GRAPHSTEER: SOFT EDGE REWEIGHTING VIA INFORMATIVE LEIDEN SUBGRAPHS FOR GRAPH OUT-OF-DISTRIBUTION GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks often exploit environment-specific correlations that do not remain stable under distribution shifts, leading to poor out-of-distribution (OOD) generalization. Existing graph OOD methods commonly rely on identifying invariant subgraphs or suppressing spurious structures, but such approaches may discard useful information when the distinction between relevant and irrelevant graph components is imperfect. We propose GraphSteer, a soft structural reweighting framework that partitions graphs into candidate subgraphs using Leiden community detection, which we extend to incorporate node-feature information. GraphSteer estimates subgraph informativeness from a subgraph classifier's predictive confidence and uses this information to reweight graph edges while preserving the full graph structure for downstream message passing. This soft structural intervention avoids irreversible structural deletion when subgraph identification is imperfect, while the resulting communities and informativeness scores provide an inspectable view of how GraphSteer emphasizes different graph regions under *covariate* and *concept* shifts. Unlike many existing methods evaluated on specific graph domains or shift settings, GraphSteer is evaluated across image-derived, molecular, and synthetic graphs under both covariate and concept shifts, without requiring ground-truth environment labels. Experiments on GOOD-CMNIST, GOOD-HIV, and GOOD-Motif demonstrate competitive and robust OOD performance across these settings.

1 INTRODUCTION

Graph neural networks (GNNs) have achieved strong performance across graph learning tasks (Kipf & Welling, 2017; Veličković et al., 2018; Xu et al., 2019). However, their performance can degrade when training and test data follow different distributions, resulting in an *out-of-distribution* (OOD) setting (Li et al., 2025; Ju et al., 2026; Zhang et al., 2026). Graph OOD generalization aims to learn predictors that remain reliable under such distribution changes. Unlike Euclidean inputs, graphs jointly encode attributes and relational structure, so distribution shifts may involve node attributes, graph topology, or their associations with the prediction target (Li et al., 2025; Ju et al., 2026). Under such shifts, models may exploit correlations that are predictive in the training distribution but fail to generalize to unseen environments. This behaviour is closely related to shortcut learning, where models rely on predictive but non-essential information whose association with the target may change across environments (Kim et al., 2019; Bahng et al., 2020).

Graph OOD generalization is often framed as a problem of identifying the graph structure that should remain stable across environments. *Invariant-subgraph* methods follow this intuition by searching for subgraphs that exclude structures associated with spurious correlations (e.g. Fan et al. (2022)). However, this framing imposes a hard decision on an inherently uncertain problem: which parts of a graph should be retained, and which should be discarded? When subgraph identification is imperfect, useful edges may be removed together with spurious ones, resulting in only partial preservation of the invariant structure (Yao et al., 2025).

An alternative is to treat graph structure not as something to select or discard, but as something to steer. Rather than asking which subgraph is invariant, we ask *which parts of the graph should have*

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.

greater influence on message passing under distribution shift. This motivates **GraphSteer**, a graph OOD framework that preserves the original graph topology while softly reweighting message passing according to estimated subgraph informativeness. Instead of removing graph structure outside a selected subgraph, GraphSteer retains the complete graph and modulates the contribution of its edges during downstream GNN training.

After decomposing a given graph into subgraphs using Leiden community detection (Traag et al., 2019), we use the most informative subgraph to determine edge-weight assignments on the original graph, which is subsequently processed by a downstream Graph Isomorphism Network (GIN) and its edge-feature extension (GINE) model. This soft reconstruction provides not only a mechanism for controlling message propagation, but also an interpretable view of which graph regions are considered informative, allowing us to examine how these structural preferences differ between *covariate* and *concept* shifts. An overview of the GraphSteer framework can be found in Figure 1.

Taken together, these ideas lead to three main contributions:

- We introduce **GraphSteer**, a graph OOD framework that converts confidence-scored subgraphs into edge-level message-passing weights while preserving the original graph topology and without requiring ground-truth environment labels.
- We show how visualizing the scored subgraphs provides insights into which graph structures GraphSteer relies on under different OOD scenarios.
- We systematically evaluate GraphSteer across covariate and concept shifts on GOOD-CMNIST, GOOD-HIV, and GOOD-Motif, showing that its effectiveness varies across datasets and distribution-shift settings.

2 Related Work

We briefly review approaches to distribution generalization, with particular emphasis on methods that address distribution shifts in graph-structured data. We organize existing graph OOD approaches according to how they identify, modify, or preserve graph information, which highlights the distinction between existing hard structural interventions and our soft reweighting strategy.

General Domain-Generalization. Domain generalization aims to learn predictors that remain effective when evaluated on unseen distributions. Representative approaches include invariant risk minimization (IRM) (Arjovsky et al., 2019), Variance Risk Extrapolation (VREx) (Krueger et al., 2021), GroupDRO (Sagawa et al., 2020), domain-adversarial training DANN (Ganin et al., 2016), Deep CORAL (Sun & Saenko, 2016), and Mixup (Zhang et al., 2018). These methods encourage robustness through invariant representation, distribution alignment, worst-group optimization, adversarial domain confusion, or interpolation between training examples. However, they generally operate on vector representations or assume structures that do not explicitly capture the relational nature of graph data. Graph-specific architectural approaches such as ASAP (Ranjan et al., 2020) instead exploit graph structure through adaptive structure-aware pooling, but do not explicitly target distribution shifts or distinguish stable from environment-dependent graph.

Graph Invariant Learning. A major line of graph OOD research seeks to identify graph representations or substructures whose predictive relationship remains stable across environments (Gui et al., 2022; Li et al., 2025; Zhang et al., 2026). DIR (Wu et al., 2022), CAL (Sui et al., 2022), GSAT (Miao et al., 2022), CIGA (Chen et al., 2022), GALA (Chen et al., 2023), and LECI (Gui et al., 2023) learn causal, invariant, or stable graph representations and substructures to reduce reliance on environment-specific correlations. OOD-GNN (Li et al., 2022) and StableGNN (Fan et al., 2024) similarly seek representation that are less sensitive to distributional variation. For molecular graphs, MoleOOD (Yang et al., 2022) and iMoLD (Zhuang et al., 2023) explicitly target invariant molecular representations under distribution shifts, while CSIB (An et al., 2024) combines causal-subgraph learning with an information-bottleneck objective for OOD robustness.

Although these approaches differ substantially in how invariance is defined and learned, many ultimately rely on identifying a subset of graph information that should be retained while reducing the influence of the remainder. This introduces an important structural question: when the distinction between invariant and spurious components is uncertain, should graph information be selected discretely, or should its contribution instead be modulated continuously? GraphSteer adopts the latter perspective.

Pruning-Based Graph Methods. Instead of identifying one invariant subgraph, PrunE (Yao et al., 2025) takes the approach of pruning potentially spurious edges throughout the graph. The same question of making hard cuts versus applying soft reweighting applies here.

Disentanglement-Based Graph Learning. Disentanglement-based approaches explicitly factor graph information into components with different predictive roles. DisC (Fan et al., 2022), for example, separates causal and bias-related substructures, while CFD (Wu & Deng, 2025) disentangles predictive concepts from spurious factors for molecular OOD generalization. By separating useful and nuisance information, these methods provide a principled mechanism for reducing reliance on biased graph components. However, disentanglement still requires the learned components to be sufficiently separable for downstream selection or representation learning.

Graph Data Augmentation. Another family of methods improves robustness by exposing GNNs to perturbed or synthetically generated training graphs. DropEdge (Rong et al., 2020), GraphCL (You et al., 2020), FLAG (Kong et al., 2022), graph Mixup (Wang et al., 2021), G-Mixup (Han et al., 2022), GREA (Liu et al., 2022), LiSA (Yu et al., 2023), AIA (Sui et al., 2023), FeatX (Li et al., 2024), and MPAIACL (Zeng & Gan, 2026) modify graph structure or node features to generate diverse training examples and improve generalization. These methods demonstrate that controlled structural perturbations can improve robustness, but the modified graphs primarily serve as additional training instances.

3 METHOD

In this section, we present our method. We first set up the OOD-problem formally, then introduce the subgraph discovery that will be used to find candidates for the invariant subgraph. Then we describe how the candidate subgraphs are scored with a confidence score from a classifier that was trained on the subgraphs in the training data using inherited graph-level labels. The confidence score itself does not require labels and can be performed on the test set as well. Finally, we show how the highest-ranked subgraph determines the edge-weight assignment on the original graph.

3.1 OOD PROBLEM SETTING

Let $G = (V, E, X^V, X^E)$ denote an input graph, where V and E are the node and edge sets, $X^V \in \mathbb{R}^{|V|\times d}$ denotes the node features and $X^E \in \mathbb{R}^{|E|\times d_e}$ the edge features. Let Y denote the *graph-level prediction target*.

Following Gui et al. (2022), we distinguish between covariate and concept shifts through

$$P(G, Y) = P(Y \mid G)P(G). \tag{1}$$

Let P_{tr} and P_{te} refer to the training and the test distribution respectively. *Covariate shift* changes the graph-input distribution while preserving the label mechanism,

$$P_{\text{tr}}(G) \neq P_{\text{te}}(G), \qquad P_{\text{tr}}(Y \mid G) = P_{\text{te}}(Y \mid G), \tag{2}$$

whereas *concept shift* changes the input-label relationship,

$$P_{\text{tr}}(G) = P_{\text{te}}(G), \qquad P_{\text{tr}}(Y \mid G) \neq P_{\text{te}}(Y \mid G). \tag{3}$$

Under either type of shift, subgraphs that are predictive in the training distribution may not be in an unseen environment. Rather than assuming that the most predictive subgraph is strictly invariant and removing the remaining structure, GraphSteer uses estimated subgraph informativeness to modulate message contributions while preserving the complete graph. Thus, we construct a weighted graph

$$\widetilde{G} = (V, E, X^V, X^E, \omega), \tag{4}$$

where $\omega : E \to \mathbb{R}_{\geq 0}$ assigns a weight to each edge and controls the strength of its contribution during downstream message passing. The original node and edge sets are preserved.

3.2 CANDIDATE SUBGRAPH DISCOVERY

We first identify candidate subgraphs from each input graph using feature-aware Leiden community detection (Traag et al., 2019). We use Leiden because it produces *well-connected communities* and

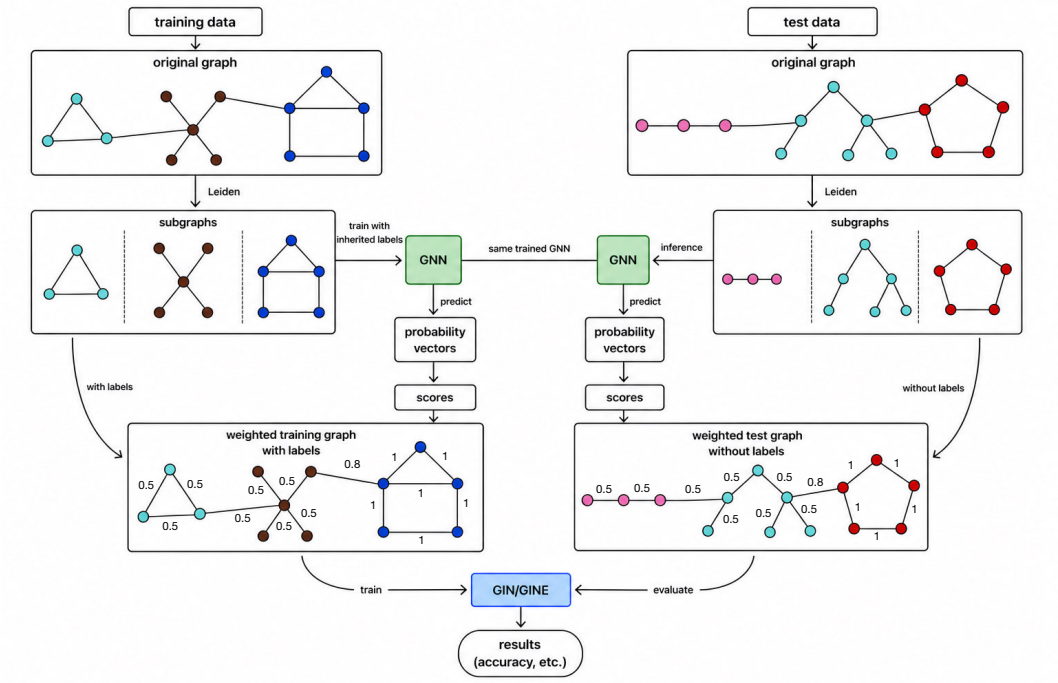

Figure 1: Overview of the GraphSteer framework. Training and test graphs are partitioned into candidate subgraphs using Leiden community detection. A subgraph classifier is trained using candidate subgraphs from the training split with inherited graph-level labels and is then reused to infer class-probability vectors for candidate subgraphs from both training and test graphs. Entropy or margin scores are computed from these probabilities to select the highest-ranked subgraph. The selected subgraph is mapped back to the original graph to assign different message-passing weights within the selected subgraph, across its boundary, and outside the selected subgraph. The resulting weighted graphs are used to train and evaluate the downstream GIN/GINE model.

can incorporate edge weights during community detection, making it suitable for combining graph connectivity with node-feature similarity. Inspired by Gaussian similarity-based graph weighting (Shi & Malik, 2000; Zhang et al., 2011), we make the Leiden partition feature-aware by assigning each existing edge $(i, j) \in E$ a similarity weight based on its endpoint node features:

$$S_{ij} = \exp\left(-\tfrac{1}{2\sigma^2}\left\|x_i^V - x_j^V\right\|_2^2\right), \tag{5}$$

where x_i^V and x_j^V denote the node-feature vectors and σ controls the similarity scale. The Gaussian similarity provides a smooth and bounded measure of feature agreement between connected nodes, with similarity values in $(0, 1]$. Edges between feature-similar nodes receive weights closer to 1, whereas edges between increasingly dissimilar nodes receive weights closer to 0.

These similarity weights are used during Leiden's weighted modularity optimization. This allows the resulting communities to reflect both structural connectivity and node-feature similarity. The resulting partition is a disjoint union of node sets:

$$V = C_1 \cup \cdots \cup C_M, \qquad C_i \cap C_j = \varnothing \quad \text{for } i \neq j, \tag{6}$$

where each community C_m induces a candidate subgraph G_m:

$$G_m = (C_m, E_m, X_m^V, X_m^E), \tag{7}$$

$$\text{where } E_m = \{(u, v) \in E \,|\, u, v \in C_m\},\ X_m^V = \{X_v^V | v \in C_m\},\ X_m^E = \{X_v^E | v \in C_m\}.$$

These candidate subgraphs are subsequently evaluated according to their predictive informativeness. Importantly, GraphSteer does not require the selected community to exactly correspond to an invariant subgraph, since the subsequent soft edge reweighting retains information from the remaining graph structure.

Table 1: Predictive-confidence scores used to estimate candidate-subgraph informativeness.

Score	Definition	Interpretation
Entropy-based confidence	$1 - \frac{-\sum_{c=1}^{C} p_{m,c} \log p_{m,c}}{\log C}$	Deviation from uniform prediction
Margin	$p_{m,(1)} - p_{m,(2)}$	Separation between top classes

3.3 Subgraph Informativeness Estimation

The candidate subgraphs G_m inherit the label Y of their parent graph and are used to train a subgraph classifier, f, which predicts Y from G_m. The subgraph classifier, f, is a GNN encoder followed by a linear classification head and is trained on the candidate subgraphs with the inherited labels using cross-entropy loss. After training, the classifier produces a class-probability vector p_m for each candidate subgraph: $p_m = f(G_m)$. This probability vector is then used to estimate subgraph informativeness through a label-free predictive confidence measure. The intuition is that, if the classifier is approximately well-calibrated, its output distribution provides information about how informative an input subgraph is for the prediction target. We therefore use the predicted probability distribution rather than the probability assigned to the true label, since the true label is unavailable for test samples, where GraphSteer must identify the most informative subgraphs before applying the learned edge reweighting.

In the covariate shift setting, Eq. (2), we expect subgraph informativeness to closely track which subgraph carries the invariant information: under covariate shift, whatever is predictive in the training set is also predictive in the test set. For the concept shift setting, Eq. (3), there is, however, a chance that subgraph informativeness misleads us: whatever is predictive in the training set is not necessarily predictive in the test set. It is for this reason that it is important to not fully prune the non-selected subgraphs, but rather to only weight the edges.

To measure informativeness, we consider two confidence measures. The entropy-based score measures how far the predicted distribution is from an uninformative uniform distribution, while the margin measures the separation between the two most probable classes. They are summarized in Table 1. For Margin, $p_{m,(1)}$ and $p_{m,(2)}$ denote the largest and second-largest class probabilities, respectively.

3.4 Soft Edge Reweighting

Having selected the candidate subgraph with the highest informativeness score, let G^* denote this selected subgraph and V^* its node set. The selection of a subgraph induces three distinct edge categories in the original graph: edges entirely within the selected subgraph (*inner edges*), edges crossing between selected and non-selected nodes (*boundary edges*), and edges entirely outside the selected subgraph (*outer edges*). We therefore introduce three weights, (w_I, w_B, w_O) — **i**nner, **b**oundary, **o**uter — so that GraphSteer can control the contribution of each of these edge categories independently during message passing. This distinction allows the model to emphasize information within the selected subgraph while separately controlling interactions across its boundary and information propagated through the remaining graph structure. Each original edge $(u, v) \in E$ is assigned one of three weights:

$$\omega_{uv} = \begin{cases} w_I, & \text{if } u \in V^* \text{ and } v \in V^*, \\ w_B, & \text{if } u \in V^* \text{ xor } v \in V^*, \\ w_O, & \text{otherwise.} \end{cases} \tag{8}$$

Here, w_I is assigned to edges whose endpoints both belong to the selected subgraph, w_B to edges connecting selected and non-selected nodes, and w_O to edges whose endpoints are both outside the selected subgraph. Thus, subgraph-level informativeness is translated into edge-level message strengths. We enforce $w_I \geq w_B \geq w_O$. The idea is that information passed within the most informative subgraph is of the highest value, information passed across the boundary is of lower value and information in the outer graph is of lowest value. We normalize the weights such that $w_I = 1$ and therefore only have two degrees of freedom.

The resulting weighted graph $\widetilde{G}$ is then used by the downstream GNN. For a GIN-style model (Xu et al., 2019), the assigned edge weight directly scales each neighbouring message:

$$h_i^{(l+1)} = \mathrm{MLP}^{(l)}\left((1+\varepsilon^{(l)})h_i^{(l)} + \sum\nolimits_{j\in\mathcal{N}(i)} \omega_{ji} h_j^{(l)}\right) \tag{9}$$

where MLP is a multi-layer perceptron, $\varepsilon^{(l)}$ controls the contribution of node i's own representation at layer l and $\mathcal{N}(i)$ denotes the set of neighbours of node i.

The weighting scheme also includes two interesting limiting cases. When $(w_I, w_B, w_O) = (1, 1, 1)$, all edges retain their original message strength, and GraphSteer reduces to the corresponding unweighted downstream GNN. When one of the weights is set to zero, messages along the corresponding edge category are completely suppressed. For example, $(1, 0, 0)$ retains message propagation only within the selected subgraph while suppressing boundary and outside-subgraph edges. However, note that isolated nodes (i.e., nodes with degree 0) still contribute to the output of the network under GIN via node pooling. Therefore, it is not the case that $(1, 0, 0)$ is equivalent to training only on the selected subgraph.

4 Experiments

We evaluate GraphSteer across diverse graph domains and distribution-shift settings. We first describe the datasets and experimental setup, followed by comparisons with existing methods and an analysis of the proposed soft structural reweighting mechanism.

4.1 Datasets

We evaluate GraphSteer on three graph-classification datasets from the GOOD benchmark (Gui et al., 2022): GOOD-CMNIST, GOOD-HIV, and GOOD-Motif. For each dataset, we consider both covariate and concept shifts using the corresponding GOOD domain definitions. GOOD-CMNIST is evaluated under colour shifts, GOOD-HIV under scaffold shifts, and GOOD-Motif under base shifts. These domains are selected to provide complementary distribution-shift settings across the three datasets and to evaluate GraphSteer under different graph characteristics. For each domain and shift setting, the data are divided into five splits: training, in-distribution (ID)-validation, out-of-distribution (OOD)-validation, ID-test, and OOD-test. The ID-validation and ID-test sets follow the same distribution as the training data, whereas the OOD-validation and OOD-test sets are drawn from shifted distributions defined by the corresponding GOOD setting. Classification accuracy is used for GOOD-CMNIST and GOOD-Motif, whereas GOOD-HIV is evaluated using ROC-AUC. See Appendix B for a more detailed description of the datasets, including how the covariate and concept shift versions are constructed.

4.2 Experimental Setup

Community Detection. GOOD-CMNIST and GOOD-HIV use node-feature-aware Leiden, while GOOD-Motif uses the original topology-based Leiden formulation, as there are no node features.

Subgraph Classifier. We use a two-layer GCN to estimate subgraph informativeness, as it provides a simple and general architecture.

GNN Architecture. For the final graph classification task, we use GIN-based architectures following the backbone choices in the GOOD benchmark, while configuring the architecture separately for each dataset. GOOD-CMNIST uses a 5-layer GIN with virtual nodes, GOOD-HIV uses a 2-layer GIN with virtual nodes, and GOOD-Motif uses a 5-layer GIN.

Detailed implementation settings and hyperparameters are provided in Appendix D. Pseudo-code for the algorithm can be found in Appendix C.

Validation-Based Model Selection. We select the edge-weight configuration based on validation performance. Let $R(\boldsymbol{w})$ denote the validation performance of $\boldsymbol{w} = (w_I, w_B, w_O)$. Rather than selecting only the best-performing configuration, we construct the near-optimal set

$$\mathcal{W}_\epsilon = \left\{\boldsymbol{w} \in [0,1]^3 \;\middle|\; R(\boldsymbol{w}) \geq R^* - \epsilon\right\}, \quad R^* = \max_{\boldsymbol{w}} R(\boldsymbol{w}). \tag{10}$$

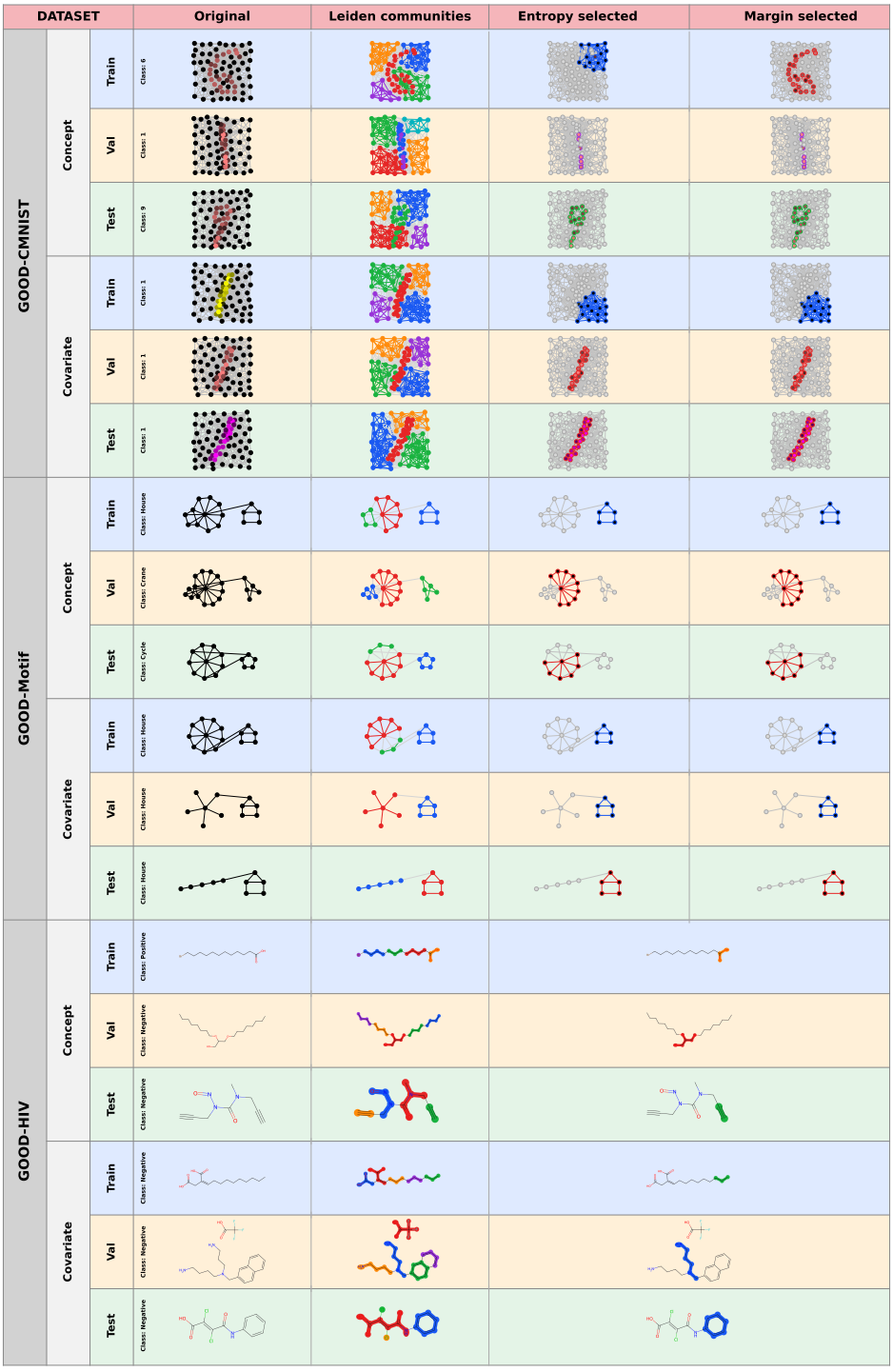

Figure 2: Representative examples of Leiden community discovery and confidence-based subgraph selection on GOOD-Motif, GOOD-CMNIST, and GOOD-HIV under concept and covariate shifts. From left to right, each row shows the original graph, the Leiden community partition, the subgraph selected using Entropy and Margin scoring. Under concept shift, the association between the graph pattern and the class label changes across data splits, such as association between digit colour and digit class in GOOD-CMNIST, between base structure and motif class in GOOD-Motif, and between molecular scaffold and activity label in GOOD-HIV. Under covariate shifts, the labels are preserved while graph characteristics change across data splits, including digit colours in GOOD-CMNIST, base structure in GOOD-Motif, and molecular scaffolds in GOOD-HIV. Non-selected graph component are faded for visualization only and are retained by GraphSteer.

Among near-optimal configurations, we prefer weights that more clearly differentiate the three edge categories, since nearly identical weights reduce the reweighting toward the no-reweighting case. We therefore select

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}\in\mathcal{W}_\epsilon} S(\boldsymbol{w}), \qquad S(\boldsymbol{w}) = (w_I - w_B)^2 + (w_B - w_O)^2. \tag{11}$$

Thus, validation performance remains the primary criterion, while $S(\boldsymbol{w})$ is used only to distinguish among near-optimal configurations. We use $\epsilon = 0.05$ percentage points for the experiments. As our sensitivity analysis for $\epsilon \in [0.05, 0.30]$ in Appendix F shows, the result is not sensitive to the exact value of ϵ. We consider both ID- and OOD-validation selection, where the weight configuration and training checkpoint are selected according to the corresponding validation performance. The test set is used only for final evaluation.

4.3 Results

We first compare the two validation criteria used for model selection. Table 2 summarizes the GraphSteer configurations selected under ID- and OOD-validation, together with their corresponding ID- and OOD-test performance.

The results show that the choice of validation criterion can lead to different selected edge-weight configurations and different OOD-test outcomes.

Table 2: Validation-based model selection with $\epsilon = 0.05$ percentage points.

Data	Shift	Sel.	Score	w^*	Val	ID Test	OOD Test
GOOD-CMNIST	Cov.	ID	Entropy	$(1, .9, .9)$	$77.51 \pm .30$	$77.35 \pm .12$	28.84 ± 4.48
		OOD	Entropy	$(1, .8, .2)$	79.21 ± 1.41	71.39 ± 2.12	38.16 ± 4.10
GOOD-CMNIST	Con.	ID	Margin	$(1, .7, .3)$	$91.63 \pm .08$	$91.93 \pm .11$	$51.34 \pm .85$
		OOD	Margin	$(1, .9, .3)$	$66.92 \pm .40$	$91.33 \pm .27$	$52.23 \pm .86$
GOOD-HIV	Cov.	ID	Entropy	$(1, .7, .7)$	$83.97 \pm .45$	$81.57 \pm .57$	69.35 ± 2.20
		OOD	Entropy	$(1, .8, .8)$	$79.91 \pm .31$	$81.34 \pm .78$	74.52 ± 1.29
GOOD-HIV	Con.	ID	Entropy	$(1, .9, .9)$	$83.65 \pm .55$	82.39 ± 1.15	65.80 ± 2.27
		OOD	Entropy	$(1, 1, .8)$	$78.51 \pm .46$	79.95 ± 1.88	75.62 ± 1.00
GOOD-Motif	Cov.	ID	Entropy	$(1, 1, .4)$	$93.14 \pm .02$	$92.81 \pm .05$	72.96 ± 9.64
		OOD	Margin	$(1, 1, 0)$	$93.17 \pm .00$	$92.04 \pm .39$	$93.31 \pm .18$
GOOD-Motif	Con.	ID	Margin	$(1, .9, .5)$	$93.22 \pm .03$	$92.24 \pm .06$	89.90 ± 1.04
		OOD	Margin	$(1, .9, .9)$	$92.75 \pm .06$	$92.40 \pm .02$	$92.77 \pm .11$

For the main comparison, we report the OOD-validation-based results to remain consistent with prior evaluation protocols. Table 3 compares GraphSteer with existing methods on GOOD-CMNIST, GOOD-HIV, and GOOD-Motif under covariate and concept shifts. GraphSteer performs particularly strongly on GOOD-Motif and remains competitive on GOOD-CMNIST and GOOD-HIV, indicating that the benefit of structural reweighting varies across datasets and shift types.

4.4 Analysis of the Soft Structural Reweighting Mechanism

Figure 2 shows examples of the subgraphs discovered by the Leiden algorithm and the individual informativeness scores assigned to each subgraph. This demonstrates the interpretability aspect of the method: We can visualize whether the system picks those subgraphs we intuitively consider the ones carrying the non-spurious information.

To further examine the design choices underlying GraphSteer, we conduct an ablation study on subgraph construction and utilization in Appendix G. The results generally support retaining the full graph through soft edge reweighting and show that node-feature-aware Leiden can improve OOD performance in some settings, particularly on GOOD-CMNIST.

Table 2 then shows how the preferred edge-weight configurations vary across validation criteria and distribution shifts. Under covariate shift, ID-validation selection tends to retain more information from the broader graph, as seen in $(1, 0.9, 0.9)$ on GOOD-CMNIST and $(1, 0.7, 0.7)$ on GOOD-HIV. In contrast, OOD-validation selection can favour stronger attenuation, such as $(1, 0.8, 0.2)$ on GOOD-CMNIST and $(1, 1, 0)$ on GOOD-Motif. This is likely explained by the fact that when validating under the same distribution (ID-validation), more of the graph structure discoverable during training remains predictive. Under concept shift, the difference between ID- and OOD-validation selection is less characterized by strong suppression of the surrounding graph. Instead, OOD-validation selection tends to retain a substantial contribution from the broader graph, as seen in $(1, 1, 0.8)$ on GOOD-HIV and $(1, 0.9, 0.9)$ on GOOD-Motif. For GOOD-CMNIST, the outside-subgraph weight remains at 0.3, while the boundary weight increases from 0.7 under ID selection to 0.9 under OOD selection. Overall, we see that concept shift tends to favour milder attenuation of the graph, whereas stronger suppression can be beneficial in some covariate-shift settings.

Table 3: OOD-test performance on GOOD-CMNIST, GOOD-HIV, and GOOD-Motif under covariate and concept shifts. For GraphSteer, the scoring function and edge-weight configuration are selected using the OOD-validation near-optimal selection rule described in Section 4.2, while the training checkpoint is selected according to OOD-validation performance. GOOD-CMNIST and GOOD-Motif report accuracy (%), while GOOD-HIV reports ROC-AUC (%). Results are reported as mean ± standard deviation where available. The best result among methods without environment labels is **bolded**; a better result from a method using environment labels, if any, is *italicized*. A dash indicates that the corresponding result is not available.

Category	Method	Env.	GOOD-CMNIST		GOOD-HIV		GOOD-Motif	
			Covariate	Concept	Covariate	Concept	Covariate	Concept
General baselines	ERM	No	28.60 ± 2.01	42.87 ± 0.72	69.58 ± 1.99	72.33 ± 1.04	68.66 ± 3.43	81.44 ± 0.45
	IRM	Yes	27.83 ± 1.84	42.80 ± 0.38	67.97 ± 2.46	72.59 ± 0.45	70.65 ± 3.18	80.71 ± 0.46
	VREx	Yes	28.48 ± 2.08	43.31 ± 0.78	70.77 ± 1.35	72.60 ± 0.82	71.47 ± 2.75	81.56 ± 0.35
	GroupDRO	Yes	29.07 ± 2.62	43.32 ± 0.75	70.64 ± 1.72	73.64 ± 0.86	68.24 ± 1.94	81.43 ± 0.70
	DANN	Yes	29.14 ± 2.93	43.11 ± 0.64	70.63 ± 1.82	71.92 ± 1.23	65.47 ± 5.35	81.33 ± 0.52
	Deep CORAL	Yes	29.05 ± 2.19	43.16 ± 0.56	68.61 ± 1.70	72.97 ± 1.04	68.88 ± 3.61	81.37 ± 0.42
	Mixup	No	26.47 ± 1.73	40.96 ± 0.81	68.88 ± 2.40	72.03 ± 0.53	70.08 ± 2.06	77.63 ± 0.57
	ASAP	No	23.53 ± 0.67	–	68.44 ± 0.49	–	–	–
Graph invariant learning	DIR	No	20.60 ± 4.26	22.69 ± 2.85	67.47 ± 2.61	71.40	61.50 ± 15.69	72.14 ± 7.29
	CAL	No	–	–	67.37 ± 3.61	72.49	65.63 ± 4.29	–
	GSAT	No	35.02 ± 2.78	–	68.66 ± 1.35	72.51	60.42 ± 9.32	76.07 ± 3.48
	OOD-GNN	No	–	–	70.46 ± 1.97	–	–	–
	StableGNN	No	–	–	68.44 ± 1.33	–	–	–
	CIGA	No	25.06 ± 3.07	–	69.40 ± 2.39	71.65	68.71 ± 10.90	81.68 ± 3.01
	GALA	No	$\mathbf{53.30 \pm 2.32}$	–	67.37 ± 4.58	–	54.78 ± 6.05	–
	MoleOOD	No	–	–	69.39	69.08	–	–
	iMoLD	No	–	–	72.93	74.32	–	–
	CSIB	Yes	–	–	72.53 ± 2.01	–	–	–
	LECI	Yes	–	–	74.43 ± 1.69	–	–	–
Disentanglement	DisC	No	24.99 ± 1.78	–	68.07 ± 1.75	64.82	51.08 ± 3.08	–
	CFD	No	–	–	**76.42**	**77.83**	–	–
Graph data augmentation	FLAG	No	37.74 ± 7.88	–	68.45 ± 2.30	–	61.12 ± 5.39	–
	GREA	No	29.02 ± 3.26	–	67.79 ± 2.56	70.76	56.74 ± 9.23	78.27 ± 4.29
	DropNode	No	33.01 ± 0.12	–	–	–	–	–
	DropEdge	No	26.83 ± 0.81	–	70.78 ± 1.38	–	45.08 ± 4.46	–
	MaskFeature	No	44.85 ± 2.42	–	–	–	–	–
	Graph Mixup	No	26.47 ± 1.73	–	–	–	–	–
	LiSA	No	29.63 ± 2.82	–	70.38 ± 1.45	–	54.59 ± 4.81	–
	G-Mixup	No	26.40 ± 1.47	–	70.01 ± 2.52	–	59.66 ± 7.03	–
	GraphCL	No	10.39 ± 1.10	–	50.71 ± 3.71	–	33.11 ± 7.85	–
	AIA	No	36.37 ± 4.44	–	71.15 ± 1.81	–	72.91 ± 5.62	82.51 ± 2.81
	FeatX	Yes	*62.49 ± 2.12*	–	–	–	–	–
	MPAIACL	No	42.35 ± 0.14	–	69 ± 2.11	–	69.77 ± 0.79	–
Pruning-based	PrunE	No	33.89 ± 1.65	–	71.84 ± 0.61	–	91.48 ± 0.40	90.28 ± 1.72
Our method	**GraphSteer**	**No**	38.16 ± 4.1	$\mathbf{52.23 \pm 0.86}$	74.52 ± 1.29	75.62 ± 1.00	$\mathbf{93.31 \pm 0.18}$	$\mathbf{92.77 \pm 0.11}$

Notes. Env. = use of ground-truth environment labels during model training. A dash indicates that the corresponding result is not available. The GraphSteer weight configurations reported here are selected according to the OOD-validation procedure described in Section 4.3.

5 Conclusion and Discussion

Graph OOD methods often rely on identifying or isolating predictive substructures, but aggressively suppressing the remaining graph structure can remove information that is still useful for generalization. To address this, GraphSteer uses confidence-scored Leiden subgraphs to softly steer message passing while preserving the original graph topology. Experiments on GOOD-CMNIST, GOOD-HIV, and GOOD-Motif show that GraphSteer achieves competitive OOD performance across both covariate and concept shifts without requiring ground-truth environment labels. Our results further indicate that the selected subgraph should guide, rather than replace, the remaining graph structure, since overly strong suppression of non-selected edges can degrade performance in some settings. This observation distinguishes GraphSteer from pruning-based approaches such as PrunE, which explicitly remove edges considered spurious. Instead, GraphSteer preserves the original graph topology and adjusts the contribution of different edge categories, outperforming PrunE across the overlapping benchmark settings reported in our experiments. The preferred degree of reweighting varies across datasets and shifts, suggesting that different settings benefit from different levels of structural emphasis. Future work includes extending GraphSteer to other graph-learning settings, such as graph regression.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, organization of the manuscript, and code-related troubleshooting. All AI-assisted text, code, and technical claims were reviewed by the authors, and experimental results were obtained from the authors' own computational pipeline. The authors take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses existing benchmark graph datasets and does not involve human participants or the collection of personally identifiable information. The method is intended for research on graph OOD generalization. As with other machine-learning systems, predictions on molecular data should not be treated as substitutes for domain-expert or experimental validation.

REPRODUCIBILITY STATEMENT

The manuscript describes the GraphSteer pipeline, confidence scores, reconstruction mechanism, and downstream weighted message passing. Complete hyperparameter and architecture comparisons are provided in the appendix. Code, configuration files, random seeds, and preprocessing details should be included in the supplementary material or an anonymous repository for reproduction of the reported experiments.

REFERENCES

Weizhi An, Wenliang Zhong, Feng Jiang, Hehuan Ma, and Junzhou Huang. Causal subgraphs and information bottlenecks: Redefining OOD robustness in graph neural networks. In *Computer Vision – ECCV 2024*. Springer, 2024.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22131–22148, 2022.

Yongqiang Chen, Yatao Bian, Kaiwen Zhou, Binghui Xie, Bo Han, and James Cheng. Does invariant graph learning via environment augmentation learn invariance? In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. Debiasing graph neural networks via learning disentangled causal substructure. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):322–337, 2024. doi: 10.1109/TPAMI.2023.3321097.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2059–2073, 2022.

Shurui Gui, Meng Liu, Xiner Li, Youzhi Luo, and Shuiwang Ji. Joint learning of label and environment causal independence for graph out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-Mixup: Graph data augmentation for graph classification. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, Xinwang Liu, Philip S. Yu, and Ming Zhang. A survey of graph neural networks in real world: Imbalance, noise, privacy and OOD challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3036–3055, 2026. doi: 10.1109/TPAMI.2025.3630673.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9012–9020, 2019.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 60–69, 2022. doi: 10.1109/CVPR52688.2022.00016.

David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5815–5826. PMLR, 2021.

Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. OOD-GNN: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022. doi: 10.1109/TKDE.2022.3193725.

Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11): 10490–10512, 2025. doi: 10.1109/TPAMI.2025.3593897.

Xiner Li, Shurui Gui, Youzhi Luo, and Shuiwang Ji. Graph structure extrapolation for out-of-distribution generalization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1069–1078, 2022. doi: 10.1145/3534678.3539347.

Siqi Miao, Miaoyuan Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. ASAP: Adaptive structure aware pooling for learning hierarchical graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1696–1705, 2022. doi: 10.1145/3534678.3539366.

Yongduo Sui, Qitian Wu, Jiancan Wu, Qing Cui, Longfei Li, Jun Zhou, Xiang Wang, and Xiangnan He. Unleashing the power of graph data augmentation on covariate distribution shift. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, pp. 443–450. Springer, 2016. doi: 10.1007/978-3-319-49409-8_35.

Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9:5233, 2019. doi: 10.1038/s41598-019-41695-z.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of The Web Conference 2021*, pp. 3663–3674, 2021.

Aming Wu and Cheng Deng. CFD: Learning generalized molecular representation via concept-enhanced feedback disentanglement. In *International Conference on Learning Representations*, 2025.

Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. In *International Conference on Learning Representations*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. Learning substructure invariance for out-of-distribution molecular representations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12964–12978, 2022.

Tianjun Yao, Haoxuan Li, Yongqiang Chen, Tongliang Liu, Le Song, Eric P. Xing, and Zhiqiang Shen. Pruning spurious subgraphs for graph out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2025.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Junchi Yu, Jian Liang, and Ran He. Mind the label shift of augmentation-based graph OOD generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Fanlong Zeng and Wensheng Gan. Graph data augmentation with contrastive learning on covariate distribution shift. *Neural Networks*, 205:109388, 2026. doi: 10.1016/j.neunet.2026.109388.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Kexin Zhang, Shuhan Liu, Song Wang, Weili Shi, Chen Chen, Pan Li, Sheng Li, Jundong Li, and Kaize Ding. A survey of deep graph learning under distribution shifts: From graph out-of-distribution generalization to adaptation. *ACM Transactions on Knowledge Discovery from Data*, 20(2):1–38, 2026. doi: 10.1145/3785475.

Xianchao Zhang, Jingwei Li, and Hong Yu. Local density adaptive similarity measurement for spectral clustering. *Pattern Recognition Letters*, 32(2):352–358, 2011. doi: 10.1016/j.patrec.2010.09.014.

Xiang Zhuang, Qiang Zhang, Keyan Ding, Yatao Bian, Xiao Wang, Jingsong Lv, Hongyang Chen, and Huajun Chen. Learning invariant molecular representation in latent discrete space. In *Advances in Neural Information Processing Systems*, volume 36, pp. 78435–78452, 2023.

Appendix of GraphSteer

Contents

A Notations **15**

B Detailed Dataset Description **16**

C Algorithmic Pseudocode of GraphSteer **17**

D Detailed Experimental Settings **17**

E Complete Experiments **18**

E.1 GOOD-CMNIST 18

E.2 GOOD-HIV 20

E.3 GOOD-Motif 21

F Sensitivity Analysis of the Near-Optimality Threshold **23**

G Ablation Study on Subgraph Construction and Selection **23**

Table 4: Summary of the main notation used in GraphSteer.

Symbol	Definition
G	Input graph.
V	Set of nodes in graph G.
E	Set of edges in graph G.
X^V	Node-feature matrix of graph G.
X^E	Edge-feature matrix of graph G.
d	Dimension of the node features.
d_e	Dimension of the edge features.
Y	Graph-level prediction target.
$P(G, Y)$	Joint distribution over input graphs and graph-level labels.
P_{tr}	Training distribution.
P_{te}	Test distribution.
$\widetilde{G}$	Edge-weighted graph obtained from the original graph after GraphSteer reweighting.
ω	Edge-weighting function $\omega : E \to \mathbb{R}_{\geq 0}$.
x_i^V, x_j^V	Node-feature vectors of nodes i and j.
S_{ij}	Gaussian node-feature similarity assigned to an existing edge (i, j) during feature-aware Leiden community detection.
σ	Gaussian bandwidth controlling the scale of node-feature similarity.
C_m	Node set of the m-th Leiden community.
M	Number of communities discovered by Leiden.
G_m	Candidate subgraph induced by community C_m.
E_m	Set of edges whose endpoints both belong to community C_m.
X_m^V	Node features associated with candidate subgraph G_m.
X_m^E	Edge features associated with candidate subgraph G_m.
f	Subgraph classifier used to estimate the predictive informativeness of candidate subgraphs.
p_m	Predicted class-probability vector for candidate subgraph G_m.
$p_{m,c}$	Predicted probability of class c for candidate subgraph G_m.
$p_{m,(1)}$	Largest class probability predicted for candidate subgraph G_m.
$p_{m,(2)}$	Second-largest class probability predicted for candidate subgraph G_m.
C	Number of prediction classes.
G^*	Highest-ranked candidate subgraph selected according to the informativeness score.
V^*	Node set of the selected subgraph G^*.
w_I	Weight assigned to inner edges whose endpoints are both inside the selected subgraph.
w_B	Weight assigned to boundary edges connecting selected and non-selected nodes.
w_O	Weight assigned to outer edges whose endpoints are both outside the selected subgraph.
ω_{uv}	Message-passing weight assigned to edge (u, v).
$h_i^{(l)}$	Representation of node i at GNN layer l.
$\mathcal{N}(i)$	Set of neighbouring nodes of node i.
$\varepsilon^{(l)}$	GIN parameter controlling the contribution of the node's own representation at layer l.
$\boldsymbol{w}$	GraphSteer edge-weight configuration (w_I, w_B, w_O).
$R(\boldsymbol{w})$	Validation performance obtained using edge-weight configuration $\boldsymbol{w}$.
R^*	Best validation performance among the evaluated edge-weight configurations.
ϵ	Tolerance used to define the near-optimal set of edge-weight configurations.
$\mathcal{W}_\epsilon$	Set of edge-weight configurations whose validation performance is within ϵ of R^*.
$S(\boldsymbol{w})$	Secondary weight-separation criterion, $(w_I - w_B)^2 + (w_B - w_O)^2$.
$\boldsymbol{w}^*$	Selected edge-weight configuration after validation-based model selection.

A NOTATIONS

For clarity, Table 4 summarizes the main mathematical notation used throughout the paper.

B Detailed Dataset Description

We evaluate GraphSteer on three graph-classification datasets from the GOOD benchmark: GOOD-CMNIST, GOOD-HIV, and GOOD-Motif. For each dataset, GOOD constructs separate covariate- and concept-shift settings based on a predefined domain variable. In the covariate-shift setting, the distribution of the domain variable changes across data splits while its relationship with the target is not intentionally altered. In the concept-shift setting, GOOD changes the spurious correlation between the domain variable and the target across environments. The specific construction for each dataset is described below.

GOOD-CMNIST. GOOD-CMNIST contains handwritten MNIST digits converted into graphs using superpixels, with the digit class as the prediction target and colour as the domain variable. The original greyscale graphs are coloured by assigning three-channel RGB values to the node features.

For the covariate-shift setting, seven different colours are used. The training set contains graphs from five colour domains, while the OOD validation and OOD test sets use the sixth and seventh colours, respectively. Thus, the shift is induced by changing the distribution of the colour feature across environments, while colour itself is not intended to determine the digit label.

For the concept-shift setting, ten colours are used, with each digit class associated with one colour. During training, each colour is highly correlated with a particular digit label. This colour-label correlation is weakened in the OOD validation set and removed in the OOD test set. The shift therefore arises from changing the spurious relationship between colour and the target label across environments.

GOOD-HIV. GOOD-HIV is a molecular graph dataset in which nodes represent atoms and edges represent chemical bonds. The prediction task is to determine whether a molecule inhibits HIV replication. We use the Bemis-Murcko scaffold as the domain variable.

For the covariate-shift setting, molecules are grouped according to their scaffold domains. The graphs are sorted according to the selected domain and divided such that the training, OOD validation, and OOD test sets contain different scaffold domains. The resulting distribution shift therefore arises from changes in the molecular scaffold distribution across the data splits.

For the concept-shift setting, the molecular graphs themselves are not modified. Instead, GOOD constructs different concepts using a screening procedure based on scaffold-label probabilities. Each molecule has a probability of being included in a particular concept according to its scaffold domain and HIV activity label. The training concepts are constructed to have relatively similar scaffold-label correlations, whereas the correlations differ substantially between the training, OOD validation, and OOD test sets. The shift is therefore induced by changing the spurious relationship between scaffold and HIV activity across environments.

GOOD-Motif. GOOD-Motif is a synthetic graph-classification dataset designed to evaluate structural distribution shifts. Each graph is generated by connecting a label-determining motif to a label-irrelevant base graph. The prediction target is one of three motifs—house, cycle, and crane—while five possible base graphs are used: wheel, tree, ladder, star, and path. We use the base-graph type as the domain variable.

For the covariate-shift setting, the training set contains graphs with wheel, tree, and ladder bases, while the OOD validation and OOD test sets contain star and path bases, respectively. The base graph is independent of the motif label, so the shift is induced by exposing the model to base structures that are not observed during training.

For the concept-shift setting, only wheel, tree, and ladder bases are used. In the training environments, the house, cycle, and crane motifs are strongly correlated with wheel, tree, and ladder bases, respectively. The corresponding correlations are constructed with probabilities of 99%, 97%, and 95%. These motif–base correlations are weakened in the OOD validation set and removed in the OOD test set. Thus, the base structure is highly predictive of the target during training but becomes progressively less informative outside the training distribution.

Table 5 summarizes the number of graphs in each split for the covariate- and concept-shift settings.

Table 5: Dataset split sizes for the covariate- and concept-shift settings.

Dataset	Shift	Train	ID-Validation	ID-Test	OOD-Validation	OOD-Test
GOOD-CMNIST	Covariate	42000	7000	7000	7000	7000
	Concept	29400	6300	6300	14000	14000
GOOD-HIV	Covariate	24682	4112	4112	4113	4108
	Concept	15209	3258	3258	9365	10037
GOOD-Motif	Covariate	18000	3000	3000	3000	3000
	Concept	12600	2700	2700	6000	6000

Algorithm 1 GraphSteer

Require: Graph dataset splits $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}, \mathcal{D}_{\mathrm{te}}$; confidence score $q(\cdot)$; edge weights (w_I, w_B, w_O)
Ensure: Trained downstream GNN g_θ

Estimate σ from connected-node feature distances in $\mathcal{D}_{\mathrm{tr}}$
for each graph $G = (V, E, X^V, X^E)$ in all data splits **do**
    Partition G into Leiden communities $\{C_1, \ldots, C_M\}$
    Construct candidate subgraphs $\{G_1, \ldots, G_M\}$
end for
Train subgraph classifier f_ϕ on candidate subgraphs from $\mathcal{D}_{\mathrm{tr}}$ using inherited graph labels
Freeze f_ϕ
for each graph G in all data splits **do**
    for each candidate subgraph G_m of G **do**
        Compute class probabilities $p_m \leftarrow f_\phi(G_m)$
        Compute informativeness score $s_m \leftarrow q(p_m)$
    end for
    Select $m^* \leftarrow \arg\max_m s_m$
    $V^* \leftarrow C_{m^*}$
    for each edge $(u, v) \in E$ **do**
        if $u \in V^*$ **and** $v \in V^*$ **then**
            $\omega_{uv} \leftarrow w_I$
        else if $u \in V^*$ **xor** $v \in V^*$ **then**
            $\omega_{uv} \leftarrow w_B$
        else
            $\omega_{uv} \leftarrow w_O$
        end if
    end for
    Construct weighted graph $\widetilde{G} \leftarrow (V, E, X^V, X^E, \omega)$
end for
Train downstream GNN g_θ on the weighted training graphs
Select the model using validation performance
Evaluate g_θ on the weighted test graphs
return g_θ

C Algorithmic Pseudocode of GraphSteer

For completeness, Algorithm 1 summarizes the complete GraphSteer pipeline, from candidate-subgraph discovery to downstream training on the edge-weighted graphs.

D Detailed Experimental Settings

Leiden Community Detection. For GOOD-CMNIST and GOOD-HIV, we use node-feature-aware Leiden community detection. The Gaussian bandwidth σ is set to the median Euclidean distance between connected-node features computed from the training split and is subsequently fixed across all splits. For GOOD-CMNIST, the original node features are used, whereas for GOOD-HIV the

Table 6: GNN architectures and training hyperparameters used for each dataset.

Dataset	Model	Pooling	Layers	LR	WD	Dropout	Max Epoch
GOOD-CMNIST	GIN + Virtual Node	Mean	5	10^{-3}	0	0.5	500
GOOD-HIV	GINE + Virtual Node	Max	2	10^{-5}	10^{-5}	0.5	200
GOOD-Motif	GIN	Sum	5	10^{-3}	0	0.5	200

categorical node features are first one-hot encoded. For GOOD-Motif, since all nodes share identical node features, we use the original Leiden algorithm based on the graph adjacency structure.

Subgraph Classifier. The subgraph classifier consists of a two-layer GCN encoder with a ReLU activation between layers, followed by global mean pooling and a linear classification head. For GOOD-HIV, we use class-weighted cross-entropy with weights of 1 and 10 for the negative and positive classes, respectively, assigning greater loss weight to the positive class.

Edge-weight Search. For all datasets and shifts, we fix the inner-edge weight to $w_I = 1$ and evaluate 22 predefined combinations of (w_B, w_O) while enforcing $w_I \geq w_B \geq w_O$. The evaluated pairs are $(0.7, 0)$, $(0.7, 0.1)$, $(0.7, 0.3)$, $(0.7, 0.5)$, $(0.7, 0.7)$, $(0.8, 0)$, $(0.8, 0.2)$, $(0.8, 0.4)$, $(0.8, 0.6)$, $(0.8, 0.8)$, $(0.9, 0)$, $(0.9, 0.1)$, $(0.9, 0.3)$, $(0.9, 0.5)$, $(0.9, 0.7)$, $(0.9, 0.9)$, and $(1, 0)$, $(1, 0.2)$, $(1, 0.4)$, $(1, 0.6)$, $(1, 0.8)$, $(1, 1)$. The configuration $(1, 1, 1)$ corresponds to the no-reweighting control, where all original edges retain equal message-passing strength. A weight value of 0 suppresses message passing along the corresponding edge category, while the associated nodes and their node features remain in the graph.

GNN Architecture and Training. The downstream GNN architecture and training hyperparameters are configured separately for each dataset, as summarized in Table 6. GOOD-CMNIST and GOOD-HIV use GIN with virtual nodes, whereas GOOD-Motif uses GIN without virtual nodes. Results for GOOD-CMNIST and GOOD-HIV are averaged over three random seeds, whereas GOOD-Motif results are averaged over five random seeds. All experiments were conducted on an NVIDIA A100 GPU.

E Complete Experiments

E.1 GOOD-CMNIST

We evaluate GOOD-CMNIST under colour concept and covariate shifts using classification accuracy. Both Entropy- and Margin-based subgraph scoring are evaluated across the considered edge-weight configurations using a five-layer mean-pooling downstream architecture. Results are reported under ID- and OOD-validation-based checkpoint selection.

E.1.1 Concept Shift

Table 7 summarizes the GOOD-CMNIST colour concept-shift results under ID- and OOD-validation-based checkpoint selection.

Table 7: GOOD-CMNIST colour concept-shift results under ID- and OOD-validation-based checkpoint selection. Accuracy is reported as mean ± sample standard deviation across three runs. Italic indicates the highest validation mean within each scoring function, while bold indicates the results corresponding to the configuration selected by the model-selection procedure. Bold italic indicates cases satisfying both criteria.

Score	Architecture	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
Entropy	5L Mean	1	0.7	0	89.30±0.43	90.12±0.27	35.04±1.53	56.58±1.22	89.39±0.62	38.16±1.64
		1	0.7	0.1	90.22±0.49	90.99±0.31	41.85±2.48	60.20±1.82	90.09±0.49	43.01±2.76
		1	0.7	0.3	91.01±0.11	91.40±0.07	48.21±1.21	*65.04±0.75*	91.05±0.08	49.62±1.61
		1	0.7	0.5	90.81±0.02	90.98±0.06	46.83±0.39	63.98±0.11	90.87±0.28	47.65±0.55
		1	0.7	0.7	90.16±0.14	90.38±0.16	44.01±0.34	61.83±0.43	90.15±0.15	44.80±0.92
		1	0.8	0	89.49±0.19	90.09±0.25	35.41±1.68	56.72±0.40	89.77±0.11	37.59±0.70
		1	0.8	0.2	*91.10±0.08*	91.32±0.05	49.32±1.27	64.94±0.85	91.11±0.42	50.51±0.97
		1	0.8	0.4	90.76±0.10	91.09±0.06	47.33±0.84	64.16±0.37	91.02±0.16	48.43±1.06
		1	0.8	0.6	90.22±0.13	90.23±0.12	43.73±0.86	61.94±0.43	89.82±0.39	44.69±1.35

Continued on the next page

Table 7 continued from the previous page

Score	Architecture	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
		1	0.8	0.8	89.52±0.18	89.68±0.09	42.12±0.86	60.14±0.73	89.67±0.06	42.48±0.86
		1	0.9	0	89.67±0.21	90.54±0.10	37.64±1.28	58.21±0.67	89.88±0.13	40.45±1.40
		1	0.9	0.1	90.16±0.33	90.49±0.25	39.45±2.05	59.98±1.40	89.96±0.15	42.66±1.90
		1	0.9	0.3	90.77±0.16	90.86±0.23	47.43±0.80	64.19±0.17	90.39±0.20	49.33±0.43
		1	0.9	0.5	90.57±0.16	90.75±0.12	45.84±0.38	63.05±0.40	90.47±0.34	46.44±0.63
		1	0.9	0.7	89.76±0.08	89.90±0.13	40.97±0.32	60.14±0.05	88.88±0.23	42.56±0.26
		1	0.9	0.9	89.68±0.09	89.80±0.28	41.33±0.17	60.16±0.28	89.66±0.34	42.22±0.45
		1	1	0	88.94±0.07	89.64±0.11	32.73±0.34	55.49±1.45	88.71±0.32	36.55±2.51
		1	1	0.2	90.84±0.25	91.02±0.18	47.35±1.63	63.76±0.87	90.69±0.29	48.01±1.76
		1	1	0.4	90.63±0.06	90.70±0.29	46.19±0.75	63.30±0.64	90.31±0.21	47.14±1.28
		1	1	0.6	90.04±0.16	90.21±0.27	43.39±0.63	61.38±0.31	89.90±0.73	44.08±0.34
		1	1	0.8	89.51±0.14	89.59±0.14	40.00±0.58	59.07±0.27	89.41±0.38	41.17±0.91
		1	1	1	89.59±0.08	89.99±0.13	41.42±0.56	60.42±0.07	89.77±0.48	42.49±0.28
Margin	5L Mean	1	0.7	0	89.24±0.58	90.01±0.50	36.37±3.71	57.50±1.89	89.16±0.86	39.24±2.80
		1	0.7	0.1	90.67±0.39	91.10±0.24	45.09±3.17	63.39±1.70	90.95±0.40	46.92±2.16
		1	0.7	0.3	***91.63±0.08***	91.93±0.11	51.34±0.85	66.79±0.26	91.67±0.17	52.20±0.72
		1	0.7	0.5	91.23±0.25	91.60±0.14	50.07±1.37	65.88±0.96	91.42±0.35	50.82±0.92
		1	0.7	0.7	90.45±0.15	90.57±0.18	46.23±0.52	63.01±0.28	90.09±0.62	46.54±0.18
		1	0.8	0	89.35±0.48	90.15±0.19	38.20±0.88	57.47±1.45	90.25±0.36	38.87±1.74
		1	0.8	0.2	91.56±0.20	91.99±0.07	51.62±1.29	66.58±0.83	91.68±0.17	52.16±1.67
		1	0.8	0.4	91.42±0.08	91.66±0.14	49.83±0.33	66.17±0.75	91.56±0.21	51.23±0.96
		1	0.8	0.6	90.77±0.22	90.83±0.06	46.01±0.56	63.46±0.29	90.88±0.20	46.83±0.45
		1	0.8	0.8	89.84±0.16	90.00±0.20	42.41±0.68	60.74±0.19	89.51±0.74	43.56±0.23
		1	0.9	0	89.44±0.37	90.00±0.40	38.02±1.79	58.53±0.18	90.06±0.83	40.30±0.85
		1	0.9	0.1	89.82±0.66	90.69±0.76	39.98±4.55	60.13±1.39	89.66±1.61	43.02±1.64
		1	0.9	0.3	91.50±0.38	91.56±0.30	51.31±0.23	***66.92±0.40***	91.33±0.27	52.23±0.86
		1	0.9	0.5	91.14±0.11	91.13±0.16	47.76±0.45	64.69±0.24	91.09±0.23	48.86±0.62
		1	0.9	0.7	90.12±0.23	90.03±0.38	42.58±1.29	60.96±0.59	90.34±0.25	43.29±1.00
		1	0.9	0.9	89.68±0.22	89.95±0.08	41.89±0.61	59.97±0.29	89.75±0.37	42.09±0.58
		1	1	0	89.63±0.56	90.24±0.36	39.43±4.01	59.33±1.11	89.77±0.97	41.89±1.04
		1	1	0.2	91.23±0.29	91.72±0.43	49.73±1.94	65.78±0.97	91.49±0.62	50.98±1.52
		1	1	0.4	91.56±0.06	91.68±0.22	49.80±0.54	66.27±0.28	91.50±0.25	51.19±0.20
		1	1	0.6	90.79±0.16	90.66±0.21	45.70±0.73	63.27±0.47	90.48±0.10	46.92±0.93
		1	1	0.8	89.64±0.03	90.21±0.12	41.98±0.16	60.05±0.18	89.97±0.46	42.19±0.44
		1	1	1	89.50±0.09	89.66±0.19	40.56±1.11	60.34±0.13	89.47±0.31	42.68±0.62

E.1.2 COVARIATE SHIFT

Table 8 summarizes the GOOD-CMNIST colour covariate-shift results under ID- and OOD-validation-based checkpoint selection.

Table 8: GOOD-CMNIST colour covariate-shift results under ID- and OOD-validation-based checkpoint selection. Accuracy is reported as mean ± sample standard deviation across three runs. Italic indicates the highest validation mean within each scoring function, while bold indicates the results corresponding to the configuration selected by the model-selection procedure. Bold italic indicates cases satisfying both criteria.

Score	Architecture	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
Entropy	5L Mean	1	0.7	0	59.67±1.21	59.70±1.11	29.83±3.83	69.96±1.89	55.08±2.03	44.92±4.37
		1	0.7	0.1	73.89±1.35	73.81±0.69	47.75±2.13	78.69±0.93	70.61±2.15	44.92±5.24
		1	0.7	0.3	76.63±0.20	76.21±0.22	35.97±1.98	78.77±0.15	74.81±0.11	36.51±2.90
		1	0.7	0.5	76.15±0.34	77.00±0.36	25.45±3.37	70.95±1.38	74.09±2.36	28.85±4.41
		1	0.7	0.7	76.11±0.31	76.24±0.27	27.20±1.94	61.09±3.67	73.30±1.86	28.79±0.15
		1	0.8	0	58.78±2.21	58.41±2.01	27.12±3.21	67.33±1.65	55.29±1.72	36.25±5.92
		1	0.8	0.2	75.21±0.49	74.96±0.93	35.61±2.71	***79.21±1.41***	71.39±2.12	38.16±4.10
		1	0.8	0.4	75.53±0.42	75.72±0.28	28.08±3.52	74.80±1.22	72.61±2.30	29.09±3.58
		1	0.8	0.6	75.74±0.25	76.05±0.70	26.44±2.48	68.36±2.36	74.99±0.79	26.37±1.66
		1	0.8	0.8	76.44±0.82	76.35±0.74	27.81±2.33	60.32±3.25	74.11±1.11	26.74±2.84
		1	0.9	0	58.97±0.55	58.28±0.37	27.02±5.73	67.57±1.93	52.72±5.01	35.70±4.01
		1	0.9	0.1	70.94±1.63	70.75±1.61	39.65±7.61	69.73±4.50	64.22±7.21	38.82±4.66
		1	0.9	0.3	75.99±0.14	75.47±0.35	31.07±5.84	78.20±0.40	73.21±1.09	38.20±1.62
		1	0.9	0.5	75.07±0.77	74.97±0.69	24.25±2.42	70.15±3.36	73.12±0.99	25.25±2.48
		1	0.9	0.7	76.31±0.22	76.36±0.26	24.74±3.72	67.09±2.19	75.04±0.89	27.08±2.74
		1	0.9	0.9	**77.51±0.30**	77.35±0.12	28.84±4.48	63.84±0.11	75.69±1.65	28.63±1.02
		1	1	0	59.59±0.93	58.64±0.87	26.36±7.41	68.31±1.45	55.16±0.26	33.99±5.11
		1	1	0.2	73.09±1.98	72.64±2.18	27.23±0.20	74.66±2.71	71.04±1.74	29.30±7.58
		1	1	0.4	75.04±0.66	74.27±0.53	26.04±4.66	74.41±3.12	72.13±1.96	31.47±6.88
		1	1	0.6	75.22±0.42	75.05±0.39	27.03±0.82	67.44±4.77	73.39±0.99	25.32±2.51
		1	1	0.8	77.04±0.38	76.55±0.36	30.81±1.74	65.24±1.78	75.02±1.38	27.44±3.66
		1	1	1	*78.03±0.11*	77.40±0.53	27.61±1.27	63.43±0.63	75.02±2.07	29.61±2.34
Margin	5L Mean	1	0.7	0	52.13±3.70	51.61±4.14	33.19±2.41	65.14±1.59	48.89±5.94	38.32±2.86
		1	0.7	0.1	75.02±0.49	75.10±0.66	34.24±4.06	64.94±2.27	65.40±4.83	35.70±3.14
		1	0.7	0.3	73.01±0.13	73.16±0.23	27.17±2.63	66.92±2.16	72.11±1.17	29.11±6.09
		1	0.7	0.5	74.28±0.23	74.67±0.13	26.24±2.11	*68.80±1.67*	74.39±0.34	27.32±1.15
		1	0.7	0.7	76.36±0.24	75.86±0.48	30.06±4.67	63.97±1.47	74.78±0.51	30.01±1.11
		1	0.8	0	52.14±1.97	51.88±1.86	35.25±4.44	65.03±1.55	46.42±1.78	37.27±7.98
		1	0.8	0.2	73.57±0.62	73.34±0.57	29.53±2.97	67.42±1.54	69.69±1.84	29.57±4.38
		1	0.8	0.4	72.61±0.28	73.09±0.24	23.40±1.71	67.44±1.20	70.24±1.26	35.65±4.04
		1	0.8	0.6	74.40±0.11	74.61±0.76	20.78±1.62	66.58±1.16	73.97±0.52	25.28±2.20
		1	0.8	0.8	77.37±0.38	77.29±0.13	31.43±4.12	63.80±1.68	75.63±0.77	33.96±2.73
		1	0.9	0	51.85±1.17	51.40±1.57	26.13±8.39	64.19±1.98	49.29±2.74	33.54±7.99
		1	0.9	0.1	71.88±0.65	71.81±0.87	35.64±5.17	60.03±3.58	43.49±2.03	27.40±1.45

Continued on the next page

Table 8 continued from the previous page

Score	Architecture	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
		1	0.9	0.3	71.72±0.41	71.28±0.59	26.24±0.57	62.76±2.91	67.72±0.96	29.69±0.98
		1	0.9	0.5	72.87±0.27	72.61±0.50	26.65±2.63	68.26±0.73	71.62±1.03	25.23±2.05
		1	0.9	0.7	75.06±0.27	75.15±0.42	23.74±1.46	64.76±0.65	73.90±0.58	27.72±2.57
		1	0.9	0.9	77.45±0.26	77.30±0.43	24.78±2.38	63.18±3.34	75.00±1.04	30.25±1.90
		1	1	0	52.95±1.34	53.01±0.64	25.83±4.39	64.16±0.91	49.69±3.20	29.80±4.24
		1	1	0.2	72.91±0.55	72.15±0.75	28.86±2.20	67.56±2.39	70.45±1.09	25.33±1.77
		1	1	0.4	71.88±0.69	71.80±0.90	18.95±1.02	67.62±1.22	70.79±0.17	23.80±1.01
		1	1	0.6	72.57±0.32	73.11±0.56	24.56±1.75	66.23±1.10	71.74±0.65	25.59±2.49
		1	1	0.8	75.90±0.29	75.87±0.33	28.72±3.59	63.04±1.86	75.64±0.42	26.55±3.34
		1	1	1	*78.10±0.18*	77.66±0.31	25.85±0.69	64.06±1.57	75.22±1.02	29.39±0.25

E.2 GOOD-HIV

We evaluate GOOD-HIV under scaffold-based concept and covariate shifts using ROC–AUC. Entropy-based subgraph scoring is used throughout these experiments. Results are reported under ID- and OOD-validation-based checkpoint selection.

E.2.1 SCAFFOLD CONCEPT SHIFT

Table 9 summarizes the GOOD-HIV scaffold concept-shift results under ID- and OOD-validation-based checkpoint selection.

Table 9: GOOD-HIV scaffold concept-shift results under ID- and OOD-validation-based checkpoint selection using entropy-based subgraph scoring. ROC–AUC is reported as mean ± sample standard deviation across three runs. The learning rate is indicated below the architecture. Italic indicates the highest validation mean, while bold indicates the results corresponding to the configuration selected by the model-selection procedure. Bold italic indicates cases satisfying both criteria.

Score	Architecture	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
Entropy	2L Max	1	0.7	0	76.69±0.43	82.27±1.03	58.01±4.06	64.89±0.39	79.68±1.36	63.69±0.32
	LR=10^{-5}	1	0.7	0.1	77.75±0.46	82.14±0.45	61.90±1.31	66.86±0.09	80.85±0.67	64.10±0.15
		1	0.7	0.3	80.28±0.61	82.61±0.69	64.24±0.94	69.72±1.15	81.56±1.47	66.75±0.98
		1	0.7	0.5	81.65±0.66	83.42±1.46	66.47±1.19	72.44±0.36	81.48±1.71	69.63±0.76
		1	0.7	0.7	82.49±0.64	84.05±0.64	66.17±0.35	73.86±0.07	82.69±1.05	70.68±0.68
		1	0.8	0	76.86±0.18	80.86±0.95	60.07±2.03	64.33±0.65	80.29±0.87	63.01±0.29
		1	0.8	0.2	78.85±0.27	82.78±1.10	62.50±1.44	68.52±0.67	81.83±1.29	65.11±0.58
		1	0.8	0.4	80.98±0.56	83.15±1.47	65.41±0.43	71.35±0.59	81.54±1.97	68.32±0.64
		1	0.8	0.6	81.90±0.58	83.52±0.95	65.14±1.22	72.54±0.09	82.55±0.69	68.94±0.76
		1	0.8	0.8	82.50±0.08	84.01±0.22	67.40±0.85	74.25±0.39	83.44±0.65	70.93±0.67
		1	0.9	0	76.76±0.15	80.77±0.81	60.93±1.31	64.41±0.64	80.36±0.95	62.89±0.52
		1	0.9	0.1	77.49±0.13	83.03±0.11	60.34±0.32	66.95±0.68	81.96±0.33	63.77±0.65
		1	0.9	0.3	79.68±0.33	83.55±0.37	63.80±1.60	69.27±0.98	81.75±1.26	66.63±0.72
		1	0.9	0.5	81.13±0.45	83.57±0.96	65.31±1.93	71.92±0.25	82.02±1.98	68.97±0.20
		1	0.9	0.7	82.06±0.40	83.58±0.72	66.17±1.24	73.17±0.46	82.34±1.22	69.81±0.21
		1	0.9	0.9	82.48±0.40	84.41±0.50	66.49±0.78	74.30±0.41	84.12±0.46	70.88±1.12
		1	1	0	76.77±0.16	82.01±0.40	59.06±2.10	64.06±0.77	80.40±1.07	62.46±0.66
		1	1	0.2	78.72±0.35	83.63±0.52	61.52±0.91	67.75±1.38	82.28±0.68	64.26±1.38
		1	1	0.4	80.80±0.52	83.15±0.45	64.16±1.42	70.79±0.77	81.81±0.89	67.41±0.52
		1	1	0.6	81.51±0.31	83.28±1.64	65.78±0.90	72.15±0.86	82.73±2.04	69.02±0.70
		1	1	0.8	82.40±0.05	84.00±0.79	65.73±1.00	73.90±0.14	83.71±1.21	70.29±0.74
		1	1	1	82.73±0.09	84.88±0.27	66.11±0.31	74.22±0.71	84.50±0.85	70.76±1.49
	2L Max	1	0.7	0	77.53±0.70	79.11±1.71	63.64±1.12	70.89±0.70	77.31±1.90	68.02±0.47
	LR=5×10^{-4}	1	0.7	0.1	78.22±0.33	80.30±1.06	63.49±0.77	71.36±1.30	77.49±0.86	67.26±0.89
		1	0.7	0.3	80.23±0.85	79.16±1.27	62.32±2.22	75.12±1.81	78.37±1.57	72.57±2.06
		1	0.7	0.5	81.94±1.15	80.74±0.69	62.12±1.67	77.91±2.19	78.93±2.38	73.53±1.87
		1	0.7	0.7	82.45±0.23	81.42±2.93	63.23±6.77	77.83±1.53	78.51±0.90	74.52±0.65
		1	0.8	0	76.58±0.62	80.76±1.40	62.80±3.20	70.70±0.25	77.12±0.94	69.12±1.26
		1	0.8	0.2	79.46±0.90	81.96±0.06	63.67±2.34	73.26±1.14	79.09±0.38	69.84±0.76
		1	0.8	0.4	81.28±0.05	81.95±2.75	63.13±3.62	75.03±0.56	79.37±1.32	71.22±1.04
		1	0.8	0.6	82.15±0.31	83.15±1.31	61.26±2.27	78.19±1.36	79.03±1.31	75.11±1.91
		1	0.8	0.8	83.25±0.58	82.67±0.62	65.86±1.03	77.85±0.64	78.90±1.30	75.77±1.13
		1	0.9	0	76.70±0.29	80.58±1.66	63.56±1.36	70.97±0.71	78.56±0.47	68.73±0.60
		1	0.9	0.1	77.71±0.82	81.15±1.13	60.12±0.28	71.83±0.47	77.94±0.84	68.22±1.70
		1	0.9	0.3	79.72±0.56	80.62±2.37	64.21±3.83	74.35±0.17	78.58±1.68	70.73±2.00
		1	0.9	0.5	81.58±0.76	81.21±1.40	66.70±3.76	77.35±0.54	78.93±1.03	72.71±1.35
		1	0.9	0.7	82.11±0.85	80.61±1.01	67.67±2.63	78.42±1.02	78.47±1.65	74.73±1.41
		1	0.9	0.9	***83.65±0.55***	82.39±1.15	65.80±2.27	77.57±0.51	78.99±1.12	74.54±0.82
		1	1	0	77.32±0.39	80.43±2.08	64.13±1.96	70.94±0.31	76.53±0.73	68.64±0.87
		1	1	0.2	78.77±0.03	80.24±0.44	62.37±2.28	72.33±0.32	79.40±1.15	68.31±0.82
		1	1	0.4	80.58±0.53	81.36±0.68	65.16±3.14	75.15±0.75	78.00±1.74	72.06±1.38
		1	1	0.6	81.99±0.60	82.14±1.09	65.71±2.36	77.60±0.54	78.68±1.22	73.65±0.43
		1	1	0.8	82.56±0.70	81.79±0.80	69.67±2.15	***78.51±0.46***	79.95±1.88	75.62±1.00
		1	1	1	83.14±0.33	82.05±0.96	67.55±2.91	76.64±1.01	80.37±1.26	74.13±1.81

E.2.2 SCAFFOLD COVARIATE SHIFT

Table 10 summarizes the GOOD-HIV scaffold covariate-shift results under ID- and OOD-validation-based checkpoint selection.

Table 10: GOOD-HIV scaffold covariate-shift results under ID- and OOD-validation-based checkpoint selection using entropy-based subgraph scoring. ROC–AUC is reported as mean ± sample standard deviation across three runs. The learning rate is indicated below the architecture. Italic indicates the highest validation mean, while bold indicates the results corresponding to the configuration selected by the model-selection procedure. Bold italic indicates cases satisfying both criteria.

Score	Architecture	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
Entropy	2L Max	1	0.7	0	79.13±0.35	77.15±1.40	66.87±2.25	73.65±0.24	77.02±1.27	66.14±1.55
	LR=10^{-5}	1	0.7	0.1	79.69±0.29	78.57±0.70	66.36±2.10	74.75±0.41	78.09±0.87	66.40±2.54
		1	0.7	0.3	82.14±0.27	81.58±0.66	66.91±0.74	76.15±0.43	79.84±0.98	68.65±0.85
		1	0.7	0.5	83.30±0.49	81.66±0.71	69.41±0.99	77.75±0.37	81.08±0.52	70.59±0.36
		1	0.7	0.7	83.60±0.37	81.97±0.29	71.80±1.00	78.82±0.79	81.89±0.26	71.91±1.04
		1	0.8	0	78.73±0.28	76.77±0.70	66.68±0.86	73.90±0.11	76.84±0.67	66.01±1.45
		1	0.8	0.2	80.76±0.10	79.72±0.20	66.66±0.72	75.71±0.10	79.36±0.55	68.15±2.16
		1	0.8	0.4	82.61±0.16	81.21±0.22	68.43±0.39	76.83±0.38	80.38±1.42	69.13±0.95
		1	0.8	0.6	83.20±0.20	82.18±0.61	70.10±0.08	78.39±0.09	82.18±0.74	71.57±0.87
		1	0.8	0.8	83.10±0.21	82.35±0.55	73.49±1.24	78.84±0.25	82.99±0.29	73.56±0.86
		1	0.9	0	78.83±0.11	77.61±0.58	66.58±1.17	74.14±0.17	76.92±1.11	66.72±1.99
		1	0.9	0.1	79.52±0.43	78.10±0.87	67.16±1.63	74.73±0.28	77.37±0.64	66.29±1.86
		1	0.9	0.3	81.65±0.53	80.67±0.11	67.72±1.57	76.28±0.28	79.27±0.60	68.14±1.50
		1	0.9	0.5	82.84±0.09	81.95±0.13	68.97±1.33	77.45±0.62	81.07±1.10	69.33±1.15
		1	0.9	0.7	83.18±0.11	82.39±0.50	71.15±0.79	78.69±0.38	82.68±0.66	72.26±1.38
		1	0.9	0.9	83.69±0.38	83.48±0.75	73.44±1.54	78.90±0.32	83.54±0.76	73.42±0.49
		1	1	0	78.91±0.22	76.90±1.02	67.21±1.50	74.06±0.25	76.37±1.66	66.78±1.86
		1	1	0.2	80.16±0.45	79.36±1.04	67.73±1.56	75.95±0.71	78.55±0.76	67.20±1.52
		1	1	0.4	82.13±0.06	81.27±0.06	68.45±1.50	76.64±0.50	80.97±0.46	69.50±1.01
		1	1	0.6	83.04±0.18	82.11±1.60	68.78±0.73	77.68±0.50	81.20±1.03	70.88±1.04
		1	1	0.8	83.20±0.28	82.71±0.51	73.08±0.86	78.48±0.15	82.35±0.44	72.07±0.77
		1	1	1	83.11±0.23	82.75±0.67	73.51±1.05	78.51±0.22	82.65±0.63	73.89±1.04
	2L Max	1	0.7	0	79.31±0.47	75.69±0.28	66.14±0.70	74.94±0.40	73.95±0.69	65.97±3.87
	LR=5×10^{-4}	1	0.7	0.1	79.28±0.83	77.49±0.44	64.81±1.98	75.69±0.18	78.13±0.43	65.23±3.01
		1	0.7	0.3	81.82±0.75	80.20±0.72	67.54±1.37	77.90±0.49	80.11±1.56	72.00±1.25
		1	0.7	0.5	83.69±0.56	81.46±0.08	67.60±1.22	79.35±0.63	81.32±0.66	70.22±5.13
		1	0.7	0.7	***83.97±0.45***	81.57±0.57	69.35±2.20	79.47±1.11	81.48±1.35	73.86±5.01
		1	0.8	0	78.96±0.45	75.90±0.86	64.27±3.09	74.84±0.38	75.74±1.41	64.77±1.61
		1	0.8	0.2	80.77±0.44	81.69±0.35	68.43±0.94	77.64±0.29	79.30±1.30	67.90±0.77
		1	0.8	0.4	82.11±0.90	81.27±0.54	68.24±2.64	77.79±0.84	80.45±0.71	72.49±1.53
		1	0.8	0.6	83.69±0.25	80.87±1.02	66.73±2.80	78.76±0.57	81.68±1.06	70.40±2.06
		1	0.8	0.8	83.77±0.68	82.32±0.62	71.35±1.97	***79.91±0.31***	81.34±0.78	74.52±1.29
		1	0.9	0	79.38±0.86	75.09±0.59	66.87±1.17	74.45±1.26	75.08±0.99	65.96±0.46
		1	0.9	0.1	79.11±0.12	76.44±0.90	63.86±1.29	76.31±0.48	76.62±1.44	64.07±0.96
		1	0.9	0.3	80.70±0.46	80.95±1.74	65.10±2.36	78.06±1.85	79.60±2.16	69.81±2.66
		1	0.9	0.5	82.68±0.69	82.46±0.26	67.70±2.06	79.23±0.29	80.69±2.02	68.16±1.60
		1	0.9	0.7	83.66±0.60	81.03±0.62	71.18±2.30	78.88±0.40	81.37±0.60	71.03±3.77
		1	0.9	0.9	83.56±0.12	82.38±0.68	72.24±2.25	79.59±0.52	82.31±1.33	72.78±3.36
		1	1	0	78.38±0.33	76.22±0.48	64.64±2.32	74.67±0.81	75.31±0.91	65.00±1.38
		1	1	0.2	80.65±0.11	79.69±0.99	65.17±1.63	77.56±0.70	79.07±0.96	65.35±1.87
		1	1	0.4	82.30±0.82	81.47±1.14	69.38±4.05	78.35±0.44	81.78±0.49	72.30±1.47
		1	1	0.6	82.66±0.67	81.67±0.30	70.30±3.10	79.17±0.38	81.72±0.50	73.35±2.59
		1	1	0.8	83.27±0.55	82.09±2.28	72.86±3.06	79.77±0.33	80.87±1.97	74.89±1.92
		1	1	1	83.15±0.29	81.89±0.80	71.18±2.54	79.75±0.73	81.13±1.36	74.53±2.11

E.3 GOOD-MOTIF

We evaluate GOOD-Motif under base concept and covariate shifts using classification accuracy. Both Entropy- and Margin-based subgraph scoring are evaluated across the considered edge-weight configurations. Results are reported under ID- and OOD-validation-based checkpoint selection.

E.3.1 BASE CONCEPT SHIFT

Table 11 summarizes the GOOD-Motif base concept-shift results under ID- and OOD-validation-based checkpoint selection.

Table 11: GOOD-Motif base concept-shift results under ID- and OOD-validation-based checkpoint selection. Accuracy is reported as mean ± sample standard deviation across six runs. Italic indicates the highest validation mean within each scoring function, while bold indicates the results corresponding to the configuration selected by the model-selection procedure. Bold italic indicates cases satisfying both criteria.

Score	Arch.	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
Entropy	5L Sum	1	0.7	0	92.74±0.05	91.84±0.13	77.97±2.67	87.14±0.50	91.51±0.40	83.88±0.94
		1	0.7	0.1	93.04±0.03	92.25±0.04	87.04±1.51	89.33±0.80	92.23±0.12	87.58±1.46
		1	0.7	0.3	93.12±0.02	92.23±0.06	88.30±1.07	90.67±0.15	92.29±0.04	89.72±0.21
		1	0.7	0.5	93.19±0.02	92.24±0.07	90.21±1.25	92.25±0.20	92.28±0.03	92.15±0.19
		1	0.7	0.7	93.20±0.04	92.28±0.05	90.88±0.83	91.84±0.33	92.28±0.09	91.73±0.27
		1	0.8	0	92.79±0.03	91.77±0.06	79.21±1.32	87.55±0.52	91.69±0.07	84.38±0.75
		1	0.8	0.2	93.05±0.05	92.26±0.05	87.26±0.76	89.63±0.46	92.17±0.20	88.10±0.59
		1	0.8	0.4	93.18±0.03	92.27±0.04	89.85±0.67	91.37±0.12	92.26±0.04	90.87±0.23
		1	0.8	0.6	93.19±0.03	92.28±0.06	90.16±1.05	91.88±0.26	92.28±0.04	91.69±0.28
		1	0.8	0.8	93.25±0.02	92.34±0.07	91.66±0.75	92.62±0.23	92.40±0.02	92.69±0.28
		1	0.9	0	92.78±0.05	91.85±0.11	78.84±1.46	86.98±0.53	91.61±0.07	83.43±0.79
		1	0.9	0.1	93.02±0.03	92.21±0.04	84.28±0.49	87.77±0.22	92.15±0.08	84.77±0.21
		1	0.9	0.3	93.09±0.02	92.24±0.05	88.39±0.43	90.21±0.24	92.25±0.07	88.94±0.43
		1	0.9	0.5	93.20±0.02	92.18±0.11	89.63±0.90	91.86±0.32	92.31±0.04	91.67±0.47
		1	0.9	0.7	93.25±0.02	92.36±0.05	91.06±0.94	92.36±0.14	92.40±0.02	92.19±0.25
		1	0.9	0.9	*93.26±0.00*	92.33±0.03	91.40±0.81	*92.75±0.07*	92.38±0.03	92.76±0.09
		1	1	0	92.73±0.03	91.94±0.10	78.18±1.27	87.27±0.59	91.60±0.12	83.97±0.80
		1	1	0.2	92.96±0.11	92.22±0.05	86.98±2.26	89.51±0.96	92.16±0.09	87.82±1.76
		1	1	0.4	93.18±0.02	92.21±0.07	88.26±0.90	90.95±0.47	92.20±0.07	90.19±0.71
		1	1	0.6	93.24±0.04	92.27±0.04	90.97±1.05	92.05±0.39	92.28±0.05	91.78±0.33
		1	1	0.8	93.25±0.02	92.32±0.08	90.46±1.91	92.54±0.17	92.41±0.00	92.63±0.20
		1	1	1	93.24±0.02	92.35±0.04	91.49±0.48	92.74±0.12	92.35±0.03	92.62±0.19
Margin	5L Sum	1	0.7	0	92.75±0.07	91.94±0.10	77.02±2.71	86.15±0.74	91.48±0.09	82.46±1.29
		1	0.7	0.1	93.05±0.02	92.24±0.04	85.00±1.71	88.55±0.87	92.10±0.19	86.54±1.43
		1	0.7	0.3	93.16±0.04	92.28±0.04	88.31±1.24	90.78±0.23	92.23±0.04	89.84±0.37
		1	0.7	0.5	93.16±0.02	92.26±0.05	90.14±0.50	91.93±0.11	92.31±0.01	91.86±0.16
		1	0.7	0.7	93.22±0.03	92.28±0.03	91.28±0.61	91.88±0.21	92.28±0.04	91.73±0.28
		1	0.8	0	92.79±0.11	91.84±0.18	77.66±2.29	86.81±0.43	91.48±0.12	83.44±0.68
		1	0.8	0.2	93.04±0.02	92.21±0.04	85.96±1.17	88.97±0.27	92.16±0.10	87.07±0.46
		1	0.8	0.4	93.19±0.03	92.25±0.09	89.02±0.65	91.42±0.36	92.27±0.05	90.99±0.58
		1	0.8	0.6	93.20±0.02	92.26±0.04	90.13±0.92	91.58±0.28	92.28±0.05	91.22±0.31
		1	0.8	0.8	93.25±0.02	92.35±0.12	91.60±1.65	92.54±0.29	92.39±0.02	92.53±0.31
		1	0.9	0	92.81±0.06	91.88±0.11	78.35±1.41	86.62±0.49	91.46±0.12	83.27±0.78
		1	0.9	0.1	93.05±0.07	92.17±0.06	83.38±0.99	87.48±0.24	91.91±0.18	84.45±0.29
		1	0.9	0.3	93.09±0.05	92.22±0.08	87.48±1.68	90.35±0.38	92.20±0.06	89.11±0.53
		1	0.9	0.5	**93.22±0.03**	92.24±0.06	89.90±1.04	91.79±0.17	92.31±0.03	91.55±0.22
		1	0.9	0.7	*93.26±0.00*	92.36±0.04	90.64±0.93	91.97±0.33	92.38±0.03	91.85±0.41
		1	0.9	0.9	93.25±0.02	92.30±0.12	90.98±1.18	***92.75±0.06***	92.40±0.02	92.77±0.11
		1	1	0	92.78±0.05	91.75±0.15	77.80±1.65	86.52±0.72	91.42±0.13	82.98±1.13
		1	1	0.2	93.01±0.07	92.17±0.09	86.01±0.97	88.99±0.25	92.14±0.13	87.14±0.46
		1	1	0.4	93.19±0.00	92.17±0.11	88.94±0.69	91.18±0.25	92.28±0.03	90.54±0.53
		1	1	0.6	93.22±0.03	92.26±0.06	90.06±1.43	91.67±0.32	92.26±0.05	91.28±0.42
		1	1	0.8	93.24±0.02	92.32±0.08	90.40±1.61	92.44±0.35	92.39±0.02	92.49±0.43
		1	1	1	93.25±0.02	92.30±0.07	90.41±1.46	92.72±0.09	92.34±0.03	92.59±0.22

E.3.2 BASE COVARIATE SHIFT

Table 12 summarizes the GOOD-Motif base covariate-shift results under ID- and OOD-validation-based checkpoint selection.

Table 12: GOOD-Motif base covariate-shift results under ID- and OOD-validation-based checkpoint selection. Accuracy is reported as mean ± sample standard deviation across six runs. Italic indicates the highest validation mean within each scoring function, while bold indicates the results corresponding to the configuration selected by the model-selection procedure. Bold italic indicates cases satisfying both criteria.

Score	Arch.	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
Entropy	5L Sum	1	0.7	0	93.01±0.03	92.69±0.01	93.41±0.27	92.60±0.00	91.99±0.23	93.08±0.27
		1	0.7	0.1	93.05±0.02	92.66±0.05	93.83±0.02	92.65±0.06	92.49±0.09	93.58±0.24
		1	0.7	0.3	93.09±0.02	92.69±0.03	93.14±0.27	92.71±0.18	92.38±0.42	93.23±0.25
		1	0.7	0.5	93.13±0.00	92.81±0.02	77.83±2.05	93.17±0.00	92.81±0.10	77.13±11.95
		1	0.7	0.7	93.13±0.00	92.89±0.01	45.34±5.59	92.90±0.22	92.82±0.16	52.29±10.85
		1	0.8	0	93.03±0.00	92.66±0.02	93.74±0.21	92.60±0.00	91.87±0.61	93.18±0.11
		1	0.8	0.2	93.05±0.02	92.72±0.03	93.67±0.26	92.98±0.14	92.40±0.62	93.74±0.24
		1	0.8	0.4	93.12±0.03	92.82±0.02	66.66±15.30	93.17±0.00	92.58±0.04	93.44±0.06
		1	0.8	0.6	93.13±0.00	92.88±0.02	54.24±7.50	93.12±0.02	92.31±0.91	84.95±16.31
		1	0.8	0.8	93.13±0.00	92.82±0.04	58.88±10.09	93.18±0.02	92.77±0.08	64.12±9.54
		1	0.9	0	93.01±0.01	92.69±0.04	93.73±0.25	93.12±0.08	91.07±0.23	93.08±0.26
		1	0.9	0.1	93.02±0.02	92.64±0.10	93.76±0.20	92.98±0.20	92.12±0.58	93.59±0.30
		1	0.9	0.3	93.05±0.02	92.78±0.05	93.18±0.27	93.17±0.00	92.14±0.37	93.22±0.18
		1	0.9	0.5	93.13±0.00	92.86±0.07	50.75±6.92	*93.19±0.02*	92.57±0.11	89.24±7.70
		1	0.9	0.7	93.13±0.00	92.83±0.03	52.15±8.84	93.13±0.05	92.55±0.66	72.02±25.12
		1	0.9	0.9	93.13±0.00	92.89±0.03	63.56±3.39	93.16±0.02	92.83±0.15	58.28±12.41
		1	1	0	93.01±0.01	92.71±0.04	93.84±0.02	93.15±0.02	91.41±0.23	93.18±0.10

Continued on the next page

Table 12 continued from the previous page

Score	Arch.	w_I	w_B	w_O	ID-Val Best			OOD-Val Best		
					ID Val	ID Test	OOD Test	OOD Val	ID Test	OOD Test
		1	1	0.2	93.05±0.02	92.76±0.05	93.79±0.16	93.17±0.02	92.57±0.07	93.67±0.24
		1	1	0.4	***93.14±0.02***	92.81±0.05	72.96±9.64	93.18±0.01	92.43±0.14	93.70±0.12
		1	1	0.6	93.13±0.00	92.88±0.05	68.62±14.26	93.09±0.03	91.58±1.72	82.90±18.96
		1	1	0.8	93.13±0.00	92.80±0.02	57.78±2.55	93.16±0.04	92.76±0.13	65.74±14.68
		1	1	1	92.84±0.80	92.66±0.63	52.68±7.24	93.15±0.02	92.98±0.05	52.56±2.47
Margin	5L Sum	1	0.7	0	93.00±0.02	92.67±0.02	93.40±0.04	93.17±0.00	92.17±0.17	93.15±0.20
		1	0.7	0.1	93.02±0.01	92.66±0.03	93.91±0.05	93.17±0.00	92.51±0.06	93.84±0.22
		1	0.7	0.3	93.06±0.02	92.65±0.03	93.12±0.21	93.18±0.01	92.54±0.10	93.51±0.05
		1	0.7	0.5	93.14±0.02	92.82±0.02	58.51±8.09	93.18±0.02	92.75±0.04	79.49±16.38
		1	0.7	0.7	*93.17±0.00*	92.82±0.05	43.83±5.01	93.06±0.14	90.36±3.21	57.50±17.80
		1	0.8	0	93.02±0.02	92.65±0.05	93.89±0.19	93.17±0.00	91.87±0.26	93.21±0.19
		1	0.8	0.2	93.03±0.02	92.65±0.04	93.73±0.36	93.19±0.02	92.53±0.24	93.65±0.29
		1	0.8	0.4	93.11±0.01	92.73±0.05	68.86±16.40	93.17±0.00	92.45±0.23	93.47±0.08
		1	0.8	0.6	93.14±0.02	92.89±0.04	72.16±15.43	93.12±0.03	92.53±0.25	93.31±0.34
		1	0.8	0.8	*93.17±0.00*	92.83±0.05	51.18±4.64	93.16±0.05	92.32±1.01	57.17±11.07
		1	0.9	0	92.98±0.01	92.64±0.03	93.97±0.03	93.17±0.00	91.24±0.29	93.31±0.04
		1	0.9	0.1	92.98±0.02	92.67±0.04	93.88±0.23	93.17±0.00	92.46±0.05	93.84±0.23
		1	0.9	0.3	93.05±0.02	92.69±0.04	93.38±0.21	93.19±0.02	92.58±0.14	93.53±0.04
		1	0.9	0.5	93.14±0.02	92.81±0.02	46.49±3.42	*93.20±0.00*	92.74±0.08	68.22±17.65
		1	0.9	0.7	*93.17±0.00*	92.81±0.05	55.18±15.89	93.13±0.06	92.22±1.21	85.98±15.03
		1	0.9	0.9	*93.17±0.00*	92.95±0.02	55.18±3.47	93.18±0.02	91.31±3.86	72.82±11.40
		1	1	0	92.98±0.01	92.67±0.03	93.98±0.03	**93.17±0.00**	92.04±0.39	93.31±0.18
		1	1	0.2	93.04±0.02	92.67±0.04	93.88±0.17	93.18±0.01	92.56±0.24	93.73±0.22
		1	1	0.4	93.09±0.03	92.78±0.02	67.76±14.84	93.19±0.02	92.61±0.15	85.91±15.24
		1	1	0.6	93.15±0.02	92.86±0.03	69.23±15.23	93.10±0.05	92.49±0.24	93.82±0.20
		1	1	0.8	*93.17±0.00*	92.81±0.03	58.76±7.98	93.18±0.02	92.70±0.20	64.36±12.69
		1	1	1	*93.17±0.00*	92.92±0.02	58.76±3.28	93.13±0.04	92.99±0.04	55.46±1.93

F Sensitivity Analysis of the Near-Optimality Threshold

The model-selection procedure uses a predefined tolerance ϵ to identify near-optimal edge-weight configurations. In the main experiments, we use $\epsilon = 0.05$ percentage points. To examine sensitivity to this choice, we vary ϵ from 0.05 to 0.30 percentage points and apply the same validation-based model-selection procedure described in Section 4.2. As ϵ increases, more configurations are considered near-optimal. We therefore treat ϵ as a predefined tolerance rather than selecting it based on validation or test performance.

Figure 3 shows that validation and matched test performance are generally stable across the evaluated values of ϵ, particularly under OOD-validation-based selection. Larger variations are mainly observed in the cross-setting evaluation, especially for GOOD-Motif. Overall, the results indicate that the main conclusions are not strongly dependent on the exact choice of ϵ, supporting the use of $\epsilon = 0.05$ as a small predefined tolerance.

G Ablation Study on Subgraph Construction and Selection

We perform an ablation study to examine two design choices in GraphSteer: (i) whether retaining the full graph through soft edge reweighting is beneficial compared with using only the selected subgraph, and (ii) whether incorporating node-feature information into Leiden community detection improves performance over the original topology-based Leiden algorithm. Table 13 compares the following variants:

- **Selected Subgraph Only**: We retain the GraphSteer pipeline for constructing and selecting the informative subgraph. Specifically, candidate communities are first identified using Leiden, scored by the subgraph scoring network, and the top-ranked subgraph is selected. Instead of applying the GraphSteer edge-reweighting scheme to the original graph, we then prune all nodes and edges outside the selected subgraph and train the downstream GNN only on the resulting subgraph. For GOOD-CMNIST and GOOD-HIV, the selected subgraph is obtained using the node-feature-aware Leiden variant, whereas for GOOD-Motif, the original Leiden algorithm is used because node features are unavailable.
- **Original Leiden**: GraphSteer with the standard Leiden algorithm, where communities are constructed solely from graph topology. To isolate the effect of community construction, we do not perform an additional weight search for this variant. Instead, for each dataset, shift, and validation criterion, we use the same weight configuration (w_I, w_B, w_O) selected for the corresponding Node-Feature-Aware Leiden setting.

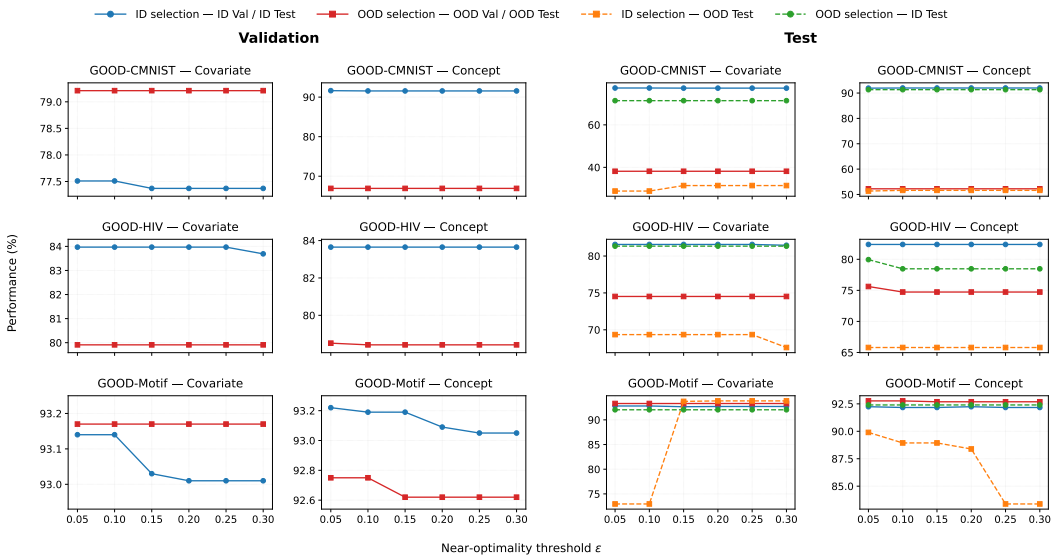

Figure 3: Sensitivity of validation and test performance to the near-optimality threshold ϵ under ID- and OOD-validation-based model selection. Solid lines denote matched selection and evaluation settings, while dashed lines denote cross-setting test evaluation. Results are shown for GOOD-CMNIST, GOOD-HIV, and GOOD-Motif under covariate and concept shifts. GOOD-CMNIST and GOOD-Motif report accuracy (%), while GOOD-HIV reports ROC–AUC (%).

- **Node-Feature-Aware Leiden**: GraphSteer with our node-feature-aware extension of Leiden, where node-feature similarity is incorporated into community construction in addition to graph topology. For each dataset, shift, and validation criterion, the weight configuration (w_I, w_B, w_O) is selected using the validation-based selection procedure described in Section 4.2, with $\epsilon = 0.05$ percentage points.

Figure 4 provides a qualitative illustration of the difference between the original and node-feature-aware Leiden variants. While the original Leiden algorithm forms communities based only on graph connectivity, the node-feature-aware variant also incorporates node-feature similarity during community construction. As illustrated on a GOOD-CMNIST example, this can lead to different community boundaries even when both methods operate on the same input graph.

Overall, the results show that retaining the full graph through soft edge reweighting is generally preferable to using only the selected subgraph. On GOOD-CMNIST and GOOD-HIV, the Selected Subgraph Only variant consistently performs worse than the soft-reweighting variants, particularly on OOD test data. This suggests that information outside the selected subgraph can remain useful for prediction and is better downweighted than completely removed.

An exception occurs on GOOD-Motif under covariate shift. Under ID-based checkpoint selection, the Selected Subgraph Only variant achieves higher OOD-test performance than soft reweighting. We attribute this to the synthetic construction of GOOD-Motif: the prediction target is determined by the motif, while the attached base graph does not provide useful information about the target. Consequently, hard pruning to the selected predictive subgraph can be feasible for this dataset.

Table 13: Ablation study on subgraph construction and utilization. *Selected Subgraph Only* retains the GraphSteer community detection and top-ranked subgraph selection procedure but removes the remaining graph instead of applying soft edge reweighting. *Original Leiden* and *Node-Feature-Aware Leiden* retain the full GraphSteer reweighting procedure and differ only in how candidate communities are constructed. For each dataset and shift, ID selection uses the checkpoint with the best ID-validation performance, while OOD selection uses the checkpoint with the best OOD-validation performance. GOOD-CMNIST and GOOD-Motif report accuracy (%), while GOOD-HIV reports ROC–AUC (%). Node-feature-aware Leiden is not applicable to GOOD-Motif because node features are unavailable.

Data	Shift	Selection	Variant	ID Test	OOD Test
CMNIST	Covariate	ID	Selected Subgraph Only	37.10 ± 0.37	25.34 ± 6.57
			Original Leiden	77.80 ± 0.17	28.39 ± 4.18
			Node-Feature-Aware Leiden	77.35 ± 0.12	28.84 ± 4.48
		OOD	Selected Subgraph Only	36.59 ± 0.63	30.49 ± 2.37
			Original Leiden	70.23 ± 0.59	32.96 ± 2.99
			Node-Feature-Aware Leiden	71.39 ± 2.12	38.16 ± 4.10
	Concept	ID	Selected Subgraph Only	87.94 ± 0.27	25.29 ± 2.25
			Original Leiden	87.88 ± 0.39	32.21 ± 0.48
			Node-Feature-Aware Leiden	91.93 ± 0.11	51.34 ± 0.85
		OOD	Selected Subgraph Only	87.47 ± 0.19	29.12 ± 0.55
			Original Leiden	87.10 ± 0.64	35.41 ± 0.44
			Node-Feature-Aware Leiden	91.33 ± 0.27	52.23 ± 0.86
HIV	Covariate	ID	Selected Subgraph Only	66.38 ± 0.14	55.81 ± 0.78
			Original Leiden	83.21 ± 0.90	71.15 ± 1.47
			Node-Feature-Aware Leiden	81.57 ± 0.57	69.35 ± 2.20
		OOD	Selected Subgraph Only	66.16 ± 0.43	55.47 ± 2.65
			Original Leiden	81.26 ± 0.75	73.75 ± 1.30
			Node-Feature-Aware Leiden	81.34 ± 0.78	74.52 ± 1.29
	Concept	ID	Selected Subgraph Only	69.36 ± 1.10	55.62 ± 1.33
			Original Leiden	82.51 ± 1.17	68.03 ± 4.99
			Node-Feature-Aware Leiden	82.39 ± 1.15	65.80 ± 2.27
		OOD	Selected Subgraph Only	68.44 ± 0.63	57.24 ± 0.91
			Original Leiden	80.98 ± 1.12	74.03 ± 0.41
			Node-Feature-Aware Leiden	79.95 ± 1.88	75.62 ± 1.00
Motif	Covariate	ID	Selected Subgraph Only	92.66 ± 0.02	92.47 ± 0.00
			Original Leiden	92.81 ± 0.05	72.96 ± 9.64
			Node-Feature-Aware Leiden	–	–
		OOD	Selected Subgraph Only	91.62 ± 0.47	92.41 ± 0.16
			Original Leiden	92.04 ± 0.39	93.31 ± 0.18
			Node-Feature-Aware Leiden	–	–
	Concept	ID	Selected Subgraph Only	91.54 ± 0.08	83.62 ± 0.59
			Original Leiden	92.24 ± 0.06	89.90 ± 1.04
			Node-Feature-Aware Leiden	–	–
		OOD	Selected Subgraph Only	91.62 ± 0.02	84.57 ± 0.36
			Original Leiden	92.40 ± 0.02	92.77 ± 0.11
			Node-Feature-Aware Leiden	–	–

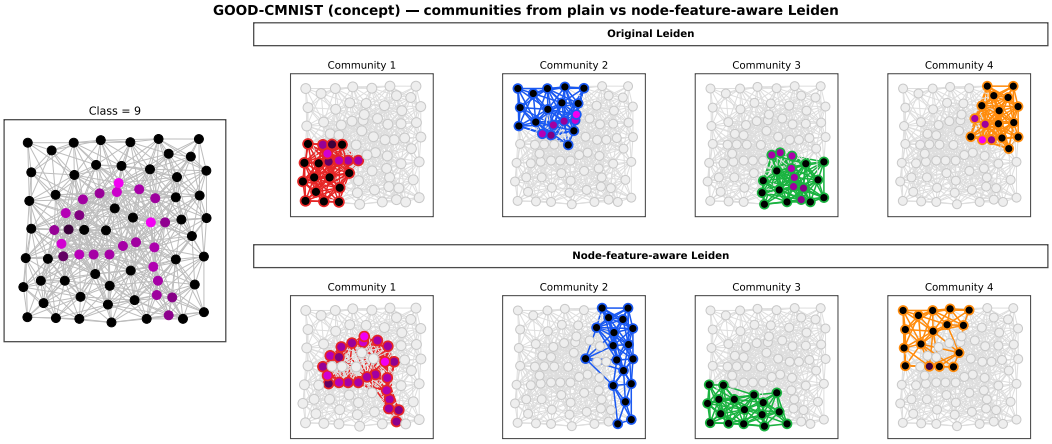

Figure 4: Qualitative comparison of community construction using original Leiden and node-feature-aware Leiden on a GOOD-CMNIST concept-shift example. The left panel shows the original graph, while the top and bottom rows show the communities identified by original Leiden and node-feature-aware Leiden, respectively. Original Leiden uses graph topology only, whereas node-feature-aware Leiden additionally incorporates node-feature similarity.