GO-FLOW: POSITIVE-ONLY GRAPH LEARNING FROM CONDITIONAL FLOW FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

In many graph prediction problems, the observed positives are reliable, whereas the unobserved graph objects are ambiguous: a missing edge may be an undiscovered relation, and an unlabeled node may still belong to the target class. We study positive-only graph learning, where trainable parameters are fit from observed positives without assigning negative labels to the remaining graph objects or using them as unlabeled training examples. Conditional flow matching provides a positive-density model, but density estimation alone does not determine how candidates should be ranked. We introduce GO-Flow (**G**raph learning from **O**bserved positives with conditional **Flow**), which fits a conditional flow *only* on observed positive graph objects and uses it to score the unlabeled candidates; for relations, the score contrasts conditional and background densities so that it reflects what the candidate context adds. Relation prediction is the primary instantiation, evaluated across standard link prediction and drug–target interaction, while node positive–unlabeled ranking and a class-as-positive study apply the same conditional flow to nodes. With positive-only training and per-dataset selection, which uses sampled validation non-edges, GO-Flow improves over the learned one-class baselines on all eight link-prediction datasets and compares favorably with the positive-only baselines on seven; methods that additionally use unlabeled or negative examples remain stronger in some regimes, since the unlabeled pool and reliable negative labels carry information that positives alone do not, whereas hidden positives labeled negative degrade these methods.

1 INTRODUCTION

Graph-learning problems often contain an asymmetry in the available labels: positive observations are reliable, while the status of the remaining graph objects is uncertain. In link prediction, an observed edge confirms a relation, but a missing edge may simply be undiscovered. In node prediction, observed positives identify members of a target class, while many other nodes remain unresolved. This ambiguity is common in discovery settings such as drug–target interaction (DTI) (Peng et al., 2017), knowledge-graph completion (Yang et al., 2022), citation discovery (Färber & Jatowt, 2020; Long et al., 2024), and graph positive–unlabeled learning (Wu et al., 2024).

Learned rankers commonly resolve this ambiguity by labeling the remainder. Link predictors sample non-edges as negatives, and positive–unlabeled (PU) methods keep the remainder in the objective as unlabeled examples. Both train on the objects whose status is uncertain, and when some of them are undiscovered positives, the labels they supply are partly incorrect, as our contamination experiment in Section 4.3 illustrates. In this work, we instead fit the model on the observed positives alone and rank the remaining candidates without training on them.

Prior work offers valuable tools for this setting. PU learning characterizes when positive and unlabeled data identify a classifier and how to estimate its risk (Elkan & Noto, 2008; Kiryo et al., 2017; Mordelet & Vert, 2014), and graph PU methods extend it to nodes under heterophily (Wu et al., 2024) and to edge-incomplete link prediction (Kim et al., 2025); these methods, however, train on the unlabeled candidates. One-class methods are the closest supervision match and estimate the support or compactness of positive examples (Schölkopf et al., 2001; Tax & Duin, 2004; Ruff et al., 2018), but a candidate relation depends on both endpoints and their neighborhoods (Zhang & Chen, 2018; Zhang et al., 2021), and membership in a fixed positive region need not capture this

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.

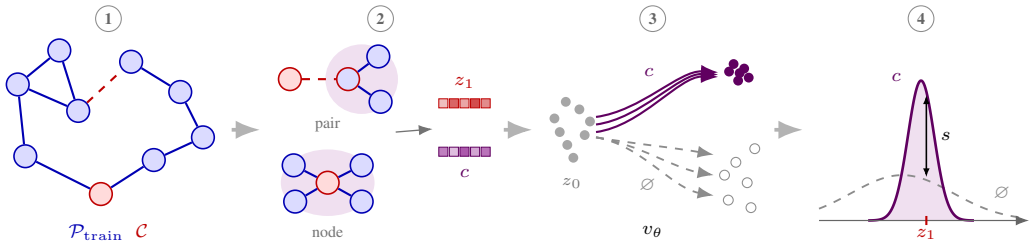

Figure 1: **Overview of GO-Flow.** (1) Only observed positives (blue) fit the model, and candidates (red) stay unlabeled. (2) Each object yields a target z_1 (red) and a graph context c (violet). (3) One conditional flow transports noise z_0 to targets given c and given the null context $\varnothing$. (4) A candidate relation is scored by the gap s between the density of its target under its context and under the null context.

conditional structure. Work on out-of-distribution detection, a different topic, shows that generative likelihood is confounded by background statistics and that a ratio against a background model corrects it (Nalisnick et al., 2019; Ren et al., 2019), and conditional flow matching (CFM) learns conditional densities without simulating the flow during training (Lipman et al., 2023; Tong et al., 2024). Ranking graph objects by what their graph context adds to a positive-only conditional density connects these lines of work, and it is the idea we develop here.

Our approach. We introduce GO-Flow (**G**raph learning from **O**bserved positives with conditional **Flow**s), a method for positive-only graph learning whose graph objects may be relations or nodes. The guiding idea is that a true relation makes the attributes of one endpoint more plausible in the graph context of the other. The endpoints are the two nodes u and v of a candidate pair; in the cross-endpoint construction, the target is the attributes of u and the context is the representation of v from a graph encoder trained jointly with the flow, and both directions are averaged. GO-Flow fits one conditional flow on observed positives only, trained with conditional flow matching, and, by occasionally hiding the context during training, also learns the background density of targets without context. It then scores a candidate relation by how much its graph context raises the density of its target over this background, as illustrated in Figure 1; for nodes, the target is the attributes of a node, the context its graph representation, which aggregates its neighbors and, through the residual connection, also contains its own attributes, and the same family of readouts applies. We prove that ranking by this conditional-to-background contrast is Bayes-optimal when non-relations pair a target with an independent context, and that at full weight it equals the pointwise mutual information of the positive distribution. Empirically, a better fit of the positive density does not by itself rank better, whereas the contrast does. GO-Flow also fuses this score with a positive-only structural encoding (PSE) of the candidate, with a fusion weight selected on validation, and we ablate both components.

Contributions. Our main contributions are as follows:

- We study positive-only graph learning, in which the trainable parameters are fit from observed positives while the unlabeled candidates stay out of the objective, and we instantiate it for relations and nodes in Section 3.
- We introduce GO-Flow, which fits a conditional flow on observed positives and scores candidates by a conditional-to-background contrast that is Bayes-optimal under a stated model of non-relations and equals the pointwise mutual information of the positive distribution at full weight, as we prove in Proposition 1.
- We show that positive-density quality and ranking are distinct objectives: minibatch optimal-transport coupling raises held-out positive log-likelihood on all five tested graphs yet lowers the area under the ROC curve (ROC-AUC) on four of them, as we report in Section 4.4.
- Across eight link-prediction datasets, GO-Flow compares favorably with the positive-only baselines on seven with per-dataset model selection, and on the four DTI families it improves over the strongest positive-only baseline on each family and compares favorably with the published PU methods on three, as we show in Sections 4.1 and 4.2; the PU, sampled-negative, contamination, and hard-negative comparisons of Section 4.3 characterize when extra supervision helps, and Appendix A maps every appendix to these questions.

2 Related Work

Positive-unlabeled and one-class learning. PU learning assumes labeled positives together with unlabeled examples. nnPU stabilizes empirical risk estimation for flexible models (Kiryo et al., 2017), while Bagging-PU trains ensembles against different subsets of the unlabeled data (Mordelet & Vert, 2014). On graphs, GPL studies node-level PU learning under heterophily (Wu et al., 2024), while PULL treats unconnected node pairs as unlabeled candidates and introduces latent variables for edge-incomplete link prediction (Kim et al., 2025). These methods use the unlabeled candidates during training, whereas GO-Flow fits its trainable parameters from observed positives only. One-class methods provide the closest supervision match: OC-SVM (Schölkopf et al., 2001), isolation forest (Liu et al., 2008), Deep SVDD (Ruff et al., 2018), and OCGNN (Wang et al., 2021) estimate the support or compactness of positive examples. These methods offer a principled account of what positives alone determine. Nonetheless, they score a candidate by its position relative to a fixed positive region, whereas GO-Flow scores how much the graph context of a candidate raises the density of its target.

Flow matching and likelihood-based scoring. Flow matching learns a velocity field that transports a base distribution to the data distribution without simulating the flow's ordinary differential equation during training (Lipman et al., 2023). Optimal-transport couplings and reflow modify the geometry of this transport and are commonly studied for generative quality and sampling efficiency (Tong et al., 2024; Liu et al., 2022). Continuous normalizing flows recover likelihoods through the instantaneous change-of-variables formula (Chen et al., 2018). Likelihood alone is not always a useful discriminative score: generative models can assign high likelihood to out-of-distribution inputs because of background input statistics (Nalisnick et al., 2019) and input complexity (Serrà et al., 2020). Likelihood ratios against a background model reduce this effect (Ren et al., 2019), and diffusion likelihoods can also serve directly as classifiers (Li et al., 2023a). GO-Flow applies the background idea conditionally on graph context: for a candidate relation, it compares the conditional density of the target with its background density under a null context.

Graph link prediction. Modern link predictors combine learned node representations with structural information about the candidate pair. SEAL learns from enclosing subgraphs (Zhang & Chen, 2018); BUDDY uses precomputed subgraph sketches for scalable prediction (Chamberlain et al., 2023); NCNC studies how graph incompleteness affects common-neighbor features and learns to complete the missing structure (Wang et al., 2024); and LPFormer learns pairwise encodings with attention (Shomer et al., 2024). These methods are trained with sampled negatives, as is SGDiff, which scores a candidate by the likelihood of its enclosing subgraph under a diffusion model (Li et al., 2024). HeaRT replaces easy random negatives at evaluation time with heuristic-related hard negatives, producing a more demanding ranking test (Li et al., 2023b). BGRL (Thakoor et al., 2023) provides a negative-free representation baseline and helps test whether the gain comes from the relation score or from the graph encoder alone.

3 Positive-Only Graph Learning with GO-Flow

Positive-only graph learning starts from a set of graph objects whose positive members are observed and whose remaining members are unlabeled candidates. Let $G = (V, E_{\text{train}})$ be the observed graph, with $n = |V|$ nodes, node attributes $\mathbf{X} \in \mathbb{R}^{n \times d_x}$, and observed edges E_{train}, and let $\Omega(G)$ denote the objects to be ranked: in this paper, $\Omega(G)$ is either a set of nodes or a set of node pairs. Let $\mathcal{P}_{\text{train}} \subset \Omega(G)$ be the observed positives, which are the edges E_{train} themselves for relations and a subset of V for nodes, and $\mathcal{C} \subset \Omega(G)$ the candidate set. GO-Flow fits its trainable parameters using $\mathcal{P}_{\text{train}}$ only. Each positive object induces a target–context pair (z_1, c), and the learned conditional flow scores a candidate by the conditional density of its target, optionally contrasted with its background density under a null context. The task enters through the construction of the target and context, while the training objective and the family of readouts are shared.

Relation ranking is the primary instantiation and requires an additional pair construction, which we spell out in Sections 3.1 and 3.2. For node prediction, $\Omega(G) = V$ and a positive node i yields $(z_1, c) = (\mathbf{X}[i], \mathbf{H}[i])$, with $\mathbf{H}$ computed as below. Candidate nodes are scored with the readouts of Equation (8), and model selection picks $w = 0$, the conditional log-density, on both node graphs; we give the node protocol in Appendix J. GO-Flow also fuses its score with a positive-only structural

encoding (PSE) of the candidate, which we define in Section 3.2; for nodes, the PSE counts walks into the observed positives, as we detail in Appendix D.

Notation. Bold capitals denote matrices, and $\mathbf{Z}[u]$ is the row of $\mathbf{Z}$ for node u. For vectors, $\|\cdot\|_2$ is the Euclidean norm, $\odot$ the elementwise product, $\|$ concatenation, and $|\cdot|$ the elementwise absolute value; tr is the trace of a square matrix, $\mathbb{E}$ the expectation, $\mathcal{N}(0, \mathbf{I}_d)$ the standard Gaussian on $\mathbb{R}^d$, and $U[0,1]$ the uniform distribution on $[0,1]$.

3.1 Positive-Only Conditional Modeling

Relation instantiation: graph representation and positive pairs. For relation prediction, candidate pairs form the set $\mathcal{C}$ and are not assigned labels in the training objective. We encode the observed graph as

$$\mathbf{H} = \mathrm{Enc}(G, \mathbf{X}) \in \mathbb{R}^{n \times d_h}, \tag{1}$$

where $\mathbf{H}[u]$ is the graph-conditioned representation of node u. The encoder is optimized jointly with the velocity field. Self-loops are removed, but the residual connection still adds $\mathbf{X}[u]$ to $\mathbf{H}[u]$. Training edges stay in the message-passing graph, whereas validation and test pairs are kept out of it, so no evaluated pair contributes its own edge to its context, as we detail in Appendix C.

Each observed positive relation defines target–context examples (z_1, c) with $z_1 \in \mathbb{R}^{d_z}$ and $c \in \mathbb{R}^{d_c}$, whose construction we give in Section 3.2. For every example we draw $z_0 \sim \mathcal{N}(0, \mathbf{I}_{d_z})$ and $t \sim U[0,1]$, set $z_t = (1-t)z_0 + tz_1$, and train the conditional velocity field $v_\theta : \mathbb{R}^{d_z} \times [0,1] \times \mathbb{R}^{d_c} \to \mathbb{R}^{d_z}$, with trainable parameters θ, by minimizing

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}_{(z_1,c),z_0,t}\left[\|v_\theta(z_t, t, c) - (z_1 - z_0)\|_2^2\right]. \tag{2}$$

No sampled non-edge or unlabeled candidate pair enters Equation (2). Following Contrastive Flow Matching (Stoica et al., 2025), we retain a small field correction built from positive tuples,

$$\mathcal{L} = \mathcal{L}_{\mathrm{CFM}} - \lambda\, \mathbb{E}\left[\|v_\theta(z_t, t, c) - (\tilde{z}_1 - \tilde{z}_0)\|_2^2\right], \tag{3}$$

where $(\tilde{z}_0, \tilde{z}_1)$ is formed by permuting source–target tuples within the positive minibatch and λ is a small weight, a hyperparameter listed in Appendix E. This term contrasts positive tuples without declaring any candidate pair negative, and we analyze how it shifts the velocity field in Appendix C.

Conditional and null contexts. A positive density can be high for two different reasons: the candidate relation may explain the target, or the target may simply be common among positives. The ranking score should separate these effects. During training we therefore replace c by a distinguished null context $\varnothing$ with probability 0.2, which we call conditioning dropout. The same velocity field then defines both the conditional density $p_\theta(z \mid c)$, which measures the target in the context of the candidate relation, and the background density $p_\theta(z \mid \varnothing)$, which measures how plausible the target is without relation-specific information.

3.2 Scoring and the Relation Instantiation

Conditioning a relation. The main parameterization predicts across the endpoints of an observed relation. For each positive edge (u, v), cross-endpoint conditioning creates

$$(z_1, c) = (\mathbf{X}[u], \mathbf{H}[v]), \qquad (z_1, c) = (\mathbf{X}[v], \mathbf{H}[u]). \tag{4}$$

Each direction asks whether the attributes of one endpoint are compatible with the graph context of the other. We also use joint conditioning, a symmetric parameterization, on datasets where it validates better (Appendix Q). For any representation matrix $\mathbf{Z} \in \mathbb{R}^{n \times d}$, define $\phi(\mathbf{Z}, u, v) \in \mathbb{R}^{2d}$ as

$$\phi(\mathbf{Z}, u, v) = [\mathbf{Z}[u] \odot \mathbf{Z}[v] \,\|\, |\mathbf{Z}[u] - \mathbf{Z}[v]|]. \tag{5}$$

Joint conditioning uses $z_1 = \phi(\mathbf{X}, u, v)$ and $c = \phi(\mathbf{H}, u, v)$, so $d_z = 2d_x$ and $d_c = 2d_h$, whereas cross-endpoint conditioning has $d_z = d_x$ and $d_c = d_h$.

From conditional density to a candidate score. For a fixed context c, the velocity field defines

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = v_\theta(z(t), t, c). \tag{6}$$

With $p_0 = \mathcal{N}(0, \mathbf{I}_{d_z})$ and $z(t)$ the solution of Equation (6) with $z(1) = z_1$, continuous change of variables gives

$$\log p_\theta(z_1 \mid c) = \log p_0(z(0)) - \int_0^1 \operatorname{tr}\left(\frac{\partial v_\theta}{\partial z}(z(t), t, c)\right) \mathrm{d}t, \tag{7}$$

computed with 20 Euler steps and two Hutchinson probes, which estimate the trace stochastically (Grathwohl et al., 2019), on attributes compressed by principal component analysis (PCA) to 64 standardized dimensions, as we detail in Appendix C; readouts that avoid this integration are in Appendix Y and runtimes in Appendix X. The conditional log-density alone still rewards targets that are likely regardless of the candidate context. We therefore use

$$r_w(z_1, c) = \log p_\theta(z_1 \mid c) - w \log p_\theta(z_1 \mid \varnothing), \qquad w \in \{0, 0.25, 0.5, 0.75, 1\}. \tag{8}$$

We call r_w the conditional-to-background contrast; it is the core scoring principle of GO-Flow. At $w = 0$ it is the conditional log-density, at $w = 1$ it retains only the evidence that the candidate context adds beyond the null context, and intermediate values interpolate between the two. If positives are selected completely at random, r_1 is the pointwise mutual information of the positive distribution, and ranking by r_w is Bayes-optimal when non-relations pair a target drawn from a marginal proportional to $p(z_1)^w$ with an independent context, where p is the distribution of positive target–context pairs, as we prove in Proposition 1 of Appendix B.

For relation prediction, the node-pair score averages the two directed scores under cross-endpoint conditioning, or scores the symmetric pair once under joint conditioning:

$$s_w(u, v) = \begin{cases} \dfrac{1}{2}\left[r_w(\mathbf{X}[u], \mathbf{H}[v]) + r_w(\mathbf{X}[v], \mathbf{H}[u])\right], & \text{cross-endpoint,} \\ r_w(\phi(\mathbf{X}, u, v), \phi(\mathbf{H}, u, v)), & \text{joint.} \end{cases} \tag{9}$$

Explicit pairwise structure through a positive-only encoding. The context $\mathbf{H}[v]$ is a node-level representation, and scores built from node-level representations do not always distinguish pairs that differ only in the structure between them (Zhang et al., 2021). GO-Flow therefore adds a parameter-free positive-only structural encoding (PSE) of the pair. Its 13 coordinates are standard structural features: path counts of length 2 to 4 between the endpoints, as in the Katz and local-path indices (Katz, 1953; Lü et al., 2009), endpoint degrees (Liben-Nowell & Kleinberg, 2003), and random-walk return probabilities, as in random-walk structural encodings (Dwivedi et al., 2022; Rampášek et al., 2022; Cantürk et al., 2024). We compute them from the observed graph alone, mask walks through the candidate edge (Zhu et al., 2024), and average their ranks over the candidate set into a PSE score $\psi(u, v)$, so no weight is learned and no pair is labeled negative; we motivate each choice in Appendix D. With $\mathrm{rk}_{\mathcal{C}}(\cdot) \in [0, 1]$ the normalized rank among the candidates in $\mathcal{C}$, GO-Flow ranks candidates by

$$S_\alpha(u, v) = (1 - \alpha)\,\mathrm{rk}_{\mathcal{C}}(s_w(u, v)) + \alpha\,\mathrm{rk}_{\mathcal{C}}(\psi(u, v)), \qquad \alpha \in \{0, 0.25, 0.5, 0.75\}, \tag{10}$$

where the fusion weight α is a hyperparameter selected on validation and $\alpha = 0$ recovers the flow score s_w. We ablate both components in Section 4.4.

Training supervision and model selection. All trainable encoder and flow parameters are fit using observed positive objects only: observed edges in the relation experiments and observed positive nodes in the node experiments. Neither sampled non-edges nor unlabeled candidates are training examples in Equations (2) and (3). The main protocol selects the encoder, the conditioning parameterization and its implementation, the readout and its weight, the number of reflow steps, and the fusion weight α by validation ROC-AUC on held-out positives and sampled validation non-edges. We therefore use *positive-only training* for the supervision that fits the learned parameters, not for model selection. We list the hyperparameters in Appendix E.

4 Experiments

In this section, we evaluate GO-Flow on relation and node ranking from observed positives. Our experiments seek to address the following questions:

(Q1) Does GO-Flow rank relations and nodes better than positive-only baselines?

Table 1: **GO-Flow is competitive with positive-only baselines.** ROC-AUC ↑, all rows fit on observed positives only; blue marks GO-Flow, best in **bold**, second underlined; 3-path is defined only on bipartite graphs; standard deviations are reported in Appendix I.

Method	Cora	CiteSeer	UMLS	WN18RR	DISEASES	Drug–Disease	PPI	Ohmnet	Avg.
Parameter-free scores									
Feature cosine	.763	.833	.781	.573	.522	.518	.473	.620	.635
Adamic–Adar	.718	.664	.923	.565	.500	.500	.867	.882	.702
3-path	–	–	–	–	.914	**.825**	–	–	–
PSE-only	.695	.592	.856	.735	.708	.800	.884	.870	.768
Learned one-class models									
OC-SVM	.743	.876	.420	.651	.653	.503	.462	.750	.632
Isolation forest	.529	.572	.808	.573	.687	.503	.527	.864	.633
Deep SVDD / OCGNN	.680	.673	.815	.541	.796	.661	.800	.759	.716
GO-Flow (Ours)	**.913**	**.935**	**.937**	**.751**	**.920**	.797	**.889**	**.894**	**.880**

(Q2) Does GO-Flow transfer to DTI discovery?

(Q3) How does GO-Flow compare with methods that also train on unlabeled candidates or sampled negatives, and when do these labels hurt?

(Q4) Which components of GO-Flow produce its ranking signal?

Baselines. We compare GO-Flow with three categories of baselines, grouped by the pair information used for fitting: (1) *positive-only methods*, namely feature cosine similarity, the Adamic–Adar common-neighbor index (Adamic & Adar, 2003), 3-path counts on bipartite graphs (Kovács et al., 2019), PSE-only, OC-SVM, isolation forest, and Deep SVDD/OCGNN, which receive the same observed positives as GO-Flow and assess whether the conditional flow improves on parameter-free scores and one-class objectives; (2) *PU methods*, namely Naive PU, which labels all unlabeled candidates negative (Elkan & Noto, 2008), nnPU, Bagging-PU, PULL, and GPL for nodes, which also train on the unlabeled candidates and measure what this information adds; and (3) *sampled-negative methods*, namely SEAL, BUDDY, LPFormer, and NCNC, which train on non-edges labeled negative. The PU and one-class baselines with an encoder select it from the same four encoders as GO-Flow, on the same validation split and by the same validation ROC-AUC; the sampled-negative methods and PULL run from their official code with their published hyperparameters, as detailed in Appendix G, and main-text tables report means over three held-out seeds (ten for PULL and for three of the five contamination graphs), with population standard deviations in the appendix.

Datasets. Relation prediction uses eight link-prediction datasets: two citation graphs (Cora and CiteSeer (Sen et al., 2008)), two knowledge graphs (UMLS (Kok & Domingos, 2007) and WN18RR (Dettmers et al., 2018)), two bipartite biomedical association graphs (DISEASES gene–disease (Pletscher-Frankild et al., 2015) and Drug–Disease (Gottlieb et al., 2011)), and two protein interaction graphs (PPI (Hamilton et al., 2017) and Ohmnet (Zitnik & Leskovec, 2017)). Node prediction uses node PU ranking on Cora and CiteSeer and a class-as-positive study of 45 tasks from eight graphs. We provide the datasets and protocols in Appendix F and the baselines in Appendix G.

4.1 Relation and Node Ranking

Link prediction. As Table 1 shows, GO-Flow compares favorably with the positive-only baselines on seven of eight datasets and improves over the learned one-class baselines on all eight, with an average ROC-AUC of 0.880 compared with 0.768 for the strongest positive-only baseline. For instance, on Cora GO-Flow reaches 0.913, compared with 0.763 for the next method, and the 3-path count is higher only on Drug–Disease. A negative-free BGRL encoder scored by endpoint cosine similarity reaches 0.902 and 0.853 on Cora and CiteSeer, below GO-Flow, as we detail in Appendix L.

Node ranking. We report node PU ranking in Appendix J. GO-Flow is competitive with the positive-only baselines on both graphs, reaching 0.925 on Cora and 0.924 on CiteSeer, 0.024 above the strongest positive-only baseline on Cora and on par with feature cosine on CiteSeer. Bagging-PU,

Table 2: **GO-Flow improves over the positive-only baselines, and is competitive with the PU methods.** ROC-AUC ↑ under the PUDTI protocol (Peng et al., 2017); PU results as published; best in **bold**, second underlined; standard deviations are reported in Appendix J.

Method	Enzyme	Ion channel	GPCR	Nuclear receptor	Avg.
Positive-only baselines					
Strongest positive-only baseline	.798	.825	.790	.739	.788
Published PU methods					
WNN-GIP (van Laarhoven & Marchiori, 2013)	.861	.775	.872	.839	.837
PUDT-Lan (Lan et al., 2016)	.884	.831	.878	.885	.870
PUDTI (Peng et al., 2017)	.898	.875	.894	**.907**	.894
GO-Flow (Ours)	**.968**	**.969**	**.956**	.894	**.947**

which also trains on the unlabeled candidates, remains higher at 0.956 and 0.932, and we discuss our reimplementation of GPL in Appendix H.

4.2 Drug–Target Interaction

DTI discovery is a second relation-ranking domain in which an unobserved pair need not be a true negative. We evaluate on the four families of Yamanishi et al. (2008) under the PUDTI protocol and report the results in Table 2. GO-Flow improves over the positive-only baselines on all four families, by 0.170 on Enzyme, 0.144 on Ion channel, 0.166 on GPCR, and 0.155 on Nuclear receptor, and compares favorably with the published PU methods on three of the four, with an average ROC-AUC of 0.947 compared with 0.894 for PUDTI; on Nuclear receptor, the smallest family, PUDTI is higher (0.907 against 0.894). The selected fusion weight is $\alpha = 0$ on all four families, so these gains come from the conditional flow alone. One of its two settings also compares favorably in mean reciprocal rank (MRR) and Hits@10 on three of the four families (Appendix K), with features, model selection, and training settings in Appendix J.

4.3 Methods with Additional Pair Information

We next compare with methods that train on more pair information than GO-Flow, first under their standard protocols and then under controlled changes to the reliability of negative labels.

Edge-incomplete graphs. PULL is the closest graph PU method because it learns from the unlabeled candidates directly. On PULL's own candidate sets for Cora-full (Bojchevski & Günnemann, 2018), PubMed (Sen et al., 2008), and the chameleon, crocodile, and Facebook graphs (Rozemberczki et al., 2021), with missing-edge ratio $m = 0.10$, the fraction of edges removed from the observed graph, GO-Flow exceeds PULL on PubMed and crocodile, is on par on chameleon, and is lower on Cora-full and Facebook (Table 3); the same pattern holds at $m = 0.20$ (Appendix N).

Table 3: **GO-Flow is competitive with PULL on its edge-incomplete protocol.** ROC-AUC ↑ at $m = 0.10$; the higher value per graph is in **bold**; standard deviations in Appendix N.

Graph	PULL (unlabeled)	**GO-Flow (Ours)** (positives only)
Cora-full	**.9597**	.9437
PubMed	.9623	**.9684**
chameleon	**.9774**	.9773
crocodile	.9826	**.9856**
Facebook	**.9780**	.9687

PU and sampled-negative methods. In Table 4, GO-Flow remains within 0.024 to 0.057 ROC-AUC of the strongest method with additional pair information on five of six datasets; on WN18RR, sampled negatives add a much larger gain.

False-negative contamination. In this experiment, one third of the training positives is hidden and a fraction of each method's sampled-negative training set is replaced by those hidden true edges. GO-Flow does not use the affected labels, so its training data are unchanged, and its model selection uses clean validation non-edges, so the comparison isolates the training channel. At 40% contamination (Table 5a), PULL loses the most on Cora and CiteSeer, BUDDY loses 6.3% on PPI and 6.2% on WN18RR, NCNC and SEAL are less sensitive, and LPFormer loses 3.9% to 7.5% on all five

Table 4: **GO-Flow is competitive with methods that use additional pair information.** ROC-AUC ↑; each heading states the training signal; standard deviations in Appendix I.

Method	Cora	CiteSeer	UMLS	WN18RR	PPI	Ohmnet
Positives and unlabeled candidates						
Naive PU	.764	.779	.985	.744	.888	.904
nnPU	.737	.743	.964	.666	.888	.897
Bagging-PU	.845	.858	.994	.755	.908	.915
Positives and sampled negatives						
SEAL	.889	.861	.987	.930	.913	.933
BUDDY	.945	.964	.986	.777	.836	.884
LPFormer	.917	.946	.964	.827	.882	.903
NCNC	.965	.966	.970	.837	.896	.928
Observed positives only						
GO-Flow (Ours)	.913	.935	.937	.751	.889	.894

Table 5: **Two regimes for additional pair information.** (a) Retention under false-negative contamination, ROC-AUC ↑. (b) HeaRT MRR ↑, 500 hard negatives per positive. Full results in Appendices M and O; best in **bold** and second underlined, in (a) among methods trained with negatives.

(a) False-negative contamination

Method	Cora	CiteSeer	PPI	WN18RR	Ohmnet
Trained with negatives					
NCNC	.954	.976	**.999**	**.995**	.997
BUDDY	.966	.976	.937	.938	.991
LPFormer	.946	.945	.925	.938	.961
SEAL	**.977**	**.983**	.995	.990	.995
PULL	.861	.822	.990	.949	**1.016**
Observed positives only					
GO-Flow (Ours)	1.000	1.000	1.000	1.000	1.000

(b) HeaRT hard negatives

Method	Cora	CiteSeer	PubMed
Trained with negatives			
LPFormer	**.1452**	**.2714**	**.0864**
BUDDY	.1375	.2225	.0692
NCNC	.1344	.1980	.0378
Observed positives only			
GO-Flow (Ours)	.1011	.1824	.0458

graphs. Negative-trained methods are therefore not uniformly fragile; rather, positive-only fitting avoids this sensitivity by construction, and GO-Flow stays at 0.881 on Cora at every contamination level (Appendix M).

Hard negatives. HeaRT reverses the intervention: each true edge is ranked against 500 heuristic-related hard negatives by MRR, and the sampled-negative methods have seen negative pairs during training. Here this extra information helps: LPFormer reaches an MRR of 0.1452 on Cora and 0.2714 on CiteSeer, compared with 0.1011 and 0.1824 for GO-Flow (Table 5b). On PubMed, GO-Flow is above our NCNC reproduction but below LPFormer and BUDDY, and we report Hits@10 and Hits@20 in Appendix O.

4.4 Ablation and Analysis

We study which components of GO-Flow produce its ranking signal on relation prediction, where the density model and the pair construction can be varied directly.

Conditional-to-background contrast. Subtracting the background density improves ROC-AUC on all five datasets, as can be seen from Table 6: relative to the conditional log-density ($w = 0$), the full contrast ($w = 1$) raises ROC-AUC from 0.803 to 0.911 on Cora, from 0.736 to 0.808 on PPI, and from 0.561 to 0.622 on Drug–Disease, and selecting w adds further gains on PPI and Drug–Disease. We report the component and encoder studies in Appendix P and the conditioning mode in Appendix Q, because their preferred choices vary by dataset.

Flow and PSE. Removing each component of GO-Flow shows that the two carry different signals (Appendix P.1). Removing the PSE leaves Cora, CiteSeer, DISEASES, and the four DTI families unchanged, because the selected fusion weight there is $\alpha = 0$, and lowers ROC-AUC most on Drug–Disease, from 0.797 to 0.639, Ohmnet, from 0.894 to 0.746, and WN18RR, from 0.751 to 0.656.

Table 6: **The contrast improves on the conditional log-density on all five datasets.** ROC-AUC ↑; blue: the readout of GO-Flow (w^* selected); row settings and standard deviations in Appendix P.

Readout	Cora	UMLS	DISEASES	PPI	Drug–Dis.
Log-density, $w = 0$	.803	.876	.908	.736	.561
Contrast, $w = 1$	**.911**	**.915**	**.915**	.808	.622
Contrast, w^*	**.911**	.908	.912	**.823**	**.640**

Removing the flow, which leaves PSE-only, lowers it most on CiteSeer, from 0.935 to 0.592, and on Cora, from 0.913 to 0.695.

Generative quality and ranking. A better positive-density fit does not by itself improve ranking. Minibatch optimal-transport (OT) coupling (Tong et al., 2024) improves held-out positive log-likelihood by 4.7 to 13.6 nats on all five tested datasets, yet ROC-AUC decreases on four of them, by 0.009 on CiteSeer, 0.073 on UMLS, 0.058 on DISEASES, and 0.034 on PPI, while Cora changes by $+0.003$, within noise, and one reflow pass changes it by at most 0.002 (Appendix R).

Direct estimators. A direct positive-only alternative trains a classifier to separate matched positive target–context pairs from recombined positive pairs. GO-Flow is higher than the better of a bilinear and a multilayer perceptron (MLP) direct estimator on six of eight datasets, while the direct estimators are higher on UMLS and WN18RR and score candidates over $100\times$ faster (Appendices S and X).

When recovery is possible. Feature separability predicts recovery. The class-as-positive study, which treats 45 node classes from eight graphs as discovery tasks and exceeds the chance area under the precision–recall curve (AUPRC) in 95.4% of its 1,800 runs, characterizes when positive-only recovery is easiest; we give its protocol in Appendix U. The lift of GO-Flow at $\alpha = 0.5$, its AUPRC divided by the positive rate, has Spearman correlation $\rho = 0.822$ with feature separability over the 45 tasks, measured by a feature-only logistic probe, compared with 0.366 for label informativeness and 0.180 for adjusted homophily (Platonov et al., 2023). Feature separability also has the largest coefficient after controlling for dataset identity (Appendix V).

The gain of PSE-only over GO-Flow with $\alpha = 0$ instead correlates with adjusted homophily ($\rho = 0.809$), mainly between graphs (Appendix W), in agreement with the observation that graph structure is not uniformly useful (Bechler-Speicher et al., 2024; Luan et al., 2023; Lee et al., 2024).

Model selection. Re-selecting configurations with positive validation statistics only lowers mean test ROC-AUC by 0.065 without the PSE and 0.049 with it, while several datasets are unchanged or improve; model selection without trusted negative or unlabeled validation pairs remains open (Appendix T).

5 Conclusions

Many discovery problems on graphs share one asymmetry: the observed relations can be trusted, while an unobserved relation may simply be undiscovered. In this paper, we introduced GO-Flow, which builds on this asymmetry rather than working around it: it learns a conditional flow from the observed positives alone and ranks each candidate relation by how much its graph context raises the density of its target over a background, without ever labeling a candidate as negative. This conditional-to-background contrast turns a conditional generative model into a principled ranker, Bayes-optimal under a stated model of non-relations, and ranking improves with the contrast rather than with a closer fit of the positive density. Across link prediction, drug–target interaction, and node ranking, GO-Flow is strong among positive-only methods and competitive with methods that also train on negative or unlabeled pairs, and, unlike them, its training data are unchanged when undiscovered positives are mislabeled as negatives, because it does not train on negative labels. More broadly, our results show that the positives a field has already confirmed carry a strong signal for what remains to be found, and that progress in conditional generative modeling can carry over directly to discovery on graphs. We envision a positive impact in drug discovery and related screening settings, where GO-Flow can prioritize the candidates most worth testing next.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization and presentation, and literature-oriented checks. All scientific ideas, methodological choices, experimental decisions, mathematical statements, interpretations, and conclusions included in the final manuscript were independently evaluated and approved by the authors. AI-assisted outputs were reviewed by the authors for correctness. The authors take full responsibility for the final content of the paper, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We describe the method in Section 3, prove Proposition 1 in Appendix B, and give the datasets and evaluation protocols in Appendix F, the baselines in Appendix G, and the hyperparameters in Appendix E. The code will be published upon acceptance.

REFERENCES

Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003. ISSN 0378-8733. doi: https://doi.org/10.1016/S0378-8733(03)00009-1. URL https://www.sciencedirect.com/science/article/pii/S0378873303000091.

Maya Bechler-Speicher, Ido Amos, Ran Gilad-Bachrach, and Amir Globerson. Graph neural networks use graphs when they shouldn't. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking, 2018. URL https://arxiv.org/abs/1707.03815.

Semih Cantürk, Renming Liu, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Benjamin Paul Chamberlain, Sergey Shirobokov, Emanuele Rossi, Fabrizio Frasca, Thomas Markovich, Nils Hammerla, Michael M. Bronstein, and Max Hansmire. Graph neural networks for link prediction with subgraph sketching, 2023. URL https://arxiv.org/abs/2209.15486.

Hailin Chen and Zuping Zhang. A semi-supervised method for drug-target interaction prediction with consistency in networks. *PloS one*, 8(5):e62975, 2013.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

Feixiong Cheng, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Weixing Zhou, Jin Huang, and Yun Tang. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS computational biology*, 8(5):e1002503, 2012.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations, 2022. URL https://arxiv.org/abs/2110.07875.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 213–220, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1401920. URL https://doi.org/10.1145/1401890.1401920.

Michael Färber and Adam Jatowt. Citation recommendation: approaches and datasets. *Int. J. Digit. Libr.*, 21(4):375–405, December 2020. ISSN 1432-5012. doi: 10.1007/s00799-020-00288-2. URL https://doi.org/10.1007/s00799-020-00288-2.

Assaf Gottlieb, Gideon Y Stein, Eytan Ruppin, and Roded Sharan. Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology*, 7: 496 – 496, 2011. URL https://api.semanticscholar.org/CorpusID:13806083.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

Mehmet Gönen. Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310, 09 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts360. URL https://doi.org/10.1093/bioinformatics/bts360.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

Junghun Kim, Ka Hyun Park, Hoyoung Yoon, and U Kang. Accurate link prediction for edge-incomplete graphs via pu learning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i17.33966. URL https://doi.org/10.1609/aaai.v39i17.33966.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. URL https://arxiv.org/abs/1609.02907.

Ryuichi Kiryo, Gang Niu, Marthinus C. du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1674–1684, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 433–440, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273551. URL https://doi.org/10.1145/1273496.1273551.

István A Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, Tong Hao, et al. Network-based prediction of protein interactions. *Nature communications*, 10(1):1240, 2019.

Wei Lan, Jianxin Wang, Min Li, Jin Liu, Yaohang Li, Fang-Xiang Wu, and Yi Pan. Predicting drugtarget interaction using positive-unlabeled learning. *Neurocomput.*, 206(C):50–57, September 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2016.03.080. URL https://doi.org/10.1016/j.neucom.2016.03.080.

Meng-Chieh Lee, Haiyang Yu, Jian Zhang, Vassilis N. Ioannidis, Xiang Song, Soji Adeshina, Da Zheng, and Christos Faloutsos. Netinfof framework: Measuring and exploiting network usable information, 2024. URL https://arxiv.org/abs/2402.07999.

Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier, 2023a.

Hang Li, Wei Jin, Geri Skenderi, Harry Shomer, Wenzhuo Tang, Wenqi Fan, and Jiliang Tang. Subgraph based diffusion model for link prediction, 2024. URL https://arxiv.org/abs/2409.08487.

Juanhui Li, Harry Shomer, Haitao Mao, Shenglai Zeng, Yao Ma, Neil Shah, Jiliang Tang, and Dawei Yin. Evaluating graph neural networks for link prediction: current pitfalls and new benchmarking. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023b. Curran Associates Inc.

David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pp. 556–559, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581137230. doi: 10.1145/956863.956972. URL https://doi.org/10.1145/956863.956972.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Kehan Long, Shasha Li, Pancheng Wang, Chenlong Bao, Jintao Tang, and Ting Wang. Recommending missed citations identified by reviewers: A new task, dataset and baselines, 2024. URL https://arxiv.org/abs/2403.01873.

Linyuan Lü, Ci-Hang Jin, and Tao Zhou. Similarity index based on local paths for link prediction of complex networks. *Phys. Rev. E*, 80:046122, Oct 2009. doi: 10.1103/PhysRevE.80.046122. URL https://link.aps.org/doi/10.1103/PhysRevE.80.046122.

Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. When do graph neural networks help with node classification? investigating the impact of homophily principle on node distinguishability. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Haitao Mao, Juanhui Li, Harry Shomer, Bingheng Li, Wenqi Fan, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. Revisiting link prediction: A data perspective, 2024. URL https://arxiv.org/abs/2310.00793.

F. Mordelet and J.-P. Vert. A bagging svm to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209, 2014. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2013.06.010. URL https://www.sciencedirect.com/science/article/pii/S0167865513002432. Partially Supervised Learning for Pattern Recognition.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know?, 2019. URL https://arxiv.org/abs/1810.09136.

Lihong Peng, Wen Zhu, Bo Liao, Yunxin Duan, Min Chen, Yi Chen, and Jialiang Yang. Screening drug-target interactions with positive-unlabeled learning. *Scientific Reports*, 7, 2017. URL https://api.semanticscholar.org/CorpusID:2195652.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. Characterizing graph datasets for node classification: homophily-heterophily dichotomy and beyond. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress?, 2024. URL https://arxiv.org/abs/2302.11640.

Sune Pletscher-Frankild, Albert Pallejà, Kalliopi Tsafou, Janos X. Binder, and Lars Juhl Jensen. Diseases: Text mining and data integration of disease–gene associations. *Methods*, 74:83–89, 2015. ISSN 1046-2023. doi: https://doi.org/10.1016/j.ymeth.2014.11.020. URL https://www.sciencedirect.com/science/article/pii/S1046202314003831. Text mining of biomedical literature.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. *Likelihood ratios for out-of-distribution detection*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 04 2021. ISSN 2051-1329. doi: 10.1093/comnet/cnab014. URL https://doi.org/10.1093/comnet/cnab014.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ruff18a.html.

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. doi: 10.1162/089976601750264965.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, September 2008. ISSN 0738-4602. doi: 10.1609/aimag.v29i3.2157. URL https://doi.org/10.1609/aimag.v29i3.2157.

Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models, 2020. URL https://arxiv.org/abs/1909.11480.

Harry Shomer, Yao Ma, Haitao Mao, Juanhui Li, Bo Wu, and Jiliang Tang. Lpformer: An adaptive graph transformer for link prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 2686–2698, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3672025. URL https://doi.org/10.1145/3637528.3672025.

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching, 2025. URL https://arxiv.org/abs/2506.05350.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, USA, 1st edition, 2012. ISBN 0521190177.

David M. J. Tax and Robert P. W. Duin. Support vector data description. *Mach. Learn.*, 54(1): 45–66, January 2004. ISSN 0885-6125. doi: 10.1023/B:MACH.0000008084.60811.49. URL https://doi.org/10.1023/B:MACH.0000008084.60811.49.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping, 2023. URL https://arxiv.org/abs/2102.06514.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. URL https://arxiv.org/abs/2302.00482.

Twan van Laarhoven and Elena Marchiori. Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. *PLoS ONE*, 8, 2013. URL https://api.semanticscholar.org/CorpusID:9057310.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL https://arxiv.org/abs/1710.10903.

Xiyuan Wang, Haotong Yang, and Muhan Zhang. Neural common neighbor with completion for link prediction, 2024. URL https://arxiv.org/abs/2302.00890.

Xuhong Wang, Baihong Jin, Ying Du, Ping Cui, Yingshui Tan, and Yupu Yang. One-class graph neural networks for anomaly detection in attributed networks. *Neural Comput. Appl.*, 33(18): 12073–12085, September 2021. ISSN 0941-0643. doi: 10.1007/s00521-021-05924-9. URL https://doi.org/10.1007/s00521-021-05924-9.

Yuhao Wu, Jiangchao Yao, Bo Han, Lina Yao, and Tongliang Liu. Unraveling the impact of heterophilic structures on graph positive-unlabeled learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zheng Xia, Ling-Yun Wu, Xiaobo Zhou, and Stephen TC Wong. Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC systems biology*, 4(Suppl 2): S6, 2010.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019. URL https://arxiv.org/abs/1810.00826.

Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 07 2008. ISSN 1367-4803. doi: 10.1093/bioinformatics/btn162. URL https://doi.org/10.1093/bioinformatics/btn162.

Haotong Yang, Zhouchen Lin, and Muhan Zhang. Rethinking knowledge graph evaluation under the open-world assumption. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Zhouxin Yu, Feng Huang, Xiaohan Zhao, Wenjie Xiao, and Wen Zhang. Predicting drug–disease associations through layer attention graph convolutional network. *Briefings in Bioinformatics*, 22(4):bbaa243, 07 2021. ISSN 1477-4054. doi: 10.1093/bib/bbaa243. URL https://doi.org/10.1093/bib/bbaa243.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 5171–5181, Red Hook, NY, USA, 2018. Curran Associates Inc.

Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: a theory of using graph neural networks for multi-node representation learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'03, pp. 321–328, Cambridge, MA, USA, 2003. MIT Press.

Jing Zhu, Yuhang Zhou, Vassilis N. Ioannidis, Shengyi Qian, Wei Ai, Xiang Song, and Danai Koutra. Pitfalls in link prediction with graph neural networks: Understanding the impact of target-link inclusion & better practices. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pp. 994–1002, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635786. URL https://doi.org/10.1145/3616855.3635786.

Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 07 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx252. URL https://doi.org/10.1093/bioinformatics/btx252.

A Appendix Roadmap

The appendix is organized in four parts, method details, the experimental setup, full results, and analyses, and we map each appendix to the part of the paper it supports in Table 7. Evaluations with different split definitions or candidate sets are reported separately throughout.

Table 7: **Appendix roadmap.** Each appendix and the method component or research question of Section 4 that it supports.

Appendix	Content	Supports
Method details		
Appendix B	Proposition 1 and its proof	Method
Appendix C	Message passing, the λ term, and density numerics	Method
Appendix D	PSE construction	Method
Appendix E	Hyperparameters	All
Experimental setup		
Appendix F	Datasets and evaluation protocols	All
Appendix G	Descriptions of all compared methods	All
Appendix H	Reimplemented GPL and SGDiff	Q1, Q3
Full results		
Appendix I	Full standard link-prediction results with standard deviations	Q1, Q3
Appendix J	Node PU ranking and DTI protocols and full results	Q1, Q2
Appendix K	Ranking metrics for DTI	Q2
Appendix L	Representation and supervision controls	Q1
Appendix M	False-negative contamination on five graphs	Q3
Appendix N	PULL candidate sets at a second missing-edge ratio	Q3
Appendix O	HeaRT MRR, Hits@10, and Hits@20	Q3
Analysis		
Appendix P	Component, readout, PSE, and encoder ablations	Q4
Appendix Q	Conditioning mode	Q4
Appendix R	Generative quality against ranking	Q4
Appendix S	Direct estimators of the contrast	Q4
Appendix T	Positive-only model selection	Q4
Appendix U	Class-as-positive study, oracle, and GPL	Q1, Q3, Q4
Appendix V	Dataset controls on the recoverability correlation	Q4
Appendix W	When the PSE helps	Q4
Appendix X	Runtime measurements	Q4
Appendix Y	Scoring budget and the quality–cost frontier	Q4

B What the Score Estimates

Let $p(z_1, c)$ be the distribution of target–context examples induced by true relations, with marginals $p(z_1)$ and $p(c)$. Conditioning dropout trains $p_\theta(z_1 \mid \varnothing)$ on the same positive targets with the context removed, so it estimates the positive target marginal $p(z_1)$, while $p_\theta(z_1 \mid c)$ estimates $p(z_1 \mid c)$.

Proposition 1. *Assume (i) the observed positives are selected completely at random from the true relations, so their target–context distribution is $p(z_1, c)$; and (ii) a candidate is a true relation with probability $\pi \in (0, 1)$ and otherwise a non-relation whose target and context are independent, $q(z_1, c) = q_w(z_1)\, p(c)$ with $q_w(z_1) = p(z_1)^w/\zeta_w$ for some $w \in [0, 1]$ and finite $\zeta_w = \int p(z)^w \, \mathrm{d}z$, which for $w = 0$ requires a bounded target domain. Then the posterior log-odds that a candidate is a true relation equal $r_w(z_1, c) + \log\{\pi\zeta_w/(1-\pi)\}$. Ranking candidates by r_w computed with exact densities is therefore Bayes-optimal and maximizes the population ROC-AUC. At $w = 1$, $r_1 = \log\{p(z_1, c)/(p(z_1)p(c))\}$ is the pointwise mutual information of the positive distribution.*

Proof. By Bayes' rule the log-odds are

$$\log\{\pi\, p(z_1, c)\} - \log\{(1-\pi)\, q_w(z_1)\, p(c)\} = \log p(z_1 \mid c) - w \log p(z_1) + \log\{\pi\zeta_w/(1-\pi)\},$$

where the $\log p(c)$ terms cancel. The last term is the same for every candidate, and by the Neyman–Pearson lemma any strictly increasing function of the likelihood ratio between the two classes gives the ROC-optimal ordering. □

The two ends of the grid correspond to two models of what a non-relation looks like. At $w = 1$ its target is distributed like a positive target, so only the dependence on the context separates the classes; at $w = 0$ the non-relation target distribution is flat and typical targets are rewarded as well. Intermediate values describe non-relations whose targets are less concentrated than positive ones, as when candidate endpoints are less typical than observed endpoints, for example under degree bias. Because this distribution is unknown, w is selected rather than derived. The statement concerns exact densities: the learned velocity field approximates them, the λ term shifts it slightly, as we show in Appendix C, and cross-endpoint scoring averages two directional scores, each covered by the proposition separately.

C Protocol Details: Message Passing, the λ Term, and Density Numerics

Construction (Figure 1). Only the observed positives $\mathcal{P}_{\text{train}}$ fit the model, and no candidate pair or node in $\mathcal{C}$ is labeled negative. Each object yields a target z_1, the attributes of one node, and a context c, the encoder representation of a node. For a pair (u, v), $z_1 = \mathbf{X}[u]$ and $c = \mathbf{H}[v]$, together with the reverse direction; for a node i, $z_1 = \mathbf{X}[i]$ and $c = \mathbf{H}[i]$. The velocity field v_θ is trained by conditional flow matching to transport noise z_0 to the targets given their context, and for relations conditioning dropout also teaches it the null context $\varnothing$, under which it produces the broader background distribution. A relation candidate is scored by the gap s between $\log p_\theta(z_1 \mid c)$ and $\log p_\theta(z_1 \mid \varnothing)$, the contrast of Equation (8); for nodes, model selection picks $w = 0$, so the background density is not subtracted. We fit a separate model for each task.

Target edges in message passing. Training edges remain in E_{train} while their own contexts are computed, so for a training positive (u, v) the context $\mathbf{H}[v]$ has aggregated $\mathbf{X}[u]$. Validation and test pairs are kept out of the message-passing graph, and during model selection the validation positives are removed from it as well, so no evaluated pair contributes to its own context and validation ROC-AUC is not inflated by this path; the effect is a shift between training and evaluation contexts, the distribution shift that Zhu et al. (2024) describe for target links in message passing. Because this path does not reach evaluated pairs, it does not raise test scores directly; we measure its effect on training in Table 8. We remove each minibatch's target edges from message passing at every training step, with one configuration on all three graphs (GCN encoder, FiLM conditioning, cross-endpoint conditioning, the $s_{w=1}$ readout, no reflow) and three seeds. Because masking also acts as edge dropout, a control drops the same number of uniformly chosen training edges. Without the PSE ($\alpha = 0$), masking raises ROC-AUC by 0.025 on Cora and 0.034 on CiteSeer, where the random drop already adds 0.012 and 0.030, so removing the target edges adds 0.013 and 0.004 beyond edge dropout. On PPI, masking lowers ROC-AUC by 0.112, or 0.082 relative to the random drop, because the model uses the training edges during fitting. With the PSE ($\alpha = 0.5$), masking is within 0.021 of the random drop on all three graphs.

Residual connection. Self-loops are removed, but the encoder adds each layer's input to its output when the widths match, and GIN adds a learned self term, so $\mathbf{H}[u]$ contains $\mathbf{X}[u]$. For nodes this places the target inside its own context.

The λ term. At a point (z_t, t, c) the objective of Equation (3) is $\mathbb{E}\|v - y\|_2^2 - \lambda\,\mathbb{E}\|v - \tilde{y}\|_2^2$ with $y = z_1 - z_0$ and $\tilde{y} = \tilde{z}_1 - \tilde{z}_0$. For $\lambda < 1$ it is strictly convex in v, with gradient $2(1-\lambda)v - 2(y^\star - \lambda\tilde{y}^\star)$, where $y^\star = \mathbb{E}[y \mid z_t, c]$ and $\tilde{y}^\star = \mathbb{E}[\tilde{y} \mid z_t, c]$, so it is minimized at $v = (y^\star - \lambda\tilde{y}^\star)/(1 - \lambda)$. With $\lambda = 0.05$ this is a small shift of the CFM field, and Equation (7) is the density of the shifted velocity field. In the component study of Appendix P, the $w = 1$ readout with $\lambda = 0$ and with $\lambda = 0.05$ differ by at most 0.006 ROC-AUC on all five graphs.

Density numerics. Log-densities use 20 Euler steps and two Rademacher Hutchinson probes (Grathwohl et al., 2019). Targets are continuous: attributes are PCA-compressed to 64 standardized dimensions (PPI's 50 are standardized; graphs without attributes use a 64-dimensional SVD of the training adjacency), so d_z is 64 for cross-endpoint and 128 for joint targets (50 and 100 on PPI),

Table 8: **Target-edge masking during training.** Test ROC-AUC ↑, mean ± population standard deviation over three seeds, one configuration on all three graphs; "Random drop" removes the same number of uniformly chosen training edges.

Training graph	Cora	CiteSeer	PPI
Without the PSE ($\alpha = 0$)			
Unmasked	$.917_{\pm.003}$	$.925_{\pm.004}$	$.642_{\pm.008}$
Target edges masked	$.942_{\pm.005}$	$.959_{\pm.001}$	$.530_{\pm.011}$
Random drop	$.929_{\pm.003}$	$.955_{\pm.002}$	$.613_{\pm.011}$
With the PSE ($\alpha = 0.5$)			
Unmasked	$.855_{\pm.005}$	$.798_{\pm.007}$	$.812_{\pm.004}$
Target edges masked	$.884_{\pm.007}$	$.833_{\pm.004}$	$.783_{\pm.006}$
Random drop	$.879_{\pm.005}$	$.847_{\pm.002}$	$.804_{\pm.005}$

with context width 32 and 64. The targets are therefore not raw binary bag-of-words vectors, and no dequantization is applied. One observation qualifies the density reading: on Ohmnet, ROC-AUC rises as the number of integration steps falls, from 0.785 at twenty steps to 0.832 at two, as reported in Appendix Y, so integration error plausibly contributes to the ranking on that graph.

D Positive-Only Structural Encoding Details

Let $\mathbf{A} \in \{0,1\}^{n\times n}$ be the adjacency matrix of the training graph, $\deg u$ the degree of node u, and $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ the random-walk transition matrix with $\mathbf{D}$ the diagonal degree matrix. The PSE collects standard structural features, and we do not claim them as new. We choose them so that they cover the local and global structural factors of link formation, which complement the feature proximity that the node-level context of GO-Flow captures and which rarely dominate the same edges (Mao et al., 2024); our contribution is how they are computed and combined without negative labels.

Path counts. Three coordinates, $\log(1 + \mathbf{A}^k[u, v])$ for $k \in \{2, 3, 4\}$, count walks between the endpoints. Length 2 is the common-neighbor count, and longer lengths add the wider connectivity that the Katz and local-path indices weight (Katz, 1953; Lü et al., 2009). Both parities are needed. In the bipartite association graphs, DISEASES, Drug–Disease, and the DTI graph, a candidate pair joins nodes of different types, so only odd-length walks connect it and length 3 carries the signal, as in the length-three principle for protein interactions (Kovács et al., 2019).

Endpoint statistics. Five coordinates per endpoint: $\log(1 + \deg u)$, the degree signal behind preferential attachment (Liben-Nowell & Kleinberg, 2003), and the random-walk return probabilities $(\mathbf{P}^k)[u, u]$ for $k = 1, \ldots, 4$ (at $k = 1$ this is zero for every node because self-loops are removed, so the coordinate is constant and leaves the ranking unchanged), the random-walk structural encoding of Dwivedi et al. (2022), also used in graph transformers (Rampášek et al., 2022; Cantürk et al., 2024), which describes the cycle structure around each endpoint.

Masking. Walks that would traverse the pair itself are removed at $k = 3$, where every observed edge contributes them; at $k = 4$ an observed edge contributes only through triangles at its endpoints, which cannot occur on the bipartite graphs; $k = 1$ is excluded because it is 1 for every training edge and 0 for every candidate, which would be a train/test shift rather than a signal. This follows the practice of keeping a target link out of its own structural features (Zhu et al., 2024). At $k = 4$ these walks are not removed on the non-bipartite graphs, so a training edge keeps this small triangle-based contribution to its own coordinate.

Combination. The 13 coordinates are rank-transformed over the candidate set and then averaged. Ranks put coordinates of very different scales, such as counts and probabilities, on one scale, and the unweighted average needs no fitted weight, so the PSE has no parameters and uses neither negative labels nor unlabeled candidates as training examples.

Node PSE. For nodes, the PSE is the node analogue of the path counts: for each candidate node i and $k \in \{1, 2, 3\}$ it counts the walks of length k from i into the observed positives, $\log\big(1 + \sum_{j\in\mathcal{P}_{\text{train}}} (\mathbf{A}^k)[i, j]\big)$, rank-normalizes each count over the candidate set, and averages the

three. This is a truncated, unnormalized form of propagating the positive labels along walks, as in label propagation (Zhou et al., 2003), and at $k = 1$ it is the observed-positive neighbor count. Like the relation PSE, it reads only the observed graph and the observed positives and does not label any node negative.

E Hyperparameters and Reproducibility

The score grid is $w \in \{0, 0.25, 0.5, 0.75, 1\}$; conditioning dropout, as in classifier-free guidance (Ho & Salimans, 2022), is 0.2; the contrastive weight is $\lambda = 0.05$, and the reported DTI runs use both, as noted in Appendix J; and the fusion weight is selected from $\alpha \in \{0, 0.25, 0.5, 0.75\}$, where $\alpha = 0$ turns the PSE off. Encoders are selected from GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), GIN (Xu et al., 2019), and a multi-hop alternative, and are trained jointly with the velocity field under a single optimizer. The context enters the velocity field either by concatenation at the input or through FiLM modulation (Perez et al., 2018), which generates a per-block scale and shift from c; FiLM is selected on six of the eight standard link-prediction datasets, and a null context then yields the context-free scale and shift of the block, as Equation (8) requires. Reflow (Liu et al., 2022) is available as a post-training pass that refits the velocity field on its own simulated pairs; it is off on seven of the eight datasets and set to 400 steps on Ohmnet. Model selection uses a validation subset of the positives, with sampled non-edges used only to compute validation ROC-AUC. Each reported model is an ensemble of three flows that differ only in their seed; the search trains a single flow per configuration and retrains the selected one as an ensemble. The encoder and the velocity field are trained jointly with AdamW (weight decay 10^{-4}), a cosine-annealed learning rate starting at 2×10^{-3} for relations and 10^{-3} for nodes, gradient clipping at norm 1, and $1{,}500$ steps on minibatches of 512 relations or 256 nodes. Every encoder has two layers of width 64, an output width of 32 for relations and 64 for nodes, and no self-loops; GAT uses four heads, and the multi-hop encoder aggregates three hops. The velocity field projects z_t, a 32-dimensional sinusoidal embedding of t, and, without FiLM, the context to width 128, applies three residual blocks of width 128 with SiLU activations, and maps linearly back to the target dimension. The search trains 32 configurations per relation dataset, from four encoders, two conditioning modes, FiLM on or off, and reflow on or off, and 16 per node dataset, and scores each under every readout and fusion weight. It holds out 15% of the training positives for validation, with ten sampled non-edges per validation positive and at most $20{,}000$; seed 0 is used only for this search and is excluded from the reported seeds. Experiments ran on one NVIDIA RTX 5090 GPU and NVIDIA RTX 6000 Ada GPUs.

F Data and Evaluation Details

The standard link-prediction experiment uses Cora and CiteSeer (Sen et al., 2008), UMLS (Kok & Domingos, 2007), WN18RR (Dettmers et al., 2018), the DISEASES gene–disease associations (Pletscher-Frankild et al., 2015), the Drug–Disease associations of Gottlieb et al. (2011) as distributed with LAGCN (Yu et al., 2021), the PPI graphs (Hamilton et al., 2017) built from the tissue networks of Zitnik & Leskovec (2017), and Ohmnet, the BioSNAP release of those tissue networks. Ten percent of edges are held out as test positives and paired with sampled non-edges for evaluation. SEAL on PPI and Ohmnet uses one hop; LPFormer on WN18RR uses a personalized PageRank tolerance of $\epsilon = 10^{-5}$. The HeaRT experiment uses the HeaRT splits of Cora, CiteSeer, and PubMed with 500 hard negatives per positive, MRR and Hits metrics, and pessimistic tie-breaking. The PULL comparison uses a random edge split together with PULL's own candidate sets. DTI follows the ten-trial, five-fold pair-level protocol of PUDTI on the families of Yamanishi et al. (2008). Because these evaluations use different splits and candidate sets, their values are reported separately.

G Compared Methods

This appendix describes every method we compare against, in the three supervision groups used in Section 4 plus the controls. Each entry states what the method does, not how it is written down; the original papers give the formal definitions. The grouping is by *what pair labels a method may use during training*, because that is what makes a comparison like-for-like or not. Naive PU, nnPU, and

Deep SVDD/OCGNN select their encoder from the four encoders of GO-Flow on the same validation split by validation ROC-AUC, and train with the widths, steps, and optimizer of Appendix E; Bagging-PU reuses the encoder selected for Naive PU with ten bags. OC-SVM ($\nu = 0.1$) and isolation forest are fit once on the Hadamard product of the endpoint attributes, without a search. SEAL, BUDDY, NCNC, and LPFormer run from their official repositories with the per-dataset hyperparameters their authors publish, on our splits, and are evaluated at the best-validation epoch each repository uses for its own test result. PULL runs from its official code with its default settings on its own protocol (Appendix N), and GPL is our reimplementation with fixed settings (Appendix H). GO-Flow searches a larger grid than the PU baselines, because its conditioning mode, FiLM, reflow, readout, and fusion weight are selected as well (Appendix E).

G.1 Positive-Only Scores without Learning

Feature cosine. Scores a candidate by how similar the two endpoints' attribute vectors are, with no training at all. It is the most direct form of "similar things link", and it measures how much of a benchmark is already decided by node attributes before any graph model is involved.

Adamic–Adar. A classical topological heuristic. It counts the neighbors the two endpoints share and weights each shared neighbor by how rare it is, so a common contact that links to few other nodes counts for more than a hub that links to everything. It uses no attributes and fits no parameters (Adamic & Adar, 2003).

3-path. Counts paths of length three between the two endpoints. This is the natural heuristic for bipartite association graphs, where two nodes of the same type are not adjacent and the shortest meaningful connection between them passes through two intermediate nodes. It is a raw form of the length-three principle of Kovács et al. (2019), and we report it only on the bipartite datasets.

PSE-only. Our own parameter-free topological score, reported as a standalone row so that the contribution of the PSE to GO-Flow can be attributed. It reads a small set of path-count and random-walk statistics off the observed graph, converts each one to a rank over the candidate set, and averages them. It fits no parameters and does not treat any pair as negative. We list its coordinates in Appendix D.

G.2 Learned from Observed Positives Only

One-class SVM. Fits a single boundary that encloses the training examples while keeping that boundary as far as possible from the origin, and scores a new example by which side of it the example falls on. Here the training examples are descriptors of observed edges, so the model learns what an edge looks like and scores candidates by how well they fit that description (Schölkopf et al., 2001).

Isolation forest. Builds an ensemble of random trees, each splitting on a randomly chosen coordinate at a randomly chosen threshold. An example that gets separated from the rest after only a few splits is treated as anomalous, and one that needs many splits as typical. A candidate whose descriptor is hard to isolate therefore resembles the observed edges and scores highly (Liu et al., 2008).

Deep SVDD. The neural form of the one-class boundary. A network is trained to map the observed examples as close as it can to a single fixed center in its output space, and a new example is scored by its distance from that center. Without negatives, the network learns what the positive examples have in common (Ruff et al., 2018).

OCGNN. Applies the same enclosing-region objective to graph neural network embeddings, so the region is defined over graph-aware representations instead of raw attributes (Wang et al., 2021). Our implementation shares its training loop with Deep SVDD and differs in the encoder, which is why the two appear on one row of Table 1.

G.3 Learned from Positives and Unlabeled Candidates

Naive PU. Treats every unobserved pair as a negative and trains an ordinary binary classifier. This is less a PU method than the assumption most supervised link predictors make, stated plainly, and it is the starting point that the two estimators below correct.

nnPU. Keeps that classifier but repairs its training objective. Given an estimate of what fraction of unobserved pairs are really positive, it removes the expected contribution of those hidden positives from the negative part of the objective, and floors the result at zero so the correction stays bounded. The classifier is then no longer penalized for ranking a hidden positive highly (Kiryo et al., 2017).

Bagging-PU. Trains many classifiers, each against a small fresh random sample of unlabeled pairs treated as negatives, and averages their scores. A hidden positive is drawn as a negative by only some of the classifiers, so the penalty it wrongly receives is diluted across the ensemble (Mordelet & Vert, 2014).

PULL. A graph-specific positive–unlabeled link predictor built on the same premise as ours, that an unobserved pair is unlabeled rather than negative (Kim et al., 2025). It alternates between training a link predictor and promoting that predictor's most confident unlabeled pairs to positives, so the graph it trains on is progressively completed, and it redraws its negative sample on every inner epoch. It is the closest published method to our setting that still uses the unlabeled candidates.

G.4 Learned from Positives and Sampled Negatives

SEAL. Extracts the small subgraph surrounding each candidate pair, labels every node in it by its distance to the two endpoints, and classifies the subgraph as a whole (Zhang & Chen, 2018). The classifier therefore sees local structure directly rather than through a heuristic chosen in advance. The cost is that one subgraph is extracted and processed per candidate.

BUDDY. Keeps the use of local structure but replaces the per-candidate subgraph with sketches of each node's neighborhood computed once in advance, from which the overlap statistics of any pair can be recovered cheaply (Chamberlain et al., 2023). Its computation is mainly per node rather than per candidate, which makes it scalable.

NCNC. Starts from common-neighbor counting and corrects for the observed graph being incomplete: it first predicts which neighbors are missing and then computes the common-neighbor evidence on the completed neighborhood (Wang et al., 2024). Its cost is dominated by a single encoder pass and not by per-candidate work, which is why its scaling behavior in Appendix X differs from the other three.

LPFormer. Learns the pairwise structure with attention instead of fixing it in advance. For each candidate it attends over the nodes that connect the two endpoints, so which parts of the local structure matter is learned per dataset rather than chosen by the designer (Shomer et al., 2024).

G.5 Matched Controls

BGRL. A self-supervised graph encoder trained without negatives, by making two augmented views of the graph agree (Thakoor et al., 2023). We rank candidates by the cosine similarity of the two endpoint embeddings. It is the control for whether a good negative-free encoder is already sufficient, which separates representation quality from the contribution of the relation score.

Clean-negative control. The same network trained with node pairs known to be non-edges. It is the measure of what reliable negative supervision adds on this architecture, and not a claim about supervised link prediction in general.

G.6 Reimplemented Comparators

GPL. A graph positive–unlabeled method concerned with heterophily: when linked nodes tend to be dissimilar, propagating labels along observed edges misleads the risk estimate. It learns a weighting over the existing edges that suppresses the misleading ones before propagating (Wu et al., 2024).

SGDiff. Scores a candidate by how likely its enclosing subgraph is under a diffusion model trained on subgraphs, modeling structure and features separately (Li et al., 2024). It is the closest published method to ours in its use of generative likelihood; the principal difference is supervision, since it trains with sampled negatives.

We reimplemented both from their papers, and we discuss what this implies for their numbers in Appendix H.

H Two Reimplemented Comparators

Two published methods close to this setting required reimplementation from their papers. Their numbers therefore characterize our reimplementations rather than the original methods, and we include them because a measured comparison is more informative than a citation alone.

GPL, described in Appendix G.6, studies how heterophily affects graph PU learning when the unlabeled nodes participate in training. Our reimplementation reaches 0.925 ± 0.002 on Cora and 0.891 ± 0.002 on CiteSeer in node PU ranking, on par with GO-Flow on Cora and 0.033 below it on CiteSeer, and below nnPU and Bagging-PU on both. The GPL paper does not specify six implementation details, for which we use three propagation steps, 20 inner and 200 outer steps over five rounds, an edge mask initialized close to the given graph, the known positive rate as the class prior, a held-out half of the believed positives at each inner step so that the label-propagation loss is not trivially satisfied, and ROC-AUC as the metric. We identified and corrected an implementation issue that initially prevented gradients from reaching the learnable edge weights, and we report the corrected version.

SGDiff, described in Appendix G.6, scores a candidate by the conditional likelihood of its enclosing subgraph under a diffusion model, which makes it the closest published method to ours in its use of generative likelihood; the principal difference is supervision, since SGDiff trains with sampled negatives. Our reimplementation reaches 0.637 on Cora against 0.913 for GO-Flow. The SGDiff paper does not specify five implementation details, for which we use a fixed stratified set of diffusion times at inference rather than a Monte Carlo evidence bound, so that a candidate receives the same score on repeated evaluation; a marginal-prior transition matrix for the discrete structure diffusion; 10 structure and 50 feature diffusion steps; our compressed node features; and the same sampled non-edges as the other sampled-negative methods.

I Full Standard-Link Results

This appendix supports Sections 4.1 and 4.3. We report every standard link-prediction result with its standard deviation, one table per supervision group: positive-only methods in Table 9, PU methods in Table 10, and sampled-negative methods in Table 11. GO-Flow remains within 0.024 to 0.052 of the strongest sampled-negative method on five of six datasets; the exception is WN18RR.

J Node PU Ranking and DTI Evaluations

This appendix supports Sections 4.1 and 4.2.

J.1 Node Positive–Unlabeled Ranking

For node PU ranking, the largest class is positive, 25% of its nodes are held out, and all remaining nodes form the candidate set. GO-Flow is instantiated for nodes by taking the target to be the attributes $\mathbf{X}[i]$ of a node and the context to be its representation $\mathbf{H}[i]$ from an encoder without self-loops; the residual connection still adds $\mathbf{X}[i]$ to $\mathbf{H}[i]$, so the context is not restricted to the neighborhood, as we discuss in Appendix C. The conditional flow is trained on the observed positive nodes alone, with the same conditioning dropout (0.2) and λ as for relations. The readout is selected from the same grid as for relations, and model selection picks $w = 0$, the conditional log-density, on both graphs. The class-as-positive study instead trains without conditioning dropout and fixes $w = 0$ in advance; it uses a different construction, which we report separately in Appendix U. As Table 12 shows, GO-Flow is 0.024 above the strongest positive-only baseline on Cora and on par

Table 9: **Standard link prediction, positive-only methods.** ROC-AUC ↑, mean ± standard deviation over three held-out seeds; best in **bold**, second underlined; 3-path is defined only on the two bipartite graphs.

Method	Cora	CiteSeer	UMLS	WN18RR	DISEASES	Drug–Dis.	PPI	Ohmnet
Parameter-free scores								
Feature cosine	.763$_{\pm.004}$	.833$_{\pm.002}$	.781$_{\pm.011}$	.573$_{\pm.004}$	.522$_{\pm.012}$	.518$_{\pm.005}$	.473$_{\pm.004}$	.620$_{\pm.003}$
Adamic–Adar	.718$_{\pm.008}$	.664$_{\pm.006}$	.923$_{\pm.007}$	.565$_{\pm.001}$	.500$_{\pm.000}$	.500$_{\pm.000}$	.867$_{\pm.005}$	.882$_{\pm.001}$
3-path	–	–	–	–	.914$_{\pm.013}$	**.825**$_{\pm.002}$	–	–
PSE-only	.695$_{\pm.004}$	.592$_{\pm.007}$	.856$_{\pm.003}$	.735$_{\pm.003}$	.708$_{\pm.011}$	.800$_{\pm.003}$	.884$_{\pm.001}$	.870$_{\pm.002}$
Learned one-class models								
OC-SVM	.743$_{\pm.021}$	.876$_{\pm.002}$	.420$_{\pm.106}$	.651$_{\pm.003}$	.653$_{\pm.043}$	.503$_{\pm.004}$	.462$_{\pm.005}$	.750$_{\pm.003}$
Isolation forest	.529$_{\pm.005}$	.572$_{\pm.016}$	.808$_{\pm.013}$	.573$_{\pm.004}$	.687$_{\pm.010}$	.503$_{\pm.004}$	.527$_{\pm.010}$	.864$_{\pm.002}$
Deep SVDD / OCGNN	.680$_{\pm.013}$	.673$_{\pm.023}$	.815$_{\pm.017}$	.541$_{\pm.033}$	.796$_{\pm.047}$	.661$_{\pm.062}$	.800$_{\pm.037}$	.759$_{\pm.015}$
GO-Flow (Ours)	**.913**$_{\pm.005}$	**.935**$_{\pm.001}$	**.937**$_{\pm.004}$	**.751**$_{\pm.002}$	**.920**$_{\pm.009}$	.797$_{\pm.003}$	**.889**$_{\pm.000}$	**.894**$_{\pm.002}$

Table 10: **GO-Flow is competitive with PU methods that also train on the unlabeled candidate pairs.** ROC-AUC ↑, mean ± standard deviation over three held-out seeds; each heading states the training signal, and GO-Flow is repeated from Table 9 for comparison. Best in **bold**, second underlined.

Method	Cora	CiteSeer	UMLS	WN18RR	DISEASES	Drug–Dis.	PPI	Ohmnet
Positives and unlabeled candidates								
Naive PU	.764$_{\pm.012}$	.779$_{\pm.011}$	.985$_{\pm.002}$	.744$_{\pm.007}$	.850$_{\pm.016}$	.842$_{\pm.002}$	.888$_{\pm.001}$	.904$_{\pm.003}$
nnPU	.737$_{\pm.009}$	.743$_{\pm.005}$	.964$_{\pm.003}$	.666$_{\pm.001}$	.837$_{\pm.019}$	.840$_{\pm.002}$	.888$_{\pm.002}$	.897$_{\pm.004}$
Bagging-PU	.845$_{\pm.004}$	.858$_{\pm.007}$	**.994**$_{\pm.001}$	**.755**$_{\pm.002}$	.903$_{\pm.014}$	**.869**$_{\pm.001}$	**.908**$_{\pm.001}$	**.915**$_{\pm.003}$
Observed positives only								
GO-Flow (Ours)	**.913**$_{\pm.005}$	**.935**$_{\pm.001}$	.937$_{\pm.004}$	.751$_{\pm.002}$	**.920**$_{\pm.009}$	.797$_{\pm.003}$	.889$_{\pm.000}$	.894$_{\pm.002}$

Table 11: **Standard link prediction, methods trained with sampled negatives.** ROC-AUC ↑, mean ± standard deviation over three held-out seeds; not run on the two bipartite graphs; GO-Flow repeated from Table 9; best in **bold**, second underlined.

Method	Cora	CiteSeer	UMLS	WN18RR	PPI	Ohmnet
Positives and sampled negatives						
SEAL	.889$_{\pm.005}$	.861$_{\pm.008}$	**.987**$_{\pm.002}$	**.930**$_{\pm.005}$	**.913**$_{\pm.002}$	**.933**$_{\pm.001}$
BUDDY	.945$_{\pm.005}$	.964$_{\pm.001}$	.986$_{\pm.001}$	.777$_{\pm.005}$	.836$_{\pm.003}$	.884$_{\pm.002}$
LPFormer	.917$_{\pm.008}$	.946$_{\pm.006}$	.964$_{\pm.004}$	.827$_{\pm.005}$	.882$_{\pm.004}$	.903$_{\pm.003}$
NCNC	**.965**$_{\pm.004}$	**.966**$_{\pm.003}$	.970$_{\pm.001}$	.837$_{\pm.002}$	.896$_{\pm.002}$	.928$_{\pm.001}$
Observed positives only						
GO-Flow (Ours)	.913$_{\pm.005}$	.935$_{\pm.001}$	.937$_{\pm.004}$	.751$_{\pm.002}$	.889$_{\pm.000}$	.894$_{\pm.002}$

with feature cosine on CiteSeer within the reported spread. Bagging-PU is higher on both datasets and also uses the unlabeled candidates during training.

J.2 DTI UNDER THE PUDTI PROTOCOL

The DTI experiment uses the four families of Yamanishi et al. (2008) with ten trials of five-fold pair-level cross-validation, as in PUDTI (Peng et al., 2017). We report the full comparison in Table 13. Published PUDTI results are taken from the original evaluation and are not remeasured. For the methods evaluated in our experiments, the reported standard deviations do not specify which

Table 12: **GO-Flow is competitive with the positive-only baselines for node PU ranking, while the strongest PU method remains higher.** ROC-AUC ↑, mean ± standard deviation over three held-out seeds; PU methods also train on the unlabeled candidates. Among positive-only methods, best in **bold**, second underlined.

Method	Cora	CiteSeer
Observed positives only		
Feature cosine	$.890_{\pm.002}$	$\underline{.922}_{\pm.008}$
Feature-centroid distance	$.599_{\pm.007}$	$.507_{\pm.016}$
Observed-positive neighbor rate	$\underline{.901}_{\pm.015}$	$.844_{\pm.019}$
PSE-only	$.868_{\pm.021}$	$.841_{\pm.012}$
OC-SVM	$.831_{\pm.006}$	$.864_{\pm.002}$
Isolation forest	$.406_{\pm.016}$	$.480_{\pm.014}$
Deep SVDD / OCGNN	$.833_{\pm.025}$	$.737_{\pm.007}$
GO-Flow (Ours)	$\mathbf{.925}_{\pm.008}$	$\mathbf{.924}_{\pm.004}$
Positives and unlabeled candidates		
Naive PU	$.920_{\pm.005}$	$.895_{\pm.007}$
nnPU	$.944_{\pm.009}$	$.914_{\pm.007}$
Bagging-PU	$.956_{\pm.003}$	$.932_{\pm.006}$
GPL (reimpl.)	$.925_{\pm.002}$	$.891_{\pm.002}$

population they aggregate, so we do not combine those uncertainty values with the published rows. Nuclear receptor is the smallest family, with 90 known interactions and 18 per test fold.

Features and selection. Drugs and targets are nodes of one bipartite graph. Node attributes are block-diagonal similarity profiles (a drug's row holds its drug–drug chemical similarities and a target's row its target–target genomic similarities), jointly PCA-compressed to 64 standardized dimensions, and in each fold the encoder graph is that fold's training interactions. One configuration per family is selected on the search seed from a validation split of 15% of the interactions with ten sampled non-interactions each, then held fixed for the 10×5 cross-validation of the reported seeds. Because the validation interactions are drawn from all interactions, they also appear in the cross-validation test folds: no model is trained on a test pair, but the configuration choice carries a small optimistic bias. The flow is trained with conditioning dropout 0.2 and $\lambda = 0.05$, as for the other relation tasks.

Table 13: **DTI under the PUDTI protocol, full results.** ROC-AUC ↑, mean ± standard deviation for the methods we evaluate; published PU results as reported by their authors; best in **bold**, second underlined.

Method	Enzyme	Ion channel	GPCR	Nuclear receptor
Observed positives only				
Feature cosine	$.545_{\pm.000}$	$.510_{\pm.000}$	$.477_{\pm.000}$	$.484_{\pm.000}$
Weighted profile	$.758_{\pm.000}$	$.709_{\pm.000}$	$.790_{\pm.000}$	$.739_{\pm.000}$
PSE-only	$.798_{\pm.000}$	$.825_{\pm.000}$	$.661_{\pm.000}$	$.488_{\pm.008}$
Deep SVDD / OCGNN	$.798_{\pm.008}$	$.805_{\pm.005}$	$.732_{\pm.006}$	$.554_{\pm.021}$
GO-Flow (Ours)	$\mathbf{.968}_{\pm.002}$	$\mathbf{.969}_{\pm.001}$	$\mathbf{.956}_{\pm.000}$	$\underline{.894}_{\pm.004}$
Published PU methods				
DBSI (Cheng et al., 2012)	.806	.803	.803	.759
NetLapRLS (Xia et al., 2010)	.802	.793	.826	.761
KBMF2K (Gönen, 2012)	.812	.802	.840	.810
NetCBP (Chen & Zhang, 2013)	.825	.803	.823	.838
WNN-GIP (van Laarhoven & Marchiori, 2013)	.861	.775	.872	.839
PUDT-Lan (Lan et al., 2016)	.884	.831	.878	.885
PUDTI (Peng et al., 2017)	<u>.898</u>	<u>.875</u>	<u>.894</u>	**.907**

K Ranking Metrics for DTI

This appendix supports Section 4.2. ROC-AUC measures whether positives separate from the candidates, whereas a screening campaign is concerned with the top of the ranking. We therefore computed MRR, Hits@k, and Precision@k on the same cross-validation folds as the ROC-AUC values, recomputing the whole DTI table on a single machine so that both metric families share one set of folds. Ties are broken pessimistically, as in the HeaRT evaluation, so a constant score earns an MRR of $1/(N+1)$ rather than 1, where N is the number of negatives ranked against each positive.

One of the two settings of GO-Flow compares favorably in MRR, Hits@10, and Precision@10 on three of the four families. On MRR, the better setting reaches 0.052 against 0.036 for the strongest positive-only baseline on Enzyme, 0.016 against 0.005 on GPCR, and 0.114 against 0.082 on Nuclear receptor. On Ion channel, the weighted-profile baseline ranks the top candidates higher, with an MRR of 0.039 against 0.030, a Hits@10 of 0.064 against 0.056, and a Precision@10 of 0.873 against 0.664. Training the null context lowers the MRR of the flow alone while it raises ROC-AUC, for example from 0.076 to 0.038 on Enzyme: the global ordering improves while fewer true interactions reach the very top, and the setting with the PSE retains most of its top-of-ranking performance. The PU DTI literature reports ROC-AUC only, so these values characterize ranking behavior and do not extend a published comparison.

L Matched Representation and Supervision Controls

This appendix supports Section 4.1. BGRL (Thakoor et al., 2023) provides a negative-free encoder, and we rank candidate pairs by the cosine similarity of their endpoint embeddings. It gives 0.902 on Cora and 0.853 on CiteSeer, compared with 0.913 and 0.935 for GO-Flow. On Ohmnet and Drug–Disease, BGRL gives 0.774 and 0.559, compared with 0.894 and 0.797 for GO-Flow. Standard deviations are not available for this control.

A second control keeps the same network but trains it with known non-edges as negatives. It gives 0.868 on Cora, 0.872 on CiteSeer, and 0.919 on Ohmnet, compared with 0.913, 0.935, and 0.746 for GO-Flow with $\alpha = 0$. UMLS does not admit the same construction because most unordered pairs are already training edges or evaluation candidates. These comparisons separate representation quality from the effect of reliable negative supervision; they are not intended to claim superiority over supervised link prediction in general.

M Contamination Details

This appendix supports Section 4.3. For the contamination experiment, one third of the training positives is hidden. A fraction $\gamma \in \{0, 0.05, 0.10, 0.20, 0.40\}$ of each fixed training-negative set is then replaced by hidden true edges. Runs use 10 seeds with nested contaminated subsets. Because GO-Flow does not train on sampled negatives, its training set is unchanged across contamination levels. We report the full Cora sweep in Table 14. Retention is ROC-AUC at $\gamma = 0.40$ divided by ROC-AUC at $\gamma = 0$, computed from seed means, so a value below 1 means that a method lost accuracy as its sampled negatives were replaced by hidden true edges; Table 15 reports it on all five graphs. The reference row is GO-Flow with each graph's selected configuration and $\alpha = 0$; with the PSE, its retention would also be 1, because the PSE is computed from observed positives only. At $\gamma = 0.40$, 1,017 of the 2,542 Cora training negatives are hidden true edges. PULL's own negative sampler gives 0.827 ± 0.014 at $\gamma = 0$, compared with 0.666 ± 0.021 using the frozen pool. We interpret the contamination experiment through each method's change from its own clean value rather than through absolute scores across different samplers. LPFormer samples its own training negatives, so in its runs the contaminated negatives are passed to that sampler, and every method trains on the same contaminated set.

We ran the same intervention on four further graphs so that the pattern does not rest on one citation network, and we report retention per method and graph in Table 15 because absolute scores are not comparable across methods that draw their negatives differently.

PULL on Ohmnet is the only case with retention above 1 (1.016), that is, the only cell whose ROC-AUC increases as sampled negatives are replaced by hidden true edges; with three seeds, we do not

Table 14: **False-negative contamination on Cora.** ROC-AUC ↑ at each contamination level γ, mean ± standard deviation over ten seeds, and retention; among methods trained with negatives, best in **bold** and second underlined.

Method ↓ / γ →	0	0.05	0.10	0.20	0.40	retention
Methods trained with negatives						
NCNC	**.933**$_{\pm.005}$	**.926**$_{\pm.006}$	**.922**$_{\pm.007}$	**.912**$_{\pm.007}$	**.891**$_{\pm.006}$	.954
BUDDY	.922$_{\pm.005}$	.914$_{\pm.005}$	.910$_{\pm.005}$	.902$_{\pm.004}$	.890$_{\pm.005}$	.966
LPFormer	.878$_{\pm.007}$	.868$_{\pm.008}$	.852$_{\pm.010}$	.838$_{\pm.010}$	.830$_{\pm.010}$	.946
SEAL	.750$_{\pm.019}$	.751$_{\pm.019}$	.744$_{\pm.015}$	.748$_{\pm.021}$	.733$_{\pm.016}$	**.977**
PULL (frozen negatives)	.666$_{\pm.021}$	.641$_{\pm.024}$	.631$_{\pm.020}$	.607$_{\pm.020}$	.574$_{\pm.011}$	.861
Ours, observed positives only						
GO-Flow (Ours), $\alpha = 0$	.881$_{\pm.005}$	.881$_{\pm.005}$	.881$_{\pm.005}$	.881$_{\pm.005}$	.881$_{\pm.005}$	1.000

Table 15: **Retention under contamination on five graphs.** Retention from seed means; among methods trained with negatives, best in **bold** and second underlined, ties underlined.

Method	Cora	CiteSeer	PPI	WN18RR	Ohmnet
Trained with negatives					
NCNC	0.954	0.976	**0.999**	**0.995**	0.997
BUDDY	0.966	0.976	0.937	0.938	0.991
LPFormer	0.946	0.945	0.925	0.938	0.961
SEAL	**0.977**	**0.983**	0.995	0.990	0.995
PULL (frozen)	0.861	0.822	0.990	0.949	**1.016**
Observed positives only					
GO-Flow (Ours), $\alpha = 0$	1.000	1.000	1.000	1.000	1.000

interpret this increase. WN18RR and Ohmnet use three seeds against ten for the other graphs, so their retention values are noisier.

N PULL Candidate Sets

This appendix supports Section 4.3. GO-Flow selects its configuration, including the fusion weight α, on PULL's own validation candidates. We report PULL and GO-Flow at both missing-edge ratios, with standard deviations over ten seeds, in Table 16. At $m = 0.20$, GO-Flow minus PULL is -0.0118 on Cora-full, $+0.0083$ on PubMed, approximately 0 on chameleon, $+0.0049$ on crocodile, and -0.0078 on Facebook.

Table 16: **GO-Flow is competitive with PULL at both missing-edge ratios.** ROC-AUC ↑, mean ± population standard deviation over ten seeds; the higher value on each graph is in **bold**.

Method	Cora-full	PubMed	chameleon	crocodile	Facebook
Missing-edge ratio $m = 0.10$					
PULL	**.9597**$_{\pm.0022}$	.9623$_{\pm.0030}$	**.9774**$_{\pm.0020}$	.9826$_{\pm.0011}$	**.9780**$_{\pm.0006}$
GO-Flow (Ours)	.9437$_{\pm.0027}$	**.9684**$_{\pm.0016}$	.9773$_{\pm.0024}$	**.9856**$_{\pm.0008}$	.9687$_{\pm.0007}$
Missing-edge ratio $m = 0.20$					
PULL	**.9548**$_{\pm.0024}$	.9583$_{\pm.0017}$	**.9760**$_{\pm.0013}$	.9800$_{\pm.0010}$	**.9764**$_{\pm.0004}$
GO-Flow (Ours)	.9429$_{\pm.0012}$	**.9666**$_{\pm.0014}$	.9759$_{\pm.0013}$	**.9849**$_{\pm.0005}$	.9685$_{\pm.0008}$

The three graphs of the multi-scale attributed node embedding collection (Rozemberczki et al., 2021), chameleon, crocodile, and Facebook, use raw features because PULL's preprocessed features were not available to us. The candidate sets are taken from PULL's own runs and reused

unchanged, so comparisons within a row are valid, while absolute values are not comparable to PULL's published table.

O HeaRT Hard-Negative Results

This appendix supports Section 4.3. We report MRR, Hits@10, and Hits@20 on the HeaRT splits in Table 17. The readout and the fusion weight $\alpha \in \{0, 0.25, 0.5, 0.75\}$ of GO-Flow are selected on HeaRT's validation pools, and validation selects $\alpha = 0$ in all nine runs, so the rows report the flow score. SEAL uses one-hop instead of three-hop subgraphs here, because the 500 hard negatives per positive multiply its per-candidate subgraph extraction and three hops exceeded 120 GB of memory; on PubMed (§) it also used per-batch extraction and checkpoint selection every fifth epoch to fit in 60 GB. Our reproduced LPFormer result on Cora is below the published value of 0.168 MRR

Table 17: **HeaRT hard-negative evaluation.** MRR, Hits@10, and Hits@20 ↑, mean ± population standard deviation over three seeds; best in **bold** and second underlined per graph and metric; "1-hop" and "§" mark the SEAL deviations described in the text.

Method	MRR	Hits@10	Hits@20
Cora			
LPFormer	**.1452**$_{\pm.0167}$	.3175$_{\pm.0063}$	.4497$_{\pm.0102}$
BUDDY	.1375$_{\pm.0005}$	**.3245**$_{\pm.0015}$	**.4775**$_{\pm.0009}$
NCNC	.1344$_{\pm.0036}$	.3213$_{\pm.0085}$	.4769$_{\pm.0065}$
SEAL (1-hop)	.0737$_{\pm.0051}$	.1638$_{\pm.0245}$	.2701$_{\pm.0258}$
GO-Flow (Ours), $\alpha = 0$	.1011$_{\pm.0035}$	.2441$_{\pm.0036}$	.3681$_{\pm.0031}$
CiteSeer			
LPFormer	**.2714**$_{\pm.0062}$	**.5201**$_{\pm.0180}$	**.6410**$_{\pm.0231}$
BUDDY	.2225$_{\pm.0003}$	.4769$_{\pm.0018}$	.6125$_{\pm.0010}$
NCNC	.1980$_{\pm.0029}$	.4586$_{\pm.0010}$	.6059$_{\pm.0045}$
SEAL (1-hop)	.0807$_{\pm.0103}$	.1729$_{\pm.0208}$	.2916$_{\pm.0265}$
GO-Flow (Ours), $\alpha = 0$	.1824$_{\pm.0091}$	.4044$_{\pm.0207}$	.5114$_{\pm.0122}$
PubMed			
LPFormer	**.0864**$_{\pm.0076}$	**.1868**$_{\pm.0145}$	**.3111**$_{\pm.0125}$
BUDDY	.0692$_{\pm.0004}$	.1360$_{\pm.0007}$	.2522$_{\pm.0001}$
NCNC	.0378$_{\pm.0019}$	.0748$_{\pm.0037}$	.1694$_{\pm.0117}$
SEAL (1-hop)§	.0536$_{\pm.0094}$	.1091$_{\pm.0203}$	.1760$_{\pm.0266}$
GO-Flow (Ours), $\alpha = 0$	.0458$_{\pm.0011}$	.1022$_{\pm.0050}$	.1995$_{\pm.0065}$

(Shomer et al., 2024), and our NCNC result on PubMed is below the 0.0858 reported there, so we use both only for comparison within our reproduced HeaRT setting. SEAL reaches 0.0536 ± 0.0094 MRR on PubMed; its mean is above the 0.0458 of GO-Flow while its seed spread covers it, so we do not claim an ordering for that cell. It is also the noisiest cell, with MRR 0.0599, 0.0402, and 0.0606 across seeds, consistent with its coarser checkpoint grid.

P Component and Encoder Studies

This appendix supports Section 4.4. We report the component and readout study in Table 18, the ablation of the flow and the PSE in Appendix P.1, and the encoder ablation in Appendix P.2.

P.1 Ablation of the Two Components

GO-Flow fuses the conditional flow with the PSE through the weight α of Equation (10), selected on validation from $\{0, 0.25, 0.5, 0.75\}$. We report the selected configuration, the selected configuration without the PSE ($\alpha = 0$), the selected configuration with it ($\alpha > 0$), and PSE-only, which removes the flow, for link prediction in Table 19 and for node PU ranking and DTI in Table 20. The selected configuration coincides with one of the two middle rows on every dataset.

Table 18: **Component and readout study.** ROC-AUC ↑, mean ± population standard deviation over three seeds; rows are separate configurations, grouped by the design choice that changes; $\lambda = 0$ in the construction, $w = 0$, and $w = 1$ rows, and $\lambda = 0.05$ in the others.

Configuration ↓ / Dataset →	Cora	UMLS	DISEASES	PPI	Drug–Disease
Relation construction					
Transport, joint	$.554_{\pm.007}$	$.259_{\pm.012}$	$.502_{\pm.019}$	$.724_{\pm.006}$	$.574_{\pm.008}$
Relative displacement, joint	$.750_{\pm.006}$	$.786_{\pm.046}$	$.860_{\pm.021}$	$.648_{\pm.012}$	$.561_{\pm.007}$
Transport, cross-endpoint	$.779_{\pm.005}$	$.890_{\pm.010}$	$.800_{\pm.003}$	$.506_{\pm.008}$	$.526_{\pm.008}$
Density readout					
Conditional log-density, $w = 0$	$.803_{\pm.005}$	$.876_{\pm.006}$	$.908_{\pm.005}$	$.736_{\pm.004}$	$.561_{\pm.010}$
Contrast, $w = 1$	$.911_{\pm.004}$	$.915_{\pm.015}$	$.915_{\pm.003}$	$.808_{\pm.006}$	$.622_{\pm.003}$
Contrast, selected w^*	$.911_{\pm.004}$	$.908_{\pm.007}$	$.912_{\pm.002}$	$.823_{\pm.002}$	$.640_{\pm.004}$
Training / auxiliary variants					
+ contrastive field term ($\lambda = 0.05$)	$.911_{\pm.004}$	$.919_{\pm.005}$	$.912_{\pm.002}$	$.814_{\pm.005}$	$.627_{\pm.001}$
+ PSE, $\alpha > 0$	$.894_{\pm.005}$	$.909_{\pm.008}$	$.913_{\pm.006}$	$.883_{\pm.001}$	$.797_{\pm.002}$

Table 19: **Flow and PSE ablation, link prediction.** ROC-AUC ↑, mean ± standard deviation over three held-out seeds; rows are the selected configuration, the selected configuration without the PSE ($\alpha = 0$) and with it ($\alpha > 0$), and PSE-only.

Configuration	Cora	CiteSeer	UMLS	WN18RR	DISEASES	Drug–Dis.	PPI	Ohmnet
GO-Flow (Ours) selected α	$.913_{\pm.005}$	$.935_{\pm.001}$	$.937_{\pm.004}$	$.751_{\pm.002}$	$.920_{\pm.009}$	$.797_{\pm.003}$	$.889_{\pm.000}$	$.894_{\pm.002}$
Without the PSE $\alpha = 0$	$.913_{\pm.005}$	$.935_{\pm.001}$	$.911_{\pm.007}$	$.656_{\pm.001}$	$.920_{\pm.009}$	$.639_{\pm.002}$	$.829_{\pm.002}$	$.746_{\pm.110}$
With the PSE selected $\alpha > 0$	$.908_{\pm.005}$	$.906_{\pm.004}$	$.937_{\pm.004}$	$.751_{\pm.002}$	$.922_{\pm.006}$	$.797_{\pm.003}$	$.889_{\pm.000}$	$.894_{\pm.002}$
Without the flow PSE-only	$.695_{\pm.004}$	$.592_{\pm.007}$	$.856_{\pm.003}$	$.735_{\pm.003}$	$.708_{\pm.011}$	$.800_{\pm.003}$	$.884_{\pm.001}$	$.870_{\pm.002}$

Table 20: **On node PU ranking the PSE adds to the flow, and on DTI the flow alone is selected.** ROC-AUC ↑, mean ± standard deviation over three held-out seeds, with the same rows as Table 19.

	Node PU ranking		DTI			
Configuration	Cora	CiteSeer	Enzyme	Ion channel	GPCR	Nuclear receptor
GO-Flow (Ours) selected α	$.925_{\pm.008}$	$.924_{\pm.004}$	$.968_{\pm.002}$	$.969_{\pm.001}$	$.956_{\pm.000}$	$.894_{\pm.004}$
Without the PSE $\alpha = 0$	$.916_{\pm.001}$	$.919_{\pm.006}$	$.968_{\pm.002}$	$.969_{\pm.001}$	$.956_{\pm.000}$	$.894_{\pm.004}$
With the PSE selected $\alpha > 0$	$.925_{\pm.008}$	$.924_{\pm.004}$	$.952_{\pm.002}$	$.964_{\pm.001}$	$.926_{\pm.000}$	$.873_{\pm.006}$
Without the flow PSE-only	$.868_{\pm.021}$	$.841_{\pm.012}$	$.798_{\pm.000}$	$.825_{\pm.000}$	$.661_{\pm.000}$	$.488_{\pm.008}$

P.2 Encoder Ablation

In Table 21, we compare the three encoders with the other selected settings fixed. The 0.097 gap between GAT and GCN on PPI shows that performance on PPI is sensitive to the encoder choice.

Table 21: **The encoder matters most on PPI, where GAT is 0.097 above GCN.** ROC-AUC ↑, mean ± population standard deviation over three seeds, with the other selected settings fixed; best in **bold**, second underlined, ties underlined.

Encoder	Cora	CiteSeer	UMLS	PPI
GCN	**.917**$_{\pm.003}$	.922$_{\pm.001}$	.908$_{\pm.003}$	.726$_{\pm.003}$
GAT	.907$_{\pm.006}$	**.937**$_{\pm.002}$	**.911**$_{\pm.007}$	**.823**$_{\pm.002}$
GIN	.838$_{\pm.011}$	.800$_{\pm.002}$	.908$_{\pm.007}$	.700$_{\pm.008}$

Q Conditioning Mode

This appendix supports Section 4.4. Without the PSE, cross-endpoint conditioning is selected on five of the eight standard link-prediction datasets, namely CiteSeer, Cora, DISEASES, UMLS, and WN18RR, while PPI, Drug–Disease, and Ohmnet use joint conditioning. Among configurations with $\alpha > 0$, cross-endpoint conditioning is selected on every graph except Drug–Disease. As we show with the component study in Table 18, the choice depends on the dataset: cross-endpoint conditioning raises the ROC-AUC of the transport readout on UMLS from 0.259 to 0.890 but lowers it on PPI from 0.724 to 0.506. Other conditioning results use different readouts or configurations, and we do not combine them with these numbers.

R Generative Quality versus Ranking

This appendix supports Section 4.4. The generative-quality experiment compares independent coupling, minibatch OT coupling, and one reflow pass on the same five graphs and three seeds. OT improves held-out positive log-likelihood on all five datasets, from $+4.7$ nats on PPI to $+13.6$ nats on CiteSeer. Candidate ROC-AUC changes by -0.009 on CiteSeer, -0.034 on PPI, -0.058 on DISEASES, -0.073 on UMLS, and $+0.003$ on Cora. Reflow remains within ± 2 nats in log-likelihood and ± 0.002 ROC-AUC. We express the comparison as paired changes relative to independent coupling and do not compare absolute log-likelihood and ROC-AUC values across couplings.

S Direct Estimators of the Contrast

This appendix supports Section 4.4. To test whether a conditional flow is needed to estimate the contrast, we train classifiers that distinguish matched positive target–context pairs from independently recombined positive pairs, which estimates the $w = 1$ contrast as a density ratio by classification (Sugiyama et al., 2012). Both sides of this binary problem are constructed only from observed positive relations, and no non-edge is introduced as a training example. The encoder, graph, node features, conditioning mode, and dataset-specific configuration are kept fixed relative to GO-Flow, and we report the results in Table 22.

Table 22: **Direct estimators of the contrast.** ROC-AUC ↑, mean ± population standard deviation over three held-out seeds; best in **bold**, second underlined; GO-Flow is shown without the PSE ($\alpha = 0$) because the direct estimators use positive relations only.

Estimator	Cora	CiteSeer	UMLS	WN18RR	DISEASES	Drug–Dis.	PPI	Ohmnet
Direct estimation								
Bilinear	.817$_{\pm.012}$	.805$_{\pm.009}$	**.985**$_{\pm.003}$	.671$_{\pm.003}$	.678$_{\pm.012}$	.632$_{\pm.012}$	.676$_{\pm.005}$	.617$_{\pm.009}$
MLP	.811$_{\pm.006}$	.918$_{\pm.004}$	.974$_{\pm.005}$	**.715**$_{\pm.002}$	.782$_{\pm.019}$	.628$_{\pm.009}$	.711$_{\pm.018}$	.635$_{\pm.017}$
Flow-based scoring								
GO-Flow (Ours), $\alpha = 0$	**.913**$_{\pm.005}$	**.935**$_{\pm.001}$	.911$_{\pm.007}$	.656$_{\pm.001}$	**.920**$_{\pm.009}$	**.639**$_{\pm.002}$	**.829**$_{\pm.002}$	**.746**$_{\pm.110}$

The direct estimators contain 8k to 104k parameters, fit in 11 to 54 s, and score 100,000 candidate pairs in 0.03 to 0.07 s, whereas GO-Flow requires 9.68 s for the same scoring batch. The conditional flow is therefore the stronger estimator on six of eight datasets, and direct estimation is a cheaper alternative that is higher on UMLS and WN18RR.

T Model Selection with Positive-Only Validation Statistics

This appendix supports Section 4.4. The main configurations are selected by validation ROC-AUC computed from validation positives and sampled non-edges. We also tested two criteria that use positive examples only: the mean validation-positive score and a standardized gap between training-positive and validation-positive scores. Re-selecting the standard link-prediction configurations with these criteria lowers mean test ROC-AUC by 0.065 without the PSE and 0.049 with it, with a 0.264 drop on PPI. Cora improves by 0.006, UMLS by 0.011, and Ohmnet by 0.034, and DISEASES is unchanged. A single positive-only criterion is difficult to use across readouts because their score scales differ considerably; on CiteSeer, three readouts have mean validation-positive scores of -117.3, 21.0, and -12.5. For this reason, we distinguish positive-only training from positive-only model selection in the main text.

U Class-as-Positive Discovery

This appendix supports Section 4.4. The class-as-positive study contains 45 tasks from eight graphs, namely the five heterophilous benchmarks of Platonov et al. (2024) (Roman-empire, Amazon-ratings, Minesweeper, Tolokers, and Questions) and Cora, CiteSeer, and PubMed (Sen et al., 2008), with four reveal ratios $\kappa \in \{0.05, 0.10, 0.20, 0.50\}$, the fraction of the class that is observed, 10 seeds, and 1,800 runs. The main metric is lift, the AUPRC divided by the positive rate of the evaluation set, with ROC-AUC as a secondary summary. We fix one recipe in advance for every task: the node configuration of Appendix J, scored by the conditional log-density ($w = 0$, the readout selected in node PU ranking) and rank-fused at $\alpha = 0.5$ with the node PSE, the k-hop walk counts into the observed positives. This is GO-Flow at a fixed fusion weight $\alpha = 0.5$. The construction differs from node PU ranking, which we report in Appendix J.

What labeled negatives add. To measure how much of the remaining gap is due to supervision, we train one supervised network per cell on the same observed positives plus labeled negatives that no other row in this study uses, reporting GCN and GAT separately and without encoder selection, for 1,080 cells over the same classes, reveal ratios, and three seeds. The comparison is paired over (graph, class, reveal ratio, seed). A supervised GCN gains 0.071 ROC-AUC and 2.21 lift on average and leads on 366 of 540 shared cells; GAT gains 0.059 and 1.56 and leads on 344. **The supervised network is an aggregate ceiling rather than a cell-by-cell one**: on 32% to 36% of cells, GO-Flow, which uses no negative labels, matches or exceeds it. The two metrics also move in opposite directions as more positives are observed. The ROC-AUC gain falls from 0.095 to 0.029 between $\kappa = 0.05$ and $\kappa = 0.50$, while the lift gain rises from 0.86 to 3.50, so clean negatives add global separation when positives are scarce and precision at the top of the ranking when they are plentiful. We report this network as an oracle. Its candidate set is smaller than that of the positive-only cells by the negatives it trains on, so lift, not AUPRC, is the comparable quantity.

What the unlabeled candidates add. GPL (Wu et al., 2024) is a graph PU method: it trains on the observed positives and the unlabeled nodes, with the observed-positive rate as its class prior. We ran our reimplementation, described in Appendix G.6, on all 1,800 cells and scored the same held-out and negative nodes. GPL exceeds chance AUPRC in 98.1% of cells, against 95.4% for GO-Flow. In paired comparison, GPL leads by 1.48 lift and 0.057 ROC-AUC on average, and GO-Flow matches or exceeds it on 33% of cells, close to its share against the negative-label oracle above. The gap lies mainly on the citation graphs, with -3.11 lift on Cora and -0.87 on CiteSeer, while the two are level on Amazon-ratings and Tolokers. On Roman-empire, GO-Flow with $\alpha = 0$ is 7.56 lift above GPL, while GO-Flow at $\alpha = 0.5$ is 2.10 below it, which reflects the fusion cost discussed below. GPL belongs to the PU group, so it measures the benefit of training on the unlabeled candidates, not where GO-Flow ranks among positive-only methods.

We report lift by dataset in Table 23. Across all runs, 95.4% exceed chance AUPRC. Fusion is not uniformly beneficial: on Cora, PSE-only reaches 6.67× chance against 3.48× for GO-Flow

Table 23: **Lift grows with the reveal ratio on the citation graphs and stays close to chance on most heterophilous benchmarks.** Mean lift of GO-Flow at $\alpha = 0.5$ over each dataset's classes and ten seeds, for four reveal ratios κ, followed by the mean ROC-AUC at the lowest and highest κ.

Dataset	$\kappa = 0.05$	0.10	0.20	0.50	ROC-AUC, 0.05	ROC-AUC, 0.50
Citation graphs						
Cora	1.9	2.1	2.8	7.2	.624	.880
CiteSeer	1.5	1.7	2.4	5.6	.534	.852
PubMed	2.5	2.7	3.0	4.0	.857	.917
Heterophilous benchmarks						
Roman-empire	5.6	5.3	5.0	4.7	.705	.803
Amazon-ratings	1.4	1.5	1.8	2.0	.569	.659
Questions	1.3	1.4	1.5	1.6	.523	.554
Tolokers	1.2	1.3	1.3	1.4	.560	.565
Minesweeper	1.1	1.1	1.2	1.2	.572	.597

at $\alpha = 0.5$ and is higher in 88% of the 280 Cora runs, while on Roman-empire GO-Flow with $\alpha = 0$ reaches $14.83\times$, PSE-only $1.06\times$, and GO-Flow at $\alpha = 0.5$ reaches $5.17\times$. Pooled feature separability correlates with the reported lift at Spearman $\rho = 0.822$, compared with $\rho = 0.180$ for adjusted homophily (Platonov et al., 2023). The pooled correlation between the PSE gain and adjusted homophily is high ($\rho = 0.809$) but weak within the one graph with enough classes to test it, as we show in Appendix W, so we do not interpret it as a within-graph effect.

V Dataset Controls on the Recoverability Claim

This appendix supports Section 4.4. The recoverability correlation in the main text pools 45 class-as-positive tasks from eight graphs. A pooled correlation over eight datasets does not distinguish a property that predicts recovery from a between-dataset difference that happens to co-vary with it. We therefore apply three controls to the 1,800 individual runs behind these tasks, with no new runs. At this run level the pooled correlations with lift are 0.686 for feature separability, 0.187 for label informativeness, and 0.010 for adjusted homophily; the task-level values of the main text average each task's 40 runs.

Within-dataset correlations. Computed separately inside each graph, feature separability stays positive on all five graphs with at least three classes, and is largest on Roman-empire ($+0.657$, 18 classes) and PubMed ($+0.782$). Adjusted homophily (Platonov et al., 2023) behaves differently: $+0.158$ within CiteSeer, $+0.201$ within Roman-empire, -0.100 within Amazon-ratings and -0.546 within PubMed, while its pooled value over the runs is $+0.010$.

Leave-one-dataset-out. Refitting the pooled correlation eight times, each time holding out one graph, feature separability stays in $[+0.616, +0.740]$ so no single dataset drives it. Adjusted homophily ranges over $[-0.099, +0.608]$ and label informativeness over $[+0.098, +0.627]$; both are therefore statements about which graphs are in the pool.

Dataset fixed effects. To control for dataset membership and the other diagnostics simultaneously, we regress the standardized lift of GO-Flow with $\alpha = 0$ on all five standardized diagnostics plus dataset indicators, with standard errors clustered by class (45 clusters, CR1 finite-sample correction). Feature separability has the largest and most precisely estimated coefficient, $\beta = +1.419$ with t-statistic 3.63. Adjusted homophily has $\beta = +1.093$ with t-statistic 1.83 and is not distinguishable from zero. Label informativeness has a pooled run-level correlation of $+0.187$ but a fitted coefficient of -1.007 with t-statistic -2.11: its sign flips once graph identity is controlled for, the pattern expected of a between-dataset artifact. The two size variables are small and negative: mean degree within the class has $\beta = -0.265$ with t-statistic -2.08, and class prevalence has $\beta = -0.308$ with t-statistic -2.47, so denser and more common classes are marginally harder to recover once the other diagnostics are held fixed.

Summary. Feature separability is both the largest pooled correlate and the one that holds under all three controls, and we report the adjusted homophily comparison as a within-graph observation only.

W Which Component Helps, and Where

This appendix supports Section 4.4. The class-as-positive study records GO-Flow at $\alpha = 0$ and $\alpha = 0.5$ and PSE-only, which lets us compare the contributions of the flow and the PSE. We define, per class and in lift units, the PSE gain as the lift of PSE-only minus the lift of GO-Flow with $\alpha = 0$. Pooled over the 45 classes, it correlates with adjusted homophily at $\rho = +0.809$ and with label informativeness at $+0.631$, and negatively with feature separability at -0.332; the cost of fusing is largest where features are strong ($\rho = -0.695$). Within graphs, however, the association does not hold: the correlation is $+0.106$ over the 18 classes of Roman-empire and -0.071 over the seven of Cora, while CiteSeer ($+0.714$) and Amazon-ratings ($+0.700$) have only six and five classes, too few for a within-graph estimate. The pooled association is therefore largely between datasets, and it should not be used to select the fusion weight per class.

X Runtime Measurements

This appendix supports Section 4.4. We time one scoring call on Cora as the number of candidates grows, for GO-Flow and for the sampled-negative baselines, each timed at its own prediction step so that both sides of the comparison are measured, and we report the medians of three repeats in Table 24. BUDDY, LPFormer, and SEAL scale linearly with the candidate set, whereas NCNC and GO-Flow have a fixed cost followed by linear growth; the fixed cost of GO-Flow, about 1.34 s per call, dominates up to roughly 10k candidates. At 1k candidates GO-Flow has the highest cost of the five; past its knee it is cheaper than SEAL beyond about $7{,}400$ candidates and more expensive than LPFormer at all sizes. The comparison is not fully symmetric: the SEAL time is a lower bound because it covers the forward pass over already extracted subgraphs, and the NCNC encoder forward pass (0.4 ms) lies outside its timed region, whereas the GO-Flow time includes all computation, which inflates its constant. In Appendix Y, we vary the scoring budget itself and find that the default budget is about four times larger than the knee of the quality–cost curve on six of the eight graphs.

Table 24: **Scoring time on Cora.** Seconds per scoring call, median of three repeats; the last column is the cost of $100\times$ more candidates. ‡SEAL covers only the forward pass over already extracted subgraphs, so it is a lower bound.

Method	1k	5k	10k	50k	100k	$100\times$ cost
NCNC	0.002	0.002	0.002	0.003	0.006	$3.5\times$
BUDDY	0.006	0.024	0.045	0.354	0.556	$94.5\times$
LPFormer	0.026	0.123	0.238	1.181	2.372	$92.6\times$
SEAL‡	0.186	0.913	1.819	9.065	18.219	$97.9\times$
GO-Flow (Ours)	1.342	1.344	1.373	4.863	9.676	$7.2\times$

Y Scoring Budget and the Quality–Cost Frontier

This appendix supports Sections 3.2 and 4.4. The scoring cost is set by a Monte Carlo budget chosen at evaluation time, so it can be varied without retraining. We re-score the same candidate sets with the same trained ensembles at each setting, so the comparison is paired: the same weights and candidates, with only the budget changing. These runs use the search seed, which is not one of the reported evaluation seeds, so the ROC-AUC values below describe the shape of the frontier rather than the values reported elsewhere.

The relevant budget depends on the selected readout. The residual readouts, selected on CiteSeer and WN18RR, spend their time on the number of time samples K, default 32, whereas the other six graphs select s_w, which is built from the conditional log-density and is governed by the number of integration steps of the change-of-variables solver, default 20.

Residual readouts. The residual readouts avoid integrating Equation (7). They use the per-example value of Equation (2) as a transport residual,

$$R(z_1, c) = \mathbb{E}_{z_0,t}\left[\|v_\theta(z_t, t, c) - (z_1 - z_0)\|_2^2\right], \tag{11}$$

estimated with K independent draws of (z_0, t); it is small when the velocity field already transports noise onto z_1 under context c. The transport-cost readout scores $-R(z_1, c)$, and two contrasts play the role that Equation (8) plays for the log-density,

$$r^{\mathrm{tr}}(z_1, c) = R(z_1, \varnothing) - R(z_1, c), \qquad r^{\mathrm{rel}}(z_1, c) = R(z_1, \tilde{c}) - R(z_1, c), \tag{12}$$

for the transport-contrast and relative-cost readouts, where $\tilde{c}$ is the context of another candidate in the same batch rather than a sampled non-edge. The first asks whether the relation context helps at all, the second whether this context helps more than an unrelated one. Both are combined across endpoints as in Equation (9).

Table 25: **Integration-step budget.** ROC-AUC at each number of integration steps for the six graphs whose readout integrates Equation (7); the default of 20 steps is in bold; single seed (the search seed).

Dataset	Readout	2	5	10	**20**	40
Cora	$s_{w=1}$	.8952	.9130	.9130	**.9132**	.9129
UMLS	$s_{w=0.75}$	.8462	.8807	.8891	**.8954**	.8994
DISEASES	$s_{w=0.75}$	.9117	.9208	.9220	**.9263**	.9263
PPI	$s_{w=0.75}$	.7825	.8116	.8189	**.8209**	.8221
Drug–Disease	$s_{w=0.75}$	.5732	.6185	.6368	**.6458**	.6502
Ohmnet	$s_{w=1}$	.8324	.8259	.7962	**.7851**	.7810

Table 26: **Time-sample budget of the residual readouts.** ROC-AUC at each number of time samples K for the two graphs that select a residual readout; the default $K = 32$ is in bold; single seed (the search seed).

Dataset	Readout	2	4	8	16	**32**	64
CiteSeer	transport contrast	.9072	.9179	.9307	.9340	**.9356**	.9406
WN18RR	relative cost	.6519	.6524	.6533	.6556	**.6607**	.6600

We report the integration-step sweep in Table 25 and the time-sample sweep in Table 26. The residual readouts change by at most 0.007 ROC-AUC beyond $K = 16$ time samples. Three observations follow. First, the default budget is about $4\times$ larger than the knee on the likelihood path: at 5 steps the six graphs run 3.3 to $4.0\times$ faster for a median ROC-AUC cost of 0.007, and on Cora the cost is 0.0002. Below 5 steps the solver degrades on every graph, by 0.018 on Cora and 0.073 on Drug–Disease. Second, two graphs are slightly under-budgeted: CiteSeer gains 0.0050 at $K = 64$ and Drug–Disease gains 0.0044 at 40 steps, so the default has not converged there. Third, Ohmnet inverts: its curve decreases monotonically in the budget, and the cheapest setting is its most accurate by 0.047. Ohmnet selects $s_{w=1}$, the full contrast, in which both terms are integrated, so their errors may partially cancel; we report this as a single-graph observation, not as a mechanism. Peak memory is unaffected by either budget on all eight graphs and varies only with the candidate-set size, from 79 MB at 1,366 candidates to 778 MB at 26,853, because both budgets are accumulated and not materialized, so only compute time is traded here.