Test-Time Adaptation for Graph Learning: A Systematic Survey

Jiayi Chen[1], **Xin Zheng**[2*], **Bo Li**[3], **Zeyu Wang**[3,4], **Yanqing Guo**[1], **Feng Xia**[2*]
[1]Dalian University of Technology, China
[2]RMIT University, Australia
[3]Griffith University, Australia
[4]Zhejiang University of Technology, China
{chenjy@mail.dlut.edu.cn}, {xin.zheng2@rmit.edu.au}, {bo.li6@griffithuni.edu.au}, {vencent_wang@outlook.com}, {guoyq@dlut.edu.cn}, {f.xia@ieee.org}

Abstract

Graph distribution shifts between training and test graphs pose severe challenges to the generalization of graph neural networks (GNNs). In real-world deployment, application environments are continuously evolving, while retraining or redesigning GNNs is often costly and impractical. In light of this, **test-time adaptation on graphs**, which aims to dynamically adapt well-trained GNNs or adjust test graphs to improve inference performance, has attracted growing attention as a practical solution. In this survey, we provide a comprehensive review of test-time adaptation on graphs, an emerging yet underexplored research direction. We identify two fundamental challenges: (1) <u>Data-level</u>: *complex graph distribution shifts*; and (2) <u>Model-level</u>: *limited test-time learning information*. Upon this, we present a systematic taxonomy of existing methods into ❶ model-centric, ❷ data-centric, and ❸ hybrid methods, followed by a summary of representative applications, benchmarks, and open opportunities. We aim to bridge the gap between laboratory GNN development and real-world deployment via test-time adaptation. The repository of papers, code, and datasets is at https://github.com/jiayichen1121/Test-Time-Adaptation-for-Graph-Learning.

1 Introduction

Graphs have become a fundamental data type for representing complex relational systems across a wide range of real-world applications, including social networks [Tran *et al.*, 2025], recommender systems [Liu *et al.*, 2025], and chemical networks [Wang *et al.*, 2025], etc. To effectively model diverse and complex characteristics of graphs, graph neural networks (GNNs) have become a learning model, with expressive performance across various graph learning tasks [Zheng *et al.*, 2023a]. By jointly learning with node attributes and graph structure, GNNs capture complex dependencies of relational entities, leading to powerful node-level and graph-level representations [Pan *et al.*, 2025; Zheng *et al.*, 2023c].

Despite the promising performance of existing GNNs,

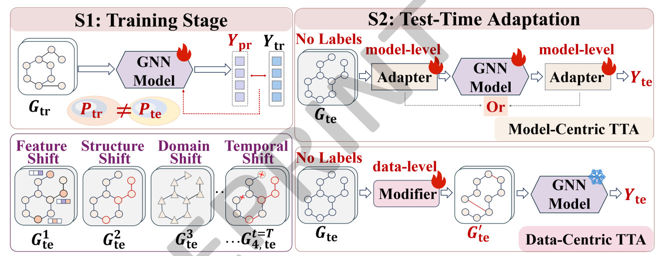

Figure 1: Overview of test-time adaptation on graphs.

graph distribution shifts, i.e., where the statistical characteristics of the test graphs deviate from those of the training graphs [Jin *et al.*, 2023], pose the severe challenge to the *generalization ability* of well-trained GNNs. In this case, a GNN model that performs well on the training distribution often experiences significant performance degradation when deployed on distribution-shifted test graphs at test time. However, in the real world, graph data are continuously evolving due to changes in the environment, and it is a natural case that previously unseen test distributions potentially differ from those in training. Meanwhile, it is impractical to retrain GNNs or develop new GNN architectures to address potential distribution shifts.

In light of these, *test-time adaptation (TTA)*, which aims to dynamically fine-tune the pre-trained model to generalize and adapt well to the test data [Liang *et al.*, 2025], has been increasingly explored, originally in the computer vision domain on images with convolutional neural network (CNN) models. However, when dealing with graph data, the problem becomes more complex due to its structured and non-Euclidean nature, and the fundamental difference in model architectures between GNNs and CNNs. Hence, it is necessary to systematically investigate **test-time adaptation on graphs** to enable robust generalization under distribution shifts for real-world graph learning problems.

Key Challenges of Test-Time Adaptation on Graphs. The limited generalization ability of GNNs on unseen graphs under distribution shifts can be primarily attributed to two factors, which define the key challenges for test-time adaptation on graphs:

- **(C1)** [**Data-level**]**: Complex graph distribution shifts**, which mainly come from the underlying environmental variations, with time-related attribute changes, agnostic corruptions, and inconsistent graph data collection procedures [Sui *et al.*, 2023; Jin *et al.*, 2023]. These changes would result in significant differences in node contexts, graph structures, and the overall scale of graphs during test time. Typically, graph distribution shifts can be generally categorized into the following groups: (a) Node feature and graph structure shift; (b) Graph domain shift; (c) Temporal shift; and (d) Label shift, and these types of graph distribution shifts typically affect node features, graph structure, and labels, in complex and dataset-specific ways.

- **(C2)** [**Model-level**]**: Limited information for test-time learning**, which arises from two inherent constraints in practical GNN deployment scenarios: (a) Since test graph labels & the training graphs are usually unavailable, test-time learning has to rely on unsupervised learning with self-supervised signals, inevitably imposing an upper bound on achievable performance. (b) Given that deployed GNNs are often not allowed to update a large number of model parameters, test-time learning is required to fine-tune minimal model parameters, or ideally, no modification of well-trained GNNs.

In light of these challenges, recent research attention has been shifted to the study of test-time adaptation on graphs for improving GNN generalization ability under distribution shifts, without access to the training graphs or test graph ground-truth labels at test time.

Importance of Test-Time Adaptation on Graphs. Test-time graph learning offers insightful perspectives in the stage of test-time model deployment in real-world graph learning.

First, it narrows the lab-to-practice GNN deployment gap. For many years, GNN research has mainly focused on model design, where GNNs can be continuously trained, evaluated, and adjusted in a controllable environment and tidy graph datasets. Hence, the lab research is more like a ‘safe house’ for model development. However, in real-world deployment platforms and application scenarios, graph data are often constantly evolving in complex distributions at test time, and large-scale model fine-tuning is usually not allowed once the system is online, as it may interrupt user services and incur huge costs. In this case, test-time adaptation enables GNNs to adapt to practical conditions without model retraining or disruptive parameter updates during inference. Hence, it effectively narrows the gap between idealized laboratory GNN design and the constraints of real-world GNN deployment.

Second, it promotes the cost-effective model reuse of well-trained GNNs. Given the great effort that we already put into GNN model design for practical deployment, it is important to maximize these benefits by enabling stronger model adaptability and generalization ability when facing diverse distribution shifts and evolving application scenarios. In this context, test-time adaptation provides a lightweight and effective alternative by allowing existing well-trained GNN models to be reused and adapted at inference time. This approach substantially reduces the deployment cost, extending the practical value of well-trained and well-designed GNN models.

In this paper, we present a comprehensive and systematic review of test-time adaptation on graphs for real-world graph learning tasks, aiming to provide a general blueprint of test-time graph learning research. The contributions of our work are summarized as follows:

- **Comprehensive Overview**: We provide a comprehensive overview of current test-time adaptation methods for graph learning in terms of challenges, methods, applications.
- **Systematic Taxonomy**: We provide a systematic taxonomy of test-time adaptation methods for graph learning and categorize them into three groups, i.e., model-centric methods, graph data-centric methods, and hybrid methods.
- **Diverse Application Scenarios**: We provide a thorough discussion of test-time adaptation methods for various graph learning scenarios in real-world applications.
- **Future Directions**: We suggest promising future research directions and discuss the limitations of existing test-time adaptation methods on graphs from multiple perspectives, including strong adaptation ability to complex graph distribution shifts, diverse graph types, and a wide range of graph learning tasks, as well as the need for theoretical guarantees, robustness, explainability, and scalability to enable effective graph learning at test time.

Differences between our survey and existing work [Zhang *et al.*, 2024c]. While [Zhang *et al.*, 2024c] reviews deep graph learning under distribution shifts, our survey differs in three key aspects. For **scope**, it focuses on general out-of-distribution (OOD) generalization and adaptation up to 2024, whereas we specifically target test-time adaptation for graph learning up to date (early 2026). For **taxonomy**, it adopts a coarse model-/data-centric split, while we present a finer-grained technical taxonomy covering diverse TTA mechanisms and application scenarios on graphs. For **future directions**, it discusses high-level opportunities such as trustworthiness and graph foundation models, while we provide more concrete and actionable directions on TTA techniques and deployment for effective graph learning.

2 Problem Definition

Notation. Given a training graph $\mathcal{G}_{\text{tr}} = (\mathbf{X}_{\text{tr}}, \mathbf{A}_{\text{tr}}, \mathbf{Y}_{\text{tr}}) \sim \mathcal{P}_{\text{tr}}$ and a test graph $\mathcal{G}_{\text{te}} = (\mathbf{X}_{\text{te}}, \mathbf{A}_{\text{te}}) \sim \mathcal{P}_{\text{te}}$, we have $\mathbf{X}_{\{\text{tr,te}\}}$ denote node features, $\mathbf{A}_{\{\text{tr, te}\}}$ denotes adjacency matrix indicating graph structure, and $\mathbf{Y}_{\text{tr}}$ is the training set label. For node classification task, it indicates the node labels; While for graph classification or regression task, it denotes the graph label. The graph distribution shift makes $\mathcal{P}_{\text{tr}} \neq \mathcal{P}_{\text{te}}$.

Training Stage. A GNN model with learnable parameters $\boldsymbol{\theta}$ is trained on $\mathcal{G}_{\text{tr}}$ with the following objective:

$$\boldsymbol{\theta}^*_{\text{tr}} = \arg\min_{\boldsymbol{\theta}_{\text{tr}}} \mathcal{L}\big(\text{GNN}_{\boldsymbol{\theta}_{\text{tr}}}(\mathbf{X}_{\text{tr}}, \mathbf{A}_{\text{tr}}), \mathbf{Y}_{\text{tr}}\big) \Rightarrow \text{GNN}_{\boldsymbol{\theta}^*_{\text{tr}}}. \quad (1)$$

Here, $\mathcal{L}$ denotes a task-specific loss function, e.g., cross-entropy for node classification. The training process yields a well-trained model, denoted as $\text{GNN}\boldsymbol{\theta}^*_{\text{tr}}$, which is typically fixed and deployed for test-time inference.

Test-Time Inference. Without any test-time adaptation, the standard test-time inference aims to make the prediction for:

$$\widehat{\mathbf{Y}}_{\text{te}} = \text{GNN}_{\boldsymbol{\theta}^*_{\text{tr}}}(\mathbf{X}_{\text{te}}, \mathbf{A}_{\text{te}}) \Rightarrow \widehat{\mathbf{Y}}_{\text{te}}. \quad (2)$$

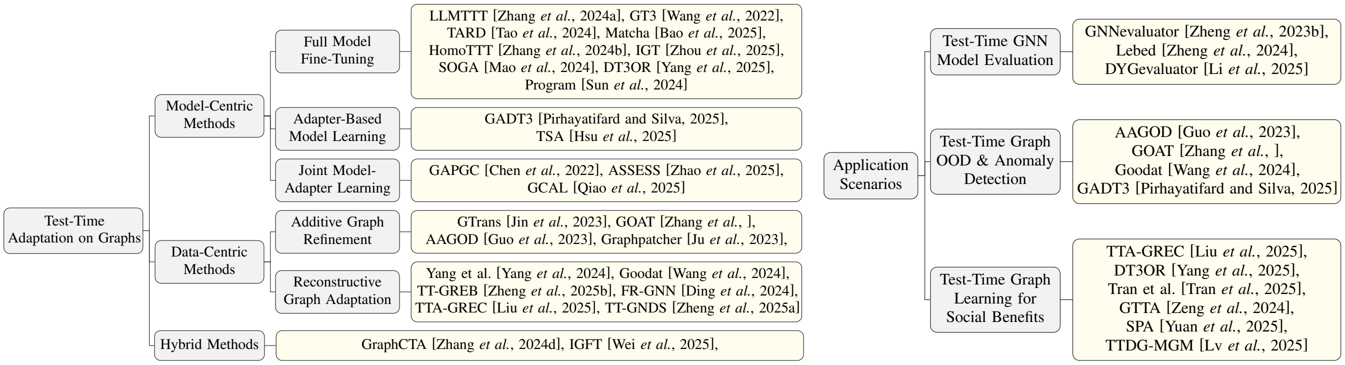

Figure 2: A comprehensive taxonomy of test-time adaptation for graph learning.

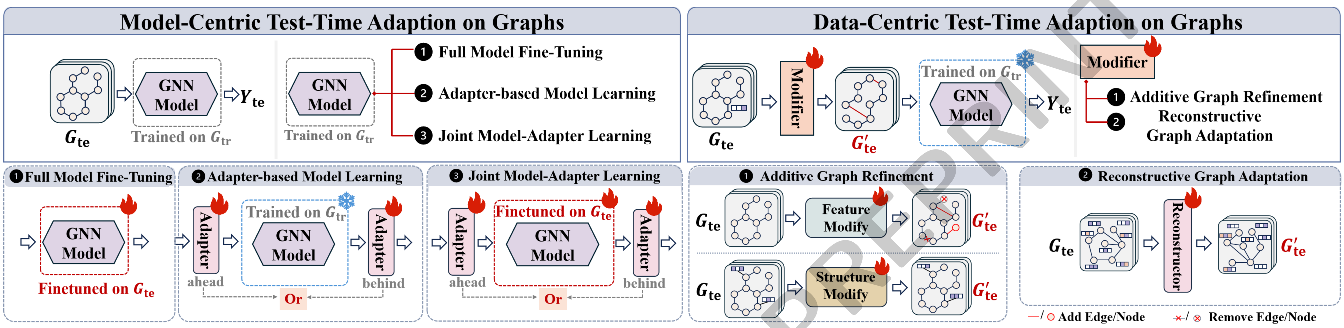

Figure 3: Methodology overview of model-centric (left) and data-centric (right) test-time adaptation on graphs.

However, under the graph distribution shift, direct inference using the fixed well-trained GNN often leads to degraded performance of $\widehat{\mathbf{Y}}_{te}$ for graph learning tasks at test time.

PROBLEM DEFINITION [TEST-TIME ADAPTATION ON GRAPHS]. *Given a well-trained* $\mathrm{GNN}_{\boldsymbol{\theta}^*_{tr}}$ *model and a test graph* $\mathcal{G}_{te}$*, test-time adaptation on graphs aims to dynamically adapt these two objectives by learning following functions:*

$$\begin{aligned} \textit{Model-Centric:} \quad & \mathcal{A}_\phi : \boldsymbol{\theta}^*_{\text{tr}} \mapsto \boldsymbol{\theta}_{\{\text{tr,te}\}}; \\ \textit{Data-Centric:} \quad & \mathcal{M}_\psi : \mathcal{G}_{\text{te}} \mapsto \mathcal{G}'_{\text{te}}, \end{aligned} \tag{3}$$

where $\mathcal{A}_\phi$ *denotes the model adapter function for optimizing the well-trained GNN to fit both training and test distributions for better test-time performance, and* $\mathcal{M}_\psi$ *denotes the test graph modification function to project the unlabeled test graph to the training distribution for effectively reusing the well-trained* $\mathrm{GNN}_{\boldsymbol{\theta}^*_{tr}}$.

In Table 1, we compare the strategies of existing methods to address graph distribution shifts, including graph domain generalization, domain adaptation, unsupervised domain adaptation, test-time training, as well as model-centric and data-centric test-time adaptation. The comparison is from four aspects: training graph availability, test-time label accessibility, training objectives, and test-time objectives.

3 Systematic Taxonomy

Fig. 2 presents a hierarchical taxonomy that systematically categorizes existing test-time adaptation methods on graphs. Specifically, existing studies are organized into three groups: (1) **Model-centric methods**, which adapt the well-trained model itself through *full model fine-tuning*, *adapter-based model learning*, or *joint model-adapter learning*; (2) **Data-centric methods**, which keep the model fixed and instead transform the input graph, further divided according to modification strength into *additive graph refinement* and *reconstructive graph adaptation*; and (3) **Hybrid methods**, which combine both strategies to jointly adapt the model and the test graph, thereby enhancing robustness under complex distribution shifts. Additionally, an application scenario branch summarizes task-specific adaptations covering GNN model evaluation, graph OOD & anomaly detection, and socially beneficial domains.

4 Model-Centric TTA Methods on Graphs

As illustrated in Fig. 3, existing model-centric approaches can be categorized according to the scope and granularity of parameter adaptation: (1) *Full model fine-tuning*, which updates the inherited parameters of well-trained models; (2) *Adapter-based model learning*, which integrates lightweight trainable modules with a frozen backbone; (3) *Joint model-adapter learning*, which jointly optimizes both inherited and newly introduced parameters.

4.1 Full Model Fine-Tuning

Latent Feature Space-Oriented Fine-Tuning. Most model-centric approaches adapt the intermediate layers of the GNN,

Components	Training Graph	Test Graph	Training Stage Loss	Test-Time Loss
Out-of-Distribution Generalization	$\{\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}\}$	$\{\mathcal{G}_{\text{te}}, \mathbf{Y}_{\text{te}}\}$	$\mathcal{L}_{\text{tr}}(\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}})$	-
Domain Adaptation	$\{\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}\}$	$\{\mathcal{G}_{\text{te}}, \mathbf{Y}_{\text{te}}\}$	$\mathcal{L}_{\text{tr}}(\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}) + \mathcal{L}_{\text{g}}(\mathcal{G}_{\text{te}}, \mathcal{G}_{\text{tr}}) + \mathcal{L}_{\text{te}}(\mathcal{G}_{\text{te}}, \mathbf{Y}_{\text{te}})$	-
Unsupervised Domain Adaptation	$\{\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}\}$	$\mathcal{G}_{\text{te}}$	$\mathcal{L}_{\text{tr}}(\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}) + \mathcal{L}_{\text{g}}(\mathcal{G}_{\text{te}}, \mathcal{G}_{\text{tr}})$	-
Test-Time Training†	$\{\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}\}$	$\mathcal{G}_{\text{te}}$	$\mathcal{L}_{\text{tr}}(\mathcal{G}_{\text{tr}}, \mathbf{Y}_{\text{tr}}) + \mathcal{L}_{\text{self}}(\mathcal{G}_{\text{tr}})$	$\mathcal{L}_{\text{tt}}(\mathcal{G}_{\text{te}})$
Test-Time Adaptation† **(Model-Centric)**	Inaccessible	$\mathcal{G}_{\text{te}}$	-	$\mathcal{L}_{\text{tt}}(\mathcal{G}_{\text{te}})$
Test-Time Adaptation (Data-Centric)	Inaccessible	$\mathcal{G}_{\text{te}}$	-	$\mathcal{L}_{\text{tt}}(\mathcal{G}_{\text{te}})$

Table 1: Comparison of different strategies for addressing graph distribution shifts. Note that † indicates that the well-trained model is allowed to be updated at test time, $\mathcal{L}_{\text{tr}}$ and $\mathcal{L}_{\text{te}}$ denote the supervised training and test loss functions, respectively (e.g., cross-entropy); $\mathcal{L}_{\text{g}}$ is for measuring the distribution distance between the training and the test graphs; $\mathcal{L}_{\text{self}}$ and $\mathcal{L}_{\text{tt}}$ denote the training graph self-supervised loss and the test graph test-time adaption loss.

which aggregate neighborhood information and form latent representations, while keeping the task head fixed. Methods in this category typically enforce consistency or alignment in the embedding space: GT3 [Wang *et al.*, 2022] performs contrastive learning on a single test graph to fine-tune the shared GNN encoder and introduces an explicit alignment constraint that reduces distribution discrepancy between source and target features; HomoTTT [Zhang *et al.*, 2024b] integrates a parameter-free, homophily-aware contrastive objective with adaptive augmentations. It explicitly employs a homophily-driven model selection mechanism to constrain the test-time fine-tuning process, thereby preventing feature distortion caused by label-agnostic adaptation. This self-supervised fine-tuning paradigm is further adapted by TARD [Tao *et al.*, 2024] and IGT [Zhou *et al.*, 2025] for dynamic rumor detection and out-of-distribution graph classification, respectively. Complementing distribution alignment, MATCHA [Bao *et al.*, 2025] addresses structural distortion by dynamically adjusting hop-wise aggregation weights, rebalancing the influence of ego-node features and multi-hop neighbors to restore representation quality under topology shifts.

Graph Learning Task-Oriented Fine-Tuning. In contrast to adapting latent representations in intermediate layers, a more conservative strategy focuses adaptation exclusively on the task-specific output layer. By restricting updates to the final layer, it preserves the general-purpose graph encoding learned during pre-training while adapting predictions to the test distribution. LLMTTT [Zhang *et al.*, 2024a] adopts this shallow adaptation paradigm to mitigate the risk of catastrophic forgetting in the representation space. Instead of standard self-training, it integrates a Large Language Model (LLM) as an annotator within a hybrid active learning framework. The adaptation pipeline proceeds in two phases: first, supervised fine-tuning using high-confidence pseudo-labels generated by the LLM for informative nodes; second, consistency-based self-training on the remaining unlabeled instances. This decoupling allows the model to align the classifier boundary with the test distribution while preserving the generalized structural encoding learned during pre-training.

Fully Fine-Tuning. Beyond partial or layer-specific adaptation, some approaches update all parameters of the well-trained model at test time for comprehensive distribution alignment. For example, SOGA [Mao *et al.*, 2024] introduces a source-free unsupervised graph domain adaptation framework. It fully fine-tunes all parameters of the well-trained GNN through a joint objective that combines information-theoretic maximization and structural-consistency regularization, enabling adaptation to the target domain while preserving both discriminative capacity and relational graph structure. PROGRAM [Sun *et al.*, 2024] adopts a prototype-based pseudo-label generation scheme coupled with a robust self-training mechanism. High-confidence hard labels are used to accelerate model convergence, while soft labels maintain consistency under label noise. In the recommendation domain, DT3OR [Yang *et al.*, 2025] extends test-time adaptation by employing two complementary self-supervised tasks: self-distillation and contrastive learning. These tasks capture the latent structure of test data, allowing the model to jointly learn invariant user preferences and dynamic behavioral shifts.

4.2 Adapter-Based Model Learning

Ahead Well-Trained Model. In contrast to direct parameter fine-tuning, adapter-based methods optimize auxiliary modules attached to a frozen backbone. When placed ahead, these modules act as an input-side enhancer that reshapes or enriches the test graph before it enters the frozen model. GADT3 [Pirhayatifard and Silva, 2025] adopts this paradigm for label-scarce cross-domain scenarios, employing a domain-specific encoder to map target features into a source-aligned latent space. By updating solely this input module via homophily-guided self-supervision, it bridges domain gaps without modifying the downstream GNN decoder.

Behind Well-Trained Model. Conversely, behind adapters are appended after the frozen GNN and operate on its fixed embeddings to produce task-specific predictions. These modules act as post-hoc interpreters that adjust decision boundaries or recalibrate latent representations without affecting the upstream feature extractor. A representative example is TSA [Hsu *et al.*, 2025], which introduces an adaptive weighting mechanism in the post-feature space. It utilizes test-time pseudo-labels to dynamically re-balance the aggregation of ego-node versus neighbor information, allowing for robust inference-time adjustment without backpropagating gradients into the upstream feature extractor.

4.3 Joint Model-Adapter Learning

Within the model-centric paradigm, a hybrid strategy jointly optimizes both the original model parameters and the added adapter modules. For example, GAPGC [Chen *et al.*, 2022] jointly optimizes a well-trained GNN's core parameters, adversarial augmenters, and contrastive projection heads during test time. Through group-pseudo-positive contrastive learning and adversarial min-max optimization, it strengthens the alignment between self-supervised and main task objectives, efficiently extracting minimal sufficient information from the graph. Recently, some works have guided adaptation explicitly through information-theoretic constraints. ASSESS [Zhao *et al.*, 2025] employs mutual information to select reliable subgraphs and integrates a Bayesian-regularized self-training loss to jointly refine class prototypes and model parameters, enhancing consistency between pseudo-labels and feature representations. GCAL [Qiao *et al.*, 2025] adopts a two-layer strategy that alternates between adaptation and memory generation. It optimizes pseudo-labels via information maximization and builds a variational memory-graph generator inspired by the information bottleneck principle, which preserves task-relevant signals while filtering domain-specific noise.

5 Data-Centric TTA Methods on Graphs

Data-centric graph TTA methods aim to improve model generalization and robustness by modifying the test graph while keeping the well-trained model frozen. As illustrated in Fig. 3, existing data-centric strategies can be categorized according to the degree of graph modification: (1) *additive graph refinement*, which enriches the original graph with auxiliary signals while preserving its core structure; (2) *reconstructive graph adaptation*, which reconstructs feature and/or structural components to produce a new adapted test graph.

5.1 Additive Graph Refinement

Additive refinement methods enhance the original test graph by injecting learnable or model-guided auxiliary components into its feature matrix or adjacency structure, while preserving the base graph intact. The resulting augmented graph is a superposition of the original input and task-specific perturbations, designed to improve robustness or restore inductive biases such as homophily or feature coherence.

A representative work is GTrans [Jin *et al.*, 2023], a pioneering data-centric test-time graph transformation framework that applies additive perturbations to both node features and graph structure. Guided by a parameter-free contrastive surrogate loss, it implicitly mitigates distribution shifts, enhancing robustness without accessing training data or modifying the model architecture. In contrast, a widely adopted class of methods introduces structured auxiliary signals into the original graph via lightweight learnable modules. AAGOD [Guo *et al.*, 2023] employs a lightweight multilayer perceptron (MLP) trained on in-distribution (ID) source data to generate a graph-specific amplifier matrix. This matrix is applied additively to the adjacency matrix of the test graph, reinforcing ID structural patterns and thereby enhancing discriminative performance for graph OOD detection; GOAT [Zhang *et al.*,] integrates low-rank attention with a multi-layer perceptron, leveraging the test graph's own augmented views and a self-supervised consistency loss to produce an OOD representation matrix. This matrix is additively combined with the original node feature matrix, thereby reducing the distribution gap without altering the graph topology; GraphPatcher [Ju *et al.*, 2023] generates a sequence of increasingly corrupted versions of each node's ego-graph during test time, using a graph convolutional network encoder (GCN)–MLP module to create virtual nodes and corresponding connections. Through an iterative patching strategy, it aligns the prediction distribution of the pre-trained GCN, effectively mitigating degree bias in GNNs.

5.2 Reconstructive Graph Adaptation

Unlike additive methods that preserve the original test graph, reconstructive adaptation fundamentally transforms the test input. These approaches view the original graph as potentially corrupted or misaligned, seeking to reconstruct its structural or feature components to better match the pre-trained model's inductive bias.

Feature-Level Reconstruction. A prominent strategy involves replacing node features with semantically grounded embeddings derived from source-domain knowledge. Yang et al. [Yang *et al.*, 2024] frame frame the GNN as a control system, utilizing a two-layer MLP to map initial predictions into class-conditioned embeddings. These refined embeddings substitute the original features during a second inference pass to correct distributional deviations. Similarly, FR-GNN [Ding *et al.*, 2024] employs a feature reconstruction module to generate class-prototype embeddings, which replace the original node features, effectively projecting the test graph into a semantic space aligned with the source domain and thereby enhancing GNN robustness under distribution shifts.

Joint Feature & Structure Reconstruction. Beyond feature-level reconstruction, a broader class of methods employs structured transformation mechanisms to partially or fully reconstruct the test graph. GOODAT [Wang *et al.*, 2024] learns a lightweight graph masker, which performs element-wise multiplication with the adjacency and feature matrices to extract an informative subgraph highly correlated with pseudo-ID labels while discarding noisy or anomalous components; TT-GREB [Zheng *et al.*, 2025b] introduces a test-time graph rebirth mechanism coordinated by dual MLP modules. Through a prototype extractor and an environment refiner, it generates re-weighting matrices to filter the test graph and then proportionally fuse the components into a new graph that retains class-core information while aligning with the source distribution; TTA-GREC [Liu *et al.*, 2025] employs pseudo-labels to filter key user-item interaction edges, uses rationality-aware scoring to mask and reconstruct semantic edges in the knowledge graph, achieving joint feature and structure reconstruction for more effective user and item representation learning.

Generative Reconstruction. Leveraging generative models, TT-GNDS [Zheng *et al.*, 2025a] employs a dual-conditioned diffusion model, using the pre-trained GNN's pseudo-labels and predictions to guide the joint denoising of node features and graph structure. It dynamically selects adapted graphs that align with the training distribution while preserving task-relevant patterns, enabling precise adaptation under distribution shifts through ensemble inference.

6 Hybrid TTA Methods on Graphs

Hybrid TTA methods integrate both the data-centric and model-centric TTA methods on graphs, achieving robust consistency between the graph and GNN model views by jointly optimizing the test-time graph and the well-trained model. For example, GraphCTA [Zhang *et al.*, 2024d] implements dual-stream adaptation through a collaborative framework: its graph-adaptation module optimizes node features and edge structure via additive and element-wise exclusive OR (XOR) perturbations, while its model-adaptation module fine-tunes the well-trained GNN with pseudo-label-guided consistency losses. IGFT [Wei *et al.*, 2025] shifts focus to the spectral domain. It adjusts the low-frequency components of the test graph via graph Fourier transform to correct input-level shifts, and simultaneously fine-tunes the well-trained model parameters through a feature-mapping-guided self-training strategy.

7 Application Scenarios

Test-Time GNN Model Evaluation. GNNevaluator [Zheng *et al.*, 2023b] pioneers test-time evaluation of GNNs on unlabeled graphs. It generates meta-graphs with simulated distribution shifts and converts them into DiscGraphs, where node features encode embedding discrepancies and graph-level labels denote model accuracy. A GNNevaluator is then trained to predict performance from these discrepancies. LeBed [Zheng *et al.*, 2024] estimates test accuracy by computing the L2 distance between the well-trained model's parameters and those of a pseudo-label-based retrained model, achieving strong correlation with true test error. DYGevaluator [Li *et al.*, 2025] extends this idea to dynamic graphs by inducing time-gap shifts through edge dropout and training a Graphormer-based evaluator to regress performance metrics from embedding discrepancy matrices between training and test graphs.

Test-Time Graph OOD & Anomaly Detection. Conventional graph OOD and anomaly detection methods usually train a GNN from scratch, which is expensive and cannot reuse well-trained models. AAGOD [Guo *et al.*, 2023] establishes a non-retraining baseline by learning an edge-amplification module on source-domain ID data. During inference, it superimposes generated amplifier matrices onto the test graph topology, explicitly widening the score margin between ID and OOD samples. GOODAT [Wang *et al.*, 2024] further enforces a strictly source-free constraint, employing a graph information bottleneck (GIB)-optimized masker to extract informative subgraphs, where the bottleneck loss serves directly as an anomaly score to quantify distributional deviations. GOAT [Zhang *et al.*,] instead injects a consistency-optimized additive residual via a low-rank adapter to rectify feature distributions on shifted test graphs without structural modification. For cross-domain scenarios, GADT3 [Pirhayatifard and Silva, 2025] utilizes structure-preserving attention to optimize a dedicated target encoder, tailoring representations specifically for domain-level anomaly detection.

Test-Time Graph Learning for Social Benefits.

(a) Recommendation System. TTA-GREC [Liu *et al.*, 2025] is the first data-centric test-time adaptation on graphs method for recommendation, reconstructing the user–item and knowledge graphs to adapt distribution shifts. It generates pseudo-interactions from the well-trained knowledge graph neural network (KGNN), samples high-attention edges, and revises graph structure through rationality-guided filtering. DT3OR [Yang *et al.*, 2025], as the first model-centric approach, enhances out-of-distribution recommendation by refining user and item embeddings through dual self-supervised adaptation, using soft K-means to form pseudo-label clusters followed by self-distillation and contrastive learning.

(b) Cybersecurity. Tran et al. [Tran *et al.*, 2025] address distribution shift in graph-based Android malware classification by introducing a semantic graph augmentation approach that integrates function metadata and LLM-Generated code embeddings. They also design strategies to handle feature-missing scenarios, thereby enhancing the generalization robustness of GNN models to unseen malware families and variants.

(c) Healthcare. GTTA [Zeng *et al.*, 2024] pioneers test-time graph learning for glaucoma diagnosis, freezing the backbone and refining classifier boundaries using pseudo-labels from target-domain feature memory to handle cross-institutional fundus imaging shifts; TTDG-MGM [Lv *et al.*, 2025] constructs foreground-region graphs and learns universal node embeddings, enabling anatomy-preserving segmentation via multi-graph matching; SPA [Yuan *et al.*, 2025] freezes vision-language models and optimizes text-fused prototypes, enhancing cross-institutional surgical phase recognition through task-graph diffusion for temporal coherence.

8 Benchmarking Landscape

In this section, we collect the mainstream benchmarking landscape of existing TTA on graphs by categorizing datasets according to the type of graph distribution shift. Major shift types include feature/structural shift, domain shift and temporal shift. A summary of corresponding dataset characteristics, task formulations, evaluation metrics, and data-split protocols is provided. As shown in Table 2, the covered application domains include citation networks, social or transaction graphs, molecular graphs, and synthetic benchmark datasets. Evaluation tasks encompass both node-level and graph-level classification, highlighting the broad applicability of test-time adaptation across diverse graph learning scenarios. All reviewed methods are evaluated during test time based on well-trained GNN models and unlabeled test graphs for adaptation.

9 Future Directions

Adaptation under Complex Graph Distributions. Most existing test-time adaptation methods for graph learning face two benchmarking challenges. *First*, current research typically collects or creates distribution-shifted graph data under relatively simple graph distribution shifts, where only a single type of shift occurs, either in graph temporal dimension or graph structure. In practical scenarios, however, test graphs often exhibit complex and intertwined distribution changes, involving mixed shifts across multiple dimensions of graph data. It is necessary to discover more complex, diverse, and mixed graph distribution shift data for existing and future test-time adaptation for graph learning evaluation. *Second*, existing evaluations almost lack quantify the severity of graph distribution shifts or assess the tolerance limits of current test-time adaptation methods for graph learning. It remains unclear how much distributional change existing methods can reliably handle. Hence, developing fine-grained metrics for complex shifts and systematically investigating the performance limits of TTA methods are critical future directions.

Adaptation under Diverse Graph Types and Learning Tasks. Most existing test-time adaptation methods for graph

Distribution Shift	Datasets	#Nodes / #Graphs	#Edges / Avg. Size	#Classes	Task Level	Metric(s)	Splits
Feature/Structural Shift	Cora [Jin *et al.*, 2023]	2,703	5,278	10	Node	Accuracy	1/1/8
	Amazon-Photo [Jin *et al.*, 2023]	7,650	119,081	10	Node	Accuracy	1/1/8
	IMDB-BINARY [Yanardag and Vishwanathan, 2015]	1,000	19	2	Graph	Accuracy	1/1/8
	PROTEINS [Borgwardt *et al.*, 2005]	1,113	39	2	Graph	Accuracy	8/1/1
	ENZYMES [Morris *et al.*, 2020]	587	33	6	Graph	Accuracy	4/1
Domain Shift	Twitch-E [Jin *et al.*, 2023]	1,912-9,498	31,299–153,138	2	Node/Graph	ROC-AUC	1/1/5
	FB-100 [Jin *et al.*, 2023]	769-41,536	16,656–1,590,655	2	Node	Accuracy	3/2/3
	ACMv9 [Wu *et al.*, 2020]	7,410	22,270	6	Node/Graph	Accuracy	1/1/369 (3-domains: A, D, C)
	DBLPv8 [Wu *et al.*, 2020]	5,578	14,682	6	Node/Graph	Accuracy	1/1/369 (3-domains: A, D, C)
	Citationv2 [Wu *et al.*, 2020]	4,715	13,466	6	Node/Graph	Accuracy	1/1/369 (3-domains: A, D, C)
	ogbg-molhiv [Hu *et al.*, 2020]	41,127	25.3	2	Graph	ROC-AUC	8/1/1
Temporal Shift	OGB-arXiv [Jin *et al.*, 2023]	169,343	1,166,243	40	Node/Graph	Accuracy	1/1/3
	Elliptic [Jin *et al.*, 2023]	203,769	234,355	2	Node	F1 Score	5/5/33

Table 2: Statistics and characteristics of graph datasets for evaluating test-time adaptation on graphs.

learning are evaluated under both limited graph types and graph learning task. *First*, regarding graph types, current methods mainly learn on single-relational and homophilic graphs, as well as temporal graphs. More diverse graph types, such as multi-relational graphs, heterophilic graphs, and complex spatio-temporal graphs, remain largely unexplored. *Second*, most research focuses solely on node classification, with few addressing graph classification or regression. This restricts applicability in real-world scenarios where data types and objectives are more diverse.

Hence, extending test-time adaptation to broader tasks—including knowledge graph completion, spatio-temporal forecasting, and structure learning—is an important research direction.

Adaptation with Theory-Grounded and Robust Supervision. Most existing test-time adaptation methods for graph learning rely on self-supervised signals to guide adaptation, without access to labeled test data or source training graphs, leading to two-fold challenges. *First*, existing methods typically include contrastive learning, pseudo-label augmentation and tuning, as well as information bottleneck regularization. However, supervision signals derived from these frameworks are often noisy and unstable, which may lead to error accumulation and performance degradation during adaptation. *Second*, most current adaptation process is largely heuristic and lacks theoretical guarantees on convergence and robustness. A potential future solution is to incorporate ground-truth factual signals from external knowledge, inspired by graph-based retrieval-augmented generation (RAG), to design new test-time adaptation objectives for graph learning. Hence, developing robust and theoretically grounded supervision signals therefore remains an open challenge for enabling powerful test-time adaptation on graphs.

Adaptation with Explainability and Scalability. Most data-centric graph TTA methods modify test graph features and structures to project shifted distributions back toward training ones, without fine-tuning the GNN. However, these modification-based methods generally lack explainability; it remains unclear why certain transformations improve adaptation, limiting our understanding of TTA under complex shifts. Incorporating causal reasoning or self-explainable models via graph decomposition is a promising direction for providing such explanations. Moreover, current methods often manipulate nodes and edges within a single test graph, which is

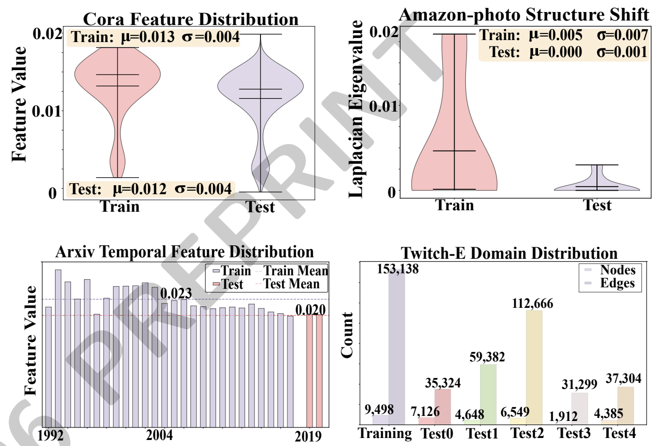

Figure 4: Visualization of graph distribution shifts categorized into four representative groups.

computationally expensive for large-scale graphs or datasets, hindering scalability. Although recent work explores learning new test distributions to improve efficiency [Zheng *et al.*, 2025a], it introduces challenges in accurately modeling complex distributions. Hence, designing scalable TTA methods that balance efficiency, explainability, and effectiveness remains an open research direction.

10 Conclusion

In this survey, we present a comprehensive review of test-time adaptation for graph learning, aiming to draw attention to the challenges of generalization under distribution shifts during real-world model deployment, i.e., at test time. We propose a systematic taxonomy to organize existing research, categorizing them into three groups: model-centric, data-centric, and hybrid methods. By collecting representative application scenarios and evaluation benchmarks, we highlight the practical potential of these methods in narrowing the gap between laboratory GNN research and robust real-world deployment. We hope this survey serves as a foundational roadmap to inspire future research toward building widely adaptable, theoretically grounded, robust, explainable, and scalable graph learning systems for test-time deployment.

Ethical Statement

There are no ethical issues.

References

[Bao *et al.*, 2025] Wenxuan Bao, Zhichen Zeng, Zhining Liu, Hanghang Tong, and Jingrui He. Matcha: Mitigating graph structure shifts with test-time adaptation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[Borgwardt *et al.*, 2005] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 2005.

[Chen *et al.*, 2022] Guanzi Chen, Jiying Zhang, Xi Xiao, and Yang Li. Graphtta: Test time adaptation on graph neural networks. *ICML 2022 Workshop on "Principles of Distribution Shift"*, 2022.

[Ding *et al.*, 2024] Rui Ding, Jielong Yang, Feng Ji, Xionghu Zhong, and Linbo Xie. Fr-gnn: Mitigating the impact of distribution shift on graph neural networks via test-time feature reconstruction. *IEEE Internet of Things Journal*, 11(13):23521–23531, 2024.

[Guo *et al.*, 2023] Yuxin Guo, Cheng Yang, Yuluo Chen, Jixi Liu, Chuan Shi, and Junping Du. A data-centric framework to endow graph neural networks with out-of-distribution detection ability. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 638–648, 2023.

[Hsu *et al.*, 2025] Hans Hao-Hsun Hsu, Shikun Liu, Han Zhao, and Pan Li. Structural alignment improves graph test-time adaptation. *arXiv preprint arXiv:2502.18334*, 2025.

[Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[Jin *et al.*, 2023] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. In *ICLR*, 2023.

[Ju *et al.*, 2023] Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. Graphpatcher: mitigating degree bias for graph neural networks via test-time augmentation. *Advances in Neural Information Processing Systems*, 36:55785–55801, 2023.

[Li *et al.*, 2025] Bo Li, Xin Zheng, Ming Jin, Can Wang, and Shirui Pan. Test-time gnn model evaluation on dynamic graphs. *The IEEE International Conference on Data Mining*, 2025.

[Liang *et al.*, 2025] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

[Liu *et al.*, 2025] Yating Liu, Xin Zheng, Yi Li, and Yanqing Guo. Test-time adaptation on recommender system with data-centric graph transformation. In *International Joint Conference on Artificial Intelligence*, 2025.

[Lv *et al.*, 2025] Xingguo Lv, Xingbo Dong, Liwen Wang, Jiewen Yang, Lei Zhao, Bin Pu, Zhe Jin, and Xuejun Li. Test-time domain generalization via universe learning: A multi-graph matching approach for medical image segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15621–15631, 2025.

[Mao *et al.*, 2024] Haitao Mao, Lun Du, Yujia Zheng, Qiang Fu, Zelin Li, Xu Chen, Shi Han, and Dongmei Zhang. Source-free graph unsupervised domain adaptation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 520–528, 2024.

[Morris *et al.*, 2020] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020.

[Pan *et al.*, 2025] Junjun Pan, Yixin Liu, Xin Zheng, Yizhen Zheng, Alan Wee-Chung Liew, Fuyi Li, and Shirui Pan. A label-free heterophily-guided approach for unsupervised graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12443–12451, 2025.

[Pirhayatifard and Silva, 2025] Delaram Pirhayatifard and Arlei Silva. Cross-domain graph anomaly detection via test-time training with homophily-guided self-supervision. *Transactions on Machine Learning Research*, 2025.

[Qiao *et al.*, 2025] Ziyue Qiao, Qianyi Cai, Hao Dong, Jiawei Gu, Pengyang Wang, Meng Xiao, Xiao Luo, and Hui Xiong. GCAL: Adapting graph models to evolving domain shifts. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pages 50113–50129, 2025.

[Sui *et al.*, 2023] 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*, 2023.

[Sun *et al.*, 2024] Haopeng Sun, Lumin Xu, Sheng Jin, Ping Luo, Chen Qian, and Wentao Liu. Program: Prototype graph model based pseudo-label learning for test-time adaptation. In *The twelfth international conference on learning representations*, 2024.

[Tao *et al.*, 2024] Xiang Tao, Mingqing Zhang, Qiang Liu, Shu Wu, and Liang Wang. Out-of-distribution rumor detection via test-time adaptation. *arXiv preprint arXiv:2403.17735*, 2024.

[Tran *et al.*, 2025] Ngoc N Tran, Anwar Said, Waseem Abbas, Tyler Derr, and Xenofon D Koutsoukos. Mitigating distribution shift in graph-based android malware classification via function metadata and llm embeddings. *arXiv preprint arXiv:2508.06734*, 2025.

[Wang *et al.*, 2022] Yiqi Wang, Chaozhuo Li, Wei Jin, Rui Li, Jianan Zhao, Jiliang Tang, and Xing Xie. Test-time training for graph neural networks. *arXiv preprint arXiv:2210.08813*, 2022.

[Wang *et al.*, 2024] Luzhi Wang, Dongxiao He, He Zhang, Yixin Liu, Wenjie Wang, Shirui Pan, Di Jin, and Tat-Seng Chua. Goodat: Towards test-time graph out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15537–15545, 2024.

[Wang *et al.*, 2025] Zeyu Wang, Tianyi Jiang, Huanchang Ma, Yao Lu, Xiaoze Bao, Shanqing Yu, Qi Xuan, Shirui Pan, and Xin Zheng. Few-shot molecular property prediction: A survey. *arXiv preprint arXiv:2510.08900*, 2025.

[Wei *et al.*, 2025] Xin Wei, Qin Yang, Yijie Fang, Mingrui Zhu, and Nannan Wang. 3d test-time adaptation via graph spectral driven point shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26762–26771, 2025.

[Wu *et al.*, 2020] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of the web conference 2020*, pages 1457–1467, 2020.

[Yanardag and Vishwanathan, 2015] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.

[Yang *et al.*, 2024] Jielong Yang, Rui Ding, Feng Ji, Hongbin Wang, and Linbo Xie. Control the gnn: Utilizing neural controller with lyapunov stability for test-time feature reconstruction. *arXiv preprint arXiv:2410.09708*, 2024.

[Yang *et al.*, 2025] Xihong Yang, Yiqi Wang, Jin Chen, Wenqi Fan, Xiangyu Zhao, En Zhu, Xinwang Liu, and Defu Lian. Dual test-time training for out-of-distribution recommender system. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[Yuan *et al.*, 2025] Kun Yuan, Tingxuan Chen, Shi Li, Joël L Lavanchy, Christian Heiliger, Ege Özsoy, Yiming Huang, Long Bai, Nassir Navab, Vinkle Srivastav, et al. Recognizing surgical phases anywhere: Few-shot test-time adaptation and task-graph guided refinement. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 467–477. Springer, 2025.

[Zeng *et al.*, 2024] Qian Zeng, Le Zhang, Yipeng Liu, Ce Zhu, and Fan Zhang. Graph-guided test-time adaptation for glaucoma diagnosis using fundus photography. *arXiv preprint arXiv:2407.04396*, 2024.

[Zhang *et al.*,] Haoxiang Zhang, Zhuofeng Li, Qiannan Zhang, Ziyi Kou, Lianyong Qi, Juncheng Li, and Shichao Pei. Map to optimal: Adapting graph out-of-distribution in test time.

[Zhang *et al.*, 2024a] Jiaxin Zhang, Yiqi Wang, Xihong Yang, Siwei Wang, Yu Feng, Yu Shi, Ruichao Ren, En Zhu, and Xinwang Liu. Test-time training on graphs with large language models (llms). In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2089–2098, 2024.

[Zhang *et al.*, 2024b] Jiaxin Zhang, Yiqi Wang, Xihong Yang, and En Zhu. A fully test-time training framework for semi-supervised node classification on out-of-distribution graphs. *ACM Transactions on Knowledge Discovery from Data*, 18(7):1–19, 2024.

[Zhang *et al.*, 2024c] 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*, 2024.

[Zhang *et al.*, 2024d] Zhen Zhang, Meihan Liu, Anhui Wang, Hongyang Chen, Zhao Li, Jiajun Bu, and Bingsheng He. Collaborate to adapt: Source-free graph domain adaptation via bi-directional adaptation. In *Proceedings of the ACM Web Conference 2024*, pages 664–675, 2024.

[Zhao *et al.*, 2025] Yusheng Zhao, Qixin Zhang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. Test-time adaptation on graphs via adaptive subgraph-based selection and regularized prototypes. In *Forty-second International Conference on Machine Learning*, 2025.

[Zheng *et al.*, 2023a] Xin Zheng, Yixin Liu, Zhifeng Bao, Meng Fang, Xia Hu, Alan Wee-Chung Liew, and Shirui Pan. Towards data-centric graph machine learning: Review and outlook. *arXiv preprint arXiv:2309.10979*, 2023.

[Zheng *et al.*, 2023b] Xin Zheng, Miao Zhang, Chunyang Chen, Soheila Molaei, Chuan Zhou, and Shirui Pan. Gnnevaluator: Evaluating gnn performance on unseen graphs without labels. *Advances in Neural Information Processing Systems*, 36:33606–33623, 2023.

[Zheng *et al.*, 2023c] Xin Zheng, Miao Zhang, Chunyang Chen, Quoc Viet Hung Nguyen, Xingquan Zhu, and Shirui Pan. Structure-free graph condensation: From large-scale graphs to condensed graph-free data. *Advances in Neural Information Processing Systems*, 36:6026–6047, 2023.

[Zheng *et al.*, 2024] Xin Zheng, Dongjin Song, Qingsong Wen, Bo Du, and Shirui Pan. Online gnn evaluation under test-time graph distribution shifts. In *The Twelfth International Conference on Learning Representations*, 2024.

[Zheng *et al.*, 2025a] Xin Zheng, Wei Huang, Chuan Zhou, Ming Li, and Shirui Pan. Test-time graph neural dataset search with generative projection. In *Forty-second International Conference on Machine Learning*, 2025.

[Zheng *et al.*, 2025b] Xin Zheng, Yu Zheng, Qin Zhang, Haishuai Wang, Yuxuan Liang, Alan Wee-Chung Liew, and Shirui Pan. Test-time graph rebirth: Serving gnn generalization under distribution shifts. *The IEEE International Conference on Data Mining*, 2025.

[Zhou *et al.*, 2025] Zhikai Zhou, Li Pi, Honghua Xu, Suyang Zhou, and Youyong Kong. Test-time training with invariant graph learning for out-of-distribution generalization. *Engineering Research Express*, 7(4):0452e5, 2025.