RoCoNA: A Robust Continual Learning Framework for Alignment of Dynamic Networks Under Distribution Shift and Domain Differences

Shruti Saxena and Joydeep Chandra

Indian Institute of Technology Patna, India
{shruti_2021cs31,joydeep}@iitp.ac.in

Abstract. Network alignment, which maps the same entities across multiple networks, has gained tremendous interest in recent years. However, most existing alignment methods propose to align static graphs that are merely a single snapshot of real-world networks. These methods fail to model the inherent dynamics of entire networks where the nodes, links, and attributes are bound to change over time, making dynamic network alignment challenging. Moreover, modeling the interaction between two different dynamic graphs comes with additional challenges: (1) catastrophic forgetting while learning the evolution of individual networks, (2) distributional shift on the same dynamic network, and (3) domain differences between the two networks. Hence, to overcome these challenges, we propose RoCoNA, an end-to-end reservoir sampling-based continual learning approach built over streaming Graph Neural Networks (GNNs) that uses a novel shift-induced regularizer to handle distribution drift and domain differences in evolving networks. We empirically show that our method outperforms the existing state-of-the-art static and dynamic alignment methods. We perform case studies on networks with high distributional shifts to strongly validate our claims. Code is available at RoCoNA

Keywords: Network alignment · Continual learning · Distribution shift.

1 Introduction

In the era of Big-data, graphs are one of the most compelling ways of representing and analyzing complex real-world interactions across several domains, such as social networks, co-authorship, and protein interaction networks. More often, the interacting entities in each network (like the persons, authors, and proteins) appear in multiple networks and exhibit different interactional behavior [13]. For example, a user's interaction style across online social networks, like Facebook, Twitter, or LinkedIn, may vary. The participation of a protein in several genetic pathways is also analogous. Network alignment, or mapping the same entities across several networks, can provide these entities a broader feature, benefiting various intriguing applications, including cross-species gene prioritization, fraud detection, and recommendation engines [2].

Comprehensive network alignment research is built upon static networks and focuses on preserving the structural and attribute consistency of the partially aligned networks [3, 12, 16, 21, 24, 25]. Despite their remarkable results, these methods fail to account for the inherent dynamics of the continuously evolving real-world networks. For example, edges are added or removed in social networks, and the attributes of nodes may also change over time. These dynamics give rise to some new patterns while retaining some existing ones. Markedly, network dynamics can help in network alignment by capturing the trends in data [1].

Motivated to capture the network dynamicity for better results in the real-world scenario, we study the problem of aligning entities in two dynamic social networks in this paper. We simultaneously deal with significant challenges that remain in modeling the interaction between two different dynamic networks:

1. *How to learn the evolving behavior of dynamic networks?* Nodes or edges are added or removed in dynamic networks, and the attributes of nodes may also change over time. These dynamics give rise to new patterns while retaining existing ones, and capturing these network dynamics remains challenging.
2. *How to consolidate the natural phenomenon of distributional shift on dynamic networks?* The network structure and node feature distribution of large networks like social networks and citation networks follow a strong time-dependent correlation [8]. For instance, the distributional characteristics of a citation network would go through significant change as new research fields grow, and authors may generate distinct types of social circles in different years.
3. *How to align networks that come from different domains?* The source and target networks can be of different domains with different evolution rates and patterns, and matching their distributional spaces for generalizing the dynamic behavior of both networks remains challenging.

Existing dynamic alignment methods [18, 19] typically focus on the first and third challenges, capturing the network dynamics via an LSTM Autoencoder framework and matching the domains by constructing a common subspace guided by the pre-known correspondences, a.k.a. anchors. However, these methods overlook the significant shift in the data distributions resulting from the temporal augmentations in an evolving network. To the best of our knowledge, we are the first to handle the distribution shifts in the face of dynamic network alignment. Moreover, the existing LSTM-based methods keep learning new patterns, overwriting, and abruptly forgetting the existing patterns, making them prone to the problem of *catastrophic forgetting*.

Given the above challenges and the need to collectively address them, we propose RoCoNA, an end-to-end continual learning-based Graph Neural Network (GNN) model for dynamic network alignment. We use a memory reservoir-based strategy to keep track of the evolution behavior coupled with a novel distribution divergence regularizer to handle distribution drift and domain differences. Since memory capacity is limited, we propose both, a novel PageRank-induced sampling strategy for identifying representative interactions to store in the memory reservoir, and an online memory updation technique that eliminates the least

representative interactions from memory. The distribution divergence regularization is a statistical moment-matching approach involving two strategies: hard distribution matching for addressing the network's distribution shift and soft distribution matching for addressing the domain differences. The end-to-end framework addresses the problem of catastrophic forgetting by ensuring that the model trained over the current timestep still applies for earlier timesteps. It also effectively captures new patterns emerging due to the distributional shift of evolving networks. We empirically validate the effectiveness of RoCoNA on several real-world datasets and compare it to several state-of-the-art static and dynamic network alignment methods.

2 Related Works

Network alignment is the task of identifying the same entities across multiple networks on their common users *a.k.a.* anchors. The earliest works include spectral methods based on a matrix decomposition formulation [10,17,27]. More recently, deep learning-based methods propose joint learning of the two networks optimized over a loss function. Methods like [3, 6, 7] impose strong topological constraints that strictly impose the anchors to be structurally and semantically the same, whereas some methods use noise and adversarial robustness, like contrastive learning [4, 16, 23] and adversarial learning-based approaches [7, 25, 28] to better capture the network variances. However, all these approaches consider networks to be static and fail to exploit their underlying dynamics. Existing studies on dynamic network alignment primarily focus on biological domains where they find similar conserved regions between two networks for function prediction [1, 22]. Hence, they differ from our goal of finding entity mappings. The problem of aligning dynamic social networks has not yet been thoroughly explored, with only a few proposed works. DNA [18] is the first framework that captures each node's dynamic and global consistency for aligning users across dynamic networks. It proposes an LSTM Autoencoder framework for capturing the intra-network dynamics while constructing a common subspace for inter-network alignment. DGA [19] extends it by replacing the Random walk with restart (RWR) encoding module with a graph attention method, reducing DNA's convergence time and parameter size. HDyNA [5] distinguishes itself by proposing a heuristic algorithm that locally updates the weight of only newly arrived nodes during the network evolution.

3 Problem Formulation

Definition 1 (Network as Graph). *We formally define a network* $G = (V, \hat{A}, X)$ *as an undirected, unweighted graph with a node set* $V = \{v_1, v_2, ..v_n\}$*, adjacency matrix* $\hat{A} \in \{0,1\}^{n \times n}$*, and an attribute matrix* $X \in \mathbb{R}^{n \times m}$ *where* $\boldsymbol{x}_i \in \mathbb{R}^m$ *represents the feature vector of the node* v_i*.*

Definition 2 (Dynamic Network). *A dynamic network is a sequence of static graph snapshots,* $\mathcal{G} = \{G^0, G^1, ..., G^T\}$*, with* T *timestamps. At each time step* $t = \{1, 2, .., T\}$*,* $G^t = (V^t, \hat{A}^t, X^t)$*.*

*Problem 1 (**Dynamic Network Alignment**).*
Given: *(1)* two dynamic networks $\mathcal{G}_s = \{G_s^0, G_s^1, .., G_s^T\}$ and $\mathcal{G}_\tau = \{G_\tau^0, G_\tau^1, .., G_\tau^T\}$, and *(2)* a set of anchor node pairs A'.
Output: An alignment matrix $A_t^\star$, $\forall t \in T$ where $A_t^\star(u, v)$ represents the similarity between nodes $u \in V_s^t$ and $v \in V_\tau^t$.

Here, A' is of fixed size, but the size of A^t varies over time.

3.1 Preliminaries: Graph Neural Networks

A typical L-layer GNN architecture iteratively aggregates information from its neighborhood using a learnable aggregator function F^θ, where θ are the learnable weight parameters. Successive layers are stacked together to generate node representations as:

$$Z^{(l)} = F^\theta(\tilde{A}Z^{(l-1)}; \theta) \tag{1}$$

where $Z^{(l)} \in [a, b]^n$, $Z^{(0)} = X$ and $Z^{(l)}(i) = \boldsymbol{h}_i^{(l)}$ is the representation of a node i at the l^{th} layer. We represent the final node representations $Z^{(L)}$ as Z throughout the work for the sake of simplicity. $\tilde{A} = D^{-\frac{1}{2}}(\hat{A} + I)D^{-\frac{1}{2}}$ and D is the degree matrix of $(\hat{A} + I)$. Moreover, graph pooling is adopted to summarize node embeddings for representing the entire graph as:

$$\boldsymbol{z} = \frac{1}{n}\sum_{j=1}^{n} Z_{:j}^T \tag{2}$$

Streaming GNNs are extensions of traditional GNNs in a dynamic environment. Given a dynamic network $\mathcal{G}$, the goal is to learn learn $\left(\theta^1, \theta^2, \ldots, \theta^T\right)$ where θ^t is the parameters of the GNN F at time t. We represent the final latent node representations at any time t as $Z^t = \{\boldsymbol{h}_0^t, \boldsymbol{h}_1^t, .., \boldsymbol{h}_n^t\}$ and the graph representation as $\boldsymbol{z}^t$.

4 Methodology

We build RoCoNA over a continual learning framework for generating node representations of the source and target networks by minimizing the following loss functions: (a) a continual loss function for adapting to the networks' time-induced structural changes and (b) an anchor alignment loss function for preserving the anchor node embedding similarities in the face of distribution shifts in the individual networks. The following sections discuss the generation of continual node representations, aligned representation, and optimization and alignment computation in detail.

4.1 Continual Representations

The dynamic network environment encounters continuous changes in the network patterns, and we aim to learn representations modeled by streaming GNNs that can capture these changes. We propose streaming GNNs via continual learning that learns (1) new patterns that may occur at each timestamped snapshot and (2) historical patterns from previous snapshots to alleviate catastrophic forgetting. We next explain the proposed learning strategy in detail.

Detecting New Patterns: The new node set, ΔV^t, of consecutive snapshots can be intuitively regarded as new patterns. However, even nodes other than ΔV^t can depict new patterns as some changes in the new nodes can affect the interactional behaviors of the existing nodes, and these nodes also need to be retrained. Hence it is crucial to mine the set of nodes that show new patterns during network evolution.

We define new patterns based on the differences in the PageRanks of nodes at each consequent snapshot. A high difference in node ranks implies the occurrence of unusual interactions during $[t-1, t]$ and hence reflects their importance. The underlying intuition is that the dynamic PageRank of ordinary nodes whose neighborhoods are not significantly affected does not change much, and it is unnecessary to recompute their representation [15]. Hence the set of affected nodes between two consecutive snapshots is given by

$$\mathcal{I} = \Delta V^t + \left\{ \left\| \mathbf{r}_u^t - \mathbf{r}_u^{t-1} \right\| > \delta \right\}_{\forall u \in V^t \cap V^{t-1}}, \tag{3}$$

where $\mathbf{r}_u^t$ is the PageRank score of a node u at time t. δ is a hyper-parameter that controls the number of nodes to be treated as new patterns. We utilize [15] for efficiently calculating the dynamic PageRank scores.

Preserving Existing Patterns: We preserve the existing patterns from the previously timestamped snapshots by following the reservoir sampling technique. We take a fixed-size memory, say $\mathcal{M}$, and sample only the most representative interactions from each snapshot. At any time t, the idea is to make the model learn additionally from the consolidated historical information preserved in the reservoir memory.

Therefore, we propose an importance-based sampling strategy based on the PageRank node centrality scores that quantify the relative influence of graph nodes. We scale the constant probability of sampling a random node, p_u^t, in the reservoir memory based on its importance factor. The underlying intuition is that the common samples have limited importance with respect to the model convergence and representing patterns, and hence are more likely to increase the redundant information in the reservoir memory. The scaled probability is calculated as follows:

$$p_{u \leftarrow \mathcal{M}}^t = 1 - \min\left(p_u^t \frac{r_{\max}^t - r_u^t}{c_{\max} - r_{mean}^t}, p_{\mathcal{M}}^t \right), \tag{4}$$

where $r_{\max}^t$ and r_{mean}^t are the maximum and average PageRank scores over all nodes, and $p_{\mathcal{M}}^t$ is a hyperparameter that controls the amount of memory $\mathcal{M}$

updated by the samples at time t. Since the memory is of fixed size, we propose an online memory-updating strategy that discards the nodes with the most stable PageRank scores across all the previous snapshots whenever it gets full. So, at time t, we calculate the probability of a node u being removed from memory by:

$$p^t_{\mathcal{M}\leftarrow u} = \frac{t - t^{\mathcal{M}}_u}{\frac{1}{t}\sum\limits_{t'\in\{t,t-1,\ldots 1\}} |r^{t'}_u - r^{t'-1}_u|}, \tag{5}$$

where $t^{\mathcal{M}}_u$ is the time at which node u was sampled into the memory.

At each timestamp, the encoder model is trained over the newly detected patterns and the sampled patterns in the memory. Our aim is to learn a GNN parameterized by θ on G^t while also maintaining low errors in the previously timestamped snapshots. To achieve this goal, we follow the Bayesian approach as used in EWC [9] and devise the continual loss function as:

$$\mathcal{L}^{CL}_t = \mathcal{L}^{cons}_t + \sum_x \frac{\lambda}{2} F^t_x \left(\theta^t_x - \theta^{(t-1)^*}_x\right)^2 \tag{6}$$

The regularization term solves the overfitting problem caused by replaying the small number of nodes in the fix-sized reservoir memory. F^t_x is the Fisher Information matrix of the x^{th} parameter of θ and is computed from the first order derivatives. $\theta^{(t-1)^*}_x$ represents the optimal set of parameters of the previous snapshot that lead to the least errors, and λ regularizes the importance of information transfer between consecutive snapshots. $\mathcal{L}^{cons}_t$ is the consistency loss for G^t only that ensures that the learned node embeddings are consistent with the network topology of G^t and is given by:

$$\mathcal{L}^{cons}_t = \sum_{i\in\mathcal{M}\cup I} l\left(\theta^t; i\right) = \sum_{i\in\mathcal{M}\cup I} \left\| \tilde{A}^t(i) - \sigma\left(\boldsymbol{h}^t_i (\boldsymbol{h}^t_i)^T\right)\right\|_F, \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The loss is computed over the embeddings at all layers to ensure node neighborhood consistency at different orders.

4.2 Aligned Representations

We leverage the pre-known anchor mappings A' to align the latent representations of the two networks. We increase the similarity between the anchor nodes of each timestamped snapshot pair, (G^t_s, G^t_τ), using a negative sampling strategy. We define the anchor alignment loss function as:

$$\mathcal{L}^{anchor}_t = \frac{1}{|A_t|} \sum_{(i,j)\in A_t} \left(s\left(i,j\right) + \frac{1}{|U_t|} \sum_{(m,n)\in U_t} \left(1 - s\left(m,n\right)\right)\right) \tag{8}$$

where U_t is the set of negative anchor links sampled from the unmapped links across the two networks at time t and $s\left(i,j\right)$ gives a measure of similarity between nodes i and j as:

$$s\left(i,j\right) = \sigma\left((\boldsymbol{h}_i)^T \cdot \boldsymbol{h}_j\right) \tag{9}$$

However, we require more than relying on the anchor alignment loss while dealing with dynamic networks of different domains, evolution rates, and data distributions, where the alignment efficacy directly depends on how effectively we handle these differences. Hence, we utilize a statistical moment-matching approach involving two strategies- hard matching and soft matching, for distribution matching with different emphases to different timestamped snapshots. While hard matching is for matching the distributions of the closely time-stamped snapshots, soft matching effectively accounts for the networks' entire evolving behavior.

Hard distribution matching: We use the central moment discrepancy (CMD) metric [26] to directly maximize the similarity between the distribution of snapshots G_s^t, G_τ^t, G_s^{t-1}, and G_t^{t-1} at time t. CMD matches higher-order moments such as variance, skewness, and kurtosis in contrast to the widely used Kullback-Leibler (KL) divergence [26], which only matches the mean. We formulate the hard regularisation term as:

$$\mathcal{L}_t^{hard} = d_{cmd}(Z_s^t, Z_\tau^t) + \sum_{g\in\{s,\tau\}} d_{cmd}(Z_g^t, Z_g^{t-1}) \tag{10}$$

where $d_{cmd} : [a,b]^n \times [a,b]^n \to \mathbb{R}^+$ is a distributional shift metric given as:

$$d_{\text{cmd}}\left(Z^t, Z^{t'}\right) = \frac{1}{b-a}\left\|\mathbf{E}\left(Z^t\right) - \mathbf{E}\left(Z^{t'}\right)\right\| + \sum_{k=2}^{\infty} \frac{1}{|b-a|^k}\left\|c_k\left(Z^t\right) - c_k\left(Z^{t'}\right)\right\| \tag{11}$$

where $\mathbf{E}(Z^t) = \frac{1}{|V^t|}\sum_{i\in V^t} h_i$ and $c_k(Z^t) = \mathbf{E}(Z^t - \mathbf{E}(Z^t))^k$ is the k-th order moment. We use moments up to the 5th order.

Soft distribution matching: We propose to minimize the distributional divergence of G^t with the previously timestamped snapshots $\{G^{t-1}, G^{t-2}, .., G^0\}$ by softly learning the consistency between their distributions. The distribution similarity of any $G^{t'}$, $t' < t$, with G^t can be calculated by:

$$p_{g\in\{s,\tau\}}^{t'} = \frac{e^{\cos\left(\boldsymbol{z}_g^t, \boldsymbol{z}_g^{t'}\right)/T}}{\sum\limits_{\hat{t}\in\{t', t'-1\ldots,0\}} e^{\cos\left(\boldsymbol{z}_g^t, \boldsymbol{z}_g^{\hat{t}}\right)/T}} \tag{12}$$

where $\boldsymbol{z}$ is the graph representation (Equation 2), $cos(\cdot,\cdot)$ is the cosine similarity of vectors and T is a temperature parameter. At any time t, we can represent the distribution of all $G^{t'}, t' < t$ using Equation 12 as

$$\boldsymbol{P}_{g\in\{s,\tau\}} = \left[p_g^{t-1}, p_g^{t-2}\cdots, p_g^0\right] \tag{13}$$

Here $\boldsymbol{P}_s$ denotes the distribution of the source network snapshots and $\boldsymbol{P}_\tau$ of the target network. Now to keep the similarity structure of the dynamic source and target network consistent, we formulate the soft regularisation term as follows:

$$L_t^{soft} = d_{KL}(\boldsymbol{P}_s \| \boldsymbol{P}_\tau) + d_{KL}(\boldsymbol{P}_\tau \| \boldsymbol{P}_s) \tag{14}$$

We here use d_{KL}, the KL divergence, instead of the CMD metric for matching distributions in a soft manner.

Finally, by combining the above two regularization terms, we obtain the goal of distribution alignment as:

$$L_t^{align} = L_t^{anchor} + \alpha L_t^{hard} + \beta L_t^{soft} \tag{15}$$

where α and β are hyper-parameters for weighting the regularization terms.

4.3 Optimization and Alignment

We combine the two loss functions for continual and aligned representations to obtain the objective function at each time-step t as:

$$\mathcal{L} = \mathcal{L}_s^{CL} + \mathcal{L}_\tau^{CL} + \mathcal{L}^{align} \tag{16}$$

We optimize the above objective jointly over both networks in a unified framework using the stochastic gradient descent optimizer. After the model converges, we derive the alignment matrix A_t^* with the generated embeddings as:

$$A_t^* = \sigma\left((Z_s^t)^T \cdot Z_\tau^t\right) \tag{17}$$

The $(u, v)^{th}$ entry in A_t^* signifies the similarity score between the corresponding nodes. Hence the highest value corresponding to a node u in the row A_t^* most likely represents the same node in the other network.

5 Experimental Details

Datasets: We evaluate the effectiveness of RoCoNA on four real datasets from different fields.

- Social Networks: We use Twitter and Foursquare, two famous social networks connecting users online [12]. We obtain a sequence of snapshots for both networks based on the friending time information of users [19].
- Academic Networks: We take two pairs of academic networks, DBLP-AMiner, and MAG-AMiner. Each is a co-authorship network where author nodes are connected based on their collaborations [20]. We obtain a sequence of snapshots via the first publication time of the collaboration among authors [19].

Baselines: We compare RoCoNA with the existing dynamic social network alignment methods, including DNA [18], DGA [19], and HDyNA [5]. We also extend the formulation of CTSA [11], a GNN-based alignment method that originally aligns snapshots of the same network, to align snapshots of different dynamic networks. Moreover, to demonstrate the significance of dynamics in aligning networks, we additionally compare RoCoNA with some recent state-of-the-art static alignment methods, including IONE [12], CENALP [3], GAlign [21], BRIGHT [24], HackGAN [25] and HCNA [16].

Datasets	Metric	*Static Methods*						*Dynamic methods*				*Proposed method*
		IONE	CENALP	GAlign	BRIGHT	HackGAN	HCNA	HDyNA	CTSA	DNA	DGA	**RoCoNA**
TF	Acc@1	11.26	12.32	13.45	15.06	13.01	17.14	16.77	14.98	31.04	31.17	**37.10**
	Acc@10	19.88	20.10	22.67	23.88	21.98	24.62	21.76	22.86	36.64	38.79	**44.65**
	MAP	14.34	14.64	17.11	19.98	15.00	19.12	19.00	18.89	32.37	34.82	**40.30**
TF+	Acc@1	10.06	10.86	12.35	13.45	11.10	15.34	14.39	16.42	30.12	33.58	**36.70**
	Acc@10	18.58	20.14	22.45	22.65	20.76	22.54	20.65	24.67	36.39	39.58	**42.76**
	MAP	15.03	14.73	14.32	15.35	17.92	18.89	17.03	19.41	31.85	34.93	**38.85**
DA	Acc@1	11.28	10.97	11.34	12.04	10.67	14.95	18.76	23.44	32.63	38.88	**40.04**
	Acc@10	20.65	19.84	19.89	20.06	18.96	22.48	22.38	31.69	38.73	42.27	**46.83**
	MAP	12.86	10.47	13.39	14.11	12.03	17.25	19.63	25.48	33.13	39.05	**44.55**
MA	Acc@1	10.96	11.58	13.48	13.01	12.10	15.79	17.84	20.82	32.21	37.17	**41.28**
	Acc@10	19.76	21.44	23.58	22.88	21.47	25.67	25.45	27.57	39.65	43.24	**47.89**
	MAP	12.69	13.10	14.75	14.22	14.15	18.16	18.01	23.36	34.28	39.75	**43.34**

Table 1: Comparison of network alignment methods on real-world datasets

Evaluation Metrics: Similar to several high-quality recent works [21, 24, 25], we compare the approaches using *Acc@q*, which indicates if a node's true anchor match is present in a list of top-q potential anchors. It is given as

$$Acc@q = \frac{\sum_{u_s^* \in V_s} \mathbb{1}_{P^*[u_s^*, u_t^*] \in R(u_s^*)}}{\#\{\ \textit{ground truth anchor links}\ \}} \tag{18}$$

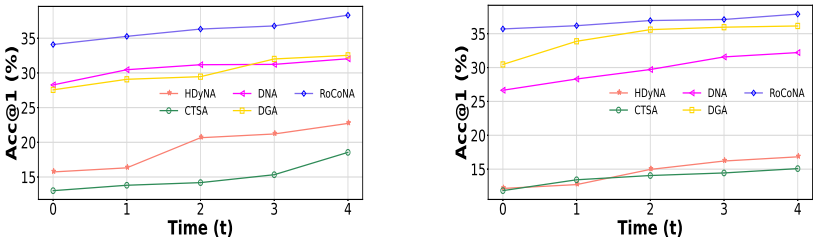
where $(u_s^*, u_t^*) \in A$ and $R(u_s)$ is a list of highest q values in the row $P^*(u_s)$. Apart from this, we also observe the *Mean Average Precision (MAP)* scores [21,25] for ranking perspective.

Implementation Details: We tune the hyper-parameters using the grid search algorithm implemented with Hyperopt. We use a GCN encoder with two hidden layers, $L = 2$, each layer of size 64. We take $\gamma = 0.005$, the size of memory $\mathcal{M}$ as 300 and search $p_{\mathcal{M}}^t$ in the range $\{0.1, 1\}$. The negative sampling size for each anchor in the training set is $K = 10$, and the temperature parameter is $T = 0.5$. We set $\lambda = (40, 30, 20, 20)$, $\alpha = (2, 3, 4, 4)$ and $\beta = (3, 3, 5, 6)$ for TF, TF+, DA and MA datasets respectively. We set the number of training epochs to 200, the embedding dimension to 128, and the initial learning rate to 0.005 for all datasets. For calculating the PageRank using [15], we set the random jump probability $\mu = 0.85$. We use the default values of hyper-parameters for the baselines except for the embedding dimensions for a fair comparison.

6 Results and Analysis

6.1 Comparison with baselines

We compare RoCoNA to static and dynamic baselines and report the results for *Acc*@1, *Acc*@10, and MAP in Table 1. We leverage five snapshots separated by 80 days, perform alignment at each time interval and then report the averaged result. For comparison with static methods, we follow [19] to generate a ϵ-overlap dataset by deleting users randomly to assess the impact of overlap on entity alignment. We align datasets with 25%, 30%, 35%, 40%, and 45% overlap and report their average. Figure 1 shows the performance of the best-performing



(a) TF Dataset (b) TF+ Dataset

(c) DA Dataset (d) MA Dataset

Fig. 1: Acc@1 of the best-performing baselines at each time step.

methods using 30% training data at each interval (or at each ϵ-overlap for static methods). RoCoNA performs best, improving the alignment accuracy, Acc@1 by a minimum margin of 19.08%, 9.28%, 3%, and 11% on the TF, TF+, DA, and MA datasets, respectively. We also observe a minimum improvement of 15.12%, 8.06%, 10.8%, and 10.7% in Acc@10 and 15.7%, 11.2%, 14.1%, and 9% in MAP scores. The results validate the effectiveness of continual learning and distribution matching for superior network alignment.

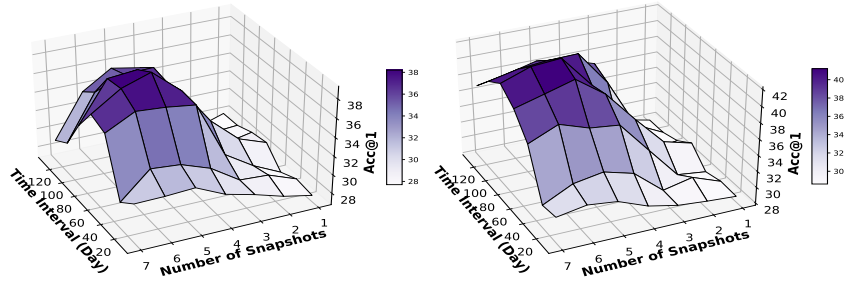
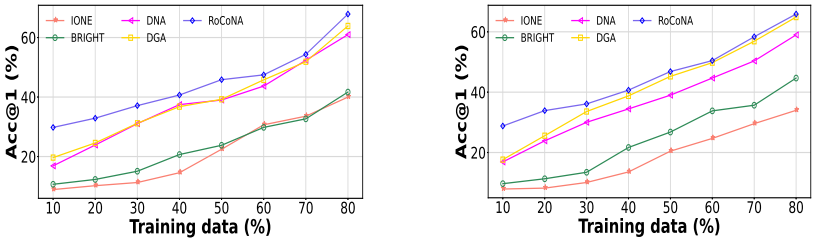


Fig. 2: Snapshotting settings for DA dataset using DGA(left) and RoCoNA(left).



(a) TF Dataset (b) TF+ Dataset

(c) DA Dataset (d) MA Dataset

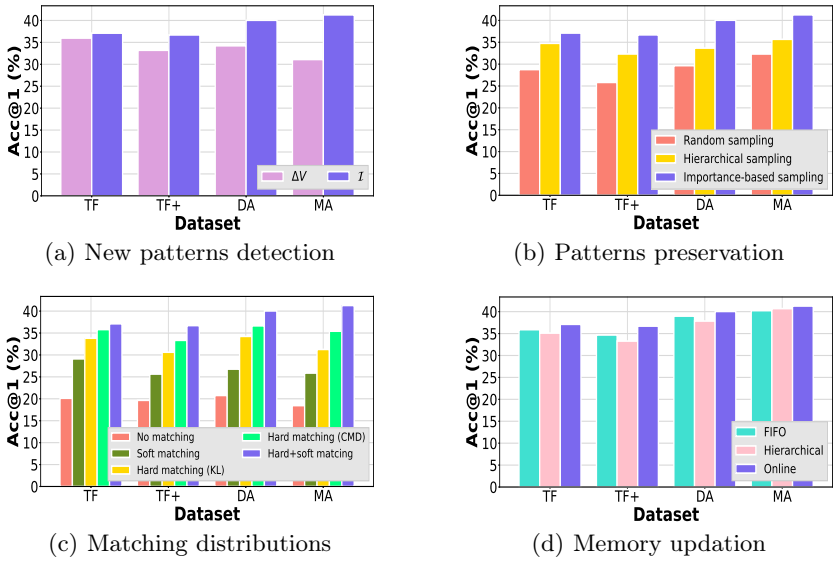
Fig. 3: Alignment results of the supervised baselines under different training rates

6.2 Analysis on the Snapshot Settings

We study the performance of RoCoNA by varying the frequency and number of snapshots and compare it with the best-performing baseline DGA. We report results for the DA dataset in Figure 2 and observe consistent results for other datasets. We observe that the *Acc*@1 score increases with the number of snapshots and time interval, reaches a maximum for around 5 snapshots with a time interval of 90 days, and then gradually goes down. It shows that only an adequate amount of information is needed to capture the network dynamicity, historical information does not add to its performance, and larger time intervals introduce distribution shifts. DGA, compared to RoCoNA, experiences a drastic performance drop when the time interval goes beyond 85 days, implying that RoCoNA can better handle distribution shifts over time.

6.3 Analysis on the Supervised Information

We observe the performance of RoCoNA and the baselines by varying the amount of supervised anchor information utilized for training. We report the results for all the datasets in Figure 3. We see that the Acc@1 score of all the methods increases with the increase in training data. RoCoNA outperforms all the methods even at very low training rates, demonstrating that it is effective in capturing network patterns and evolution behavior in the presence of some pre-known correspondences.



(a) New patterns detection (b) Patterns preservation

(c) Matching distributions (d) Memory updation

Fig. 4: Analysis of Model design

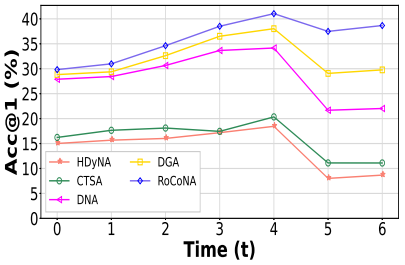
6.4 Analysis of Model Design

We analyze each proposed component accounting for the effectiveness of RoCoNA in Figure 4.

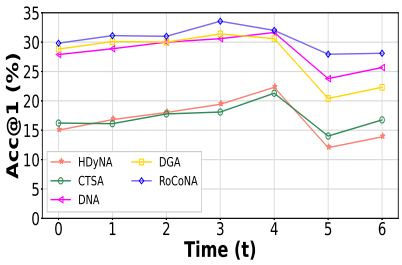
New patterns detection We compare the proposed PageRank difference-based technique, which treats $\mathcal{I}$ as in Equation 3 as new patterns, with the straightforward strategy, which treats ΔV as new patterns. Our proposed method outperforms by $3.13-32.74\%$ across all datasets, showing the importance of mining nodes other than ΔV undergoing structural changes during network evolution.

Existing patterns preservation We compare different sampling strategies for preserving existing patterns in memory. Our suggested sampling technique is importance-based sampling. Random sampling samples nodes with uniform probabilities and rank-based sampling samples the top PageRanked nodes. Our importance-based sampling yields better results, proving that it can effectively consolidate historical information compared to other methods.

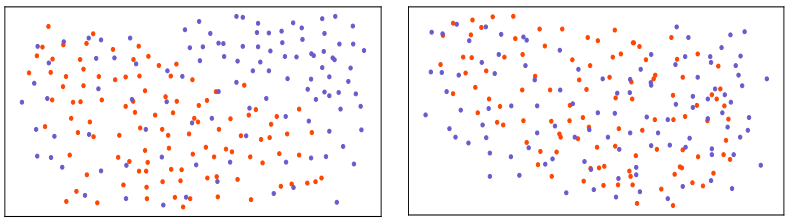
Matching distributions We analyze the effectiveness of our proposed distribution matching strategy. We compare it with no matching, which does not perform any matching, only soft matching (Equation 14), only hard matching(CMD) (Equation 10), and only hard matching(KL) that uses KL-divergence



(a) Results at each time step



(b) Results of nodes in V^0



(c) t-SNE plots of DBLP (red) and AMiner (blue) networks at time t_5 by DGA (left) and RoCoNA (right), respectively.

Fig. 5: Case Study

instead of the CMD metric. From the results, we see that matching distributions are necessary for dealing with dynamic networks. While matching higher-order moments using the CMD metric in place of KL divergence enhances the alignment accuracy, jointly performing hard and soft matching yields the best result.

Memory updation We compare the proposed online memory updation strategy in Equation 5 with the simple 'First In First Out' (FIFO), which discards the earliest sampled nodes when the memory gets full, and the hierarchical strategy, which discards the lowest PageRanked nodes. Results show that online memory updation outperforms all other methods across all datasets.

6.5 Case Study

We conduct experiments on the DA dataset to prove the effectiveness of our model in dealing with catastrophic forgetting and distribution shifts. We take the first five snapshots ($t_0 - t_4$) with a time interval of 90 days, and then at time t_5, we consider a more significant gap of 200 days. Such a split naturally introduces a distribution shift since several latent influential factors, like the popularity of research topics and author collaborations for data generation, would change over time. We then perform case studies to answer the following questions:

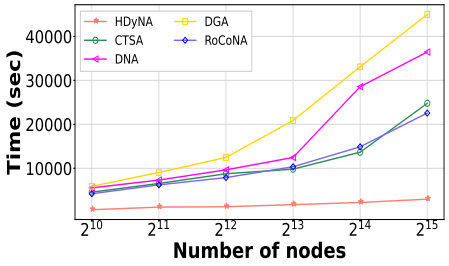


Fig. 6: Computation analysis

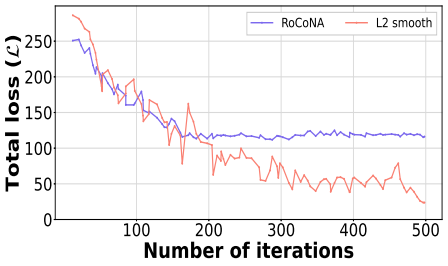


Fig. 7: Convergence analysis

(a) *How well is the distribution shift handled?* Figure 5(a) shows the performance of all the dynamic alignment methods in terms of *Acc*@1. We notice that the *Acc*@1 scores of all methods drop significantly at time t_5 as the distribution of the network structure changes. However, the performance of RocoNA remains comparably stable. Figure 5(c) shows the t-SNE visualization of the representations of the anchor nodes (100 anchors randomly extracted) for RoCoNA and the best-performing baseline DGA. We clearly see that as the network patterns change at time t_5, the distribution discrepancy between the source and target networks significantly increases for DGA. At the same time, RoCoNA still achieves a strong correspondence between the two aligned data distributions, making it robust to distribution shifts over time.
(b) *How well is catastrophic forgetting prevented?* Figure 5(b) shows the alignment performance of nodes at time t_0 using the updated GNN model as time evolves. We observe that at the $t_0 - t_2$ time steps, the alignment accuracy remains stable since the distribution of patterns is roughly unchanged. However, as the time interval increases, the GNN gets trained on new nodes and patterns, making the current model no longer applicable to the nodes at t_0. As a result of catastrophic forgetting, the alignment accuracy of all methods rapidly declines. However, RoCoNA shows stable results demonstrating its better generalization capabilities with effective historical knowledge consolidation.

6.6 Computational and Convergence Analysis

We study and compare the execution time and scalability of the network alignment methods in Figure 6. We follow [14] to use a network-generative model to increase the network size gradually. We observe that RoCoNA converges in a relatively shorter time. HDyNA takes the least time, but its alignment accuracy is not comparable to RoCoNA. We also study the convergence of RoCoNA to total loss ($\mathcal{L}$) on the DA dataset and compare it with the L2 regularised variant, i.e., $\lambda = 0$. From Figure 7, we see that our regularisation stabilizes better after a fixed number of iterations, indicating the model is convergent towards a similar alignment matrix, preserving historical knowledge constrained by λ.

7 Conclusion

In this paper, we study the dynamic network alignment problem. We focus on addressing the challenges associated when dealing with real dynamic networks- (1) catastrophic forgetting, (2) distribution shift, and (3) domain differences. We bridge them by proposing a continual learning framework, RoCoNA, built over streaming GNNs embodying regularizers to capture networks' evolving behavior. Extensive empirical evaluations show that RoCoNA effectively handles time-induced distribution shifts and exhibits better generalization capabilities by addressing the catastrophic forgetting problem. RoCoNA consistently outperforms the state-of-the-art methods by an up to 19% improvement in Acc@1 across several real datasets. Although we evaluate RoCoNA on limited datasets, its applicability can be extended to other real-world networks, including transactions and biological networks.

References

1. Aparício, D., Ribeiro, P., Milenković, T., Silva, F.: Temporal network alignment via got-wave. Bioinformatics **35**(18), 3527–3529 (2019)
2. Bayati, M., Gleich, D.F., Saberi, A., Wang, Y.: Message-passing algorithms for sparse network alignment. ACM Transactions on Knowledge Discovery from Data (TKDD) **7**(1), 1–31 (2013)
3. Du, X., Yan, J., Zha, H.: Joint link prediction and network alignment via cross-graph embedding. In: IJCAI. pp. 2251–2257 (2019)
4. Gao, S., Zhang, Z., Su, S.: Dawn: Domain generalization based network alignment. IEEE Transactions on Big Data (2022)
5. He, J., Liu, L., Yan, Z., Wang, Z., Xiao, M., Zhang, Y.: User alignment across dynamic social networks based on heuristic algorithm. In: 2021 7th International Conference on Systems and Informatics (ICSAI). pp. 1–7. IEEE (2021)
6. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: Representation learning-based graph alignment. In: Proceedings of the 27th ACM international conference on information and knowledge management. pp. 117–126 (2018)
7. Hong, H., Li, X., Pan, Y., Tsang, I.: Domain-adversarial network alignment. IEEE Transactions on Knowledge and Data Engineering (2020)
8. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems **33**, 22118–22133 (2020)
9. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
10. Koutra, D., Tong, H., Lubensky, D.: Big-align: Fast bipartite graph alignment. In: 2013 IEEE 13th international conference on data mining. IEEE (2013)
11. Liang, S., Tang, S., Meng, Z., Zhang, Q.: Cross-temporal snapshot alignment for dynamic networks. IEEE Transactions on Knowledge and Data Engineering (2021)
12. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: Ijcai. pp. 1774–1780 (2016)

13. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: Ijcai. vol. 16, pp. 1823–1829 (2016)
14. Nguyen, T.T., Pham, M.T., Nguyen, T.T., Huynh, T.T., Nguyen, Q.V.H., Quan, T.T., et al.: Structural representation learning for network alignment with self-supervised anchor links. Expert Systems with Applications **165**, 113857 (2021)
15. Ohsaka, N., Maehara, T., Kawarabayashi, K.i.: Efficient pagerank tracking in evolving networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 875–884 (2015)
16. Saxena, S., Chakraborty, R., Chandra, J.: Hcna: Hyperbolic contrastive learning framework for self-supervised network alignment. Information Processing & Management **59**(5), 103021 (2022)
17. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences **105**(35), 12763–12768 (2008)
18. Sun, L., Zhang, Z., Ji, P., Wen, J., Su, S., Philip, S.Y.: Dna: Dynamic social network alignment. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 1224–1231. IEEE (2019)
19. Sun, L., Zhang, Z., Wen, J., Wang, F., Ji, P., Su, S., Yu, P.: Aligning dynamic social networks: An optimization over dynamic graph autoencoder. IEEE Transactions on Knowledge and Data Engineering (2022)
20. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 990–998 (2008)
21. Trung, H.T., Van Vinh, T., Tam, N.T., Yin, H., Weidlich, M., Hung, N.Q.V.: Adaptive network alignment with unsupervised and multi-order convolutional networks. In: 2020 ICDE. pp. 85–96. IEEE (2020)
22. Vijayan, V., Critchlow, D., Milenković, T.: Alignment of dynamic networks. Bioinformatics **33**(14), i180–i189 (2017)
23. Xiong, H., Yan, J., Pan, L.: Contrastive multi-view multiplex network embedding with applications to robust network alignment. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (2021)
24. Yan, Y., Zhang, S., Tong, H.: Bright: A bridging algorithm for network alignment. In: Proceedings of the Web Conference 2021. pp. 3907–3917 (2021)
25. Yang, L., Wang, X., Zhang, J., Yang, J., Xu, Y., Hou, J., Xin, K., Wang, F.Y.: Hackgan: Harmonious cross-network mapping using cyclegan with wasserstein-procrustes learning for unsupervised network alignment. IEEE Transactions on Computational Social Systems (2022)
26. Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., Saminger-Platz, S.: Central moment discrepancy (cmd) for domain-invariant representation learning. arXiv preprint arXiv:1702.08811 (2017)
27. Zhang, S., Tong, H.: Final: Fast attributed network alignment. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1345–1354 (2016)
28. Zhou, Y., Ren, J., Jin, R., Zhang, Z., Zheng, J., Jiang, Z., Yan, D., Dou, D.: Unsupervised adversarial network alignment with reinforcement learning. ACM Transactions on Knowledge Discovery from Data (TKDD) **16**(3), 1–29 (2021)