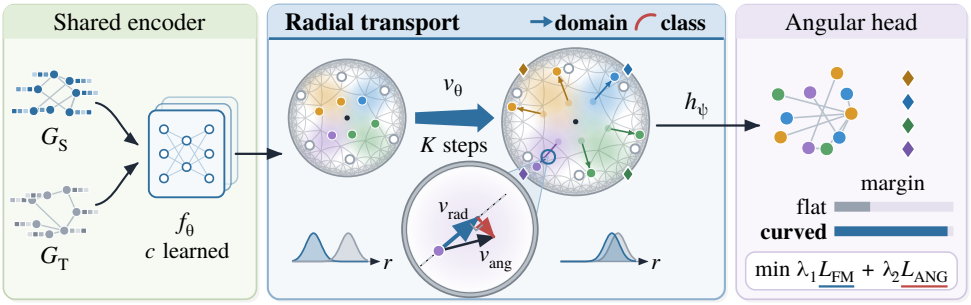

Figure 2: **Compass** encodes both graphs, transports source nodes radially and classifies them by direction. *(1) Shared encoder*. f_θ, with learned curvature, maps both graphs into one Poincaré disk. *(2) Radial transport*. Colored regions are class sectors; filled points are source nodes and hollow points target nodes. The field v_θ moves each node along its own direction in K steps, and the curves show the radius distributions, separated before transport and aligned after it. The magnifier shows the velocity split (Eq. (5)) into v_θ^{rad}, which closes the gap, and v_θ^{ang}, which $\mathcal{L}_{\text{ANG}}$ suppresses. *(3) Angular head*. h_ψ reads the class from direction alone, with one prototype per class (diamonds); the bars contrast the class separation in flat and curved geometry (Eq. (9)). No component uses a target label.

(Dan et al., 2024), and inherit their estimation error; our constraint is geometric and requires none of them.

Hyperbolic representation learning. Hyperbolic space embeds hierarchies with low distortion (Nickel & Kiela, 2017), which hyperbolic neural and graph networks exploit (Ganea et al., 2018; Chami et al., 2019; Bachmann et al., 2020; Chen et al., 2022). Hyperbolic flow matching has been used to keep class-specific transport paths apart in few-shot vision-language adaptation (Li et al., 2026), and hyperbolic embeddings have been applied to hierarchy-imbalanced node classification (Fu et al., 2023) and to active learning under shift (Franco et al., 2024). We use the geometry as the medium in which source-to-target transport between graphs takes place, and justify that choice by measuring where the domain gap lies.

3 Methodology

Compass seeks to replace the one-shot alignment of existing GDA methods by a *transport*, a velocity field that moves each representation individually while constraining the direction in which it may move. We first formalize the problem and introduce two diagnostics that locate the domain gap (Section 3.1). Concretely, we then separate magnitude from direction (Section 3.2), learn a radial transport whose angular deviation is one scalar, the angular action (Section 3.3), and show that this scalar keeps transported labels valid (Section 3.4) and makes a few integration steps sufficient (Section 3.5); Section 3.6 assembles the objective (Fig. 2).

3.1 Preliminaries

Problem Statement. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ be a graph with adjacency $\boldsymbol{A}$, node features $\boldsymbol{X}$ and neighborhoods $\mathcal{N}(v)$. Unsupervised GDA is given a source graph $\mathcal{G}_S$ with labels $\boldsymbol{Y}_S$ over C classes and an unlabeled target graph $\mathcal{G}_T$ on a disjoint node set, which differs from $\mathcal{G}_S$ through ❶ attribute and ❷ structural shift. The goal is a predictor $h \circ f$, composed of an encoder f and a classifier h, that minimizes the target risk

$$\mathcal{R}_T(h \circ f) = \mathbb{E}_{v \sim \mathcal{V}_T}\big[\,\mathbf{1}\{h(f(v)) \neq y_v\}\,\big], \tag{1}$$

without access to any target label, pseudo-label or estimate of the target class proportions.

Alignment-based GDA. Prevailing methods train a shared encoder f_θ and a source classifier h_ψ jointly with an alignment term,

$$\mathcal{L}_{\mathrm{GDA}} = \underbrace{\mathcal{L}_{\mathrm{task}}\big(h_\psi \circ f_\theta,\ \boldsymbol{Y}_S\big)}_{\text{source supervision}} + \lambda\ \underbrace{\mathcal{D}\big(\mathbb{P}_S^{f_\theta},\ \mathbb{P}_T^{f_\theta}\big)}_{\text{one-shot alignment}}, \tag{2}$$

where $\mathcal{D}$ is an adversarial, discrepancy-based, reweighted or stepwise divergence (Section 2). Clearly, $\mathcal{D}$ is a functional of the two marginals: it fixes where the representations should end up, yet leaves free the direction along which each of them travels. This omission is consequential on graphs. A message-passing layer $\boldsymbol{h}_v = \sigma(\boldsymbol{W} \sum_{u \in \mathcal{N}(v)} \boldsymbol{h}_u)$ records *how many* neighbors a node has chiefly in the magnitude of $\boldsymbol{h}_v$ and *which* neighbors chiefly in its direction, whereas in flat space an arbitrary displacement alters both at once.

Diagnostics. To test the hypothesis of Section 1, that in a hyperbolic representation the domain gap is radial while class identity is angular, we introduce two diagnostics. Writing each representation in polar form $(r, \boldsymbol{u})$ (Section 3.2) and coupling source and target nodes by π, the *radial share* measures the fraction of the displacement that is radial,

$$\rho_{\mathrm{rad}} \ = \ \frac{\mathbb{E}_\pi\big[\,|r_T - r_S|^2\,\big]}{\mathbb{E}_\pi\big[\,|r_T - r_S|^2 + \bar{r}^2\, d_{\mathbb{S}}(\boldsymbol{u}_S, \boldsymbol{u}_T)^2\,\big]} \ \in [0,1], \tag{3}$$

where $d_{\mathbb{S}}$ is the geodesic distance on the unit sphere and $\bar{r}$ the mean radius. The *path entanglement*, a hazard that Li et al. (2026) name but do not measure, quantifies how closely classes approach one another during transport. Let $\gamma_y(t)$ be the Fréchet mean of the transported source nodes of class y at time t; then

$$\mathcal{E} \ = \ \frac{2}{C(C-1)} \sum_{y<y'} \int_0^1 \exp\Big(-\tfrac{1}{\tau}\, d_{\mathbb{M}_c}\big(\gamma_y(t), \gamma_{y'}(t)\big)\Big)\, \mathrm{d}t, \tag{4}$$

with a length scale $\tau > 0$. As such, $\mathcal{E}$ is large whenever two classes come close at any instant.

3.2 Geometry: Separating Magnitude from Direction

Hyperbolic Representation Space. We work in the κ-stereographic model $\mathbb{M}_c^d$ (Bachmann et al., 2020), which is the Poincaré ball for $c < 0$ and Euclidean space for $c = 0$, with learnable curvature c. Each point admits polar coordinates $r = d_c(\mathbf{o}, \mathbf{z})$ and $\boldsymbol{u} = \mathbf{z}/\|\mathbf{z}\|$; in hyperbolic embeddings of tree-like graphs, r grows with depth in the hierarchy and $\boldsymbol{u}$ encodes the branch (Nickel & Kiela, 2017).

Polar Decomposition. Since the metric $g_{\mathbf{z}}^c = (\lambda_{\mathbf{z}}^c)^2 g^E$, with $\lambda_{\mathbf{z}}^c = 2/(1 + c\|\mathbf{z}\|^2)$, is conformal to the Euclidean one, every tangent vector $\boldsymbol{w} \in T_{\mathbf{z}}\mathbb{M}_c^d$ admits a g^c-orthogonal split,

$$\boldsymbol{w} = \underbrace{\langle \boldsymbol{w}, \boldsymbol{u} \rangle\, \boldsymbol{u}}_{\boldsymbol{w}^{\mathrm{rad}}\text{: magnitude}} + \underbrace{\big(\mathbf{I} - \boldsymbol{u}\boldsymbol{u}^\top\big)\boldsymbol{w}}_{\boldsymbol{w}^{\mathrm{ang}}\text{: direction}}, \qquad \big\langle \boldsymbol{w}^{\mathrm{rad}}, \boldsymbol{w}^{\mathrm{ang}} \big\rangle_{g_{\mathbf{z}}^c} = 0, \tag{5}$$

where $\boldsymbol{w}^{\mathrm{rad}}$ is tangent to the geodesic ray through the origin (Appendix C). As such, the split costs $O(d)$, needs no exponential or logarithmic map, and supplies exactly what Section 3.1 found missing: coordinates in which magnitude and direction are separately addressable.

3.3 Transport Along the Axis That Shifts

Riemannian Flow Matching. A shared encoder f_θ maps both graphs into $\mathbb{M}_c^d$ by tangent-space message passing followed by $\exp_{\mathbf{o}}^c$ (Chami et al., 2019). We then learn a velocity field $v_\theta(\cdot, t)$ whose flow ϕ_t, given by $\partial_t \phi_t = v_\theta(\phi_t, t)$ and $\phi_0 = \mathrm{id}$, carries the source representations onto the target. With a coupling π from minibatch entropic optimal transport (Pooladian et al., 2023), the geodesic interpolant, its velocity and the matching objective read

$$\begin{aligned} \mathbf{z}_t &= \exp_{\mathbf{z}_S}^c\big(t\, \log_{\mathbf{z}_S}^c \mathbf{z}_T\big), \qquad u_t = \mathrm{PT}_{\mathbf{z}_S \to \mathbf{z}_t}^c\big(\log_{\mathbf{z}_S}^c \mathbf{z}_T\big), \\ \mathcal{L}_{\mathrm{FM}} &= \mathbb{E}_{t\sim\mathcal{U}[0,1],\,(\mathbf{z}_S,\mathbf{z}_T)\sim\pi} \left\| v_\theta(\mathbf{z}_t, t) - u_t \right\|_{g_{\mathbf{z}_t}^c}^2, \end{aligned} \tag{6}$$

so that at its minimum ϕ_t reproduces the marginals of $\mathbf{z}_t$ (Chen & Lipman, 2024; Lipman et al., 2023).

Table 1: Radial share of the domain gap and radius–depth correlation on every benchmark. δ: Gromov hyperbolicity (lower is more tree-like). τ: Spearman correlation of the learned radius with depth (inverse degree / k-core; hubs lie near the origin). ρ_{rad}: radial share (Eq. (3)), drawn as the radial (■) and angular (■) parts of the displacement. Gain: ADAlign with the angular constraint (Section 4.3).

	Graph			Radius ↔ depth		Displacement		Plug-in
Dataset	C	$\lvert\mathcal{V}\rvert$	$\delta\downarrow$	τ_{deg}	τ_{core}	ρ_{rad}	radial / angular	gain
Airport	4	1,190	1.0	0.86	0.81	**0.94**		▲1.1
ArnetMiner	5	9,360	1.5	0.71	0.68	0.83		▲0.5
Blog	6	5,196	2.0	0.63	0.60	0.74		▲0.7
Twitch	2	9,498	1.5	0.68	0.64	0.79		▲0.3
MAG	20	12,661	2.0	0.66	0.62	0.81		▲0.9
ogbn-arxiv	40	169,343	2.5	0.61	0.58	0.77		▲0.8

Angular Action. Because π is built from marginals, $\mathcal{L}_{\mathrm{FM}}$ alone cannot distinguish motion that closes the gap from motion that crosses a class boundary, and Eq. (5) separates the two. Writing $v_\theta = v_\theta^{\mathrm{rad}} + v_\theta^{\mathrm{ang}}$, we define the *angular action* of the trajectory from $\mathbf{z}_S$ as the arc length swept by its direction,

$$\varepsilon(\mathbf{z}_S) = \int_0^1 \frac{\big\|v_\theta^{\mathrm{ang}}\big(\phi_t(\mathbf{z}_S), t\big)\big\|_{g^c}}{\rho\big(\phi_t(\mathbf{z}_S)\big)}\,\mathrm{d}t, \qquad \rho(\mathbf{z}) = \lambda_{\mathbf{z}}^c\|\mathbf{z}\|, \tag{7}$$

where ρ converts a tangential speed into an angular one. In a nutshell, ε measures how far the class-bearing coordinate drifts, and governs label validity and integration cost alike.

3.4 An Angular Head That Keeps Labels Valid

Busemann Angular Head. Following Atigh et al. (2021), class y owns an ideal prototype $\boldsymbol{p}_y \in \mathbb{S}^{d-1}$ on the boundary of the ball, and h_ψ scores $\mathbf{z}$ by its Busemann function ($c = -1$),

$$p_\psi(y \mid \mathbf{z}) = \mathrm{softmax}_y\big(-B_{\boldsymbol{p}_y}(\mathbf{z})\big), \qquad B_{\boldsymbol{p}}(\mathbf{z}) = \log\frac{\|\boldsymbol{p}-\mathbf{z}\|^2}{1-\|\mathbf{z}\|^2}. \tag{8}$$

Since every prototype has unit norm, $\arg\max_y p_\psi(y \mid \mathbf{z}) = \arg\max_y \langle \boldsymbol{p}_y, \boldsymbol{u}\rangle$: the prediction depends on $\mathbf{z}$ only through $\boldsymbol{u}$, while r scales the gaps between class scores and hence the confidence. Each class thus owns a spherical cell $\mathcal{S}_y$, and $m(\mathbf{z}) = d_{\mathbb{S}}(\boldsymbol{u}, \partial\mathcal{S}_y)$ is the angular margin of a node.

Angular Margin Budget. With C cells of equal size, each has angular radius $\Theta(C^{-1/(d-1)})$, and two nodes at radius r in neighboring cells are separated by a geodesic distance (Appendix C)

$$s_{\mathbb{E}}(r, C) = \Theta\big(r\,C^{-1/(d-1)}\big)\ \ (c=0), \qquad s_{\mathbb{H}}(r, C) = 2r - \tfrac{2}{d-1}\log C + O(1)\ \ (c<0). \tag{9}$$

Flat geometry divides the separation by $C^{1/(d-1)}$, so separating more classes requires a larger r, the very coordinate the domain gap occupies; negative curvature subtracts only a $\log C$ term. Hence the benefit of **`Compass`** should grow with C, which Section 4.4 tests.

Label Validity and Entanglement. A transported source node carries its label into the target geometry, and the label remains valid as long as the node stays in its cell. Since ε is the arc length its direction sweeps, the label survives whenever $\varepsilon < m$, and Markov's inequality bounds the invalidated fraction,

$$\Pr\big[\exists\, t:\ \boldsymbol{u}_t \notin \mathcal{S}_y\big] \ \le\ \frac{\mathbb{E}[\varepsilon^2]}{m_{\min}^2}, \qquad \mathbb{E}[\varepsilon^2] \ \le\ \frac{1}{\rho_{\min}^2}\underbrace{\mathbb{E}_t\big\|v_\theta^{\mathrm{ang}}(\mathbf{z}_t, t)\big\|_{g^c_{\mathbf{z}_t}}^2}_{=:\,\mathcal{L}_{\mathrm{ANG}}}, \tag{10}$$

where $m_{\min}$ is the smallest margin among source nodes, $\rho_{\min}$ the smallest ρ along the paths, and $\mathcal{L}_{\mathrm{ANG}}$ is evaluated on the interpolant, whose marginals the flow reproduces. As such, $\mathcal{L}_{\mathrm{ANG}}$ bounds the rate at which the transport corrupts the labels it carries, and it requires no target label, pseudo-label or label-ratio estimate since it constrains only how mass moves. The same penalty also bounds the entanglement of Eq. (4), which is monotone in the largest class-mean angular action (Appendix C).

Table 2: Adding the angular constraint to existing methods. **+ ours**: the angular head and $\mathcal{L}_{\mathrm{ANG}}$ are attached; encoder, alignment objective and hyperparameters are unchanged. Averages per benchmark, Micro-F1 except Macro-F1 on MAG; base rows as in Tables 3 to 5 (DiffGDA on Twitch is our run).

Method	ArnetMiner	Blog	Airport	Twitch	MAG	MAG tail
DiffGDA [ICLR'26]	78.02	51.24	61.41	62.85	35.62	25.55
+ ours	78.73 ▲0.7	52.17 ▲0.9	62.68 ▲1.3	63.27 ▲0.4	36.76 ▲1.1	27.17 ▲1.6
ADAlign [ICLR'26]	78.71	53.75	61.86	63.81	36.40	26.72
+ ours	79.23 ▲0.5	54.41 ▲0.7	62.95 ▲1.1	64.10 ▲0.3	37.33 ▲0.9	28.10 ▲1.4
`Compass` (ours)	79.69	54.87	63.55	64.48	37.94	28.92

3.5 Why a Few Steps Suffice

Geodesic Euler Integration. In practice ϕ_1 is approximated by K geodesic Euler steps from $\hat{\mathbf{z}}_0 = \mathbf{z}_S$, and we write $\Phi_K(\mathbf{z}_S) = \hat{\mathbf{z}}_K$. If $v_\theta^{\mathrm{ang}} \equiv 0$, a trajectory is a reparameterized geodesic ray, which one exponential map integrates exactly; in general (Appendix C)

$$\hat{\mathbf{z}}_{k+1} = \exp^c_{\hat{\mathbf{z}}_k}\Big(\tfrac{1}{K}\, v_\theta\big(\hat{\mathbf{z}}_k, \tfrac{k}{K}\big)\Big), \qquad d_c\big(\Phi_K(\mathbf{z}_S), \phi_1(\mathbf{z}_S)\big) \;\le\; \frac{C_1\, \varepsilon(\mathbf{z}_S)}{K} + O\big(K^{-2}\big), \tag{11}$$

where C_1 depends on the curvature and on a bound of v_θ along the path. In this way, the first-order integration error is proportional to the angular action, and the constraint that protects the classes also makes a few steps sufficient.

3.6 Objective and Complexity

Training Objective. The encoder, the curvature and the field are optimized jointly, with the prototypes fixed,

$$\min_{\theta,\, c} \underbrace{\mathbb{E}_{(v,y)\sim\mathcal{G}_S}\big[-\log p_\psi\big(y \,|\, \Phi_K(f_\theta(v))\big)\big]}_{\mathcal{L}_{\mathrm{task}}:\ \text{source supervision}} + \lambda_1 \underbrace{\mathcal{L}_{\mathrm{FM}}}_{\text{where mass goes}} + \lambda_2 \underbrace{\mathcal{L}_{\mathrm{ANG}}}_{\text{how it gets there}}. \tag{12}$$

Here $\mathcal{L}_{\mathrm{task}}$ applies the head of Eq. (8) to source nodes carried into the target geometry, whose labels Eq. (10) keeps valid, following the logic of DiffGDA, which trains on source graphs generated toward the target distribution (Chen et al., 2026b).

Inference and Complexity. Target nodes are encoded and classified by h_ψ directly (Algorithm 1). One step costs $O(|\mathcal{V}|d)$ and encoding $O(|\mathcal{E}|d + |\mathcal{V}|d^2)$, so adaptation costs $O(K|\mathcal{V}|d + |\mathcal{E}|d)$ and never forms a dense $|\mathcal{V}| \times |\mathcal{V}|$ object.

4 Experiments

4.1 Setup

Datasets and metrics. Following Chen et al. (2026b;a), we evaluate on sixteen transfers over **ArnetMiner** (ACMv9 A, Citationv1 C, DBLPv7 D; 5 classes) (Dai et al., 2023), **Blog** (B1, B2; 6 classes) (Li et al., 2015), **Airport** (USA U, Brazil B, Europe E; 4 classes) (Ribeiro et al., 2017), and **Twitch** (England EN, Germany DE; 2 classes) (Rozemberczki et al., 2021). To test the class-count prediction of Eq. (9) we add **MAG** (20 classes) (Sinha et al., 2015) and **ogbn-arxiv** (40 classes, temporal split) (Hu et al., 2020). We report Micro-F1, and Macro-F1 on MAG and ogbn-arxiv (Appendix E).

Baselines and implementation. We compare with source-only GNNs and the GDA methods evaluated by both ICLR 2026 studies, including DiffGDA and ADAlign, taking published values under the same splits (Chen et al., 2026b;a). We use $\lambda_1 = 1.0$, $\lambda_2 = 0.5$ and $K = 8$, and report means over five seeds; architectures and optimization settings are in Appendix D.

Table 3: Micro-F1 (%, mean and standard deviation over five seeds) on the attributed benchmarks. **T.**: one-shot alignment, intermediate states or a continuous trajectory; **Geo.**: Euclidean, hyperbolic. **Best**, second best; gray: published before 2026; *: significant against the strongest baseline (Welch's t-test, $p < 0.05$). Baselines as published (Chen et al., 2026a;b); Table 9 lists all twenty methods.

Method	T.	Geo.	ArnetMiner						Blog	
			A→C	A→D	C→A	C→D	D→A	D→C	B1→B2	B2→B1
GCN [ICLR'17]			70.82 $\pm$1.26	65.05 $\pm$2.15	65.44 $\pm$1.14	69.46 $\pm$0.83	59.92 $\pm$0.72	66.83 $\pm$0.94	23.17 $\pm$1.10	23.46 $\pm$0.48
UDAGCN [WWW'20]			80.68 $\pm$0.31	74.66 $\pm$0.93	73.46 $\pm$0.40	76.97 $\pm$0.31	69.36 $\pm$0.31	77.81 $\pm$0.50	33.87 $\pm$2.09	31.86 $\pm$1.08
GraphAlign [KDD'24]			75.18 $\pm$0.62	68.81 $\pm$0.78	65.21 $\pm$0.55	72.21 $\pm$0.45	61.66 $\pm$1.23	68.85 $\pm$0.56	45.58 $\pm$1.15	43.21 $\pm$0.62
A2GNN [AAAI'24]			80.93 $\pm$0.52	75.94 $\pm$0.33	75.09 $\pm$0.43	77.16 $\pm$0.23	73.21 $\pm$0.44	79.72 $\pm$0.63	47.10 $\pm$3.40	44.01 $\pm$2.01
GGDA [TIST'26]			79.33 $\pm$1.00	73.58 $\pm$1.73	73.50 $\pm$0.60	76.62 $\pm$0.59	70.94 $\pm$0.11	77.95 $\pm$0.54	48.25 $\pm$2.45	44.85 $\pm$3.13
DiffGDA [ICLR'26]			82.28 $\pm$0.46	76.70 $\pm$0.92	75.75 $\pm$0.30	78.11 $\pm$0.40	74.55 $\pm$0.20	80.71 $\pm$0.20	53.52 $\pm$2.93	48.95 $\pm$2.48
ADAlign [ICLR'26]			82.21 $\pm$0.36	78.06 $\pm$0.61	76.24 $\pm$0.29	79.51 $\pm$0.20	74.38 $\pm$0.98	81.84 $\pm$0.10	55.08 $\pm$0.25	52.42 $\pm$0.46
Compass (ours)			**83.18*** $\pm$0.29	**79.16*** $\pm$0.57	**77.17*** $\pm$0.26	**80.37*** $\pm$0.18	**75.57** $\pm$1.22	**82.71*** $\pm$0.21	**56.17*** $\pm$0.24	**53.57*** $\pm$0.43

Table 4: Micro-F1 on Airport, whose nodes carry no informative attributes, and Twitch. Symbols as in Table 3; GGDA and DiffGDA do not report Twitch; all methods in Table 10.

Method	T.	Geo.	Airport						Twitch	
			U→B	U→E	B→U	B→E	E→U	E→B	DE→EN	EN→DE
GCN [ICLR'17]			56.34 $\pm$3.75	47.11 $\pm$0.52	40.22 $\pm$1.21	48.92 $\pm$1.88	44.67 $\pm$0.99	58.47 $\pm$2.39	57.59 $\pm$0.26	62.82 $\pm$0.33
UDAGCN [WWW'20]			58.17 $\pm$2.07	44.51 $\pm$0.41	42.17 $\pm$1.14	47.97 $\pm$1.72	42.79 $\pm$1.47	62.29 $\pm$3.79	59.37 $\pm$0.63	63.61 $\pm$0.17
GraphAlign [KDD'24]			62.54 $\pm$0.80	52.18 $\pm$0.17	50.33 $\pm$0.77	55.23 $\pm$2.33	54.35 $\pm$0.33	71.02 $\pm$0.48	52.58 $\pm$0.07	60.12 $\pm$0.01
A2GNN [AAAI'24]			64.24 $\pm$1.37	46.62 $\pm$1.61	47.66 $\pm$5.51	47.67 $\pm$1.35	52.72 $\pm$1.16	54.81 $\pm$1.63	55.77 $\pm$0.47	60.92 $\pm$0.56
GGDA [TIST'26]			62.44 $\pm$2.36	48.17 $\pm$1.72	47.03 $\pm$1.29	48.67 $\pm$0.92	49.94 $\pm$1.52	65.80 $\pm$1.70	–	–
DiffGDA [ICLR'26]			71.76 $\pm$1.25	54.18 $\pm$0.58	54.37 $\pm$0.88	57.15 $\pm$0.85	56.20 $\pm$0.63	74.81 $\pm$1.63	–	–
ADAlign [ICLR'26]			72.52 $\pm$0.41	53.23 $\pm$0.68	55.71 $\pm$0.59	58.49 $\pm$1.19	55.41 $\pm$0.29	75.82 $\pm$0.47	61.92 $\pm$0.12	65.69 $\pm$0.26
Compass (ours)			**74.02*** $\pm$0.30	**55.52*** $\pm$0.61	**57.03*** $\pm$0.72	**59.99** $\pm$0.92	**57.56*** $\pm$0.29	**77.19*** $\pm$0.27	**62.56*** $\pm$0.14	**66.40*** $\pm$0.32

4.2 Where the Domain Gap Resides

Table 1 locates the domain gap on every benchmark before any method is compared. All graphs are tree-like ($\delta \leq 2.5$), the learned radius tracks structural depth (Spearman correlation between 0.58 and 0.86, with hubs near the origin), and the source-to-target displacement is predominantly radial (ρ_{rad} between 0.74 and 0.94, 0.81 on average). The radial and angular coordinates therefore separate the part of the representation that shifts across domains from the part that carries the class, a separation that Euclidean GDA methods have no coordinate to express.

4.3 The Radial Axis Is Orthogonal to Existing Alignment

If the radial and angular split identifies a property of graph shift that existing methods leave unused, it should improve them as well. We add only the angular head and the angular penalty to the two strongest published methods, DiffGDA, a Euclidean continuous transport, and ADAlign, a one-shot alignment, and keep everything else as released (Appendix D). Both improve on every benchmark (Table 2), most on the purely structural Airport and on the tail classes of MAG ($+1.6$ and $+1.4$ points of Macro-F1). One method transports and the other aligns in a single step, yet both gain almost equally, so the gain comes from constraining a direction both leave free.

4.4 Comparison with Published Methods

Compass attains the highest mean on all sixteen transfers (Tables 3 and 4), with margins of 0.6 to 1.5 points of Micro-F1, significant on fourteen of them; the exceptions, D→A and B→E, have the largest seed variance. The margin is largest on Airport, whose domain gap is entirely structural, and there the ranking of the baselines shifts as well: among the twenty methods of Appendix G, the

Table 5: Macro-F1 (%, five seeds) on MAG and ogbn-arxiv. Head, middle and tail hold 20%, 30% and 50% of the classes by source frequency (Appendix E), so All $= 0.2\,\text{Head} + 0.3\,\text{Mid} + 0.5\,\text{Tail}$; all rows are our runs (Appendix D).

Method	MAG (20 classes)				ogbn-arxiv (40 classes)			
	All	Head	Mid	Tail	All	Head	Mid	Tail
A2GNN [AAAI'24]	32.64 $\pm$0.61	54.91 $\pm$0.53	36.00 $\pm$0.88	21.72 $\pm$1.50	39.08 $\pm$0.59	62.12 $\pm$0.29	42.38 $\pm$0.58	27.88 $\pm$1.22
TDSS [AAAI'25]	33.27 $\pm$0.83	55.38 $\pm$0.57	36.40 $\pm$0.80	22.55 $\pm$1.95	39.55 $\pm$0.49	62.56 $\pm$0.45	42.96 $\pm$0.66	28.30 $\pm$1.41
DGSDA [ICML'25]	34.16 $\pm$0.53	55.63 $\pm$0.43	37.11 $\pm$0.75	23.80 $\pm$1.72	40.21 $\pm$0.35	62.91 $\pm$0.28	43.20 $\pm$0.58	29.34 $\pm$1.03
HGDA [ICML'25]	33.85 $\pm$0.68	55.78 $\pm$0.65	36.92 $\pm$0.87	23.24 $\pm$1.94	39.94 $\pm$0.55	62.69 $\pm$0.44	43.18 $\pm$0.56	28.90 $\pm$1.08
DiffGDA [ICLR'26]	35.62 $\pm$0.48	56.57 $\pm$0.39	38.44 $\pm$0.81	25.55 $\pm$1.84	41.77 $\pm$0.54	63.05 $\pm$0.31	44.61 $\pm$0.41	31.55 $\pm$0.81
ADAlign [ICLR'26]	36.40 $\pm$0.51	56.76 $\pm$0.31	38.96 $\pm$0.38	26.72 $\pm$1.07	42.31 $\pm$0.33	63.57 $\pm$0.16	45.14 $\pm$0.37	32.11 $\pm$0.60
Compass (ours)	**37.94** $\pm$0.40	**57.36** $\pm$0.22	**40.02** $\pm$0.41	**28.92** $\pm$0.79	**43.68** $\pm$0.26	**63.78** $\pm$0.18	**45.93** $\pm$0.30	**34.29** $\pm$0.79

Table 6: Ablation of the geometry, the form of adaptation and the angular penalty on ArnetMiner (A→C, Micro-F1), Airport (U→B, Micro-F1) and MAG (Macro-F1). ▼: drop from the full model; $\mathcal{E}$: path entanglement of Eq. (4), defined for variants that transport.

	Components			Accuracy ↑			Entanglement
#	Geo.	Adaptation	$\mathcal{L}_{\text{ANG}}$	ArnetMiner	Airport	MAG	$\mathcal{E}$ ↓
1	$\mathbb{H}$	transport	✓	**83.18**	**74.02**	**37.94**	**0.14**
2	$\mathbb{H}$	transport	✗	80.92 ▼2.3	69.88 ▼4.1	33.05 ▼4.9	0.41
3	$\mathbb{E}$	transport	✓	80.87 ▼2.3	70.16 ▼3.9	34.71 ▼3.2	0.33
4	$\mathbb{S}$	transport	✓	79.42 ▼3.8	66.85 ▼7.2	31.84 ▼6.1	0.44
5	$\mathbb{E}$	one-shot	✓	81.63 ▼1.6	71.44 ▼2.6	35.80 ▼2.1	–
6	$\mathbb{H}$	one-shot	✗	80.95 ▼2.2	70.61 ▼3.4	33.58 ▼4.4	–

one-shot A2GNN falls from fourth on ArnetMiner to thirteenth, while GraphAlign, which models the transition, rises from thirteenth to fourth. On the attributed benchmarks the margin grows with the number of classes, from two-class Twitch to six-class Blog (Eq. (9)).

Many classes and a real long tail. Table 5 tests Eq. (9) directly: overall Macro-F1 improves by 1.5 points on MAG and 1.4 on ogbn-arxiv, but by only 0.6 and 0.2 on the head classes against 2.2 on the tail of both, whose margin flat geometry consumes first.

4.5 Ablations and Efficiency

Geometry and regularizer. Keeping only the angular penalty (row 5 of Table 6, flat one-shot alignment) or only the hyperbolic encoder (row 6) falls short of the full method, most on MAG, where the class count makes Eq. (9) binding. **Entanglement tracks accuracy.** Removing $\mathcal{L}_{\text{ANG}}$ raises $\mathcal{E}$ from 0.14 to 0.41 and costs 4.9 points of Macro-F1 on MAG; across the transport variants the rank correlation of $\mathcal{E}$ with the degradation is -0.93.

Efficiency. The same constraint lowers the cost: **Compass** needs 8 function evaluations where its flat counterpart needs 64, and is 2.1× faster end to end than DiffGDA (Tables 12 and 13).

5 Conclusion

Scene adaptation aligns two domains in one step because appearance and structure are decoupled, whereas in Euclidean GNN features the scale and the class of a graph are entangled. We locate the missing separation in a choice of coordinates: in a hyperbolic representation the domain gap is predominantly radial while class identity is angular, so a per-node radial transport closes the gap without disturbing class structure, needs no target label and integrates in a few steps. Constraining this long-neglected direction also improves existing methods. Appendices A and B discuss the limitations and broader impact of **Compass**.

Ethics Statement

This work uses only publicly released citation, social and infrastructure network benchmarks, distributed in anonymized form and containing no personally identifying information; no human subjects were involved. One consequence of the design should be noted: because **`Compass`** corrects structure while preserving class semantics, it also preserves whatever bias the source labels encode. Appendix B discusses this and the deployment settings in which it matters, and Appendix A states the conditions under which the method should not be applied.

Reproducibility Statement

Appendix E gives the statistics, the source / target partitions and the head / middle / tail grouping for every benchmark, including the temporal split we construct for ogbn-arxiv. Appendix D gives the architecture, every hyperparameter, the entropic optimal-transport settings, the numerical-stability measures and the compute used, and states how baseline numbers were obtained. Algorithm 1 states the training and inference procedures in full. Appendix G reports all twenty methods on every transfer with standard deviations over five seeds, and Appendix H reports sensitivity to the one hyperparameter the method adds. Model selection uses source validation data only; no target label is read at any point, including for early stopping. Code and configuration files will be released upon acceptance.

Use of Large Language Models

A large language model was used for language editing and for assistance with LaTeX formatting. It was not used to generate research ideas, to design or run experiments, or to produce or analyze any of the results reported here. The authors take full responsibility for all content.

References

Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic Busemann learning with ideal prototypes. In *NeurIPS*, 2021.

Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In *ICML*, 2020.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, 2019.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *ICLR*, 2024.

Wei Chen, Guo Ye, Yakun Wang, Zhao Zhang, Libang Zhang, Daixin Wang, Zhiqiang Zhang, and Fuzhen Zhuang. Smoothness really matters: A simple yet effective approach for unsupervised graph domain adaptation. In *AAAI*, 2025.

Wei Chen, Xingyu Guo, Shuang Li, Zhao Zhang, Yan Zhong, Fuzhen Zhuang, and Deqing Wang. Learning adaptive distribution alignment with neural characteristic function for graph domain adaptation. In *ICLR*, 2026a.

Wei Chen, Xingyu Guo, Shuang Li, Yan Zhong, Zhao Zhang, Fuzhen Zhuang, Hongrui Liu, Libang Zhang, Guo Ye, and Huimei He. Learning structure-semantic evolution trajectories for graph domain adaptation. In *ICLR*, 2026b.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *ACL*, 2022.

Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE TKDE*, 2023.

Jun Dan, Weiming Liu, Chunfeng Xie, Hua Yu, Shunjie Dong, and Yanchao Tan. TFGDA: Exploring topology and feature alignment in semi-supervised graph domain adaptation through robust clustering. In *NeurIPS*, 2024.

Ruiyi Fang, Bingheng Li, Zhao Kang, Qiuhao Zeng, Nima Hosseini Dashtbayaz, Ruizhi Pu, Charles Ling, and Boyu Wang. On the benefits of attribute-driven graph domain adaptation. In *ICLR*, 2025a.

Ruiyi Fang, Bingheng Li, Jingyu Zhao, Ruizhi Pu, Qiuhao Zeng, Gezheng Xu, Charles Ling, and Boyu Wang. Homophily enhanced graph domain adaptation. In *ICML*, 2025b.

Luca Franco, Paolo Mandica, Konstantinos Kallidromitis, Devin Guillory, Yu-Teng Li, Trevor Darrell, and Fabio Galasso. Hyperbolic active learning for semantic segmentation under domain shift. In *ICML*, 2024.

Xingcheng Fu, Yuecen Wei, Qingyun Sun, Haonan Yuan, Jia Wu, Hao Peng, and Jianxin Li. Hyperbolic geometric graph representation learning for hierarchy-imbalance node classification. In *WWW*, 2023.

Xingcheng Fu, Yisen Gao, Yuecen Wei, Qingyun Sun, Hao Peng, Jianxin Li, and Xianxian Li. Hyperbolic geometric latent diffusion model for graph generation. In *ICML*, 2024.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17, 2016.

Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven adaptation to test-time corruption. In *CVPR*, 2023.

Shurui Gui, Meng Liu, Xiner Li, Youzhi Luo, and Shuiwang Ji. Joint learning of label and environment causal independence for graph out-of-distribution generalization. In *NeurIPS*, 2023.

Yunhui Guo, Xudong Wang, Yubei Chen, and Stella X. Yu. Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *CVPR*, 2022.

Yifei He, Haoxiang Wang, Bo Li, and Han Zhao. Gradual domain adaptation: Theory and algorithms. *JMLR*, 25, 2024.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Yuyang Huang, Yabo Chen, Junyu Zhou, Wenrui Dai, Xiaopeng Zhang, Junni Zou, Hongkai Xiong, and Qi Tian. Diffusion-driven progressive target manipulation for source-free domain adaptation. In *NeurIPS*, 2025.

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *ICML*, 2020.

Pui Ieng Lei, Ximing Chen, Yijun Sheng, Yanyan Liu, Zhiguo Gong, and Qiang Yang. Gradual domain adaptation for graph learning. *ACM TIST*, 2026.

Gen Li and Yuling Yan. Adapting to unknown low-dimensional structures in score-based diffusion models. In *NeurIPS*, 2024.

Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. In *NeurIPS*, 2022.

Jundong Li, Xia Hu, Jiliang Tang, and Huan Liu. Unsupervised streaming feature selection in social media. In *CIKM*, 2015.

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. *arXiv preprint arXiv:2602.20479*, 2026.

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

Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In *ICML*, 2018.

Meihan Liu, Zeyu Fang, Zhen Zhang, Ming Gu, Sheng Zhou, Xin Wang, and Jiajun Bu. Rethinking propagation for unsupervised graph domain adaptation. In *AAAI*, 2024a.

Meihan Liu, Zhen Zhang, Jiachen Tang, Jiajun Bu, Bingsheng He, and Sheng Zhou. Revisiting, benchmarking and understanding unsupervised graph domain adaptation. In *NeurIPS*, 2024b.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *NeurIPS*, 2019.

Shikun Liu, Tianchun Li, Yongbin Feng, Nhan Tran, Han Zhao, Qiang Qiu, and Pan Li. Structural re-weighting improves graph domain adaptation. In *ICML*, 2023.

Shikun Liu, Deyu Zou, Han Zhao, and Pan Li. Pairwise alignment improves graph domain adaptation. In *ICML*, 2024c.

Xin Ma, Yifan Wang, Siyu Yi, Wei Ju, Junyu Luo, Yusheng Zhao, Xiao Luo, and Jiancheng Lv. Dual prototype-enhanced contrastive framework for class-imbalanced graph domain adaptation. In *NeurIPS*, 2025a.

Xin Ma, Yifan Wang, Siyu Yi, Wei Ju, Bei Wu, Ziyue Qiao, Chenwei Tang, and Jiancheng Lv. PALA: Class-imbalanced graph domain adaptation via prototype-anchored learning and alignment. In *IJCAI*, 2025b.

Gal Mishne, Zhengchao Wan, Yusu Wang, and Sheng Yang. The numerical stability of hyperbolic representation learning. In *ICML*, 2023.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, 2017.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *ICML*, 2023.

Peter Potaptchik, Iskander Azangulov, and George Deligiannidis. Linear convergence of diffusion models under the manifold hypothesis. In *COLT*, 2025.

Leonardo F. R. Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. *struc2vec*: Learning node representations from structural identity. In *KDD*, 2017.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2), 2021.

Xiao Shen, Quanyu Dai, Fu-lai Chung, Wei Lu, and Kup-Sze Choi. Adversarial deep network embedding for cross-network node classification. In *AAAI*, 2020.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. An overview of Microsoft Academic Service (MAS) and applications. In *WWW*, 2015.

Yun-Yun Tsai, Fu-Chen Chen, Albert Y. C. Chen, Junfeng Yang, Che-Chun Su, Min Sun, and Cheng-Hao Kuo. GDA: Generalized diffusion for robust test-time adaptation. In *CVPR*, 2024.

Qixun Wang, Yifei Wang, Yisen Wang, and Xianghua Ying. Dissecting the failure of invariant learning on graphs. In *NeurIPS*, 2024.

Lingfeng Wen, Xuan Tang, Mingjie Ouyang, Xiangxiang Shen, Jian Yang, Daxin Zhu, Mingsong Chen, and Xian Wei. Hyperbolic graph diffusion model. In *AAAI*, 2024.

Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *WWW*, 2020.

Liang Yang, Xin Chen, Jiaming Zhuo, Di Jin, Chuan Wang, Xiaochun Cao, Zhen Wang, and Yuanfang Guo. Disentangled graph spectral domain adaptation. In *ICML*, 2025.

Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020.

Nan Yin, Li Shen, Baopu Li, Mengzhu Wang, Xiao Luo, Chong Chen, Zhigang Luo, and Xian-Sheng Hua. DEAL: An unsupervised domain adaptive framework for graph-level classification. In *ACM MM*, 2022.

Nan Yin, Li Shen, Mengzhu Wang, Long Lan, Zeyu Ma, Chong Chen, Xian-Sheng Hua, and Xiao Luo. CoCo: A coupled contrastive framework for unsupervised domain adaptive graph classification. In *ICML*, 2023.

Nan Yin, Mengzhu Wang, Zhenghan Chen, Li Shen, Huan Xiong, Bin Gu, and Xiao Luo. DREAM: Dual structured exploration with mixup for open-set graph domain adaption. In *ICLR*, 2024.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. Graph domain adaptation via theory-grounded spectral regularization. In *ICLR*, 2023.

Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. Adversarial separation network for cross-network node classification. In *CIKM*, 2021.

Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. DANE: Domain adaptive network embedding. In *IJCAI*, 2019.

Zhen Zhang and Bingsheng He. Towards unsupervised open-set graph domain adaptation via dual reprogramming. In *NeurIPS*, 2025.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019.

Zhan Zhuang, Yu Zhang, and Ying Wei. Gradual domain adaptation via gradient flow. In *ICLR*, 2024.