EvoluNet: Advancing Dynamic Non-IID Transfer Learning on Graphs

Haohui Wang [1] **Yuzhen Mao** [1] **Yujun Yan** [2] **Yaoqing Yang** [2] **Jianhui Sun** [1] **Kevin Choi** [3] **Balaji Veeramani** [3]
Alison Hu [3] **Edward Bowen** [3] **Tyler Cody** [4] **Dawei Zhou** [1]

Abstract

Non-IID transfer learning on graphs is crucial in many high-stakes domains. The majority of existing works assume stationary distribution for both source and target domains. However, real-world graphs are intrinsically dynamic, presenting challenges in terms of domain evolution and dynamic discrepancy between source and target domains. To bridge the gap, we shift the problem to the dynamic setting and pose the question: given the *label-rich* source graphs and the *label-scarce* target graphs both observed in previous T timestamps, how can we effectively characterize the evolving domain discrepancy and optimize the generalization performance of the target domain at the incoming $T+1$ timestamp? To answer it, we propose a generalization bound for *dynamic non-IID transfer learning on graphs*, which implies the generalization performance is dominated by domain evolution and domain discrepancy between source and target graphs. Inspired by the theoretical results, we introduce a novel generic framework named EvoluNet. It leverages a transformer-based temporal encoding module to model temporal information of the evolving domains and then uses a dynamic domain unification module to efficiently learn domain-invariant representations across the source and target domains. Finally, EvoluNet outperforms the state-of-the-art models by up to 12.1%, demonstrating its effectiveness in transferring knowledge from dynamic source graphs to dynamic target graphs.

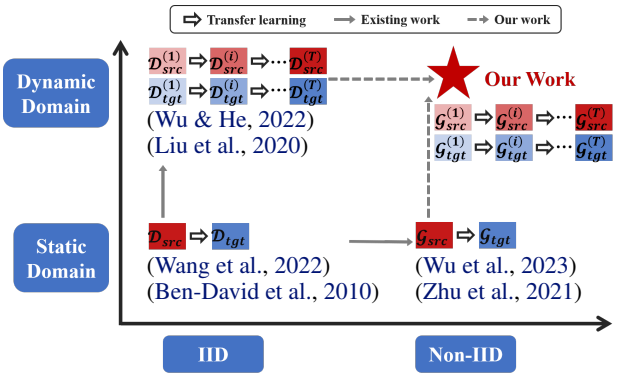

Figure 1. A paradigm shift to the dynamic non-IID transfer learning. $\mathcal{D}$ denotes IID domain; $\mathcal{G}$ denotes non-IID graph domain. Subscript src and tgt denote source and target, and superscript (i) represents the i^{th} timestamp.

1. Introduction

The recent decade has witnessed notable achievements in machine learning. Despite the exciting achievements, whether the learned model could deliver its promise in real-world scenarios heavily depends on abundant and high-quality training data. Nevertheless, the data annotation process, which requires domain-specific knowledge from human annotators, is often a costly and time-intensive endeavor (Fang et al., 2022; Cui et al., 2022; Zhou et al., 2022). Transfer learning has emerged as a promising tool, which aims to improve the generalization performance of the target domain with little or no labeled data by leveraging knowledge from the source domain with adequate labeled data (Tripuraneni et al., 2020; Wang et al., 2019; Ganin et al., 2016; Ben-David et al., 2006). However, a majority of the existing work (Ben-David et al., 2010; Zhao et al., 2019a; Wang et al., 2022) hold the assumption that data is static and independent and identically distributed (IID), as shown in the bottom-left of Figure 1. When extending transfer learning to graph-structured data, particular challenges are posed due to the non-IID nature of graphs, i.e., samples on graphs (e.g., nodes, edges, subgraphs) are naturally connected with their neighbors in certain ways (Xie et al., 2021a).

While recent research efforts have delved into non-IID transfer learning on graphs (Wu et al., 2023; Zhu et al., 2021; Hu

[1]Department of Computer Science, Virginia Tech, Blacksburg, VA, USA. [2]Department of Computer Science, Dartmouth College, Hanover, NH, USA. [3]Deloitte & Touche LLP, USA. [4]Virginia Tech National Security Institute, Arlington, VA, USA. Correspondence to: Dawei Zhou <zhoud@vt.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

et al., 2020; Wu et al., 2020; Shen et al., 2020) as shown in the bottom-right of Figure 1, the most of them have overlooked the dynamics inherent in realistic systems, where graphs in both source and target domains evolve over time. Directly applying existing static works on graphs to the dynamic setting may lead to a sub-optimal performance due to the unexplored temporal information and evolving distribution discrepancy (Greene et al., 2010; Fallani et al., 2014; Pareja et al., 2020; Song et al., 2019; Fu et al., 2020; Zhou et al., 2020a). Therefore, our paper proposes a paradigm shift in Figure 1 towards the dynamic non-IID setting, by introducing the novel problem termed as *dynamic non-IID transfer learning on graphs*. In particular, given the *label-rich* source graphs and the *label-scarce* target graphs observed in previous T timestamps, how can we effectively characterize the evolving domain discrepancy and optimize the generalization performance of the target graph at the incoming $T+1$ timestamp?

Despite the key importance, there exist three pivotal challenges in our problem setting. *C1. Generalization Bound*: There is limited theoretical analysis on how the domain discrepancy would accumulate across time and how it will affect the model performance. Carrying out theoretical analysis on the generalization bound would be crucial for understanding dynamic non-IID transfer learning on graphs. *C2. Computational Framework*: How can we develop a computational framework to characterize the evolving domain discrepancy and capture the domain-invariant information when the source and target graphs exhibit distinct distributions over time? *C3. Benchmark*: As there is little existing literature on dynamic non-IID transfer learning on graphs, it is essential to point out a set of benchmark datasets and baselines for algorithm development and evaluation.

In this paper, we make the first attempt to derive a generalization bound for dynamic transfer learning on graphs. The theoretical findings illustrate that the generalization performance is dominated by historical empirical error and domain discrepancy. It also serves as theoretical support to our proposed EvoluNet, which is a generic learning framework to enhance knowledge transfer across dynamic graphs. Moreover, we utilize a multi-resolution temporal encoding module to model domain evolution and a module to minimize domain discrepancy via dual divergence loss. In particular, the first module captures the interdependence over time and obtains the temporal graph representation in the evolving graphs, while the second module learns invariant representations to unify the source and target domains' spatial and temporal information. Our empirical results show that EvoluNet outperforms the state-of-the-art models by up to 12.1%, underscoring its effectiveness in knowledge transfer across dynamic graphs. Furthermore, we extensively surveyed existing temporal graphs and constructed benchmark datasets[1] for dynamic non-IID transfer learning, which have rich, dynamic properties regarding nodes, edges, node attributes, and labels. We conduct various evaluations on the constructed benchmark dataset, which demonstrate its validity and reliance.

2. Preliminary

In this section, we introduce the background that is pertinent to our work and give the formal problem definition. Table 1 summarizes the main notations used in this paper. We use regular letters to denote scalars (e.g., μ), boldface lowercase letters to denote vectors (e.g., $\mathbf{v}$), and boldface uppercase letters to denote matrices (e.g., $\mathbf{X}$). Next, we briefly review non-IID transfer learning on graphs and dynamic transfer learning for IID distributions.

Table 1. Symbols and notations.

Symbol	Description
$\mathcal{G}_{src}^{(i)}, \mathcal{G}_{tgt}^{(i)}$	input source and target graphs at timestamp i.
$\mathcal{V}_{src}^{(i)}, \mathcal{V}_{tgt}^{(i)}$	the set of nodes in $\mathcal{G}_{src}^{(i)}$ and $\mathcal{G}_{tgt}^{(i)}$.
$\mathcal{E}_{src}^{(i)}, \mathcal{E}_{tgt}^{(i)}$	the set of edges in $\mathcal{G}_{src}^{(i)}$ and $\mathcal{G}_{tgt}^{(i)}$.
$\mathbf{X}_{src}^{(i)}, \mathbf{X}_{tgt}^{(i)}$	the node feature matrices of $\mathcal{G}_{src}^{(i)}, \mathcal{G}_{tgt}^{(i)}$.
$\mathcal{Y}_{src}^{(i)}, \tilde{\mathcal{Y}}_{tgt}^{(i)}$	the set of labels in $\mathcal{G}_{src}^{(i)}$ and $\mathcal{G}_{tgt}^{(i)}$.
$N_{src}^{(i)}, N_{tgt}^{(i)}$	the size of sample graph $\mathcal{G}_{src}^{(i)}, \mathcal{G}_{tgt}^{(i)}$.
d_{src}, d_{tgt}	feature dimensions of $\mathbf{X}_{src}^{(i)}, \mathbf{X}_{tgt}^{(i)}, \forall i$.
T	number of timestamps.
$h(\cdot)$	node classifier for downstream task.
$\tilde{\mathfrak{R}}$	Rademacher complexity.
W_p	p-Wasserstein distance.

Non-IID Transfer Learning on Graphs. It focuses on leveraging knowledge gained from a source graph $\mathcal{G}_{src}$ to improve the performance of a target graph $\mathcal{G}_{tgt}$. Graphs are non-IID because their interconnected nodes, edges, and subgraphs exhibit inherent dependencies, and require modeling the highly complex interconnection. To applied existing theoretical guarantees of transfer learning under IID assumption to non-IID graph data, Wu et al. (2023) propose a novel graph discrepancy $d_{\text{GSD}}(\mathcal{G}_{src}, \mathcal{G}_{tgt})$ between two graphs $\mathcal{G}_{src}$ and $\mathcal{G}_{tgt}$ as follows (informal):

$$d_{\text{GSD}}(\mathcal{G}_{src}, \mathcal{G}_{tgt}) = \lim_{M\to\infty} \frac{1}{M+1} \sum_{m=0}^{M} d_b(\mathcal{G}_{src}^m, \mathcal{G}_{tgt}^m),$$

where $\mathcal{G}^m$ is the Weisfeiler-Lehman subgraph (Shervashidze et al., 2011) at depth m for an input graph $\mathcal{G}$, $d_b(\cdot,\cdot)$ is the base domain discrepancy. We refer to Definition 4 in Appendix A for formal definition. It reveals that given

[1]We publish our data and code at https://github.com/wanghh7/EvoluNet.

Weisfeiler-Lehman subtree, the subtree representations can be considered as IID samples, thus existing distribution discrepancy measures (e.g., Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) and Wasserstein Distance (Villani, 2009)) can be used to measure the distribution shift of source and target graphs.

Dynamic Transfer Learning. Let $\{\mathcal{D}_{src}^{(i)}\}_{i=1}^{T}$ and $\{\mathcal{D}_{tgt}^{(i)}\}_{i=1}^{T}$ be the labeled dynamic source domains and unlabeled (or few labeled) dynamic target domains, where superscript (i) represents the i^{th} timestamp, and there are T total timestamps. Dynamic transfer learning aims to improve the prediction performance of $\mathcal{D}_{tgt}^{(T+1)}$ using the knowledge in historical source and target domains under the domain shift $\{\mathcal{D}_{src}^{(i)}\}_{i=1}^{T} \neq \{\mathcal{D}_{tgt}^{(i)}\}_{i=1}^{T}$. Let $\mathcal{H}$ be the hypothesis class on input feature space $\mathcal{X}$ where a hypothesis is a function $h : \mathcal{X} \to \mathcal{Y}$, and $\mathcal{Y}$ is output label space. The expected error of the hypothesis h on the source domain $\mathcal{D}_{src}^{(i)}$ at timestamp i is given by $\epsilon_{src}^{(i)}(h) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{src}^{(i)}}[\mathcal{L}(h(\mathbf{x}), y)], \forall h \in \mathcal{H}$, where $\mathcal{L}(\cdot,\cdot)$ is some loss function. Its empirical estimate is defined as $\hat{\epsilon}_{src}^{(i)}(h) = \frac{1}{N_{src}^{(i)}} \sum_{j=1}^{N_{src}^{(i)}} [\mathcal{L}(h(\mathbf{x}_j), y_j)]$, where $\mathbf{x}_j$ is the feature of j^{th} sample in $\mathbf{X}_{src}^{(i)}$. We use the parallel notations $\epsilon_{tgt}^{(i)}(h)$ and $\hat{\epsilon}_{tgt}^{(i)}(h)$ for the target domain. In Wu & He (2022), the expected error on the newest target domain is derived as follows:

$$\epsilon_{tgt}^{(T+1)}(h) \leq \frac{1}{2T}\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h) + \hat{\epsilon}_{tgt}^{(i)}(h)\right) + \frac{T+2}{2}(\tilde{d} + \tilde{\lambda}) + \tilde{\Re}\left(\mathcal{H}_{\mathcal{L}}\right) + \frac{\rho}{T}\sqrt{\frac{\log\frac{1}{\delta}}{2\tilde{m}}}, \tag{1}$$

where $\tilde{d} = \rho \cdot \max\left\{\max_{1\leq i\leq T-1} d_{\text{MMD}}\left(\mathcal{D}_{src}^{(i)}, \mathcal{D}_{src}^{(i+1)}\right), d_{\text{MMD}}\left(\mathcal{D}_{src}^{(1)}, \mathcal{D}_{tgt}^{(1)}\right), \max_{1\leq i\leq T} d_{\text{MMD}}\left(\mathcal{D}_{tgt}^{(i)}, \mathcal{D}_{tgt}^{(i+1)}\right)\right\}$, d_{MMD} is the maximum mean discrepancy (Gretton et al., 2012), ρ is the Lipschitz constant, $\tilde{\lambda} = \rho \cdot \max\left\{\max_{1\leq i\leq T-1} \lambda_*\left(\mathcal{D}_{src}^{(i)}, \mathcal{D}_{src}^{(i+1)}\right), \lambda_*\left(\mathcal{D}_{src}^{(1)}, \mathcal{D}_{tgt}^{(1)}\right), \max_{1\leq i\leq T} \lambda_*\left(\mathcal{D}_{tgt}^{(i)}, \mathcal{D}_{tgt}^{(i+1)}\right)\right\}$, λ_* measures the labeling difference. $\mathcal{H}_{\mathcal{L}} = \{(\mathbf{X}, y) \mapsto \mathcal{L}(h(\mathbf{X}), y) : h \in \mathcal{H}\}$, $\tilde{\Re}\left(\mathcal{H}_{\mathcal{L}}\right)$ is a term that involves the Rademacher complexity defined on multiple domains, and $\tilde{m} = \sum_{i=1}^{T}\left(N_{src}^{(i)} + N_{tgt}^{(i)}\right)$ is the total number of training examples from historical source and target domains. However, this bound sums the errors in all timestamps without capturing domain evolution.

Problem definition. In the setting of dynamic non-IID transfer learning on graphs, the observed graph in source at timestamp i is defined as a source sample graph $\mathcal{G}_{src}^{(i)} = (\mathcal{V}_{src}^{(i)}, \mathcal{E}_{src}^{(i)})$ (parallel definition of target sample graph $\mathcal{G}_{tgt}^{(i)} = (\mathcal{V}_{tgt}^{(i)}, \mathcal{E}_{tgt}^{(i)})$), where $\mathcal{V}_{src}^{(i)}$ and $\mathcal{V}_{tgt}^{(i)}$ represent the set of nodes, and $\mathcal{E}_{src}^{(i)}$ and $\mathcal{E}_{tgt}^{(i)}$ represent the set of edges, respectively. $\mathbf{X}_{src}^{(i)}$ and $\mathbf{X}_{tgt}^{(i)}$ represent the node features in $\mathcal{G}_{src}^{(i)}$ and $\mathcal{G}_{tgt}^{(i)}$, T is the total number of timestamps that can be observed in history. Furthermore, we define labels of source at i^{th} timestamp as $\mathcal{Y}_{src}^{(i)}$ and labels of target as $\tilde{\mathcal{Y}}_{tgt}^{(i)}$, where only few target samples have labels, so $\tilde{\mathcal{Y}}_{tgt}^{(i)}$ is a sparse vector. We consider transferring knowledge from a series of time-evolving source sample graphs $\{\mathcal{G}_{src}^{(i)}\}_{i=1}^{T}$ to a series of time-evolving target sample graphs $\{\mathcal{G}_{tgt}^{(i)}\}_{i=1}^{T+1}$. Figure 2 illustrates the knowledge transfer from historical time snapshots of the book review graph and movie review graph to a more recent time snapshot of the movie review graph. Here, each node in a graph indicates an entity (user, movie, book), and the co-reviewer determines the edge between two nodes. Nodes, edges, and their attributes are evolving over time. The node label is the popularity of the movie (book) at that time and is also changing.

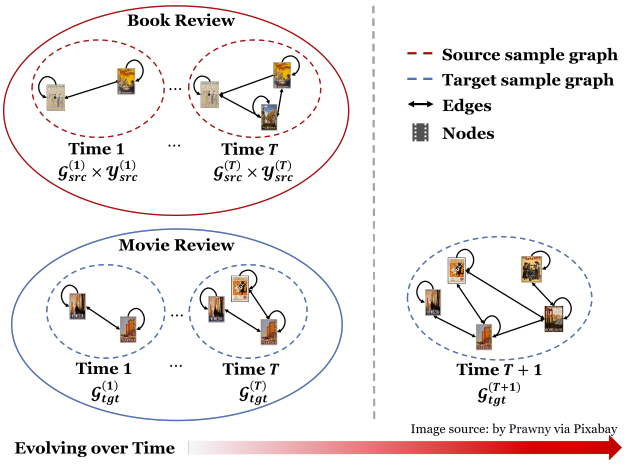

Figure 2. An illustrative example of dynamic non-IID transfer learning on book review graph and movie review graph. As an example, consider a new series launched on a movie website, where the original book of this series may have been published for decades. It is very natural to transfer knowledge from the information-rich source domain (book) to the information-scarce target domain (movie) across time in order to solve the target task (movie review prediction) at $\mathcal{G}_{tgt}^{(T+1)}$.

Given the notations above, we formally define the problem as follows.

Problem 1. ***Dynamic Non-IID Transfer Learning on Graphs***

Given: *(i) a set of source sample graphs* $\{\mathcal{G}_{src}^{(i)} = (\mathcal{V}_{src}^{(i)}, \mathcal{E}_{src}^{(i)})\}_{i=1}^{T}$ *with rich label information* $\{\mathcal{Y}_{src}^{(i)}\}_{i=1}^{T}$*, and (ii) a set of target sample graphs* $\{\mathcal{G}_{tgt}^{(i)} = (\mathcal{V}_{tgt}^{(i)}, \mathcal{E}_{tgt}^{(i)})\}_{i=1}^{T+1}$ *with few label information* $\{\tilde{\mathcal{Y}}_{tgt}^{(i)}\}_{i=1}^{T+1}$.

Find: *Accurate predictions* $\hat{\mathcal{Y}}_{tgt}^{(T+1)}$ *of unlabeled examples in the target sample graph* $\mathcal{G}_{tgt}^{(T+1)}$.

3. Model

In this section, we introduce our proposed framework EVOLUNET for dynamic non-IID transfer learning on graphs. The key idea lies in regularizing the underlying evolving domain discrepancy, which mainly stems from the distribution shift due to domain evolution and the inherent domain discrepancy between the source and target domains. In particular, we start with deriving a novel generalization bound of Problem 1, which is composed of historical empirical errors on the source and target domains, domain discrepancies across time on source and target, and Rademacher complexity of the hypothesis class. Inspired by the theoretical results, we then develop the overall learning paradigm of EVOLUNET and discuss the details of how to model domain evolution and how to unify dynamic graph distribution. Finally, we present an optimization algorithm with pseudo-code for EVOLUNET in Algorithm 1 in Appendix B.

3.1. Theoretical Analysis

Here, we propose the very first generalization guarantee under the setting of dynamic non-IID transfer learning on graphs. The existing literature (Wu & He, 2022) leads to a loosely bound in special cases when the historical empirical error at a specific timestamp is extremely large. This is because Wu and He's work simply accumulates the empirical errors across time, which results in their generalization error bound being sensitive to extreme cases.

To derive a better error bound, we propose to improve our bound mainly from the following three aspects: (1) We propose to replace $\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h)+\hat{\epsilon}_{tgt}^{(i)}(h)\right)$ with the minimum value of historical empirical errors on source and target. The conventional measurement is notably susceptible to outliers over time. This sensitivity becomes particularly evident when a machine learning model encounters failures at specific timestamps, leading to exceptionally large empirical errors in both the source and target domains. In contrast, using the minimum value demonstrates inherent resilience to such extreme cases and prevents the error bound from being impacted by some extreme cases. (2) We develop a novel dynamic Wasserstein distance to replace maximum mean discrepancy $\tilde{d}$ for better measuring the evolving domain discrepancy. (3) To accurately characterize the graph distribution shift, we propose to construct the Weisfeiler-Lehman subgraphs at each timestamp and then compute the dynamic graph discrepancy upon them.

We first introduce the definition of dynamic p-Wasserstein distance on graphs, which measures the graph discrepancy across tasks and across time stamps.

Definition 1 (Dynamic p-Wasserstein Distance on Graphs)**.** *Consider two dynamic graphs $\{\mathcal{G}_{src}^{(i)}\}_{i=1}^{T}$ and $\{\mathcal{G}_{tgt}^{(i)}\}_{i=1}^{T+1}$. For any $p \geq 1$, the dynamic p-Wasserstein distance is defined as:*

$$\tilde{W}_p = \rho\sqrt{R^2+1}\max\left(\max_{1\leq i\leq T-1} d_{GSD}(\mathcal{G}_{src}^{(i)},\mathcal{G}_{src}^{(i+1)}),\right.$$
$$\left. d_{GSD}(\mathcal{G}_{src}^{(1)},\mathcal{G}_{tgt}^{(1)}),\max_{1\leq i\leq T} d_{GSD}(\mathcal{G}_{tgt}^{(i)},\mathcal{G}_{tgt}^{(i+1)})\right),$$

where R and ρ are the Lipschitz constants, d_{GSD} denotes the graph discrepancy based on p-Wasserstein distance W_p (Wu et al., 2023).

Based on Definition 1, we can apply Lemma 1 (Error Difference over Shifted Domains) to bound the error difference on two arbitrary domains as follows.

Lemma 1 (Error Difference over Shifted Domains (Wang et al., 2022))**.** *For arbitrary classifier h and loss function $\mathcal{L}$ satisfying Assumption 1 and 2, the expected error of h on two arbitrary domain $\mathcal{D}_\mu$ and $\mathcal{D}_\nu$ satisfies*

$$|\epsilon_\mu(h)-\epsilon_\nu(h)| \leq \rho\sqrt{R^2+1}W_p(\mathcal{D}_\mu,\mathcal{D}_\nu),$$

where W_p is the p-Wasserstein distance metric and $p \geq 1$.

Intuitively, Lemma 1 yields that the expected error on the target domain at the $N+1$ timestamp $\epsilon_{tgt}^{(T+1)}$ is upper bounded with an expected error on an arbitrary domain and the maximum of measures of domain discrepancy. Based on Lemma 1, we can further generalize the difference between the expected error and the empirical error to the arbitrary domains via Lemma 2 (Algorithm Stability) as follows.

Lemma 2 (Algorithm Stability, from Lemma A.1 in Kumar et al. (2020))**.** *With the assumptions 1, 2, 3, consider empirical and expected errors on arbitrary domain with n samples, $\forall\,\delta\in(0,1)$, the following holds with probability at least $1-\delta$ for some constant $B>0$,*

$$|\hat{\epsilon}(h)-\epsilon(h)| \leq \mathcal{O}\left(\frac{\rho B+\sqrt{\log\frac{1}{\delta}}}{\sqrt{n}}\right).$$

With Lemma 2, we are able to bound $\epsilon_{tgt}^{(T+1)}$ with minimal empirical errors on the source and target and the maximum domain discrepancy. That being said, the error of the latest target domain $\epsilon_{tgt}^{(T+1)}$ can be bounded. Finally, we can derive our generalization bound for dynamic non-IID transfer learning on graphs, as stated by the following Theorem 1.

Theorem 1. *Assume classifier $h\in\mathcal{H}$ is R-Lipschitz and loss function $\mathcal{L}(\cdot,\cdot)$ is ρ-Lipschitz, where R and ρ are the Lipschitz constants. For any $\delta>0$, with probability at least $1-\delta$, the error $\epsilon_{tgt}^{(T+1)}$ is bounded by:*

$$\epsilon_{tgt}^{(T+1)}(h) \leq \frac{1}{2}\min_{1\leq i\leq T}\left(\hat{\epsilon}_{src}^{(i)}(h)+\hat{\epsilon}_{tgt}^{(i)}(h)\right)+\frac{3T}{2}\tilde{W}_p$$
$$+\tilde{\Re}\left(\mathcal{H}_{\mathcal{L}}\right)+\mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}}+\sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right) \quad (2)$$

where $\tilde{W}_p$ is dynamic Wasserstein distance on graphs, $p \geq 1$, $\mathcal{H}_{\mathcal{L}} = \{(\mathbf{X}, y) \mapsto \mathcal{L}(h(\mathbf{X}), y) : h \in \mathcal{H}\}$, $\tilde{\mathfrak{R}}(\mathcal{H}_{\mathcal{L}}) = \frac{1}{2T}\sum_{i=1}^{T}\left(\tilde{\mathfrak{R}}_{\mathcal{D}_{src}^{(i)}}(\mathcal{H}_{\mathcal{L}}) + \tilde{\mathfrak{R}}_{\mathcal{D}_{tgt}^{(i)}}(\mathcal{H}_{\mathcal{L}})\right)$, $\tilde{\mathfrak{R}}$ is Rademacher complexity, $B > 0$ is a constant, and $\tilde{n} = \min_{1\leq i\leq T}\left(N_{src}^{(i)}, N_{tgt}^{(i)}\right)$ is the minimal number of training examples in source and target domains.

Proof. The detailed proof is provided in Appendix A. □

The theorem shows that the error on the latest target domain $\epsilon_{tgt}^{(T+1)}$ is bounded in terms of (1) the minimum value of empirical errors in the historical source and target domains; (2) the maximum of domain discrepancies across time and domain; (3) the average Rademacher complexity of hypothesis class over all domains.

Remarks: Compared to the existing theoretical results on dynamic transfer learning (Wu & He, 2022), we obtain a significantly improved bound in the following aspects.

- Instead of simply averaging the errors over time as (Wu & He, 2022), we propose to use the minimum of empirical errors over time to imply domain evolution and avoid extreme errors, and we have

$$\min_{1\leq i\leq T}\left(\hat{\epsilon}_{src}^{(i)}(h) + \hat{\epsilon}_{tgt}^{(i)}(h)\right) \leq \frac{1}{T}\sum_{1\leq i\leq T}\left(\hat{\epsilon}_{src}^{(i)}(h) + \hat{\epsilon}_{tgt}^{(i)}(h)\right).$$

  Correspondingly, rather than using an accumulative method to consider all timestamps, EvoluNet uses multi-resolution temporal encoding and attention mechanisms to consider domain evolution uniformly.

- Instead of separately measuring the difference of features and the difference of labels based on MMD, we propose a dynamic Wasserstein distance on graphs to model the evolving graph discrepancy. Correspondingly, EvoluNet leverages dual-divergence unification to implicitly reduce this distance (Ganin et al., 2016; Ganin & Lempitsky, 2015).

In general, this generalization bound guarantees the transferability from evolving source domains to evolving target domains and motivates us to propose a framework for dynamic non-IID transfer learning on graphs by empirically minimizing generalization bounds with domain evolution and domain discrepancy.

3.2. EvoluNet Framework

Without loss of generality, a typical dynamic transfer learning paradigm can be formulated as follows.

$$\min_{\theta}\mathcal{L}(\theta) = \sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(\theta) + d(\mathcal{G}_{src}^{(i)}, \mathcal{G}_{tgt}^{(i)}, \theta)\right) \quad (3)$$

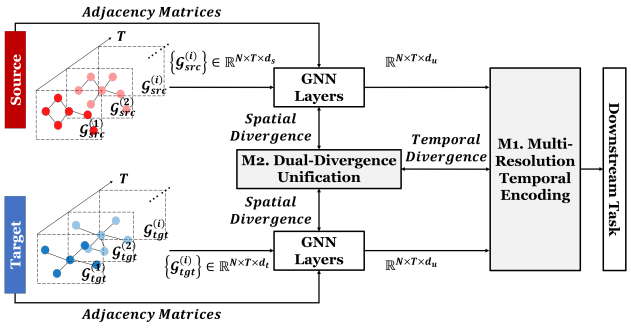

Figure 3. The proposed EvoluNet framework.

However, Eq. 3 may not well capture evolving domain discrepancy in practice due to the following two reasons. First, Eq. 3 simply sums up the empirical errors over time, which ignores the evolution process of dynamic graphs, i.e., the changes in the future snapshot $\mathcal{G}_{src}^{(t+1)}$ are often highly dependent on the structure of the current snapshot $\mathcal{G}_{src}^{(t)}$ (Kazemi et al., 2020). Second, accumulating the domain discrepancy over all timestamps might lose track of the fine-grained information on how domain discrepancies evolve, e.g., the domain discrepancy $d(\mathcal{G}_{src}^{(T)}, \mathcal{G}_{tgt}^{(T)}, \theta)$ in the last timestamp could play a key role in the success of the downstream task in the timestamp $T+1$.

As shown in Theorem 1, the generalization performance is dominated by two factors: the domain evolution across time and the domain discrepancy on source and target. Inspired by this, we propose EvoluNet, which consists of two major modules: M1. Modeling Domain Evolution via Multi-Resolution Temporal Encoding and M2. Domain-Invariant Learning via Dual-Divergence Unification. In particular, M1 introduces a multi-resolution temporal encoding for dynamic graphs, which encodes temporal information into the representation with continuous values and captures domain evolution by attention; M2 further unifies disparate spatial and temporal information of source and target into the domain-invariant hidden spaces. In addition, both M1 and M2 are absolutely necessary to overcome the main obstacles in dynamic non-IID transfer learning on graphs. M1 ensures accurate modeling domain evolution and characterizes historical temporal information for future downstream task-related representation learning, while M2 ensures extraction of domain-invariant spatial and temporal information that could be transferred to benefit the target domain. Our ablation study (Table 4) firmly attests both M1 and M2 are essential in a successful dynamic graph transfer. The overview of EvoluNet is presented in Figure 3. Next, we dive into the technical details of M1 and M2.

M1. Modeling Domain Evolution via Multi-Resolution Temporal Encoding. Different from Wu & He (2022) that

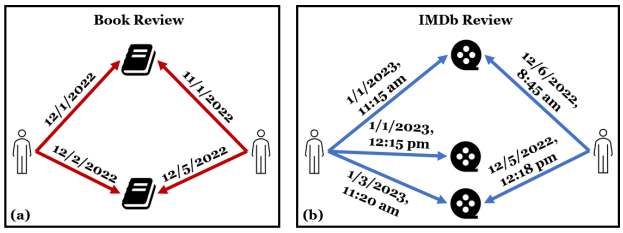

Figure 4. An illustrative example of why multi-resolution temporal encoding is important.

accumulates dynamic domain discrepancy over all timestamps, our method introduces a novel perspective by specifically focusing on the dynamic domain discrepancy within selected time windows. We treat each domain's time window as an integrated entity, utilizing Transformers (Vaswani et al., 2017) for its significant achievements in performance and computational efficiency across a variety of sequential data tasks, particularly in natural language processing. However, the positional encoding used in traditional Transformer models primarily serves to distinguish the sequential order of inputs rather than actual continuous time values. This limitation becomes particularly evident when dealing with temporal graphs that are observed at multiple timestamps, that is, the timestamps are multi-resolution and the time gap between inputs may differ.

One key challenge in temporal modeling for dynamic graphs lies in the fact that each snapshot graph is tagged with a timestamp that is both continuous and often irregular (shown in Figure 4). Such timestamps defy simple arithmetic operations, complicating the modeling process. To address this issue, we introduce an innovative approach of multi-resolution temporal encoding, which serves as a replacement for conventional positional encoding. This method enables our framework, EvoluNet, to adeptly encode temporal information across multiple resolutions into a learnable representation as follows:

$$\texttt{ENC} = \sum_{i=1}^{T} \texttt{POSITION}(\texttt{CONTEXT}(\mathcal{G}_{src}^{(i)}, \mathcal{G}_{tgt}^{(i)})) \quad (4)$$

where CONTEXT is the graph context extraction function (Starnini et al., 2012) that extracts temporal random walks from the input graphs, the POSITION is the positional encoding function (Dai et al., 2019) that considers node as a token, continuous-valued timestamp as a position to capture the multi-resolution temporal information.

Next, we introduce the cross-domain self-attention layer to obtain important temporal graph representation for domain evolution. Notably, through previous operations in this framework, node embeddings of source and target sample graphs are converted into the same dimension d_u. Thus, a parameter-shared attention layer can be used for source and target domains to learn domain-invariant temporal node embeddings and also improve the model scalability because of its parallelism. Specifically, for each node, we group its temporal embeddings across all the timestamps and pack them into a matrix where the order is consistent with the corresponding timestamps. This temporal-related matrix is passed to the self-attention layer, and the output indicates the relevance and importance of different timestamps for capturing domain evolution knowledge in terms of a specific node. Our cross-domain self-attention layer has advantages in two aspects: (i) By deploying the attention layer on the source domain (target domain), we effectively capture the temporal dynamics of each domain. This allows us to model $d_{\text{GSD}}(\mathcal{G}_{src}^{(i)}, \mathcal{G}_{src}^{(i+1)})$ ($d_{\text{GSD}}(\mathcal{G}_{tgt}^{(i)}, \mathcal{G}_{tgt}^{(i+1)})$) in error bound. (ii) By sharing the attention parameters of the source and target domains, we are able to capture $d_{\text{GSD}}(\mathcal{G}_{src}^{(1)}, \mathcal{G}_{tgt}^{(1)})$ in the error bounds.

M2. Domain-Invariant Learning via Dual-Divergence Unification. To address the aforementioned dynamic domain divergence, we aim to learn invariant representations across evolving graphs. Nonetheless, the process of transferring knowledge from graph-formatted data introduces inherent spatial and temporal divergences, necessitating the adoption of a dual-divergence unification approach to learning domain-invariant representations across both spatial and temporal dimensions. In response, we present a dual-divergence unification module shown in Figure 3. In our implementation, we first standardize the feature dimension sizes from d_{src}, d_{tgt} to a unified dimension d_u using multi-layer perceptrons (MLPs). This standardization allows for the sharing of GNN parameters between source and target sample graphs, enhancing the learning of spatial information. We unify the MLP and the GNN into one unit named GNN Layers, and then one Gradient Reversal Layer (GRL, (Ganin et al., 2016)) is utilized on this unit to obtain the spatial invariant representation across domains. In parallel, temporal invariance is secured by employing the GRL after the assimilation of domain evolution insights and the derivation of temporal graph representations through multi-resolution temporal encoding by module 1 (M1). The loss function $\mathcal{L}_{GRL}$ of M2 can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{\text{GRL}} &= \texttt{UNIF}_{spatial} + \texttt{UNIF}_{temporal} \\ &= \sum_{i=1}^{T} \text{GRL}\left(\text{GNN}(\mathcal{G}_{src}^{(i)}), \text{GNN}(\mathcal{G}_{tgt}^{(i)})\right) \\ &+ \sum_{i=1}^{T} \text{GRL}\left(\text{M1}(\mathcal{G}_{src}^{(i)}), \text{M1}(\mathcal{G}_{tgt}^{(i)})\right)\end{aligned}$$

where $\texttt{UNIF}_{spatial}$ and $\texttt{UNIF}_{temporal}$ represent the spatial divergence loss on GNN Layers and the temporal divergence loss on temporal graph representation after M1, respectively.

Overall, the objective function is defined to minimize the

dual-divergence GRL loss (for all sample graphs) and the node classification loss (for source sample graphs and the few labeled nodes in target sample graphs). We detail the optimization process for EVOLUNET, as delineated in the pseudo-code provided in Algorithm 1 in Appendix B.

4. Experiments

In this section, we evaluate the performance of EVOLUNET on six benchmark datasets. EVOLUNET exhibits superior performances compared to various state-of-the-art baselines (Section 4.2). Moreover, we conduct ablation studies (Section 4.3) and sensitivity analysis (Section 4.4) to demonstrate the necessity of each module in EVOLUNET and the reliability of EVOLUNET in various parameter settings.

4.1. Experiment Setup

Datasets: We evaluate EVOLUNET on our benchmark which is composed of three real-world graphs, including two graphs extracted from Digital Bibliography & Library Project: DBLP-3 and DBLP-5 (Fan et al., 2021), where nodes represent authors, edges represent the co-authorship between two linked nodes; and one graph generated from human connectome project: HCP (Fan et al., 2021), where nodes represent cubes of brain tissue, edges represent that two linked cubes show similar degrees of activation. Each node of these three graphs is associated with one label only.

Our benchmark follows these principles: (1) Dynamic: data follows the settings of graph and label evolution. (2) Transferability: there are existing works that have explored knowledge transfer across heterogeneous domains (Moon & Carbonell, 2017; Day & Khoshgoftaar, 2017). Some simple guesses are that the two graphs may have structural similarities allowing knowledge transfer (Zhu et al., 2021) or the attention mechanism suppressing the performance drop with heterogeneity (Moon & Carbonell, 2017). To explore the potential structural similarities of the three datasets, we employ EEE-plot (Prakash et al., 2010), which is a scatter plot of the first three singular vectors of the adjacency matrix. In Figure 5, we observe there are spokes observed on the EEE-plots of three datasets, associating with the presence of well-defined communities in graphs (Prakash et al., 2010). The results suggest a similarity in structure across the three datasets, providing insights into the possibility of knowledge transferability among them. Our experiments also prove the validity of positive knowledge transfer. The details of our benchmark are summarized in Table 2.

Baselines: We compare EVOLUNET with four classical graph neural networks, four temporal graph neural networks and two graph transfer learning methods.

- Classical GNNs: Graph Convolutional Network (GCN, Kipf & Welling (2017)), Graph Attention Network (GAT, Veličković et al. (2018)), Graph Isomorphism Network (GIN, Xu et al. (2019)), Graph SAmple and aggreGatE (GraphSAGE, Hamilton et al. (2017)) are four standard graph representation benchmark architectures.
- Temporal GNNs: Diffusion Convolutional Recurrent Neural Network (DCRNN, Li et al. (2018)) captures both spatial and temporal dependencies of graphs among time series. Dynamic Graph Encoder (DyGrEncoder, Taheri & Berger-Wolf (2019)) models embedding GNN to LSTM. Evolving Graph Convolutional Network (EvolveGCN, Pareja et al. (2020)) uses a GCN evolved by a Recurrent Neural Network (RNN) to capture the dynamism of graph sequence. Temporal Graph Convolutional Network (TGCN, Zhao et al. (2019b)) is a combination of GCN and the gated recurrent unit.
- Transfer Learning Methods: Domain-Adversarial Neural Networks (DANN, Ganin et al. (2016)) is the first method using GRL for domain adaptation. Unsupervised Domain Adaptive Graph Convolutional Network (UDAGCN, Wu et al. (2020)) is a method for domain adaptation in the static graph using the attention mechanism. GRaph ADaptive Network (GRADE, Wu et al. (2023)) is a method for cross-network knowledge transfer from the perspective of the Weisfeiler-Lehman graph isomorphism test.

The implementation details of the methods are provided in Appendix C.

Table 2. Benchmark statistics.

Benchmark	Source	Target	Benchmark	Source	Target
1	DBLP-5	DBLP-3	4	HCP	DBLP-5
2	HCP	DBLP-3	5	DBLP-3	HCP
3	DBLP-3	DBLP-5	6	DBLP-5	HCP

Dataset	#Nodes	#Edges	#Attributes	#Classes	#Timestamps
DBLP-3	4,257	23,540	100	3	10
DBLP-5	6,606	42,815	100	5	10
HCP	5,000	1,955,488	20	10	12

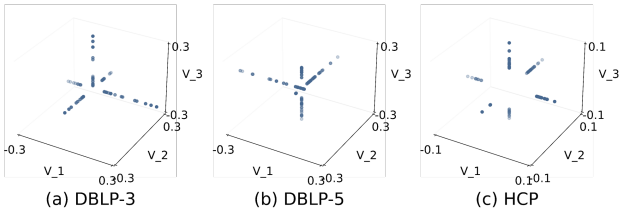

Figure 5. The EEE-plots of temporal graphs in Table 2.

4.2. Effectiveness

We compare EVOLUNET with eleven baseline methods across three real-world undirected graphs. We report the AUC of different methods on the last timestamp of the target domain in Table 3. In general, we have the following

Table 3. Comparison of different methods in node classification task using 5 labeled samples per class (area under the curve, AUC). The first four models are Classical GNN models and the next four are Temporal GNNs, we show their fine-tuned results on the target domain. The remaining three models are for transfer learning. We show results of knowledge transfer from source to target domain.

	Classical GNNs				Temporal GNNs				Transfer learning			Ours
	GCN	GAT	GIN	GraphSAGE	DCRNN	DyGrEncoder	EvolveGCN	TGCN	DANN	UDAGCN	GRADE	EvoluNet
Benchmark 1	0.5609	0.5489	0.5454	0.5452	0.5637	0.5672	0.5823	0.5640	0.5416	0.5688	0.5246	**0.6527**
Benchmark 2									0.5400	0.5523	0.5223	**0.6103**
Benchmark 3	0.5404	0.5387	0.5422	0.5390	0.5518	0.5489	0.5610	0.5482	0.5395	0.5660	0.5295	**0.5915**
Benchmark 4									0.5348	0.5651	0.5354	**0.5769**
Benchmark 5	0.6756	0.6964	0.6962	0.6798	0.5710	0.6363	0.5679	0.5695	0.6977	0.7407	0.5170	**0.7975**
Benchmark 6									0.6981	0.7320	0.5154	**0.8046**

Table 4. Ablation study (AUC).

Ablation	Benchmark 1	Benchmark 5	Benchmark 6
w/o pre-training	0.5907	0.7661	0.7234
w/o module 1	0.6487	0.7682	0.7303
w/o $\texttt{UNIF}_{spatial}$	0.6367	0.7939	0.7985
w/o $\texttt{UNIF}_{temporal}$	0.6341	0.7966	0.8021
EvoluNet	**0.6527**	**0.7975**	**0.8046**

observations: (1) EvoluNet consistently outperforms all eleven baselines on all the datasets, which demonstrates the effectiveness and generalizability of our model. Especially when adapting knowledge from DBLP-5 to DBLP-3 with five labeled samples per class, the improvement is 12.1% compared with the second-best model (EvolveGCN). (2) Classical GNNs have the worst performance on four benchmarks (1, 2, 3, 4) since they can neither learn knowledge from the previous timestamps nor transfer knowledge from other domains. EvoluNet boosts the performance compared with classical GNNs by up to 16.4% (on benchmark 1). (3) Temporal GNNs achieve second-place performance on Benchmarks 1 and 2, which means in these benchmarks, there is knowledge existing in the previous timestamps that is useful for the label prediction task in the future timestamps. Particularly, EvoluNet still outperforms these temporal GNNs on Benchmarks 1 and 2 by up to 12.1%. Notably, on Benchmarks 5 and 6, all temporal GNNs fail, while EvoluNet can still has the highest performance. (4) Transfer learning models have the second place performance on Benchmarks 5 and 6, which shows the efficacy of the domain knowledge transfer on these two benchmarks. Especially, EvoluNet still does better than this kind of model on Benchmarks 5 and 6 by up to 9.9% AUC.

4.3. Ablation Study

Considering that EvoluNet consists of various components, we set up the following experiments to study the effect of different components by removing one component from EvoluNet at a time: (1) removing the pre-training process; (2) removing module 1, multi-resolution temporal encoding and attention; (3) removing module 2, the dual-divergence losses (including $\texttt{UNIF}_{spatial}$ and $\texttt{UNIF}_{temporal}$). Due to the space limit, we use Benchmark 1, 5, and 6 to illustrate in this section. From Table 4, we have several interesting observations: (1) Pre-training can significantly boost the model performance by up to 11.2% (on Benchmark 6), which indicates the efficacy of knowledge transferring of our model across different graphs under the limited label setting. (2) Module 1 achieves impressive improvement on Benchmark 5 and 6 by up to 10.2%, which shows its strength in temporal transfer learning and also supports our theoretical analysis in Section 3.1. (3) Both dual-divergence losses help the model better adapt knowledge from the source to the target domain. especially on Benchmark 1, the removal of $\texttt{UNIF}_{spatial}$ ($\texttt{UNIF}_{temporal}$) leads to a decrease in AUC by 1.6% (1.8%), p-value < 0.001. This proves the effectiveness of dual GRLs module in alleviating the spatial and temporal divergences. (4) The improvements of M2 are not obvious in Benchmarks 5 and 6, and a simple guess is EvoluNet variation with only M1 already achieves significant improvement than our baselines, so M2 makes less contribution to the final results.

4.4. Parameter Sensitivity Analysis

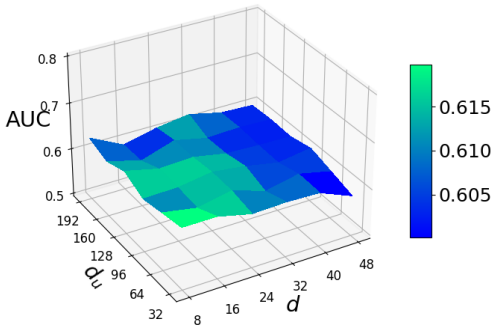

Figure 6. Hyper-parameter analysis on Benchmark 2 with respect to d_u and d.

In this section, we study two hyper-parameters of our model: (1) the size of head dimension d in M1 (modeling domain evolution via multi-resolution temporal encoding); (2) the size of the mapped features d_u of two MLPs in M2 (domain-invariant learning via dual-divergence unification). The result is shown in Figure 6. Based on that, the fluctuation

of the AUC (z-axis) is less than 3%. The AUC is slightly lower when the head dimension of module 1 becomes larger, and different values of d do not affect the AUC significantly. Overall, we find EVOLUNET is reliable and not sensitive to the hyperparameters under study within a wide range.

5. Related Work

In this section, we briefly review the existing literature in the context of transfer learning and graph neural networks.

Transfer learning has exhibited excellent performance in several areas, such as natural language processing (Yang et al., 2021; Ruder et al., 2019), computer vision (Zhang et al., 2022; Alhashim & Wonka, 2018), time series analysis (Bethge et al., 2022; Ismail Fawaz et al., 2018), and healthcare (Panagopoulos et al., 2021). Then, several works named "continuous transfer (Wang et al., 2021; 2020b; Desai et al., 2020)" or "dynamic domain adaptation (Li et al., 2021; Ke et al., 2021; Mancini et al., 2019)" are proposed to learn the evolving data. For example, Minku (2019) manually partitioned source data into several evolving parts and managed to solve the non-stationary source domain by performing transfer learning. There are also some works (Hoffman et al., 2014; Ortiz-Jiménez et al., 2019; Liu et al., 2020; Wu & He, 2020; Wang et al., 2020a; Kumar et al., 2020; Xie et al., 2021b) that addressed the scenario in which the source domain is static, and the target domain is continually evolving. Recently, Wu & He (2022) modeled the knowledge transferability with dynamic source domain and dynamic target domain and defined this problem as "dynamic transfer learning." Despite the success of dynamic transfer learning, no effort has been made to solve the problem of graph-structured data. In this paper, we aim to explore the knowledge transferability across graphs.

Graph neural networks capture the structure of graphs via message passing between nodes. Many significant efforts such as GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), GIN (Xu et al., 2019) arose and have become indispensable baseline in a wide range of downstream tasks. Here we do not intend to provide a comprehensive survey of the wide range of GNNs. Instead, we refer the reader to excellent recent surveys to get more familiar with the topics (Wu et al., 2021; Zhou et al., 2020b). Recently, several attempts have been focused on generalizing GNN from static graphs to dynamic graphs (Yu et al., 2018; Schlichtkrull et al., 2018; Rossi et al., 2020; Skarding et al., 2021; Kim et al., 2022; You et al., 2022; Cong et al., 2023; Yu et al., 2023). Specifically, Pareja et al. (2020) utilize common GCNs to learn node representations on each static graph snapshot and then aggregate these representations from the temporal dimension. While Xu et al. (2020) first propose to use time embedding and design a temporal graph attention layer to concatenate node, edge, and time features efficiently. However, the Dynamic GNN strategies often lack the capability of transferring knowledge, thus limiting their ability to leverage valuable information from other data sources. Here we further extend it to the transfer learning setting with dynamic source and target domains.

6. Conclusion

In this paper, we investigate a novel problem named dynamic non-IID transfer learning on graphs, which intends to augment knowledge transfer from dynamic source graphs to dynamic target graphs. We shed light on C1 (Generalization bound) by proposing a new generalized bound in terms of historical empirical error and domain discrepancy. We also present EVOLUNET, an end-to-end framework with two major modules: M1. modeling domain evolution via multi-resolution temporal encoding and M2. domain-invariant learning via dual-divergence unification to alleviate evolving domain discrepancy that is specified in C2 (Computational framework). Extensive experiments on our carefully prepared benchmark, where EVOLUNET consistently outperforms leading baselines, demonstrate the efficacy of our model for dynamic non-IID transfer learning on graphs.

Reproducibility: We have released our code and data at `https://github.com/wanghh7/EvoluNet`.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work is supported by 4-VA, Cisco, Commonwealth Cyber Initiative, DARPA under the contract No. HR00112490370, Deloitte & Touche LLP, DHS CINA, the National Science Foundation under Award No. IIS-2339989, and Virginia Tech. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alhashim, I. and Wonka, P. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941:arXiv:1812.11941, 2018.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 19. MIT Press, 2006.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine Learning*, 79(1–2):151–175, 2010.

Bethge, D., Hallgarten, P., Grosse-Puppendahl, T., Kari, M., Mikut, R., Schmidt, A., and Özdenizci, O. Domain-invariant representation learning from eeg with private encoders. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1236–1240, 2022.

Cong, W., Zhang, S., Kang, J., Yuan, B., Wu, H., Zhou, X., Tong, H., and Mahdavi, M. Do we really need complicated model architectures for temporal networks? In *International Conference on Learning Representations (ICLR)*, 2023.

Cui, H., Dai, W., Zhu, Y., Li, X., He, L., and Yang, C. Interpretable graph neural networks for connectome-based brain disorder analysis. In *Medical Image Computing and Computer Assisted Intervention*, pp. 375–385, Cham, 2022. Springer Nature Switzerland.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. In *Association for Computational Linguistics (ACL)*, pp. 2978–2988, 2019.

Day, O. and Khoshgoftaar, T. M. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4:29, 2017.

Desai, S., Durugkar, I., Karnan, H., Warnell, G., Hanna, J., and Stone, P. An imitation from observation approach to transfer learning with dynamics mismatch. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 3917–3929, 2020.

Fallani, F. D. V., Richiardi, J., Chavez, M., and Achard, S. Graph analysis of functional brain networks: practical issues in translational neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369 (1653):20130521, 2014.

Fan, Y., Yao, Y., and Joe-Wong, C. Gcn-se: Attention as explainability for node classification in dynamic graphs. In *IEEE International Conference on Data Mining (ICDM)*, pp. 1060–1065. IEEE, 2021.

Fang, Y., Zhang, Q., Yang, H., Zhuang, X., Deng, S., Zhang, W., Qin, M., Chen, Z., Fan, X., and Chen, H. Molecular contrastive learning with chemical element knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, volume 36, pp. 3968–3976, 2022.

Fu, D., Xu, Z., Li, B., Tong, H., and He, J. A view-adversarial framework for multi-view network embedding. In *International Conference on Information and Knowledge Management (CIKM)*, pp. 2025–2028. ACM, 2020.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1180–1189. JMLR.org, 2015.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17:59:1–59:35, 2016.

Greene, D., Doyle, D., and Cunningham, P. Tracking the evolution of communities in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, pp. 176–183, 2010.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Hoffman, J., Darrell, T., and Saenko, K. Continuous manifold based adaptation for evolving visual domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 867–874, 2014.

Hu, S., Xiong, Z., Qu, M., Yuan, X., Côté, M.-A., Liu, Z., and Tang, J. Graph policy network for transferable active learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 10174–10185, 2020.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Transfer learning for time series classification. In *IEEE International Conference on Big Data*, pp. 1367–1376, 2018.

Kazemi, S. M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., and Poupart, P. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

Ke, Z., Liu, B., Ma, N., Xu, H., and Shu, L. Achieving forgetting prevention and knowledge transfer in continual learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22443–22456, 2021.

Kim, S., Yun, S., and Kang, J. Dygrain: An incremental learning framework for dynamic graphs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3157–3163, 7 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, (ICLR)*, 2015.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Kumar, A., Ma, T., and Liang, P. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning (ICML)*, volume 119, pp. 5468–5479. PMLR, 2020.

Li, S., Zhang, J., Ma, W., Liu, C. H., and Li, W. Dynamic domain adaptation for efficient inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7832–7841, 2021.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Liang, P. Statistical learning theory, 2016.

Liu, H., Long, M., Wang, J., and Wang, Y. Learning to adapt to evolving domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mancini, M., Rota Bulò, S., Caputo, B., and Ricci, E. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6568–6577, 2019.

Minku, L. L. Transfer learning in non-stationary environments. *Learning from Data Streams in Evolving Environments*, pp. 13–37, 2019.

Moon, S. and Carbonell, J. G. Completely heterogeneous transfer learning with attention-what and what not to transfer. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2508–2514, 2017.

Ortiz-Jiménez, G., Gheche, M. E., Simou, E., Maretic, H. P., and Frossard, P. Cdot: Continuous domain adaptation using optimal transport. *ArXiv*, abs/1909.11448, 2019.

Panagopoulos, G., Nikolentzos, G., and Vazirgiannis, M. Transfer graph neural networks for pandemic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 4838–4845, 2021.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., and Leiserson, C. E. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, pp. 5363–5370, 2020.

Prakash, B. A., Sridharan, A., Seshadri, M., Machiraju, S., and Faloutsos, C. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 435–448. Springer, 2010.

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., and Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T. Transfer learning in natural language processing. In *The North American Chapter of the Association for Computational Linguistics: Tutorials*, pp. 15–18, 2019.

Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *The Extended Semantic Web Conference*, volume 10843, pp. 593–607. Springer, 2018.

Shen, X., Dai, Q., Chung, F.-l., Lu, W., and Choi, K.-S. Adversarial deep network embedding for cross-network node classification. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2020.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561, 2011.

Skarding, J., Gabrys, B., and Musial, K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., and Tang, J. Session-based social recommendation via dynamic graph attention networks. In *International Conference on Web Search and Data Mining*, pp. 555–563. ACM, 2019.

Starnini, M., Baronchelli, A., Barrat, A., and Pastor-Satorras, R. Random walks on temporal networks. *Physical Review E*, 85(5):056115, 2012.

Taheri, A. and Berger-Wolf, T. Predictive temporal embedding of dynamic graphs. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 57–64, 2019.

Tripuraneni, N., Jordan, M. I., and Jin, C. On the theory of transfer learning: The importance of task diversity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Villani, C. *Optimal transport: old and new*, volume 338. Springer, 2009.

Wang, H., He, H., and Katabi, D. Continuously indexed domain adaptation. In *International Conference on Machine Learning (ICML)*, volume 119, pp. 9898–9907. PMLR, 2020a.

Wang, H., Li, B., and Zhao, H. Understanding gradual domain adaptation: Improved analysis, optimal path and beyond. In *International Conference on Machine Learning (ICML)*, volume 162, pp. 22784–22801. PMLR, 2022.

Wang, J., Chen, Y., Feng, W., Yu, H., Huang, M., and Yang, Q. Transfer learning with dynamic distribution adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(1), 2020b.

Wang, L., Zhang, M., Jia, Z., Li, Q., Bao, C., Ma, K., Zhu, J., and Zhong, Y. Afec: Active forgetting of negative transfer in continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Wang, Z., Dai, Z., Póczos, B., and Carbonell, J. Characterizing and avoiding negative transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11285–11294, 2019.

Wu, J. and He, J. Continuous transfer learning with label-informed distribution alignment. *arXiv preprint arXiv:2006.03230*, 2020.

Wu, J. and He, J. A unified meta-learning framework for dynamic transfer learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3573–3579, 2022.

Wu, J., He, J., and Ainsworth, E. A. Non-iid transfer learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, pp. 10342–10350, 2023.

Wu, M., Pan, S., Zhou, C., Chang, X., and Zhu, X. Unsupervised domain adaptive graph convolutional networks. In *The Web Conference (WWW)*, pp. 1457–1467. ACM, 2020.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

Xie, H., Ma, J., Xiong, L., and Yang, C. J. Federated graph classification over non-iid graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 18839–18852, 2021a.

Xie, J., Huang, B., and Dubljevic, S. Transfer learning for dynamic feature extraction using variational bayesian inference. *IEEE Transactions on Knowledge and Data Engineering*, 2021b.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Yang, H., Chen, H., Zhou, H., and Li, L. Enhancing cross-lingual transfer by manifold mixup. In *International Conference on Learning Representations (ICLR)*, 2021.

You, J., Du, T., and Leskovec, J. Roland: Graph learning framework for dynamic graphs. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 2358–2366. ACM, 2022.

Yu, B., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Yu, L., Sun, L., Du, B., and Lv, W. Towards better dynamic graph learning: New architecture and unified library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhang, M., Singh, H., Chok, L., and Chunara, R. Segmenting across places: The need for fair transfer learning with satellite imagery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2915–2924. IEEE Computer Society, 2022.

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning (ICML)*, 2019a.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2019b.

Zhou, D., Zheng, L., Han, J., and He, J. A data-driven graph generative model for temporal interaction networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 401–411. ACM, 2020a.

Zhou, D., Zheng, L., Fu, D., Han, J., and He, J. Mentorgnn: Deriving curriculum for pre-training gnns. In *ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 2721–2731. ACM, 2022.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020b. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001.

Zhu, Q., Yang, C., Xu, Y., Wang, H., Zhang, C., and Han, J. Transfer learning of graph neural networks with ego-graph information maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1766–1779, 2021.

A. Algorithm Analysis

First, we have the following assumptions from the previous work.

Assumption 1 (R-Lipschitz Classifier (Wang et al., 2022)). *Assume each classifier $h \in \mathcal{H}$ is R-Lipschitz in ℓ_2 norm, i.e., $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$,*

$$|h(\mathbf{x}) - h(\mathbf{x}')| \leq R \|\mathbf{x} - \mathbf{x}'\|_2 .$$

Assumption 2 (ρ-Lipschitz Loss (Wang et al., 2022)). *Assume the loss function $\mathcal{L}(\cdot, \cdot)$ is ρ-Lipschitz if $\exists\ \rho > 0$ such that $\forall \mathbf{x} \in \mathcal{X}$, $y, y' \in \mathcal{Y}$ and $h, h' \in \mathcal{H}$, the following inequalities hold:*

$$|\mathcal{L}(h'(\mathbf{x}), y) - \mathcal{L}(h(\mathbf{x}), y)| \leq \rho |h'(\mathbf{x}) - h(\mathbf{x})| ,$$
$$|\mathcal{L}(h(\mathbf{x}), y') - \mathcal{L}(h(\mathbf{x}), y)| \leq \rho |y' - y| .$$

Assumption 3 (Bounded Model Complexity (Wang et al., 2022; Kumar et al., 2020; Liang, 2016)). *Assume the Rademachor complexity $\tilde{\Re}$ of the hypothesis class $\mathcal{H}$ is bounded, i.e., for some constant $B > 0$,*

$$\tilde{\Re}(\mathcal{H}) = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(\mathbf{x}_i)\right] \leq \frac{B}{\sqrt{n}}$$

where $\sigma_i \sim \text{Uniform}(\{-1, 1\})$ for $i = 1, \ldots, n$.

Next, we give the definition of dynamic Wasserstein distance on graphs, Wasserstein distance between domains, Weisfeiler-Lehman subtree, graph discrepancy, and Rademacher Complexity of hypothesis class.

Definition 1 (Dynamic p-Wasserstein Distance on Graphs). *Consider two dynamic graphs $\{\mathcal{G}_{src}^{(i)}\}_{i=1}^{T}$ and $\{\mathcal{G}_{tgt}^{(i)}\}_{i=1}^{T+1}$. For any $p \geq 1$, the dynamic p-Wasserstein distance is defined as:*

$$\tilde{W}_p = \rho\sqrt{R^2 + 1} \max\Bigg(\max_{1 \leq i \leq T-1} d_{GSD}(\mathcal{G}_{src}^{(i)}, \mathcal{G}_{src}^{(i+1)}),$$
$$d_{GSD}(\mathcal{G}_{src}^{(1)}, \mathcal{G}_{tgt}^{(1)}), \max_{1 \leq i \leq T} d_{GSD}(\mathcal{G}_{tgt}^{(i)}, \mathcal{G}_{tgt}^{(i+1)})\Bigg) ,$$

where R and ρ are the Lipschitz constants, d_{GSD} denotes the graph discrepancy based on p-Wasserstein distance W_p (Wu et al., 2023).

Definition 2 (p-Wasserstein Distance (Villani, 2009)). *Consider two domains $\mathcal{D}_\mu$ and $\mathcal{D}_\nu$. For any $p \geq 1$, their p-Wasserstein distance metric is defined as:*

$$W_p(\mathcal{D}_\mu, \mathcal{D}_\nu) = \left(\inf_{\gamma \in \Gamma(\mathcal{D}_\mu, \mathcal{D}_\nu)} \int d(x, y)^p \, \mathrm{d}\gamma(x, y)\right)^{1/p},$$

where $\Gamma(\mathcal{D}_\mu, \mathcal{D}_\nu)$ is the set of all measures over $\mathcal{D}_\mu \times \mathcal{D}_\nu$.

Definition 3 (Weisfeiler-Lehman subtree (Shervashidze et al., 2011)). *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the Weisfeiler-Lehman subtree of depth m rooted at $\mathbf{v} \in \mathcal{V}$ can be defined as:*

$$f_m(\mathbf{v}) = f_m\left(f_{m-1}(\mathbf{v}); \cup_{\mathbf{u} \in \mathcal{N}(\mathbf{v})} f_{m-1}(\mathbf{u})\right),$$

where $f_0(\mathbf{v})$ is the initial node attributes for node $\mathbf{v}$, $f_i, i = 1, \cdots, m, \cdots$ denotes the labeling function, $\mathcal{N}(\mathbf{v})$ denotes the neighbors of node $\mathbf{v}$.

Definition 4 (Graph Discrepancy (Wu et al., 2023)). *Given two graphs $\mathcal{G}_\mu$ and $\mathcal{G}_\nu$, the graph discrepancy between the two graphs can be represented as:*

$$d_{GSD}(\mathcal{G}_\mu, \mathcal{G}_\nu) = \lim_{M \to \infty} \frac{1}{M + 1} \sum_{m=0}^{M} d_b(\mathcal{G}_\mu^m, \mathcal{G}_\nu^m),$$

where $\mathcal{G}^m$ is the Weisfeiler-Lehman subgraph at depth m for an input graph $\mathcal{G}$, $d_b(\cdot, \cdot)$ is the base domain discrepancy, here we use the p-Wasserstein distance metric W_p.

Definition 5 (Rademacher Complexity (Bartlett & Mendelson, 2002)). *Given a sample $S = (\mathbf{X}_1, \cdots, \mathbf{X}_N) \in \mathcal{X}^N$, the empirical Rademacher complexity of $\mathcal{H}$ given S is defined as:*

$$\hat{\mathfrak{R}}_S(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \sum_{i=1}^{N} \sigma_i h\left(\mathbf{x}_i\right) \mid S = (\mathbf{x}_1, \cdots, \mathbf{x}_N)\right],$$

where $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_m)$ is a vector of independent random variables from the Rademacher distribution.

Then, we use Lemma 1 to bound the error difference between arbitrary two domains and use Lemma 2 to bound the difference between empirical and expected errors.

Lemma 1 (Error Difference over Shifted Domains (Wang et al., 2022)). *For arbitrary classifier h and loss function $\mathcal{L}$ satisfying Assumption 1 and 2, the expected error of h on two arbitrary domain $\mathcal{D}_\mu$ and $\mathcal{D}_\nu$ satisfies*

$$|\epsilon_\mu(h) - \epsilon_\nu(h)| \leq \rho\sqrt{R^2+1} W_p(\mathcal{D}_\mu, \mathcal{D}_\nu),$$

where W_p is the p-Wasserstein distance metric and $p \geq 1$.

Lemma 2 (Algorithm Stability, from Lemma A.1 in Kumar et al. (2020)). *With the assumptions 1, 2, 3, consider empirical and expected errors on arbitrary domain with n samples, $\forall\ \delta \in (0,1)$, the following holds with probability at least $1-\delta$ for some constant $B > 0$,*

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq \mathcal{O}\left(\frac{\rho B + \sqrt{\log\frac{1}{\delta}}}{\sqrt{n}}\right).$$

The proof of Lemma 2 can be found in the proof of Proposition 1 of the proof of Wang et al. (Wang et al., 2022) and Lemma A.1 of Kumar et al. (Kumar et al., 2020).

Lemma 3 (McDiarmid's inequality). *Let function f satisfies for all $1 \leq i \leq N$, and all $\mathbf{X}_1, \cdots, \mathbf{X}_N, \mathbf{X}_i' \in \mathcal{X}$,*

$$|f\left(\mathbf{X}_1, \cdots, \mathbf{X}_i, \cdots, \mathbf{X}_N\right) - f\left(\mathbf{X}_1, \cdots, \mathbf{X}_i', \cdots, \mathbf{X}_N\right)| \leq c_i, \tag{5}$$

where bound $c_1, \cdots, c_N$ are constants. Then, for any $\epsilon > 0$,

$$\Pr[f - \mathbb{E}[f] \geq \epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{N} c_i^2}\right). \tag{6}$$

Based on the above conclusion, Theorem 1 and its proof are given as follows.

Theorem 1. *Assume classifier $h \in \mathcal{H}$ is R-Lipschitz and loss function $\mathcal{L}(\cdot,\cdot)$ is ρ-Lipschitz, where R and ρ are the Lipschitz constants. For any $\delta > 0$, with probability at least $1-\delta$, the error $\epsilon_{tgt}^{(T+1)}$ is bounded by:*

$$\begin{aligned}\epsilon_{tgt}^{(T+1)}(h) \leq \frac{1}{2}\min_{1\leq i\leq T}\left(\hat{\epsilon}_{src}^{(i)}(h) + \hat{\epsilon}_{tgt}^{(i)}(h)\right) + \frac{3T}{2}\tilde{W}_p \\ + \tilde{\mathfrak{R}}\left(\mathcal{H}_\mathcal{L}\right) + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right)\end{aligned} \tag{2}$$

where $\tilde{W}_p$ is dynamic Wasserstein distance on graphs, $p \geq 1$, $\mathcal{H}_\mathcal{L} = \{(\mathbf{X}, y) \mapsto \mathcal{L}(h(\mathbf{X}), y) : h \in \mathcal{H}\}$, $\tilde{\mathfrak{R}}(\mathcal{H}_\mathcal{L}) = \frac{1}{2T}\sum_{i=1}^{T}\left(\tilde{\mathfrak{R}}_{\mathcal{D}_{src}^{(i)}}(\mathcal{H}_\mathcal{L}) + \tilde{\mathfrak{R}}_{\mathcal{D}_{tgt}^{(i)}}(\mathcal{H}_\mathcal{L})\right)$, $\tilde{\mathfrak{R}}$ is Rademacher complexity, $B > 0$ is a constant, and $\tilde{n} = \min_{1\leq i\leq T}\left(N_{src}^{(i)}, N_{tgt}^{(i)}\right)$ is the minimal number of training examples in source and target domains.

Proof. For the sake of simplicity here, we use $\mathcal{G}_{src}^{(i)}$ and $\mathcal{G}_{tgt}^{(i)}$ be the Weisfeiler-Lehman subgraphs of source domain and the target domain at i^{th} timestamp, following the discussion of Wu et al. (2023), the representations can be considered

as *conditionally independent* with respect to Weisfeiler-Lehman subgraph. $\mathcal{B} \in (\mathcal{G} \times \mathcal{Y})^{\tilde{n}}$ is the measurable subset over $\mathcal{G}_{src}^{(1)} \times \cdots \times \mathcal{G}_{src}^{(T)} \times \mathcal{G}_{src}^{(1)} \times \cdots \times \mathcal{G}_{tgt}^{(T)}$, and we define a function g over $\mathcal{B}$ as follows (Wu & He, 2022):

$$g(\mathcal{B}) = \sup_{h \in \mathcal{H}} \epsilon_{tgt}^{(T+1)}(h) - \frac{1}{2T} \sum_{i=1}^{T} \left(\hat{\epsilon}_{src}^{(i)}(h) + \hat{\epsilon}_{tgt}^{(i)}(h) \right), \tag{7}$$

where $\hat{\epsilon}_{src}^{(i)}(h) = \frac{1}{N_{src}^{(i)}} \sum_{j=1}^{N_{src}^{(i)}} [\mathcal{L}(h(\mathbf{x}_j), y_j)]$ ($\mathbf{x}_j$ is the feature of j^{th} sample in $\mathbf{X}_{src}^{(i)}$) and $\hat{\epsilon}_{tgt}^{(i)}(h) = \frac{1}{N_{tgt}^{(i)}} \sum_{j=1}^{N_{tgt}^{(i)}} [\mathcal{L}(h(\mathbf{x}_j), y_j)]$ ($\mathbf{x}_j$ is the feature of j^{th} sample in $\mathbf{X}_{tgt}^{(i)}$) are the estimate errors on graph $\mathcal{G}_{src}^{(i)}$ and $\mathcal{G}_{tgt}^{(i)}$. Let $\mathcal{B}$ and $\mathcal{B}'$ be two measurable subsets containing only one different source sample in $\mathcal{G}_{src}^{(i)}$, then we have

$$|g(\mathcal{B}) - g(\mathcal{B}')| \leq \frac{2\rho}{2N_{tgt}^{(i)}T} \leq \frac{\rho}{\tilde{n}T}.$$

The same result holds for different target samples. Based on McDiarmid's inequality (see Lemma 3), we have for any $\epsilon > 0$,

$$\Pr\left[g(\mathcal{B}) - \mathbb{E}_{\mathcal{B}}[g(\mathcal{B})] \geq \epsilon\right] \leq \exp\left(\frac{-2\tilde{n}T^2\epsilon^2}{\rho^2}\right).$$

Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds:

$$g(\mathcal{B}) \leq \mathbb{E}_{\mathcal{B}}[g(\mathcal{B})] + \frac{\rho}{T}\sqrt{\frac{\log\frac{1}{\delta}}{2\tilde{n}}}.$$

In addition, Definition 4 gives a metric to measure the graph discrepancy based on p-Wasserstein distance W_p, so we can generalize Lemma 1 to graphs. It bounds the population error difference of a classifier between a pair of shifted domains on graphs. For any $h \in \mathcal{H}$ and any $i \in \{1, \cdots, T\}$, we have

$$\begin{aligned}\epsilon_{tgt}^{(i)}(h) &= \epsilon_{src}^{(i)}(h) + \epsilon_{tgt}^{(i)}(h) - \epsilon_{src}^{(i)}(h), \\ &\leq \epsilon_{src}^{(i)} + \rho\sqrt{R^2+1}\, d_{\text{GSD}}(\mathcal{G}_{tgt}^{(i)}, \mathcal{G}_{src}^{(i)}).\end{aligned}$$

Similarly, we have

$$\begin{aligned}\epsilon_{tgt}^{(T+1)}(h) &= \epsilon_{tgt}^{(i)}(h) + \epsilon_{tgt}^{(T+1)}(h) - \epsilon_{tgt}^{(i)}(h), \\ &\leq \epsilon_{tgt}^{(i)} + \rho\sqrt{R^2+1}\, d_{\text{GSD}}(\mathcal{G}_{tgt}^{(T+1)}, \mathcal{G}_{tgt}^{(i)}).\end{aligned}$$

Then, we have

$$\begin{aligned}&\sum_{i=1}^{T} \left(\epsilon_{tgt}^{(T+1)}(h) - \epsilon_{tgt}^{(i)}(h) \right) \\ =&\epsilon_{tgt}^{(T+1)}(h) - \epsilon_{tgt}^{(T)}(h) + \cdots + \epsilon_{tgt}^{(2)}(h) - \epsilon_{tgt}^{(1)}(h) + \sum_{i=2}^{T} \left(\epsilon_{tgt}^{(T+1)}(h) - \epsilon_{tgt}^{(i)}(h) \right) \\ \leq&\rho\sqrt{R^2+1} \left(d_{\text{GSD}}(\mathcal{G}_{tgt}^{(T)}, \mathcal{G}_{tgt}^{(T+1)}) + \cdots + d_{\text{GSD}}(\mathcal{G}_{tgt}^{(1)}, \mathcal{G}_{tgt}^{(2)}) \right) + \sum_{i=2}^{T} \left(\epsilon_{tgt}^{(T+1)}(h) - \epsilon_{tgt}^{(i)}(h) \right) \\ \leq&T\tilde{W}_p + \sum_{i=2}^{T} \left(\epsilon_{tgt}^{(T+1)}(h) - \epsilon_{tgt}^{(i)}(h) \right) \leq \frac{T(T+1)}{2} \tilde{W}_p\end{aligned}$$

Then

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{B}}[g(\mathcal{B})] \\
=&\mathbb{E}_{\mathcal{B}}\left[\sup_{h\in\mathcal{H}}\epsilon_{tgt}^{(T+1)}(h)-\frac{1}{2T}\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h)+\hat{\epsilon}_{tgt}^{(i)}(h)\right)\right] \\
=&\mathbb{E}_{\mathcal{B}}\left[\sup_{h\in\mathcal{H}}\epsilon_{tgt}^{(T+1)}(h)-\frac{1}{2T}\sum_{i=1}^{T}\left(\epsilon_{src}^{(i)}(h)+\epsilon_{tgt}^{(i)}(h)\right)+\frac{1}{2T}\sum_{i=1}^{T}\left(\epsilon_{src}^{(i)}(h)-\hat{\epsilon}_{src}^{(i)}(h)\right)+\frac{1}{2T}\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(i)}(h)-\hat{\epsilon}_{tgt}^{(i)}(h)\right)\right] \\
=&\frac{1}{2T}\sup_{h\in\mathcal{H}}\left(\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(T+1)}(h)-\epsilon_{tgt}^{(i)}(h)\right)+\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(T+1)}(h)-\epsilon_{src}^{(i)}(h)\right)\right) \\
+&\mathbb{E}_{\mathcal{B}}\left[\sup_{h\in\mathcal{H}}\frac{1}{2T}\sum_{i=1}^{T}\left(\epsilon_{src}^{(i)}(h)-\hat{\epsilon}_{src}^{(i)}(h)\right)+\frac{1}{2T}\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(i)}(h)-\hat{\epsilon}_{tgt}^{(i)}(h)\right)\right] \\
\leq&\frac{1}{2T}\sup_{h\in\mathcal{H}}\left(\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(T+1)}(h)-\epsilon_{tgt}^{(i)}(h)\right)+\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(T+1)}(h)-\epsilon_{tgt}^{(i)}(h)\right)+\sum_{i=1}^{T}\left(\epsilon_{tgt}^{(i)}(h)-\epsilon_{src}^{(i)}(h)\right)\right) \\
+&\mathbb{E}_{\mathcal{B}}\left[\frac{1}{2T}\sum_{i=1}^{T}\sup_{h\in\mathcal{H}}\left(\epsilon_{src}^{(i)}(h)-\hat{\epsilon}_{src}^{(i)}(h)\right)+\frac{1}{2T}\sum_{i=1}^{T}\sup_{h\in\mathcal{H}}\left(\epsilon_{tgt}^{(i)}(h)-\hat{\epsilon}_{tgt}^{(i)}(h)\right)\right] \\
\leq&\frac{1}{2T}\left[\frac{T(T+1)}{2}\tilde{W}_p+\frac{T(T+1)}{2}\tilde{W}_p+T\tilde{W}_p\right]+\mathbb{E}_{\mathcal{B}}\left[\frac{1}{2T}\sum_{i=1}^{T}\tilde{\mathfrak{R}}_{\mathcal{D}_{src}^{(i)}}(\mathcal{H}_{\mathcal{L}})+\frac{1}{2T}\sum_{i=1}^{T}\tilde{\mathfrak{R}}_{\mathcal{D}_{tgt}^{(i)}}(\mathcal{H}_{\mathcal{L}})\right] \\
\leq&\frac{T+2}{2}\tilde{W}_p+\tilde{\mathfrak{R}}(\mathcal{H}_{\mathcal{L}}).
\end{aligned}
$$

According to (7), we have for any $h\in\mathcal{H}$,

$$
\epsilon_{tgt}^{(T+1)}(h)\leq\frac{1}{2T}\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h)+\hat{\epsilon}_{tgt}^{(i)}(h)\right)+\mathbb{E}_{\mathcal{B}}[g(\mathcal{B})]+\frac{\rho}{T}\sqrt{\frac{\log\frac{1}{\delta}}{2\tilde{n}}}. \tag{8}
$$

W.l.o.g., we assume $\hat{\epsilon}_{src}^{(1)}\leq\hat{\epsilon}_{src}^{(2)}\leq\cdots\leq\hat{\epsilon}_{src}^{(T)}$ for simplify. Consider the last term in $\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h)\right)$, for some constant $B>0$,

$$
\begin{aligned}
(lemma\ 2)\hat{\epsilon}_{src}^{(T)}\leq&\epsilon_{src}^{(T)}+\mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}}+\sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right) \\
(lemma\ 1)\leq&\epsilon_{src}^{(T-1)}+\rho\sqrt{R^2+1}d_{\text{GSD}}(\mathcal{G}_{src}^{(T)},\mathcal{G}_{src}^{(T-1)})+\mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}}+\sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right) \\
\leq&\cdots \\
\leq&\epsilon_{src}^{(1)}+(T-1)\tilde{W}_p+\mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}}+\sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right) \\
(lemma\ 2)\leq&\hat{\epsilon}_{src}^{(1)}+(T-1)\tilde{W}_p+\mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}}+\sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right).
\end{aligned} \tag{9}
$$

For the second last term in $\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h)\right)$, we have

$$\begin{aligned}(lemma\ 2)\hat{\epsilon}_{src}^{(T-1)} \leq & \epsilon_{src}^{(T-1)} + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log(1/\delta)}{\tilde{n}}}\right) \\ \leq & \epsilon_{src}^{(T)} + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right) \\ (Eq.9) \leq & \hat{\epsilon}_{src}^{(1)} + (T-1)\tilde{W}_p + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right).\end{aligned}$$

It is easy to see that this can be bounded for source or target across time. Generally,

$$\begin{aligned}\frac{1}{2T}\sum_{i=1}^{T}\left(\hat{\epsilon}_{src}^{(i)}(h)\right) &\leq \frac{1}{2}\min_{1\leq i\leq T}(\hat{\epsilon}_{src}^{(i)}) + \frac{T-1}{2}\tilde{W}_p + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right), \\ \frac{1}{2T}\sum_{i=1}^{T}\left(\hat{\epsilon}_{tgt}^{(i)}(h)\right) &\leq \frac{1}{2}\min_{1\leq i\leq T}(\hat{\epsilon}_{tgt}^{(i)}) + \frac{T-1}{2}\tilde{W}_p + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right).\end{aligned}$$

Therefore, from (8), we have

$$\epsilon_{tgt}^{(T+1)}(h) \leq \frac{1}{2}\min_{1\leq i\leq T}\left(\hat{\epsilon}_{src}^{(i)}(h) + \hat{\epsilon}_{tgt}^{(i)}(h)\right) + \frac{3T}{2}\tilde{W}_p + \tilde{\Re}(\mathcal{H}_{\mathcal{L}}) + \mathcal{O}\left(\frac{\rho B}{\sqrt{\tilde{n}}} + \sqrt{\frac{\log\frac{1}{\delta}}{\tilde{n}}}\right). \tag{10}$$

which completes the proof. □

B. Optimization and Pseudo Code

The goal of the training process is to minimize the dual-divergence GRL loss (for all sample graphs) and the node classification loss (for source sample graphs and the few labeled nodes in target sample graphs). The overall loss function can be written as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{\text{GRL}} + \gamma_1 * \mathcal{L}_{task} \tag{11}$$

where $\mathcal{L}_{GRL}$ represents the dual GRL loss, $\mathcal{L}_{task}$ represents the loss for classification on labeled nodes, and the hyperparameter γ_1 balances the contribution of the two terms. In the paper, we consider the node classification task, $\mathcal{L}_{task}$ is therefore defined as follows:

$$\begin{aligned}\mathcal{L}_{task} &= \mathcal{L}_{source} + \mathcal{L}_{target} \\ &= \sum_{i=1}^{T}\mathcal{L}_{\text{CE}}\left(h(\mathcal{G}_{src}^{(i)}), \mathcal{Y}_{src}^{(i)}\right) + \gamma_2 * \sum_{i=1}^{T+1}\mathcal{L}_{\text{CE}}\left(h(\tilde{\mathcal{G}}_{tgt}^{(i)}), \tilde{\mathcal{Y}}_{tgt}^{(i)}\right)\end{aligned} \tag{12}$$

where $h(\cdot)$ is the classifier for the downstream task, $\mathcal{L}_{source}$ and $\mathcal{L}_{target}$ represent the node classification loss on the source and target domains, here we employ cross-entropy loss $\mathcal{L}_{\text{CE}}$, and the contribution of the two terms is balanced by γ_2.

We provide the pseudo-code of EvoluNet in Algorithm 1 and we employ Adam (Kingma & Ba, 2015) as the optimizer. Given a set of source sample graphs $\{\mathcal{G}_{src}^{(i)} = (\mathcal{V}_{src}^{(i)}, \mathcal{E}_{src}^{(i)})\}_{i=1}^{T}$ with rich label information $\{\mathcal{Y}_{src}^{(i)}\}_{i=1}^{T}$, and a set of target graphs $\{\mathcal{G}_{tgt}^{(i)} = (\mathcal{V}_{tgt}^{(i)}, \mathcal{E}_{tgt}^{(i)})\}_{i=1}^{T+1}$ with few label information $\{\tilde{\mathcal{Y}}_{tgt}^{(i)}\}_{i=1}^{T+1}$, our proposed EvoluNet framework aims to predict $\hat{\mathcal{Y}}_{tgt}^{(T+1)}$ in the latest target sample graph $\mathcal{G}_{tgt}^{(T+1)}$. We initialize each of the models and the classifier in Step 1. Steps 2-7 correspond to the pre-train process: in Step 3, we map sample graphs from source and target domains to a shared latent space using two separate MLPs; then the mapped representations are passed to a domain-invariant GNN for computing domain-invariant spatial representations in Step 4; followed by a domain-invariant module 1 for computing domain-invariant temporal graph representations in Step 5; while in Step 6, models are trained by minimizing the objective function. In Steps 8-10, we fine-tune the MLP of the target domain, the domain-invariant GNN, the domain-invariant module 1, and the classifier $h(\cdot)$ on the latest target domain $\mathcal{G}_{tgt}^{(T+1)}$.

Algorithm 1 The EVOLUNET Learning Framework.

Require:

(i) a set of source sample graphs $\{\mathcal{G}_{src}^{(i)} = (\mathcal{V}_{src}^{(i)}, \mathcal{E}_{src}^{(i)})\}_{i=1}^{T}$ with rich label information $\{\mathcal{Y}_{src}^{(i)}\}_{i=1}^{T}$; (ii) a set of target sample graphs $\{\mathcal{G}_{tgt}^{(i)} = (\mathcal{V}_{tgt}^{(i)}, \mathcal{E}_{tgt}^{(i)})\}_{i=1}^{T+1}$ with few label information $\{\tilde{\mathcal{Y}}_{tgt}^{(i)}\}_{i=1}^{T+1}$.

Ensure:

Prediction $\hat{\mathcal{Y}}_{tgt}^{(T+1)}$ of unlabeled examples in $\mathcal{G}_{tgt}^{(T+1)}$.

Initialize two MLPs for source and target, the domain-invariant GNN, the domain-invariant module 1, the dual-divergence unification module, and the classifier $h(\cdot)$ for the downstream task in $\mathcal{G}_{tgt}^{(T+1)}$.
while not converge **do**
  Compute representations in a shared latent space of both $\{\mathcal{G}_{src}^{(i)}\}_{i=1}^{T}$ and $\{\mathcal{G}_{tgt}^{(i)}\}_{i=1}^{T+1}$ via two MLPs.
  Compute domain-invariant spatial representations of both $\{\mathcal{G}_{src}^{(i)}\}_{i=1}^{T}$ and $\{\mathcal{G}_{tgt}^{(i)}\}_{i=1}^{T+1}$ via the domain-invariant GNN and first GRL.
  Compute domain-invariant temporal graph representations of both $\{\mathcal{G}_{src}^{(i)}\}_{i=1}^{T}$ and $\{\mathcal{G}_{tgt}^{(i)}\}_{i=1}^{T+1}$ via the domain-invariant module 1 and second GRL.
  Update the hidden parameters of two MLPs, the GNN, module 1, and the dual-divergence unification module by minimizing the loss function in Eq. 11.
end while
while not converge **do**
  Fine-tune MLP for the target domain, the GNN, module 1, and the classifier $h(\cdot)$ for the downstream task.
end while

C. Implementation Details

We compare EVOLUNET with four classical graph neural networks GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), GIN (Xu et al., 2019), GraphSAGE (Hamilton et al., 2017); four temporal graph neural networks DCRNN (Li et al., 2018), DyGrEncoder (Taheri & Berger-Wolf, 2019), EvolveGCN (Pareja et al., 2020), TGCN (Zhao et al., 2019b)); and three graph transfer learning methods DANN (Ganin et al., 2016), UDAGCN (Wu et al., 2020), GRADE (Wu et al., 2023)). For a fair comparison, the output dimensions of all GNNs including baselines and EVOLUNET are set to 16. We conduct experiments with only five labeled samples in each class of the target dataset and test model performance based on all the rest of the unlabeled nodes. For non-temporal GNNs, since they cannot process dynamic graphs directly, we train each model on the graph of the last timestamp. Specifically, for classical GNNs, they are trained on the target dataset for 1000 epochs; for transfer learning models, after training on the source dataset for 2000 epochs, they are fine-tuned on the target dataset for 600 epochs. We use GCN as the feature extractor of DANN and follow the instructions from the original paper of UDAGCN (Wu et al., 2020) to build a union set for input features between the source and target domains by setting zeros for unshared features. The original code for GRADE does not support cross-domain transfer with different feature and class dimensions; we processed the features with a linear layer and constructed a joint label space. For four temporal GNNs, they are trained using all timestamps of the target dataset for 1000 epochs.

For EVOLUNET, it is firstly pre-trained for 2000 epochs, then fine-tuned on the target dataset for 600 epochs using limited labeled data in each class. Since the label of each node in current benchmarks is consistent in every timestamp, in this paper, the output of module 1 in EVOLUNET is aggregated using the average function over all the timestamps, but our model can easily be applied to the settings where labels of each node are changed in different timestamps by simply removing the aggregation operation. We use Adam optimizer with learning rate 3e-3. Considering the imbalanced label distribution, the area under the receiver of the characteristic curve (AUC) is used as the evaluation metric. We run all the experiments with 25 random seeds. The experiments are performed on a Ubuntu20 machine with 16 3.8GHz AMD Cores and a single 24GB NVIDIA GeForce RTX3090.