PLANETALIGN: A COMPREHENSIVE PYTHON LIBRARY FOR BENCHMARKING NETWORK ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Network alignment (NA) aims to identify node correspondence across different networks and serves as a critical cornerstone behind various downstream multi-network learning tasks. Despite growing research in NA, there lacks a comprehensive library that facilitates the systematic development and benchmarking of NA methods. In this work, we introduce PLANETALIGN, a comprehensive Python library for network alignment that features a rich collection of built-in datasets, methods, and evaluation pipelines with easy-to-use APIs. Specifically, PLANETALIGN integrates 18 datasets and 14 NA methods with extensible APIs for easy use and development of NA methods. Our standardized evaluation pipeline encompasses a wide range of metrics, enabling a systematic assessment of the effectiveness, scalability, and robustness of NA methods. Through extensive comparative studies, we reveal practical insights into the strengths and limitations of existing NA methods. We hope that PLANETALIGN can foster a deeper understanding of the NA problem and facilitate the development and benchmarking of more effective, scalable, and robust methods in the future. The source code of PLANETALIGN is available at https://anonymous.4open.science/r/PlanetAlign-E9BA

1 INTRODUCTION

Multi-sourced and multi-layer networks are becoming ubiquitous across a wide range of domains in the era of big data and AI, ranging from social network analysis (Shao et al., 2023; Rácz & Zhang, 2024; Peng et al., 2025), anti-money laundering (Zhang et al., 2021), bio-informatics (Hu et al., 2024; Zare Mirak-Abad & Ghorbanali, 2025), to knowledge graph fusion (Yan et al., 2021a; Chen et al., 2024). Identifying the same node across different networks, i.e., network alignment (NA), enables joint learning across multiple networks and serves as the key cornerstone of multi-network tasks. For example, aligning users across online social networks improve personalized services, e.g., cross-domain recommendation (Liu et al., 2023a; Zeng et al., 2023; Yu et al., 2025). In transaction networks, aligning suspicious accounts from different transaction networks facilitates the detection of fraudulent activity (Zhang et al., 2019b; Du et al., 2021; Yan et al., 2024). In protein interaction networks, alignment of proteins across different species uncovers hidden biological homologies (Clark & Kalita, 2014; Hu et al., 2024). In knowledge graphs (KG), merging incomplete KGs based on aligned entities helps construct more unified knowledge bases (Yan et al., 2021a; Liu et al., 2023b; Chen et al., 2024).

Despite growing interest in NA, there lacks a comprehensive benchmark to provide standardized evaluation of NA methods on different datasets from various aspects. The absence of such benchmarks leaves the genuine performance and usefulness of existing NA methods an open research question, hindering the standardization of research in the NA community.

Although prior efforts, which are summarized in Table 1, have been made in benchmarking NA methods (Clark & Kalita, 2014; Cao & Yu, 2016; Sun et al., 2020; Trung et al., 2020; Döpmann, 2013), they suffer from at least one of the following limitations: (1) *limited datasets* within a single domain, e.g. biological networks (Clark & Kalita, 2014) or social networks (Cao & Yu, 2016); (2) *limited methods* exclusively focusing on a single category, e.g., consistency-based methods (Döpmann, 2013) or embedding-based methods (Sun et al., 2020), while ignoring the most recent line of works, e.g., optimal transport (OT) based methods; (3) *limited and inconsistent evaluation* from a single aspect, e.g. effectiveness (Clark & Kalita, 2014; Cao & Yu, 2016), without standardized dataset splits and evaluation metrics (Clark & Kalita, 2014; Cao & Yu, 2016; Sun et al., 2020; Trung et al., 2020; Döpmann, 2013).

In response to these limitations, we introduce PLANETALIGN, an open-source PyTorch-based library designed for unified evaluation and streamlined development of NA methods, which features the following key design. Firstly, PLANETALIGN includes 18 different public datasets spanning 6 different domains which can be directly downloaded through a simple API call, including social networks (Zhang & Philip, 2015; Zhang & Tong, 2016), publication networks (Tang et al., 2008; Yang et al., 2016; Leskovec et al., 2007), biological networks (Stark et al., 2006; De Domenico et al., 2015b; Zitnik & Leskovec, 2017; Park et al., 2010), knowledge graphs (Sun et al., 2017), infrastructure networks (Yan et al., 2022; Zhu et al., 2021; Song et al., 2020), and communication networks (Zhang et al., 2017; Kunegis, 2013), covering both real-world and synthetic scenarios (*Limitation #1*). The wide range of datasets built into PLANETALIGN allows comprehensive evaluation of NA methods on different types of networks, e.g., plain and attributed networks, fostering in-depth understanding of the applicability of NA methods to different domains. Secondly, PLANETALIGN features efficient implementations of 14 different NA methods including consistency-based (Singh et al., 2008; Zhang & Tong, 2016), embedding-based (Liu et al., 2016; Heimann et al., 2018; Chu et al., 2019; Zhang et al., 2020; Gao et al., 2021; Yan et al., 2021b; Zhang et al., 2021; Liu et al., 2023a), and OT-based methods (Zeng et al., 2023; Tang et al., 2023; Zeng et al., 2024; Yu et al., 2025), covering traditional and state-of-the-art baselines (*Limitation #2*). With easy-to-use APIs, PLANETALIGN allows streamlined comparison between NA methods across diverse benchmark settings. Thirdly, PLANETALIGN highlights a comprehensive list of evaluation metrics and benchmarking tools (*Limitation #3*). For evaluation metrics, we include the most classical effectiveness metrics, Hits@K and MRR, under different pairwise alignment settings. We also include time and memory overheads for evaluating the efficiency and scalability of NA methods. For benchmarking tools, we enforce consistent dataset split through a unified API design to ensure reproducibility. PLANETALIGN also provides a rich collection of APIs and utility functions that allows fair and reproducible benchmarking across key dimensions of NA performance. Finally, PLANETALIGN implements extensible APIs and efficient utility functions which allow users to streamline the implementation of customized NA methods and the integration of customized datasets with minimal efforts. Specifically, our API design allows customized datasets and NA methods to be built upon carefully designed base classes and integrated into PLANETALIGN's pipeline with only a few lines of code. PLANETALIGN further provides commonly used utility functions such as random walk with restart (RWR) embedding, anchor-based embedding, etc. Empowered by the aforementioned features, PLANETALIGN addresses the limitations of existing NA benchmarks comprehensively.

Based on PLANETALIGN, we conduct comprehensive experiments to evaluate the effectiveness, scalability, robustness, and sensitivity to supervision of 14 built-in NA methods across 18 built-in datasets, revealing practical insights into the strength and limitations of existing NA methods. We also compare PLANETALIGN's implementation of NA algorithms with their official implementation which shows that our implementation can achieve up to 3 times speed-up while maintaining similar effectiveness performance, demonstrating the superiority of PLANETALIGN.

In summary, we introduce a unified, comprehensive, and efficient library PLANETALIGN featuring a wide range of built-in datasets and NA methods, as well as extensible and easy-to-use utility functions and APIs, facilitating the benchmarking and development of NA methods. We will continuously update PLANETALIGN upon release of new benchmark datasets and methods.

2 PROBLEM DEFINITION

An illustration of NA problems are shown in Figure 1. Given two input networks $\mathcal{G}_1 = \{\mathcal{V}_1, \mathbf{A}_1, \mathbf{X}_1, \mathbf{E}_1\}$, $\mathcal{G}_2 = \{\mathcal{V}_2, \mathbf{A}_2, \mathbf{X}_2, \mathbf{E}_2\}$ and a set of anchor node pairs $\mathcal{L} = \{(x, y)|x \in \mathcal{V}_1, y \in \mathcal{V}_2\}$ indicating pre-alignment, where $\mathcal{V}_1, \mathcal{V}_2$ denote the node sets, $\mathbf{A}_1, \mathbf{A}_2$ denote the graph [1] adjacency matrices, $\mathbf{X}_1, \mathbf{X}_2$ denote the node attribute matrices, and $\mathbf{E}_1, \mathbf{E}_2$ denote the edge attribute matrices, the *semi-supervised attributed network alignment* task aims to discover node-level correspondence across two networks inferred from an output alignment matrix $\mathbf{S}$, where $\mathbf{S}(x, y)$ indicates the likelihood of alignment between node $x \in \mathcal{V}_1$ and node $y \in \mathcal{V}_2$. If neither

[1]In this work, the terms 'network' and 'graph' are used interchangeably.

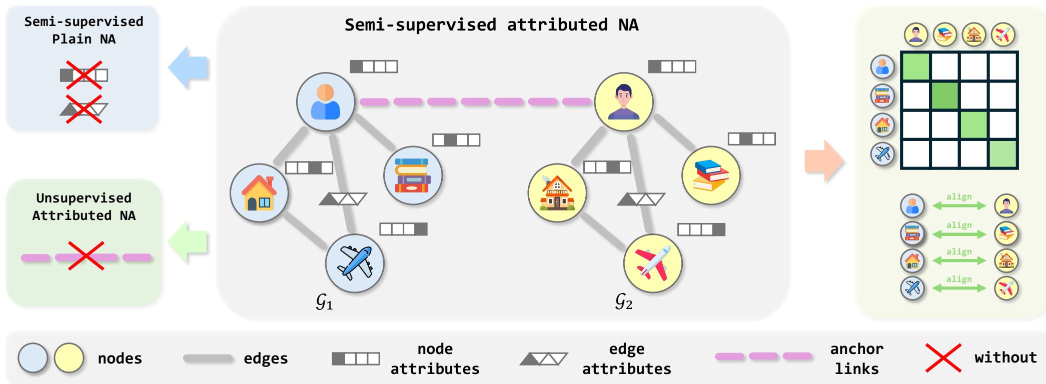

Figure 1: An illustration of NA problems.

node attributes $\mathbf{X}_1, \mathbf{X}_2$ nor edge attributes $\mathbf{E}_1, \mathbf{E}_2$ are available, this becomes the *semi-supervised plain network alignment* task; If no anchor node pairs are available, i.e., $|\mathcal{L}| = 0$, this becomes the *unsupervised attributed network alignment* task.

3 Related Work

3.1 Network Alignment Methods

Existing NA methods can be classified into three categories: consistency-based, embedding-based, and OT-based approaches (Zhang & Tong, 2020). Consistency-based methods are among the earliest approaches, formulated as optimization problems which assumes structural and/or attribute consistency between node neighborhoods across networks (Singh et al., 2008; Zhang & Tong, 2016; Zhang et al., 2019a; Bayati et al., 2009). Although recent works on NA have largely moved beyond consistency principles, consistency-based methods remain important baselines for benchmarking purpose.

Embedding-based and OT-based methods represent more recent advances in the NA community. For embedding-based methods, nodes are mapped into a shared low-dimensional space and aligned based on embedding similarity (Liu et al., 2016; Heimann et al., 2018; Chu et al., 2019; Zhang et al., 2020; Gao et al., 2021; Yan et al., 2021b; Zhang et al., 2021; Liu et al., 2023a). By leveraging advances in deep representation learning, embedding-based methods have shown strong performance and remain an active research direction. For OT-based methods, they formulate the NA problem as an optimization problem minimizing the total effort of transporting the node distribution of one graph to another under a set of pre-defined or learnable cost functions (Tang et al., 2023; Zeng et al., 2023; 2024; Yu et al., 2025). The most OT-based methods consistently achieve SOTA performance, making them a promising direction for future research. PLANETALIGN includes representative state-of-the-art methods from all three kinds of methods, providing a comprehensive benchmarking library.

3.2 Network Alignment Libraries

There are five existing benchmarks/libraries for NA, and we include a comprehensive comparison on the inclusion of datasets, NA methods, and evaluation dimensions in Table 1. Specifically, SGAPBSA (Döpmann, 2013) and CAPABN (Clark & Kalita, 2014) mainly focus on benchmarking traditional consistency-based NA methods on biological networks. ASNets (Cao & Yu, 2016) benchmarks the effectiveness of both consistency-based and embedding-based methods on social networks, leaving the scalability and robustness of NA methods an open research question. NAB (Trung et al., 2020) comprehensively evaluates the effectiveness, scalability, and robustness of both consistency-based and embedding-based methods. However, NAB only includes social networks, lacking comprehensive datasets on other domains where NA is also an important research problem. OpenEA (Sun et al., 2020) focuses on benchmarking embedding-based methods on knowledge graphs, ignoring networks in other domains. In addition, none of the existing NA libraries includes *OT-based* methods which have emerged as the most recent and effective line of work in the NA community.

Table 1: Comparison with existing NA benchmarks/libraries (Clark & Kalita, 2014; Cao & Yu, 2016; Sun et al., 2020; Trung et al., 2020; Döpmann, 2013). We denote whether a specific type of networks, methods, and evaluations is included in the benchmark/library.

Benchmark/Library		SGAPBSA	CAPABN	ASNets	NAB	OpenEA	PLANETALIGN (ours)
Networks	Social	✗	✗	✓	✓	✗	✓
	Communication	✗	✗	✗	✗	✗	✓
	Publication	✗	✗	✗	✗	✗	✓
	Biological	✓	✓	✗	✗	✗	✓
	Knowledge	✗	✗	✗	✗	✓	✓
	Infrastructure	✗	✗	✗	✗	✗	✓
Methods	Consistency-based	✓	✓	✓	✓	✗	✓
	Embedding-based	✗	✗	✓	✓	✓	✓
	OT-based	✗	✗	✗	✗	✗	✓
Evaluations	Effectiveness	✓	✓	✓	✓	✓	✓
	Scalability	✓	✗	✗	✓	✓	✓
	Robustness	✗	✗	✗	✓	✗	✓

4 DESIGN OF PLANETALIGN

In this section, we introduce the design features of PLANETALIGN, which includes comprehensive built-in datasets and NA methods (Section 4.1), unified and easy-to-use APIs (Section 4.2), as well as standardized and diverse benchmarking tools (Section 4.3).

4.1 COMPREHENSIVE DATASETS AND METHODS

PLANETALIGN collects and curates 18 NA datasets across 6 different domains, covering social networks, publication networks, biological networks, knowledge graphs, infrastructure networks, and communication networks. PLANETALIGN also implements 14 existing NA methods across all 3 categories, including consistency-based, embedding-based, and OT-based methods. An overview of built-in datasets and NA methods in PLANETALIGN is summarized in Figure 2.

Dataset Collection and Synthesis. We collect 11 real-world datasets from existing NA works and synthesize 7 additional datasets across 6 distinct domains. We follow the most classical method to synthesize NA datasets from a single network, where we insert 10% noisy edges into and delete 15% existing edges from the original network to create two permuted networks (Yang et al., 2016; Zhang et al., 2020; Yan et al., 2021b; Zhang et al., 2021; Zeng et al., 2023; Yu et al., 2025).

Specifically, for *social networks*, where NA is used to align the same user for personalized recommendation (Cao & Yu, 2016; Zhang & Philip, 2015; Liu et al., 2016), PLANETALIGN includes 4 real-world datasets: Foursquare-Twitter (Zhang & Philip, 2015), Douban (Zhang & Tong, 2016), Flickr-LastFM (Zhang & Tong, 2016), and Flickr-MySpace (Zhang & Tong, 2016); for *publication networks*, where NA is used for author disambiguation (Li et al., 2021), PLANETALIGN includes the most representative real-world dataset ACM-DBLP (Tang et al., 2008), and synthesizes 2 additional datasets from Cora (Yang et al., 2016) and ArXiv (Leskovec et al., 2007); for *biological networks*, where NA uncovers hidden biological homologies by aligning proteins of different species (Clark & Kalita, 2014; Faisal et al., 2015; Singh et al., 2008), PLANETALIGN includes 1 real-world dataset SacchCere (Stark et al., 2006; De Domenico et al., 2015b) and 2 synthetical datasets PPI (Zitnik & Leskovec, 2017) and GGI (Park et al., 2010). For *knowledge graphs*, where NA is used for knowledge fusion (Liu et al., 2023b; Chen et al., 2023; Sun et al., 2020), PLANETALIGN includes 3 variants of a real-world dataset DBP15K (Sun et al., 2017), namely DBP15K ZH-EN, JA-EN, and FR-EN. For *infrastructure networks*, where NA plays an important role in cross layer dependency inference (Yan et al., 2022), PLANETALIGN includes 1 real-world dataset Italy (Yan et al., 2022), and 2 synthetic datasets Airport (Zhu et al., 2021) and PeMS08 (Song et al., 2020). For *communication networks*, PLANETALIGN includes 1 real-world dataset Phone-Email (Zhang et al., 2017) and 1 synthetic dataset Arenas (Kunegis, 2013). Detailed dataset statistics can be found in Appendix A.

Baseline Implementations. We implement 14 existing NA methods based on a unified API, including 2 representative consistency-based methods, 8 embedding-based methods, and 4 OT-based

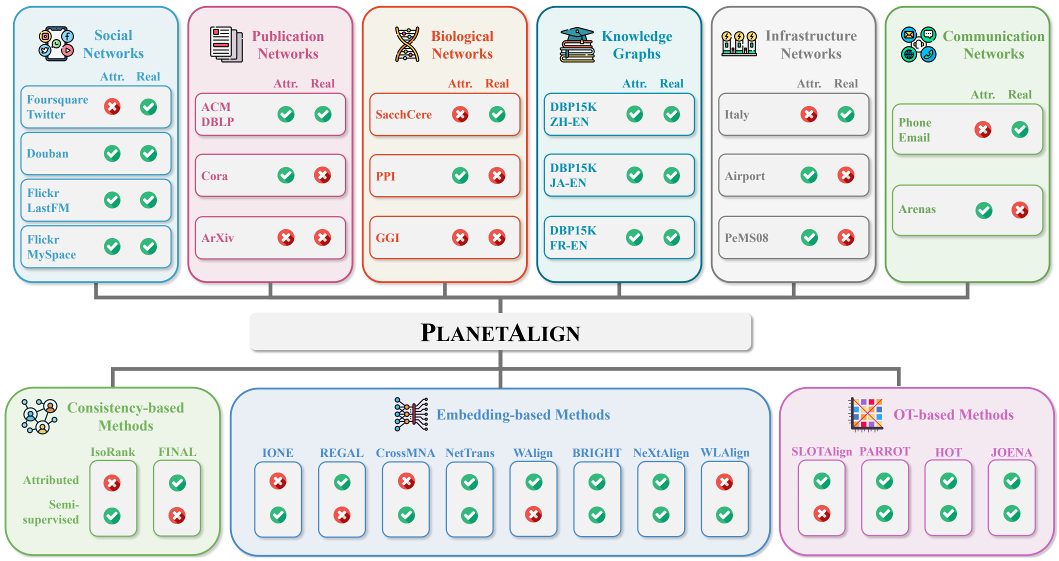

Figure 2: **An overview of built-in datasets and NA methods in PLANETALIGN.** For built-in *datasets*, we indicate if they consist of attributed (Attr.) or plain networks, and if they are real-world (Real) or synthetic datasets. For built-in *NA methods*, we indicate if they are designed for attributed or plain NA tasks, and if they are semi-supervised or unsupervised methods.

methods. Specifically, for *consistency-based* methods, PLANETALIGN includes IsoRank (Singh et al., 2008) and FINAL (Zhang & Tong, 2016); for *embedding-based* methods, PLANETALIGN includes IONE (Liu et al., 2016), REGAL (Heimann et al., 2018), CrossMNA (Chu et al., 2019), NetTrans (Zhang et al., 2020), WAlign (Gao et al., 2021), BRIGHT (Yan et al., 2021b), NeXtAlign (Zhang et al., 2021), and WLAlign (Liu et al., 2023a); for *OT-based* methods, PLANETALIGN includes SLOTAlign (Tang et al., 2023), PARROT (Zeng et al., 2023), HOT (Zeng et al., 2024), and JOENA (Yu et al., 2025). Detailed introductions of built-in NA methods can be found in Appendix B.

4.2 UNIFIED AND EASY-TO-USE APIS

PLANETALIGN is carefully designed to provide unified and easy-to-use APIs to streamline the implementation, training, and evaluation of NA algorithms on customizable datasets. An example usage of PLANETALIGN is shown in Figure 3. We also provide detailed documentation at `https://planetalign.netlify.app`, covering quick-start tutorials as well as in-depth documentations of API usage of the major components.

Specifically, to train and evaluate an NA algorithm on a specific dataset, the user of PLANETALIGN will first define a `PlanetAlign.data.BaseData` object and a `Model` object inherited from the base class `PlanetAlign.algorithm.BaseModel`. For built-in datasets, PLANETALIGN provides downloading options and reproducible train/test split with a customized training ratio; for built-in algorithms, PLANETALIGN provides hyperparameter options upon definition of the algorithm, and a unified API as PyTorch for GPU/CPU offloading. Both built-in dataset and algorithm objects can be defined neatly in a single line of code. PLANETALIGN also provides unified and intuitive base classes for defining customized datasets and algorithms, as shown in Figure 3.

Before training an NA algorithm, the user has an option to initialize a `PlanetAlign.Logger` object used to log the training process of the algorithm. Then, the user can simply call the `.train` method of the algorithm object with the dataset and logger object, IDs of graphs to be aligned, and additional configuration of training, e.g., training epochs, learning rate, etc., to start the training. Training outputs, including node embeddings, alignment matrix, and training performance are returned by the `.train` after the training process ends, providing fine-grained intermediate results of alignment that users can readily leverage for downstream tasks, such as cross-layer dependency

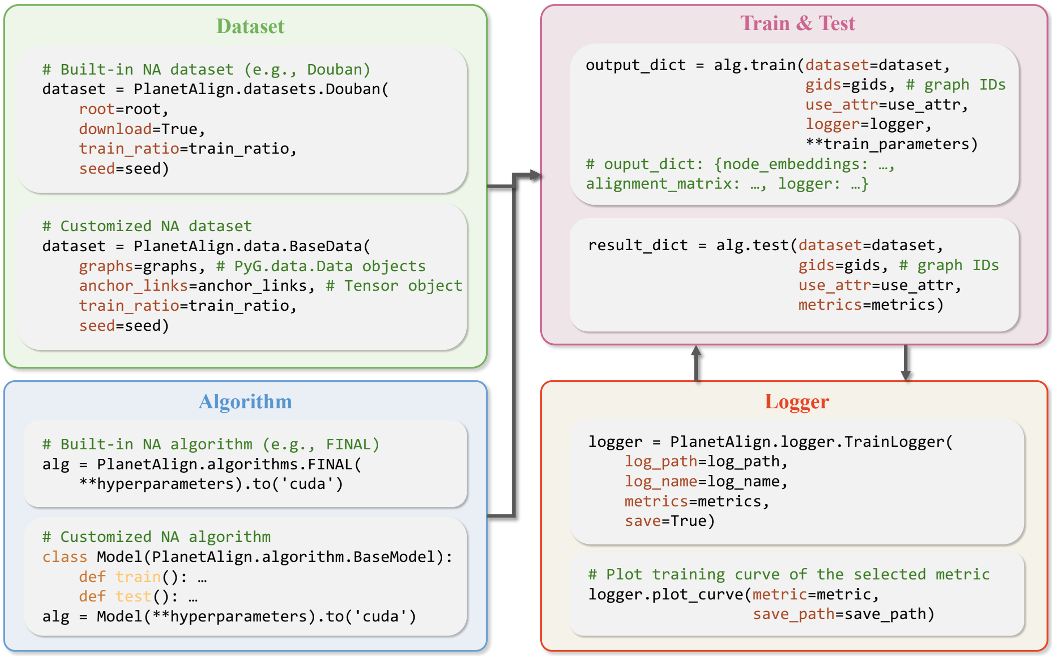

Figure 3: Example usage of PLANETALIGN for benchmarking NA. Users begin by initializing a dataset and algorithm objects, along with a logger for training-time monitoring and visualization. The training and evaluation can then be performed through simple API calls with user-defined parameters, providing substantial flexibility in controlling the training and evaluation workflows.

inference (Yan et al., 2022), knowledge integration (Yan et al., 2021a), and cross-KG modality fusion (Chen et al., 2023).

Finally, after the training process, the user can call the `.test` method of the algorithm object with customized options of evaluation metrics. The optional logger object, which records and logs comprehensive data during training, also provides a rich collection of APIs for visualizing the evolution of different metrics along training, e.g., training loss, time and memory usage, etc.

4.3 STANDARDIZED AND DIVERSE BENCHMARKING TOOLS

Standard Evaluation Metrics. PLANETALIGN provides low-level utility functions for computing standard and widely adopted evaluation metrics in the NA tasks with custom options for alignment directions, such as left-to-right for pairwise alignment scenarios where the nodes in $\mathcal{G}_1$ is aligned to $\mathcal{G}_2$, and vise versa. Specifically, PLANETALIGN includes the following metrics:

- **Hits@K**. In the case of aligning $\mathcal{G}_1$ to $\mathcal{G}_2$, Hits@K refers to the proportion of nodes in $\mathcal{G}_1$ whose correct alignment in $\mathcal{G}_2$ is ranked within the top-K candidates by a NA algorithm. Formally,
$$\text{Hits@K} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{\text{rank}_i \leq K\},$$
where N is the number of nodes in $\mathcal{G}_1$, rank_i is the rank of the correct alignment for the i-th node in $\mathcal{G}_1$, and $\mathbb{1}\{\cdot\}$ is the indicator function. Note that in NA, Precision@K (Trung et al., 2020) is equivalent to Hits@K.
- **Mean Reciprocal Rank (MRR)**. MRR refers to the average reciprocal of the rank at which the correct alignment appears in the candidate list. Formally, In the case of aligning $\mathcal{G}_1$ to $\mathcal{G}_2$,
$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\text{rank}_i},$$
where N is the number of nodes in $\mathcal{G}_1$, rank_i is the rank of the correct alignment for the i-th node in $\mathcal{G}_1$. Note that in NA, Mean Average Precision (MAP) (Trung et al., 2020) is equivalent to MRR.

Diverse Benchmark Settings. The design of PLANETALIGN enables diverse and reproducible benchmarking of existing NA algorithms with minimal efforts, providing a rich collection of APIs and built-in utility functions that allow users to easily configure, run, and evaluate experiments along key dimensions of NA performance, including effectiveness, scalability, and robustness.

Specifically, for *effectiveness*, PLANETALIGN supports custom training ratios and generates consistent, reproducible train/test splits by a user-defined random seed, ensuring fair comparisons of different NA algorithms. The APIs also supports experiments that evaluate the sensitivity of different NA methods with respect to the amount of supervision, providing valuable insights into their applicability to various supervision regimes. PLANETALIGN further provides unified utility functions to selectively introduce or remove supervision, enabling side-by-side comparisons between supervised and unsupervised algorithms under the same setting; for *scalability*, PLANETALIGN includes built-in logging functionalities that automatically track the runtime and memory usage during training and inference, allowing consistent and transparent evaluation of the efficiency of NA algorithms across datasets of varying sizes; for *robustness*, PLANETALIGN provides utility functions for injecting edge-level, attribute-level, and supervision noise into input graphs, allowing comprehensive evaluation of the robustness of NA methods under diverse graph noises or data inconsistencies.

5 EXPERIMENTS

Based on PLANETALIGN, we carry out extensive experiments to benchmark a wide range of NA algorithms across four key dimensions: effectiveness (Section 5.2), scalability (Section 5.3), robustness (Appendix D.1), and sensitivity to supervision (Appendix D.2). Additionally, we compare our implementation to the official implementation of built-in NA algorithms of PLANETALIGN to validate the correctness and efficiency our library (Appendix D.3).

5.1 EXPERIMENTAL SETUP

Datasets and methods. We benchmarks the performance of 14 NA algorithms on 18 NA datasets built into PLANETALIGN, as shown in Figure 2. Detailed dataset statistics and a brief introduction to each algorithm can be found at Appendix A and B, respectively.

Metrics. To evaluate effectiveness, we report Hits@K and MRR introduced in Section 4.3. All reported Hits@K and MRR are averaged results from both alignment directions. To evaluate scalability, we report the runtime and peak memory usage.

Additional Setup. For each experiments, we run 5 times and report the mean and standard deviation of the results. Additional experimental setup, such as the machine used to run the experiments and hyperparameter settings, are detailed in Appendix C.

5.2 EFFECTIVENESS RESULTS

We first evaluate the effectiveness of existing NA algorithms on plain networks under a semi-supervised setting with 20% training ratio. Datasets are randomly split for training and testing by a fixed random seed to ensure fair and reproducible comparison. Table 2 shows the averaged results on all 18 datasets group by their categories. Detailed results on plain and attributed NA datasets can be found in Appendix D.

We can see from Table 2 that OT-based methods, particularly PARROT (Zeng et al., 2023) and JOENA (Yu et al., 2025), consistently achieve SOTA alignment performance in Hits@K and MRR across all datasets, demonstrating the effectiveness of optimal transport in aligning distributional structures. Embedding-based methods such as IONE (Liu et al., 2016), NetTrans (Zhang et al., 2020), and BRIGHT (Yan et al., 2021b) can be effective in aligning some networks. However, their strong performance is not consistent across different datasets, potentially due to the space disparity issue (Yan et al., 2021b; Zhang et al., 2021). Consistency-based methods, while occasionally perform well on certain datasets, usually outperformed by best-performing embedding-based and OT-based methods, suggesting that relying solely on consistency principles may lead to sub-optimal alignment.

In addition to empirical observations, we further provide theoretical analysis into the superior performance of OT-based alignment methods. Compared with consistency-based methods, which are restricted by local consistency principles, OT-based methods go beyond local assumptions by solving

Table 2: Effectiveness and efficiency results of NA algorithms on plain networks with a training ratio of 20%. We group the 18 datasets in PLANETALIGN by their categories and report the averaged Hits@1, Hits@10, MRR (in %), inference time (Time), and peak memory usage (Memory). Cells that contain the 1st/2nd/3rd best results are highlighted in red/blue/green, respectively. Detailed results for each dataset can be found in Appendix D.5.

Dataset	Social			Publication			Biological			Knowledge			Infrastructure			Communication		
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**
IsoRank	4.2	19.6	9.2	18.9	59.1	31.4	21.6	45.1	29.4	11.5	50.0	23.4	14.2	43.9	24.1	22.1	44.5	30.2
FINAL	4.9	22.3	10.1	22.3	68.6	37.3	22.9	56.6	34.1	13.9	55.4	27.3	15.1	53.6	28.0	21.7	57.7	33.9
IONE	7.9	20.0	12.1	28.7	63.6	40.1	46.1	60.5	51.0	4.7	20.3	10.0	29.6	51.2	37.0	50.4	69.0	56.7
REGAL	0.3	2.2	1.2	1.8	7.8	3.9	1.0	5.3	2.6	0.8	2.9	1.6	2.8	13.4	6.7	45.3	49.5	47.2
CrossMNA	1.2	11.1	4.5	13.2	58.1	27.2	40.2	58.7	46.5	2.7	28.8	10.7	14.6	30.9	20.2	22.8	53.3	33.9
NetTrans	7.2	21.8	11.9	40.7	77.3	52.7	34.2	57.5	41.8	28.8	62.8	39.7	29.3	59.5	39.6	45.2	62.6	51.8
WAlign	4.2	8.5	6.1	31.2	49.6	37.8	20.3	27.3	22.9	19.4	28.2	22.8	29.1	47.3	35.5	49.6	56.0	52.1
BRIGHT	5.1	17.0	9.0	40.4	74.0	51.8	30.5	48.0	36.5	30.4	61.7	40.9	29.9	57.0	39.5	50.9	62.3	55.0
NeXtAlign	7.1	19.5	11.3	43.2	76.9	54.7	25.9	44.8	32.8	27.5	59.9	38.3	28.0	55.1	37.8	29.6	51.4	37.2
WLAlign	7.6	14.8	10.1	35.9	58.1	43.2	41.2	50.7	44.3	25.9	44.2	31.7	29.5	42.8	34.1	34.1	49.1	39.5
PARROT	12.6	26.3	17.2	66.6	88.6	74.4	61.6	73.4	65.5	66.0	87.2	73.1	51.8	69.2	57.8	63.3	86.7	71.3
SLOTAlign	0.9	4.0	2.2	50.7	65.5	56.1	48.6	54.5	50.7	1.5	5.8	3.1	53.2	60.8	55.7	49.3	52.6	50.9
HOT	5.3	16.0	5.2	38.1	65.6	23.7	25.4	37.9	15.2	33.9	61.2	21.4	32.1	52.1	19.5	52.1	66.2	28.5
JOENA	18.7	35.1	24.4	73.2	92.1	80.2	63.7	72.9	66.8	66.3	87.8	73.0	62.9	75.0	67.2	66.3	89.0	74.3

Dataset	Social		Publication		Biological		Knowledge		Infrastructure		Communication	
Metrics	**Time(s)**	**Memory(GB)**	**Time(s)**	**Memory(GB)**	**Time(s)**	**Memory(GB)**	**Time(s)**	**Memory(GB)**	**Time(s)**	**Memory(GB)**	**Time(s)**	**Memory(GB)**
IsoRank	25.17	3.54	57.99	6.89	13.01	2.67	1.54×10^{2}	15.97	0.28	0.66	0.17	0.80
FINAL	5.91	5.39	6.75	10.06	1.88	3.54	18.10	24.37	0.10	0.80	0.11	0.88
IONE	6.34×10^{3}	1.94	1.43×10^{4}	4.16	1.41×10^{4}	1.93	1.50×10^{4}	8.75	9.41×10^{3}	0.90	8.10×10^{3}	1.02
REGAL	9.38	1.16	16.14	3.18	7.28	1.55	30.83	5.96	0.76	0.81	1.17	0.77
CrossMNA	3.06×10^{2}	1.40	1.16×10^{3}	3.16	5.33×10^{2}	1.58	1.11×10^{3}	6.14	59.03	0.98	5.78×10^{2}	0.81
NetTrans	1.56×10^{2}	8.88	5.14×10^{2}	8.57	1.08×10^{2}	3.90	3.65×10^{2}	21.90	6.46	1.28	12.37	1.54
WAlign	0.61	2.65	9.41	9.88	2.46	3.86	10.05	15.40	0.12	1.43	0.20	1.11
BRIGHT	21.76	3.00	1.26×10^{2}	5.66	33.55	3.24	3.81×10^{2}	11.53	0.28	1.14	0.33	1.12
NeXtAlign	40.89	3.75	1.55×10^{2}	7.82	19.35	3.57	2.62×10^{3}	13.57	0.29	1.34	2.44	0.99
WLAlign	7.41×10^{2}	2.17	2.69×10^{3}	11.98	1.15×10^{3}	4.21	3.24×10^{3}	28.33	4.25×10^{2}	1.35	6.99×10^{2}	0.95
PARROT	76.76	6.26	2.99×10^{2}	11.68	84.63	3.98	8.95×10^{2}	28.47	0.76	0.85	0.82	0.90
SLOTAlign	46.31	6.40	5.64×10^{2}	11.55	67.66	3.96	1.03×10^{4}	28.27	3.85	1.03	1.23	0.80
HOT	4.01×10^{2}	3.85	7.89×10^{2}	7.75	7.08×10^{2}	4.44	2.08×10^{3}	18.43	8.95	2.12	7.04	4.46
JOENA	58.73	4.89	30.60	2.73	3.65	1.39	6.61×10^{2}	26.47	0.05	1.02	0.49	0.86

a globally constrained optimization problem. Compared with embedding-based methods, which infer alignment from noisy embedding similarities, OT-based methods directly learns a robust alignment matrix from transportation cost, thanks to the marginal constraints that naturally encourage one-to-one node alignment. Empowered by constrained optimization and informative transportation cost encoded by powerful graph proximity measures or learnable node embeddings, OT-based methods learns *robust*, *deterministic*, and *global-structure-aware* alignment.

Takeaway #1: Optimal transport demonstrates significant potentials in NA.

Best-performing OT-based methods consistently outperform consistency and embedding-based approaches by a significant margin in alignment performance across diverse domains, demonstrating the power of constrained optimization and informative transport cost which lead to robust, deterministic, and global-structure-aware alignment.

5.3 EFFICIENCY AND SCALABILITY RESULTS

We also include the efficiency results of NA algorithms on plain networks under a semi-supervised setting with a 20% training ratio in Table 2. To further evaluate the scalability of NA algorithms, we conduct another set of experiments on synthetic graphs generated by the Erdős–Rényi (ER) (Erdos et al., 1960) model with a fixed average node degree of 10 under the same semi-supervised setting, and record the inference time and peak memory usage as the number of nodes increases in Figure 4.

Embedding-based methods typically face a two-out-of-three trade-off among effectiveness, time efficiency, and memory efficiency. Specifically, WAlign (Gao et al., 2021) and BRIGHT (Yan et al., 2021b) are among the most scalable algorithms in terms of inference time thanks to simple neural

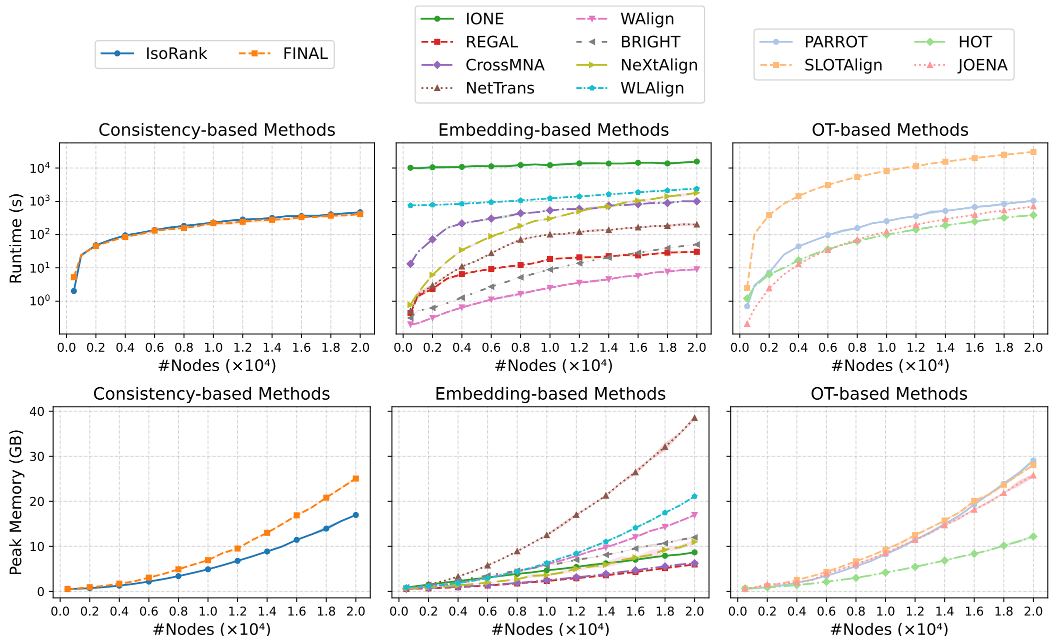

Figure 4: Scalability results on ER graphs. The **x-axis** shows the number of nodes in the ER graphs (in 10^4), and the **y-axis** of the 1st/2nd row shows the inference time and peak memory usage, respectively.

network (NN) structures which only requires a forward pass during inference. However, they tends to be less scalable in memory usage due to overheads of NN parameters. In terms of memory usage, CrossMNA (Chu et al., 2019) and IONE (Liu et al., 2016) achieve the best scalability as they learn low-dimensional embeddings without using NN. However, they tends to be less scalable in time since their transductive embeddings requires retraining for different networks (Hamilton et al., 2017). REGAL (Heimann et al., 2018) achieve both time and memory scalability by decomposition on sampled embedding matrices but is less effective compared to other NA methods.

Takeaway #2: Embedding-based methods face a two-out-of-three trade-off among effectiveness, time efficiency, and memory efficiency.

Embedding-based methods face trade-off among transductive embeddings for memory efficiency, inductive embeddings for time efficiency, and learning-based approaches for effectiveness.

Consistency-based methods (Singh et al., 2008; Zhang & Tong, 2016), on the other hand, scale moderately in terms of both time and memory usage. OT-based methods, in general, share similar scalability results as consistency-based methods since the optimizations of both kinds of methods involve matrix operations of quartic complexity. Although the original OT problem is non-convex and computationally expensive to solve by gradient descent (Tang et al., 2023), PARROT (Zeng et al., 2023) and JOENA (Yu et al., 2025) solve the OT problem efficiently by convex approximation (Peyré et al., 2019) and proximal point methods (Xu et al., 2019a). HOT (Zeng et al., 2024) further utilizes a hierarchical OT framework for cluster-level alignment to scale efficiently to large networks.

Takeaway #3: OT-based methods requires efficient optimization methods to scale similarly as consistency-based methods.

OT-based methods requires efficient optimization of OT problem, e.g, convex approximation, to scale moderately like consistency-based methods in terms of both time and memory usage.

5.4 Robustness and Sensitivity Results

We evaluate the robustness of NA methods under various types of graph noises, as well as their sensitivity to different levels of supervision. Our key findings are twofold. *First*, different NA methods show distinct sensitivities to different types of graph noises, suggesting that effective integration of different alignment techniques can potentially improve the overall robustness of NA algorithms. *Second*, current NA algorithms remain sensitive to supervision, underscoring the need for future research on self-supervised alignment approaches. Due to space constraints, detailed experimental results, analysis, and key takeaways are provided in Appendix D.1 and D.2.

6 CONCLUSION

In this paper, we introduce PLANETALIGN, a comprehensive library that facilitates the benchmarking and development of network alignment methods. PLANETALIGN highlights a collection of 18 different public datasets spanning 6 different domains, along with a unified and efficient implementation of 14 different NA algorithms of 3 different categories. With a comprehensive list of evaluation metrics, benchmarking tools, and utility functions implemented as easy-to-use APIs, PLANETALIGN not only enables fair and reproducible benchmarking of NA algorithms but also facilitates the development of new NA methods. Through extensive benchmark, we reveal practical insights into the strengths and limitations of existing NA methods which guides the development of future NA algorithms.

7 LIMITATIONS AND FUTURE WORK

While we introduce a comprehensive library for benchmarking NA, PLANETALIGN could be potentially improved from the following two directions. *First*, although PLANETALIGN features a rich collection of baselines, some variants of NA methods that targets a specific kind of network remain uncovered, e.g., entity alignment approaches (Chen et al., 2023; Liu et al., 2023b; Yan et al., 2021a) for aligning knowledge graphs. *Second*, PLANETALIGN focuses primarily on benchmarking pairwise NA problems. Although multi-network alignment methods are included in PLANETALIGN (Chu et al., 2019; Zeng et al., 2024), benchmarking under a simultaneous multi-network alignment setting remains underexplored at this stage.

As for future work, we will continuously expand PLANETALIGN to incorporate new NA datasets, algorithms, benchmark settings, and utility functions. Specifically, *for NA datasets*, we plan to include 1) multi-network alignment datasets which consist of more than two networks, such as multi-layered version of ArXiv (De Domenico et al., 2015a), Twitter (Omodei et al., 2015), and SacchCere (De Domenico et al., 2015b), 2) dynamic networks which evolves over time, such as synthetic datasets from (Vijayan et al., 2017) and (Yan et al., 2021a), and 3) cross-domain datasets which consist of networks from different domains, such as text-image network constructed by GOT (Chen et al., 2020); *for NA algorithms*, we plan to introduce 1) domain-specific alignment algorithms, such as entity alignment methods DualMatch (Liu et al., 2023b) and MEAformer (Chen et al., 2023), 2) multi-network alignment algorithms such as MrMine (Du & Tong, 2019), 3) dynamic NA algorithms such as DynaMAGNA++ (Vijayan et al., 2017) and DINGA (Yan et al., 2021a); *for benchmark settings*, we plan to add additional evaluation metrics for measure the uncertainty of the alignment (Zhou et al., 2021), which are critical for developing active or self-improving NA methods highlighted as important future directions in our paper; for utility functions, our immediate goal is to introduce scalability tools to allow easy acceleration of NA algorithms built upon PLANETALIGN, including distributed training APIs, sparse and low-rank matrix optimizations, and low-rank (Scetbon & Cuturi, 2022) & sliced OT (Liu et al., 2024) optimization tools.

Ethics statement.

Our library uses only publicly available datasets and conducts evaluation in a transparent and responsible manner in accordance with the code of ethics of ICLR. The research does not involve human subjects, animal studies, or any other procedures that may raise ethical concerns.

Reproducibility statement.

To ensure reproducibility, for **datasets** in PlanetAlign, we include their detailed statistics and description in Appendix A. For **experimental setup**, we include detailed description of adopted evaluation metrics, machines, dataset splits, and hyperparameter settings in Section 5.1 and Appendix C. The **source code** of PlanetAlign is available at `https://anonymous.4open.science/r/PlanetAlign-E9BA`, with detailed **documentation** at `https://planetalign.netlify.app`.

References

Mohsen Bayati, Margot Gerritsen, David F Gleich, Amin Saberi, and Ying Wang. Algorithms for large, sparse network alignment problems. In *2009 Ninth IEEE International Conference on Data Mining*, pp. 705–710. IEEE, 2009.

Xuezhi Cao and Yong Yu. Asnets: A benchmark dataset of aligned social networks for cross-platform user modeling. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1881–1884, 2016.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pp. 1542–1553. PMLR, 2020.

Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. Entity alignment with noisy annotations from large language models. *Advances in Neural Information Processing Systems*, 37:15097–15120, 2024.

Zhuo Chen, Jiaoyan Chen, Wen Zhang, Lingbing Guo, Yin Fang, Yufeng Huang, Yichi Zhang, Yuxia Geng, Jeff Z Pan, Wenting Song, et al. Meaformer: Multi-modal entity alignment transformer for meta modality hybrid. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3317–3327, 2023.

Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. Cross-network embedding for multi-network alignment. In *The world wide web conference*, pp. 273–284, 2019.

Connor Clark and Jugal Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, 2014.

Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 5(1):011027, 2015a.

Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature communications*, 6(1):6864, 2015b.

Christoph Döpmann. Survey on the graph alignment problem and a benchmark of suitable algorithms. *Institut für Informatik*, 2013.

Boxin Du and Hanghang Tong. Mrmine: Multi-resolution multi-network embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 479–488, 2019.

Boxin Du, Si Zhang, Yuchen Yan, and Hanghang Tong. New frontiers of multi-network mining: Recent developments and future trend. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 4038–4039, 2021.

Paul Erdos, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

Fazle E Faisal, Lei Meng, Joseph Crawford, and Tijana Milenković. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology*, 2015: 1–19, 2015.

Ji Gao, Xiao Huang, and Jundong Li. Unsupervised graph alignment with wasserstein distance discriminator. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 426–435, 2021.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 117–126, 2018.

Yunfei Hu, Manfei Xie, Yikang Li, Mingxing Rao, Wenjun Shen, Can Luo, Haoran Qin, Jihoon Baek, and Xin Maizie Zhou. Benchmarking clustering, alignment, and integration methods for spatial transcriptomics. *Genome Biology*, 25(1):212, 2024.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd international conference on world wide web*, pp. 1343–1350, 2013.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

Na Li, Renyu Zhu, Xiaoxu Zhou, Xiangnan He, Wenyuan Cai, Ming Gao, and Aoying Zhou. On disambiguating authors: Collaboration network reconstruction in a bottom-up manner. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 888–899. IEEE, 2021.

Li Liu, William K Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *Ijcai*, volume 16, pp. 1774–80, 2016.

Li Liu, Penggang Chen, Xin Li, William K Cheung, Youmin Zhang, Qun Liu, and Guoyin Wang. Wl-align: Weisfeiler-lehman relabeling for aligning users across networks via regularized representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):445–458, 2023a.

Xiaoze Liu, Junyang Wu, Tianyi Li, Lu Chen, and Yunjun Gao. Unsupervised entity alignment for temporal knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, pp. 2528–2538, 2023b.

Xinran Liu, Rocío Díaz Martín, Yikun Bai, Ashkan Shahbazi, Matthew Thorpe, Akram Aldroubi, and Soheil Kolouri. Expected sliced transport plans. *arXiv preprint arXiv:2410.12176*, 2024.

Elisa Omodei, Manlio De Domenico, and Alex Arenas. Characterizing interactions in online social networks during exceptional events. *Frontiers in Physics*, 3:59, 2015.

Daniel Park, Rohit Singh, Michael Baym, Chung-Shou Liao, and Bonnie Berger. Isobase: a database of functionally related proteins across ppi networks. *Nucleic acids research*, 39(suppl_1):D295–D300, 2010.

Yongqiang Peng, Xiaoliang Chen, Duoqian Miao, Xiaolin Qin, Xu Gu, and Peng Lu. Unveiling user identity across social media: a novel unsupervised gradient semantic model for accurate and efficient user alignment. *Complex & Intelligent Systems*, 11(1):24, 2025.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pp. 2664–2672. PMLR, 2016.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Miklós Z Rácz and Jifan Zhang. Harnessing multiple correlated networks for exact community recovery. *Advances in Neural Information Processing Systems*, 37:53834–53886, 2024.

Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 385–394, 2017.

Meyer Scetbon and Marco Cuturi. Low-rank optimal transport: Approximation, statistics and debiasing. *Advances in Neural Information Processing Systems*, 35:6802–6814, 2022.

Jiangli Shao, Yongqing Wang, Fangda Guo, Boshen Shi, Huawei Shen, and Xueqi Cheng. Cana: Causal-enhanced social network alignment. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 2219–2228, 2023.

Rohit Singh, Jinbo Xu, and Bonnie Berger. 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.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 914–921, 2020.

Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl_1): D535–D539, 2006.

Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International semantic web conference*, pp. 628–644. Springer, 2017.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *arXiv preprint arXiv:2003.07743*, 2020.

Jianheng Tang, Weiqi Zhang, Jiajin Li, Kangfei Zhao, Fugee Tsung, and Jia Li. Robust attributed graph alignment via joint structure learning and optimal transport. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1638–1651. IEEE, 2023.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 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.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pp. 613–622. IEEE, 2006.

Huynh Thanh Trung, Nguyen Thanh Toan, Tong Van Vinh, Hoang Thanh Dat, Duong Chi Thang, Nguyen Quoc Viet Hung, and Abdul Sattar. A comparative study on network alignment techniques. *Expert Systems with Applications*, 140:112883, 2020.

Vipin Vijayan, Dominic Critchlow, and Tijana Milenković. Alignment of dynamic networks. *Bioinformatics*, 33(14):i180–i189, 2017.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32, 2019a.

Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual knowledge graph alignment via graph matching neural network. *arXiv preprint arXiv:1905.11605*, 2019b.

Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. Dynamic knowledge graph alignment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4564–4572, 2021a.

Yuchen Yan, Si Zhang, and Hanghang Tong. Bright: A bridging algorithm for network alignment. In *Proceedings of the web conference 2021*, pp. 3907–3917, 2021b.

Yuchen Yan, Qinghai Zhou, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. Dissecting cross-layer dependency inference on multi-layered inter-dependent networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2341–2351, 2022.

Yuchen Yan, Yongyi Hu, Qinghai Zhou, Shurang Wu, Dingsu Wang, and Hanghang Tong. Topological anonymous walk embedding: A new structural node embedding approach. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2796–2806, 2024.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Qi Yu, Zhichen Zeng, Yuchen Yan, Lei Ying, R Srikant, and Hanghang Tong. Joint optimal transport and embedding for network alignment. In *Proceedings of the ACM on Web Conference 2025*, pp. 2064–2075, 2025.

F Zare Mirak-Abad and Z Ghorbanali. Praf framework for global protein-protein interaction network alignment. *Modares Journal of Biotechnology*, 10(2):329–334, 2025.

Zhichen Zeng, Si Zhang, Yinglong Xia, and Hanghang Tong. Parrot: Position-aware regularized optimal transport for network alignment. In *Proceedings of the ACM Web Conference 2023*, pp. 372–382, 2023.

Zhichen Zeng, Boxin Du, Si Zhang, Yinglong Xia, Zhining Liu, and Hanghang Tong. Hierarchical multi-marginal optimal transport for network alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16660–16668, 2024.

Jiawei Zhang and S Yu Philip. Integrated anchor and social link predictions across social networks. In *IJCAI*, pp. 2125–2132, 2015.

Si Zhang and Hanghang Tong. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1345–1354, 2016.

Si Zhang and Hanghang Tong. Network alignment: recent advances and future directions. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 3521–3522, 2020.

Si Zhang, Hanghang Tong, Jie Tang, Jiejun Xu, and Wei Fan. ineat: Incomplete network alignment. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 1189–1194. IEEE, 2017.

Si Zhang, Hanghang Tong, Ross Maciejewski, and Tina Eliassi-Rad. Multilevel network alignment. In *The World Wide Web Conference*, pp. 2344–2354, 2019a.

Si Zhang, Hanghang Tong, Ross Maciejewski, and Tina Eliassi-Rad. Multilevel network alignment. In *The World Wide Web Conference*, pp. 2344–2354, 2019b.

Si Zhang, Hanghang Tong, Yinglong Xia, Liang Xiong, and Jiejun Xu. Nettrans: Neural cross-network transformation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 986–996, 2020.

Si Zhang, Hanghang Tong, Long Jin, Yinglong Xia, and Yunsong Guo. Balancing consistency and disparity in network alignment. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2212–2222, 2021.

Qinkai Zheng, Xu Zou, Yuxiao Dong, Yukuo Cen, Da Yin, Jiarong Xu, Yang Yang, and Jie Tang. Graph robustness benchmark: Benchmarking the adversarial robustness of graph machine learning. *arXiv preprint arXiv:2111.04314*, 2021.

Qinghai Zhou, Liangyue Li, Xintao Wu, Nan Cao, Lei Ying, and Hanghang Tong. Attent: Active attributed network alignment. In *Proceedings of the Web Conference 2021*, pp. 3896–3906, 2021.

Qi Zhu, Carl Yang, Yidan Xu, Haonan Wang, Chao Zhang, and Jiawei Han. Transfer learning of graph neural networks with ego-graph information maximization. *Advances in Neural Information Processing Systems*, 34:1766–1779, 2021.

Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.

TABLE OF APPENDIX

A Datasets Details

A.1 Dataset Statistics

Table 3: Dataset Statistics.

Domain	Networks	# nodes	# edges	# node attr.	# edge attr.	Type
Social Networks	Foursquare Twitter	5,313 5,120	54,233 130,575	0 0	0 0	Real-world
	Douban(online) Douban(offline)	3,906 1,118	8,164 1,511	538 538	2 2	Real-world
	Flickr Lastfm	12,974 15,436	16,149 16,319	3 3	3 3	Real-world
	Flickr Myspace	6,714 10,733	7,333 11,081	3 3	3 3	Real-world
Communication Networks	Phone Email	1,000 1,003	41,191 4,628	0 0	0 0	Real-world
	Arenas1 Arenas2	1,135 1,135	10,902 10,800	50 50	0 0	Synthetic
Publication Networks	ACM DBLP	9,872 9,916	39,561 44,808	17 17	0 0	Real-world
	Cora1 Cora2	2,708 2,708	6,334 4,542	1,433 1,433	0 0	Synthetic
	ArXiv1 ArXiv2	18,722 18,722	217,921 168,394	0 0	0 0	Synthetic
Biological Networks	SacchCere1 SacchCere2	5,928 5,042	66,150 29,599	0 0	0 0	Real-world
	PPI1 PPI2	3,480 3,480	117,429 90,741	50 50	0 0	Synthetic
	GGI1 GGI2	10,403 10,403	115,755 89,448	0 0	0 0	Synthetic
Knowledge Graphs	DBP15K_ZH DBP15K_EN	19,388 19,572	70,414 95,142	300 300	0 0	Real-world
	DBP15K_JA DBP15K_EN	19,814 19,780	77,214 93,484	300 300	0 0	Real-world
	DBP15K_FR DBP15K_EN	19,661 19,993	105,997 115,722	300 300	0 0	Real-world
Infrastructure Networks	Italy1 Italy2	349 349	416 435	0 0	0 0	Real-world
	Airport1 Airport2	1,190 1,190	14,958 11,560	4 4	0 0	Synthetic
	PeMS08-1 PeMS08-2	170 170	301 233	3 3	0 0	Synthetic

A.2 Dataset Descriptions

Detailed datasets descriptions are introduced as follows

- **Foursquare-Twitter** (Zhang & Philip, 2015). A pair of online social networks, Foursquare and Twitter. Nodes represent users and an edge exists between two users if they have follower/followee relationships. Both networks are plain networks. There are 1,609 common users across two networks.
- **Douban** (Zhang & Tong, 2016). A pair of online-offline social networks constructed from Douban. Nodes represent users and edges represent user interactions on the website. The location of a suer is treated as the node attribute, and the contact/friend relationship are treated as the edge attributes. There are 1,118 common user across the two networks.

- **Flickr-LastFM** (Zhang & Tong, 2016). A pair of social networks from Flickr and LastFM. Nodes in both networks represent users, and edges represent friend / following relationships in Flickr and LastFM, respectively. The gender of a user is treated as the node attributes (male, female, unknown), and the level of people a user is connected to is treated as the edge attributes (e.g., leader with leader). There are 452 common users across two networks.
- **Flickr-MySpace** (Zhang & Tong, 2016). A pair of social networks from Flickr and MySpace. Nodes in both networks represent users, and edges represent friend / following relationships. The gender of a user is treated as the node attributes (male, female, unknown), and the level of people a user is connected to is treated as the edge attributes (e.g., leader with leader). There are 267 common users across two networks.
- **ACM-DBLP** (Tang et al., 2008). A pair of undirected co-authorship networks, ACM and DBLP. Nodes represent authors and edges an edge exists between two authors if they co-author at least one paper. Node attributes are available in both networks. There are 6,325 common authors across two networks.
- **Cora** (Yang et al., 2016). A pair of networks synthesized from the citation network Cora. Each network Nodes represent publications and an edge exists between two publications if they have a citation relationship. The two networks are noisy permutations of the original network generated by randomly inserting 10% edges (Cora1) and deleting 15% edges (Cora2) from the original network, respectively. There are in total 2,708 common nodes across two networks.
- **ArXiv** (Leskovec et al., 2007). A pair of networks synthesized from the Arxiv ASTRO-PH (Astro Physics) collaboration network (Leskovec et al., 2007). Nodes represent authors and an edge exists between two authors if they have co-authored a paper. The two networks are noisy permutations of the original network generated by randomly inserting 10% edges (ArXiv1) and deleting 15% edges (ArXiv2) from the original network, respectively. Node and edge attributes are not available. There are in total 18,722 common nodes across two networks.
- **SacchCere** (Stark et al., 2006; De Domenico et al., 2015b). A pair of direct interaction layer and association layer from the SacchCere multiplex GPI network. The SacchCere network consider different kinds of protein and genetic interactions for Saccharomyces Cerevisiae in BioGRID (Stark et al., 2006), a public database that archives and disseminates genetic and protein interaction data from humans and model organisms. There are in total 1,337 common nodes across two layers of networks.
- **PPI** (Zitnik & Leskovec, 2017). A pair of networks synthesized from the protein-protein interaction (PPI) network (Zitnik & Leskovec, 2017), where nodes represent human proteins and edges represent physical interaction between proteins in a human cell. The immunological signatures are included as node features. The two networks are noisy permutations of the original network generated by randomly inserting 10% edges (PPI1) and deleting 15% edges (PPI2) from the original network, respectively. There are in total 3,980 common nodes across two networks.
- **GGI** (Park et al., 2010). A pair of networks synthesized from the human gene-gene interaction (PPI) network from IsoBase (Park et al., 2010). Nodes represent human genes and edges represent gene-gene interactions. The two networks are noisy permutations of the original network generated by randomly inserting 10% edges (GGI1) and deleting 15% edges (GGI2) from the original network, respectively. There are in total 10,403 common nodes across two networks.
- **DBP15K ZH-EN, JA-EN, FR-EN** (Sun et al., 2017). Pairs of Chinese, Japanese, and French to English version of multi-lingual DBpedia networks. The node attributes are given by pre-trained and aligned monolingual word embeddings (Xu et al., 2019b). There are 15,000 pairs of aligned entities in DBP15K ZH-EN (Chinese to English), JA-EN (Japanese to English), and FR-EN (French to English), respectively.
- **Italy** (Yan et al., 2022). A pair of power grid networks from two regions in Italy. Nodes represent power stations and edges represent the existence of power transfer lines. Node attributes are derived from node labels. There are in total 377 common nodes across two networks inferred from the ground-truth cross-layer dependencies.
- **Airport** (Zhu et al., 2021). A pair of networks synthesized from the American air-traffic network (Ribeiro et al., 2017). Nodes represent airports and an edge exists between two aiports if there are commercial flights between them. The level of activity in each airport is used as node attributes. The two networks are noisy permutations of the original network generated by randomly inserting 10% edges (Airport1) and deleting 15% edges (Airport2) from the original network, respectively. There are in total 1,190 common nodes across two networks.
- **PeMS08** (Song et al., 2020). A pair of traffic networks synthesized from the Performance Measurement System (PeMS) Data Source. Nodes represent sensors and edges indicate traffic flow

correlation. Node attributes are averaged across all time interval. The two networks are noisy permutations of the original network generated by randomly inserting 10% edges (PeMS08-1) and deleting 15% edges (PeMS08-2) from the original network, respectively. There are in total 170 common nodes across two networks.

- **Phone-Email** (Zhang et al., 2017). A pair of communication networks among people via phone or email. Nodes represent people and an edge exists between two people if they communicate via phone or email at least once. Phone network includes 1,000 nodes and 41,191 edges. Email network includes 1,003 nodes and 4,627 edges. Both networks are plain networks. There are 1,000 common people across two networks.
- **Arenas** (Kunegis, 2013). A pair of networks synthesized from the email communication network Arenas at the University Rovira i Virgili. Nodes are users and each edge represents that at least one email was sent. The two networks are noisy permutations of each other. There are in total 1,135 common nodes across two networks.

B Network Alignment Methods

B.1 Consistency-based Methods

- **IsoRank (Singh et al., 2008).** IsoRank is originally designed for global alignment of multiple PPI networks. It is built upon neighborhood topology consistency which assumes that the neighbors of aligned anchor nodes should be aligned as well, and is formulated as an eigenvalue problem. (Yan et al., 2021b) reveals that the formulation of IsoRank can be considered as conducting random walk propagation of anchor links on the product graph to achieve topology consistency.
- **FINAL (Zhang & Tong, 2016).** FINAL interprets the alignment consistency principles as an optimization problem and introduces additional consistency principles at node/edge attribute levels to handle attributed network alignment.

B.2 Embedding-based Methods

- **IONE (Liu et al., 2016).** IONE modeled the follower/followee-ship of different nodes as input/output context vectors to learn proximity-preserving node embeddings, and solve the node embedding and network alignment problem based on a unified framework.
- **REGAL (Heimann et al., 2018).** REGAL designs an embedding learning methods called xNetMF which learns powerful node embeddings by matrix factorization on a linear combination between cross-network structural and attribute similarity matrix. Based on xNetMF embeddings, REGAL infer node-level alignment of two networks based on Euclidean distance of nodes in the embedding space.
- **CrossMNA (Chu et al., 2019).** CrossMNA leverages cross-network structural information to learn inter and intra network embeddings simultaneously. By comparing inter network embeddings across different networks, CrossMNA is capable of aligning multiple networks at the same time.
- **NetTrans (Zhang et al., 2020).** NetTrans approach the network alignment problem from a cross-network transformation perspective. It learns the transformation of both network structure and node attributes at different resolutions to identify node-level alignment.
- **WAlign (Gao et al., 2021).** WAlign learns node embeddings by a lightweight GCN model to capture both local and global graph patterns and proposes a Wasserstein distance discriminator to minimize the Wasserstein distance between node embeddings across different graphs.
- **BRIGHT (Yan et al., 2021b).** BRIGHT first generate positional node embeddings by random walk with restart (RWR) (Tong et al., 2006) against anchor links. To handle plain network alignment, BRIGHT-U learns position-aware embeddings by transforming RWR embeddings through a shared MLP. To handle attributed network alignment, BRIGHT-A use a shared GCN model for transforming node attributes and concatenates the output embeddings with RWR vectors before feeding into the shared MLP.
- **NeXtAlign (Zhang et al., 2021).** NeXtAlign designs a spatial GCN model and learns node embeddings that balance the alignment consistency and disparity by crafting the sampling strategy.
- **WLAlign (Liu et al., 2023a).** WLAlign proposes a cross-network Weisfeiler-Lehman relabeling scheme to learn embeddings that preserves long-range connectivity to the anchor pairs on plain networks.

B.3 OT-based Methods

- **SLOTAlign (Tang et al., 2023).** SLOTAlign utilizes a parameter-free GCN model to encode graph structure. By integrating output embeddings of multiple layers of GCN through a learnable linear combination, SLOTAlign encode the Gromov-Wasserstein distance between two networks via the learned embeddings and optimize the embedding and optimal transport problem alternatively to infer alignment.
- **PARROT (Zeng et al., 2023).** PARROT encodes a position-aware transportation cost by random walk with restart (RWR) (Tong et al., 2006) on separate and product graphs, and integrate consistency principle at node, edge, and neighborhood levels into the optimal transport formulation. Then, it solves the resulting optimization problem efficiently via constrained proximal point methods to infer node-level alignment.
- **HOT (Zeng et al., 2024).** HOT proposes a hierarchical multi-marginal optimal transport framework which first decomposes multiple networks to aligned clusters via the fused Gromov-Wasserstein (FGW) barycenter (Peyré et al., 2016) and then aligns node in aligned clusters simultaneous by solving optimal transport problem within clusters.
- **JOENA (Yu et al., 2025).** JOENA transforms the transport plan of optimal transport into an adaptive sampling strategy via a learnable transformation to learn node embeddings and alignment in a mutual beneficial manner.

C Detailed Experimental Setup

Machine. All experiments are conducted on a computing server equipped with dual Intel® Xeon® Gold 6240R CPUs and 4 NVIDIA Tesla V100-SXM2 GPUs with 32GB memory each.

Dataset split and hyperparameters. To mitigate the randomness introduced by a single random dataset split, we report the average metrics of 5 different dataset split based on 5 randomly selected seeds. All NA methods are evaluated under the same dataset splits to ensure a fair comparison. For each dataset split, we run a NA algorithm 5 times and report the average metrics. Hyperparameters are tuned with a fixed budget of 5 per key parameter based on the default values and hyperparameter study in the original papers. Detailed hyperparameter search spaces can be found in Table 4.

Table 4: Hyperparameter search spaces.

NA Method	Search Parameters
IsoRank	$\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$
FINAL	$\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$
IONE	out_dim$\in \{32, 64, 100, 128, 256\}$
REGAL	k$\in \{1, 5, 10, 15, 20\}$, num_layers$\in \{1, 2, 3, 4, 5\}$, $\alpha \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$
CrossMNA	$d_1 \in \{10, 50, 100, 150, 200\}$, $d_2 \in \{10, 50, 100, 150, 200\}$
NetTrans	$\alpha \in \{0.01, 0.1, 1, 10, 100\}$, $\beta \in \{0.01, 0.1, 1, 10, 100\}$, $\gamma \in \{0.01, 0.1, 1, 10, 100\}$, $L \in \{1, 2, 3, 4, 5\}$
WAlign	$h \in \{128, 256, 512, 1024, 2048\}$, $\alpha \in \{0.01, 0.02, 0.04, 0.06, 0.08\}$
BRIGHT	$\beta \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$, out_dim$\in \{32, 64, 128, 256, 512\}$, neg_sample_size$\in \{100, 300, 500, 700, 900\}$
NeXtAlign	$\beta \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$, out_dim$\in \{32, 64, 128, 256, 512\}$, neg_sample_size$\in \{100, 300, 500, 700, 900\}$
WLAlign	out_dim$\in \{32, 64, 128, 256, 512\}$, neg_sample_size$\in \{20, 40, 60, 80, 100\}$
PARROT	$\eta \in \{0.1, 0.5, 1, 5, 10\}$, $\lambda_e \in \eta\lambda_e^{\text{default}}$, $\lambda_n \in \eta\lambda_n^{\text{default}}$, $\lambda_a \in \eta\lambda_a^{\text{default}}$, $\lambda_p \in \eta\lambda_p^{\text{default}}$
SLOTAlign	$\epsilon \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, step_size$\in \{1, 2, 3, 4, 5\}$
HOT	$\epsilon \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$
JOENA	$\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma_p \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, $\lambda_0 \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$

D Additional Experimental Results

D.1 Robustness Results

To benchmark the robustness of existing NA algorithms, we conduct controlled experiments to study the impact of edge, attribute, and supervision noises to alignment performance, offering practical insights into the development of robust NA methods.

Edge noise. We introduce edge-level noise to simulate real-world edge perturbation (Jin et al., 2020). Specifically, the p% edge noise level is defined as randomly adding/deleting p% edges in the

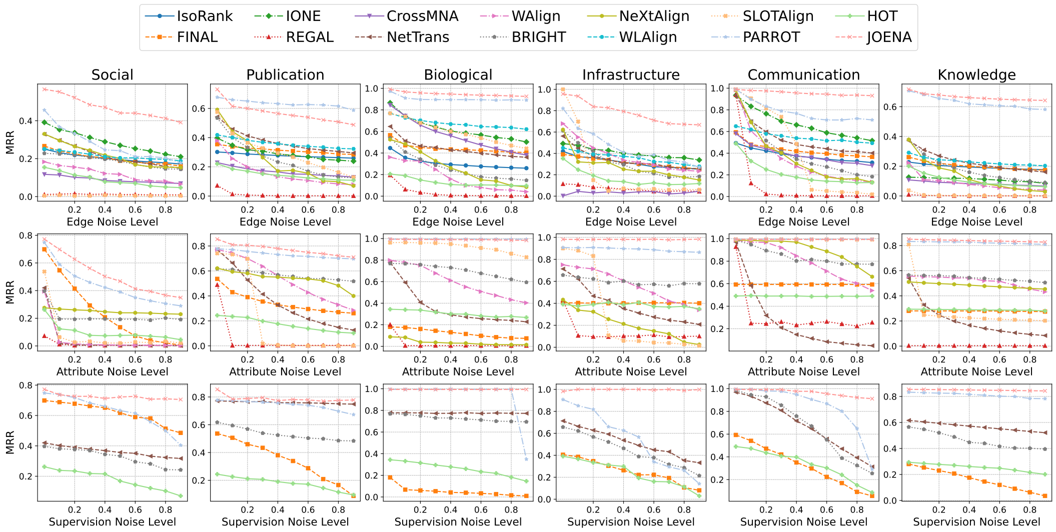

Figure 5: Robustness results of NA methods under different levels of edge, attribute, and supervision noises on representative datasets across 6 domains. The **x-axis** of the plots in the 1st/2nd/3rd row shows the noise level of edge/attribute/supervision, respectively, and the **y-axis** shows the MRR.

second network to be aligned (Tang et al., 2023; Zeng et al., 2023). We conduct evaluations of all NA methods under a semi-supervised (20% training ratio) plain NA setup to avoid potential interference of node/edge attributes.

Attribute noise. We introduce attribute-level noise to simulate real-world attribute perturbation (Zheng et al., 2021). Specifically, the p% attribute noise level is defined as randomly perturbing [2] p% node and edge attributes in the second network to be aligned (Zeng et al., 2023). We conduct evaluations of attributed NA methods under a semi-supervised attributed NA setting with a training ratio of 20%.

Supervision noise. We introduce supervision noise to evaluate the robustness of semi-supervised NA methods against noisy anchor node pairs (Yan et al., 2021b; Tang et al., 2023). Specifically, the p% supervision noise is defined as randomly setting p% anchor nodes in the second graph to non-anchor nodes. To ensure fair comparison, we only evaluate the robustness of semi-supervised attributed NA methods [3] against supervision noise under a semi-supervised attributed NA setting with a training ratio of 20%.

Analysis. Robustness results on five representative datasets are shown in Figure 5. Firstly, for **edge noise**, consistency-based methods, including IsoRank (Singh et al., 2008) and FINAL (Zhang & Tong, 2016), are among the most robust methods with the slightest performance drop across all datasets. Embedding-based methods (Liu et al., 2016; Heimann et al., 2018; Chu et al., 2019; Zhang et al., 2020; Gao et al., 2021; Yan et al., 2021b; Zhang et al., 2021; Liu et al., 2023a), while slightly less robust than consistency-based approaches, generally show descent performance degradation ratio as edge noise level increases. OT-based methods (Zeng et al., 2023; Tang et al., 2023; Zeng et al., 2024; Yu et al., 2025), on the other hand, differ significantly in terms of robustness to edge-level noisel, indicating that although OT can reduce the negative effect of graph noises by marginal constraint (Zeng et al., 2023; Yu et al., 2025), they require careful design of the transportation costs to avoid noise amplification during optimization. Nevertheless, OT-based methods PARROT (Zeng et al., 2023) and JOENA (Yu et al., 2025) consistently outperforms all other NA algorithms in alignment performance across different noise levels.

[2]We flip binary attributes and add standard gaussian noise into normalized continuous attributes.

[3]We include FINAL (Zhang & Tong, 2016) which has a semi-supervised version in its original paper.

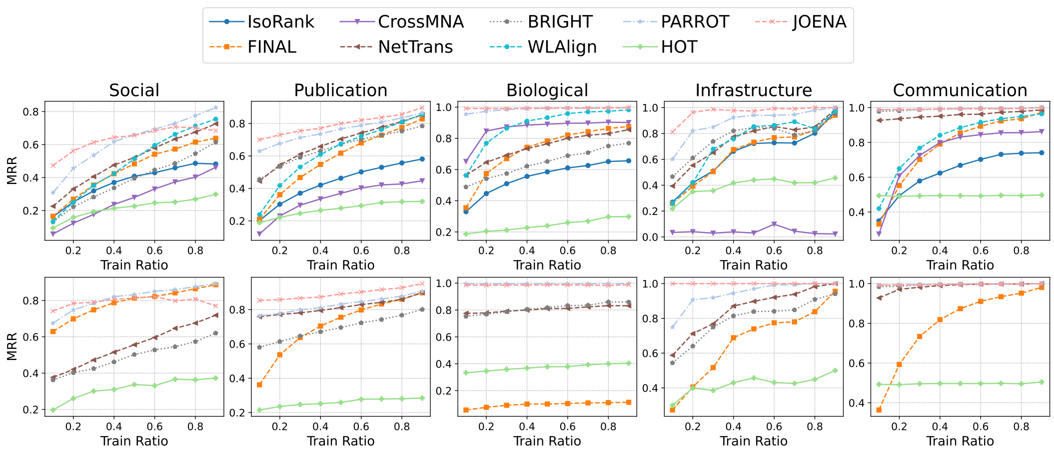

Figure 6: Sensitivity results of semi-supervised NA algorithms to different levels of supervision. The two rows correspond to plain and attributed NA settings respectively. The **x-axis** shows the training ratio (i.e., supervision level), and the **y-axis** shows the MRR.

Secondly, for **attribute noise**, PARROT (Zeng et al., 2023) and JOENA (Yu et al., 2025) becomes the most robust algorithms across all datasets. While both methods are OT-based, PARROT integrates consistency principles which further improve its robustness, and JOENA adopts an embedding-encoded OT cost learned via a MLP for robust alignment. Consistency-based methods remain robust to attribute noise on most datasets. Embedding-based methods are generally more sensitive to attribute noise than edge noise and suffer from significant performance drop under high attribute noise level, which highlights the need for more robust embedding learning approaches, potentially through the integration of optimal transport or consistency principles.

Finally, for **supervision noise**, the performance of most NA algorithms degrades significantly as the noise level increase, indicating that the effectiveness of existing semi-supervised NA methods rely heavy on the quality of anchor node pairs even when node/edge attributes are available. Nevertheless, JOENA (Yu et al., 2025) consistently shows the mildest performance drop across all datasets, demonstrating the power of effective combination of embedding and OT-based methods. Future methods may explore more effective integration of consistency, embedding, and OT-based approaches to better handle different kinds of real-world noise.

Takeaway #4: Different NA methods are sensitive to different kinds of noises. Effective integration of different NA techniques could be a way out.

Different NA methods may be sensitive to different kinds of real-world noises. Integrating different NA techniques effectively, such as consistency principles, embedding learning, and optimal transport, could potentially improve the overall robustness of NA algorithms.

D.2 Sensitivity to Supervision Results

To comprehensively evaluate the impact of supervision on the performance of NA algorithms, we conduct a set of experiments to study the sensitivity of semi-supervised NA methods to different levels of supervision. Specifically, we gradually increase the training ratio and report the MRR of semi-supervised NA methods on five representative datasets under both plain and attributed NA settings. The results are presented in Figure 6.

Analysis. Firstly, the performance of NA algorithms generally shows a growing trend as the training ratio increases, with only a few exceptions such as JOENA (Yu et al., 2025) on Douban (Zhang & Tong, 2016), potentially due to overfitting on training data or the presence of noisy anchor pairs from real-world datasets. Nonetheless, most NA methods benefit significantly from increased supervision, demonstrating its importance to the effectiveness of NA algorithms.

Secondly, the use of attribute information typically improves the performance of NA algorithms under low supervision. However, the performance gap between plain and attributed settings narrows as the training ratio increases. For example, PARROT (Zeng et al., 2023) achieve an MRR of approximately 0.7/0.3 on Douban with/without attribute information under 10% training ratio, whereas the performance rises to about 0.95/0.9 under a 90% training ratio. This suggests that while attributes can help in low-supervision scenarios, increasing supervision remains crucial even in the presence of node and edge attributes in graphs. Combined with our previous robustness study against supervision noise, we present the following findings:

Takeaway #5: Supervision greatly affect the effectiveness of NA algorithms.

The quality and quantity of supervision greatly affect the performance of NA algorithms even in the presence of node and edge attributes, suggesting that self-supervised learning methods which discover high-quality anchor pairs could be a promising directions for NA research.

D.3 COMPARISON WITH OFFICIAL IMPLEMENTATIONS

Table 5: Performance and runtime comparison with official implementations averaged against all datasets. Δ represents the absolute difference between official and PLANETALIGN's implementation.

Metrics	**MRR**			**Training Runtime(s)**			
Version	**Official**	**PLANETALIGN**	Δ	**Official**	**PLANETALIGN**	Δ	Speedup
REGAL	0.079	**0.080**	+0.001	20	**14**	-6	1.43
CrossMNA	0.220	**0.222**	+0.002	298	**210**	-88	1.42
NetTrans	0.373	**0.374**	+0.001	919	**817**	-102	1.12
WAlign	**0.271**	0.270	-0.001	79	**68**	-11	1.16
BRIGHT	0.362	**0.362**	+0.000	768	**619**	-149	1.24
NeXtAlign	0.391	**0.391**	+0.000	1319	**1234**	-85	1.07
WLAlign	**0.328**	0.322	-0.006	4018	**1276**	-2742	3.15
SLOTAlign	**0.200**	0.200	-0.001	891	**821**	-70	1.09
HOT	**0.173**	0.172	-0.001	239	**226**	-13	1.06
JOENA	**0.583**	**0.583**	+0.000	691	**679**	-12	1.02

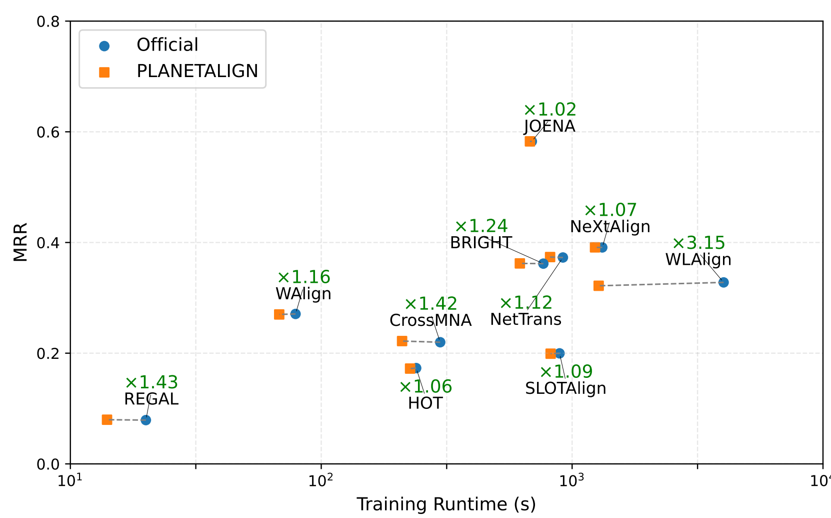

Figure 7: Performance and runtime comparison between official and PLANETALIGN's implementations. The **x-axis** shows the average training runtime, and the **y-axis** shows the average MRR of different NA algorithms across 18 datasets. The average speedup in runtime of each method are shown in green.

We conduct comparative experiments between the official and PLANETALIGN's implementations of NA algorithms to validate the correctness and efficiency of PLANETALIGN. To ensure a fair comparison, we only include algorithms that have official Python implementations to eliminate the efficiency discrepancy of different programming languages. All training parameters, including training epochs, are set as default in the official code. We report the average MRR and training runtime of official and PLANETALIGN's implementation across all 18 datasets in PLANETALIGN in Table 5 and Figure 7

We can see that PLANETALIGN's implementation show comparable performance across all baselines while achieving up to 3 times speed-up over official implementations, demonstrating the correctness and efficiency of our implementation of existing NA methods.

D.4 Scalability Results on Large Graphs

To further demonstrate the efficiency of our implementation on large-scale networks, we compare the training runtime with the official implementation on ER networks of 50K, 75K, and 100K nodes with an average node degree of 5 per network. As we can see in Figure 6, PLANETALIGN consistently outperform official implementations with up to 2.7 times speed-up.

Table 6: Runtime (s) comparison with official implementations on large-scale ER networks of 50K, 75K, and 100K nodes with average node degrees of 5. OOM represents out-of-memory.

# Nodes	**50K**			**75K**			**100K**		
Version	**Official**	**PLANETALIGN**	Speedup	**Official**	**PLANETALIGN**	Speedup	**Official**	**PLANETALIGN**	Speedup
REGAL	334	**214**	1.56	583	**390**	1.49	1.12×10^3	**731**	1.53
CrossMNA	4.12×10^3	$\mathbf{3.02\times10^3}$	1.36	7.36×10^3	$\mathbf{5.13\times10^3}$	1.43	1.37×10^4	$\mathbf{8.91\times10^3}$	1.54
WAlign	1.65×10^3	$\mathbf{1.29\times10^3}$	1.28	3.12×10^3	$\mathbf{2.54\times10^3}$	1.23	OOM	OOM	OOM
BRIGHT	2.82×10^4	$\mathbf{2.15\times10^4}$	1.31	8.95×10^4	$\mathbf{6.90\times10^4}$	1.29	3.47×10^5	$\mathbf{2.49\times10^5}$	1.40
NeXtAlign	6.55×10^4	$\mathbf{6.08\times10^4}$	1.08	3.68×10^5	$\mathbf{3.41\times10^5}$	1.08	OOM	OOM	OOM
WLAlign	2.46×10^5	$\mathbf{9.12\times10^4}$	2.70	OOM	OOM	OOM	OOM	OOM	OOM
SLOTAlign	3.01×10^4	$\mathbf{2.75\times10^4}$	1.10	OOM	OOM	OOM	OOM	OOM	OOM
HOT	3.14×10^3	$\mathbf{2.87\times10^3}$	1.10	6.17×10^3	$\mathbf{5.74\times10^3}$	1.07	1.08×10^4	$\mathbf{9.43\times10^3}$	1.15
JOENA	2.31×10^3	$\mathbf{2.28\times10^4}$	1.01	8.30×10^4	$\mathbf{8.12\times10^4}$	1.02	OOM	OOM	OOM

D.5 Detailed Effectiveness Results

Detailed effectiveness results on *plain* networks with a training ratio of 20% are shown in Table 7 and 8. Detailed effectiveness results on *attributed* networks with a training ratio of 20% are shown in Table 9.

E Statement of LLM Usage

In this paper, LLMs were used exclusively for formatting assistance and language polishing. At no point were LLMs involved significantly in research ideation and/or writing to the extent that they could be considered as a contributor. Therefore, the use of LLMs does not impact the core methodology, the scientific rigorousness, or the originality of this research.

Table 7: Detailed effectiveness results (Part I of II) on plain networks with a training ratio of 20%. The 1st/2nd/3rd best results are marked in red/blue/green respectively. Time denotes the inference time and Mem. denotes the peak memory usage.

Dataset	Foursquare-Twitter					Phone-Email				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
IsoRank	.0241±.0000	.1487±.0000	.0645±.0000	5.70±0.60	1.60±0.10	.0431±.0000	.2156±.0000	.1100±.0000	0.25±0.12	0.59±0.00
FINAL	.0474±.0000	.2407±.0000	.1062±.0000	0.96±0.03	2.30±0.00	.0494±.0000	.2725±.0000	.1257±.0000	0.10±0.04	0.63±0.01
IONE	.0202±.0052	.0985±.0051	.0481±.0045	1.10±0.02×10^{4}	1.20±0.00	.0941±.0032	.4037±.0120	.1952±.0031	2.30±0.06×10^{3}	0.80±0.00
REGAL	.0001±.0002	.0027±.0010	.0025±.0003	7.80±0.10	1.10±0.10	.0012±.0000	.0097±.0011	.0076±.0003	1.30±0.20	0.60±0.00
CrossMNA	.0162±.0034	.1011±.0061	.0456±.0039	1.10±0.06×10^{3}	1.00±0.00	.0305±.0029	.2163±.0086	.0968±.0021	1.10±0.27×10^{3}	0.62±0.00
NetTrans	.0809±.0043	.2470±.0074	.1347±.0048	5.90±0.86×10^{2}	3.50±1.20	.0216±.0027	.2546±.0020	.1020±.0016	2.40±1.70×10^{1}	0.85±0.02
WAlign	.0039±.0004	.0150±.0005	.0095±.0003	1.30±0.04	2.30±0.00	.0206±.0018	.1235±.0093	.0585±.0009	0.29±0.08	0.89±0.01
BRIGHT	.0537±.0027	.1784±.0012	.0923±.0019	9.20±0.20	1.40±0.00	.0476±.0033	.2516±.0028	.1186±.0029	0.29±0.01	0.95±0.01
NeXtAlign	.0387±.0040	.1420±.0163	.0745±.0075	7.90±0.25×10^{1}	2.30±0.00	.0570±.0045	.3012±.0116	.1411±.0036	4.60±0.10	0.86±0.01
WLAlign	.0924±.0016	.2103±.0036	.1325±.0009	1.20±0.14×10^{3}	3.00±0.00	.0764±.0012	.2669±.0047	.1412±.0014	7.50±1.10×10^{2}	1.00±0.00
PARROT	.1203±.0000	.2908±.0000	.1776±.0000	1.60±0.02×10^{1}	2.60±0.10	.2887±.0000	.7331±.0000	.4374±.0000	0.76±0.09	0.69±0.00
SLOTAlign	.0291±.0000	.1172±.0000	.0614±.0000	1.60±0.03×10^{2}	2.50±0.00	.0075±.0000	.0525±.0000	.0283±.0000	2.34±0.30	0.75±0.01
HOT	.0518±.0030	.1627±.0044	.0457±.0018	1.10±0.21×10^{2}	1.70±0.10	.0775±.0025	.3273±.0069	.0801±.0020	5.30±0.60	0.72±0.01
JOENA	.2673±.0069	.4478±.0083	.3304±.0083	2.80±0.05×10^{1}	2.50±0.06	.3468±.0029	.7809±.0037	.4973±.0018	0.76±0.08	0.72±0.01
Dataset	**ACM-DBLP**					**SacchCere1-SacchCere2**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
IsoRank	.1641±.0003	.6329±.0008	.3023±.0002	4.70±1.70×10^{1}	4.30±0.10	.0335±.0009	.2284±.0016	.0976±.0002	0.95±0.10	1.50±0.60
FINAL	.2082±.0000	.6893±.0000	.3612±.0000	3.90±0.10	6.60±0.10	.0467±.0000	.2379±.0000	.1083±.0000	0.54±0.03	1.70±0.60
IONE	.2515±.0028	.7267±.0075	.3979±.0023	1.30±0.05×10^{4}	2.60±0.13	.0458±.0037	.2100±.0055	.0992±.0016	8.30±0.04×10^{3}	1.30±0.60
REGAL	.0357±.0022	.1367±.0035	.0700±.0030	1.30±0.10×10^{1}	1.90±0.00	.0023±.0008	.0087±.0007	.0063±.0006	3.40±0.10	1.20±0.00
CrossMNA	.0742±.0034	.6108±.0031	.2290±.0025	5.50±0.13×10^{2}	1.90±0.00	.0046±.0007	.1452±.0041	.0492±.0023	1.80±0.13×10^{2}	0.98±0.00
NetTrans	.4148±.0018	.8107±.0009	.5429±.0012	1.10±0.34×10^{2}	1.50±0.50×10^{1}	.0523±.0017	.2534±.0038	.1150±.0020	1.20±0.30×10^{1}	2.20±0.20
WAlign	.2871±.0018	.5538±.0025	.3797±.0021	2.80±0.30	4.90±0.10	.0207±.0006	.0516±.0032	.0336±.0019	0.28±0.01	1.90±0.00
BRIGHT	.4052±.0011	.7957±.0011	.5346±.0013	6.80±0.10×10^{1}	3.80±0.00	.0353±.0015	.2188±.0062	.0915±.0021	4.60±0.00	1.60±0.00
NeXtAlign	.4670±.0019	.8401±.0017	.5915±.0011	1.70±0.04×10^{2}	4.00±0.00	.0292±.0035	.2075±.0075	.0886±.0040	1.50±0.40	1.30±0.10
WLAlign	.3152±.0012	.6446±.0008	.4183±.0008	1.40±0.04×10^{3}	7.00±0.00	.0535±.0011	.1639±.0014	.0888±.0007	5.80±0.99×10^{2}	1.50±0.00
PARROT	.5749±.0000	.8784±.0000	.6766±.0000	1.30±0.04×10^{2}	7.80±0.00	.0645±.0000	.2720±.0000	.1312±.0000	5.90±0.20	1.90±0.60
SLOTAlign	.4914±.0000	.7174±.0000	.5707±.0000	9.90±0.16×10^{2}	7.60±0.26	.0000±.0000	.0023±.0000	.0028±.0000	1.00±0.00	2.10±0.10
HOT	.3261±.0040	.6787±.0053	.2210±.0026	4.30±0.15×10^{2}	5.00±0.00	.0344±.0023	.1993±.0039	.0564±.0012	1.20±0.11×10^{3}	1.50±0.10
JOENA	.6149±.0458	.9062±.0219	.7136±.0389	3.50±0.50×10^{1}	3.00±2.00	.0589±.0026	.2284±.0062	.1129±.0027	0.31±0.01	1.10±0.00
Dataset	**DBP15K_ZH-EN**					**Italy1-Italy2**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
IsoRank	.1092±.0001	.4878±.0001	.2276±.0000	1.60±0.16×10^{2}	1.60±0.00×10^{1}	.0424±.0009	.2076±.0009	.0954±.0003	0.49±0.13	0.60±0.00
FINAL	.1327±.0000	.5347±.0000	.2634±.0000	1.80±0.00×10^{1}	2.40±0.00×10^{1}	.0430±.0000	.2202±.0000	.0958±.0000	0.13±0.01	0.84±0.00
IONE	.0633±.0033	.2512±.0080	.1258±.0047	1.50±0.03×10^{4}	8.70±0.00	.0245±.0027	.1570±.0149	.0679±.0028	6.20±0.03×10^{3}	0.84±0.00
REGAL	.0036±.0005	.0168±.0008	.0091±.0006	3.00±0.10×10^{1}	5.90±0.00	.0033±.0020	.0265±.0041	.0142±.0023	0.97±0.00	0.81±0.00
CrossMNA	.0303±.0012	.2816±.0046	.1077±.0017	9.90±0.26×10^{2}	6.00±0.00	.0023±.0015	.1447±.0105	.0476±.0024	3.70±0.70×10^{1}	0.98±0.01
NetTrans	.2625±.0011	.6022±.0010	.3717±.0009	5.00±0.22×10^{2}	3.00±1.30×10^{1}	.0503±.0019	.2248±.0027	.1110±.0020	0.62±0.15	1.10±0.00
WAlign	.1856±.0103	.2823±.0170	.2231±.0127	9.10±0.44	1.50±0.00×10^{1}	.0609±.0014	.1580±.0034	.0938±.0020	0.09±0.00	1.10±0.00
BRIGHT	.2715±.0007	.5938±.0015	.3789±.0007	3.20±0.03×10^{2}	1.10±0.00×10^{1}	.0904±.0025	.2566±.0045	.1443±.0010	0.33±0.03	1.00±0.00
NeXtAlign	.2695±.0098	.5981±.0095	.3790±.0104	2.60±0.03×10^{3}	1.30±0.10×10^{1}	.0861±.0048	.2580±.0029	.1466±.0045	0.11±0.04	1.20±0.00
WLAlign	.2349±.0006	.4122±.0006	.2911±.0001	2.90±0.26×10^{3}	2.70±0.00×10^{1}	.0404±.0048	.1536±.0051	.0778±.0037	2.40±0.86×10^{2}	1.40±0.00
PARROT	.6334±.0000	.8528±.0000	.7074±.0000	8.70±0.10×10^{2}	2.80±0.00×10^{1}	.0993±.0000	.2848±.0000	.1655±.0000	0.90±0.00	0.83±0.00
SLOTAlign	.0188±.0000	.0725±.0000	.0381±.0000	3.06±0.01×10^{4}	2.80±0.00×10^{1}	.0149±.0000	.0613±.0000	.0334±.0000	0.12±0.00	1.00±0.20
HOT	.3143±.0038	.5832±.0058	.2006±.0020	2.10±0.08×10^{3}	2.00±0.03×10^{1}	.0639±.0114	.2408±.0036	.0586±.0046	1.10±0.10×10^{1}	2.10±0.00
JOENA	.6476±.0037	.8496±.0034	.7170±.0037	6.70±0.08×10^{2}	2.60±0.00×10^{1}	.1010±.0049	.2930±.0161	.1697±.0061	0.06±0.00	1.00±0.10
Dataset	**Douban**					**Flickr-LastFM**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
IsoRank	.1351±.0003	.5179±.0004	.2517±.0001	0.59±0.06	0.95±0.03	.0028±.0000	.0815±.0000	.0305±.0000	9.20±0.22×10^{1}	8.13±0.21
FINAL	.1458±.0000	.5676±.0000	.2692±.0000	0.34±0.03	1.14±0.04	.0028±.0000	.0732±.0000	.0253±.0000	1.56±0.04×10^{1}	1.29±0.02×10^{1}
IONE	.2777±.0046	.6312±.0098	.3936±.0040	9.08±1.57×10^{3}	0.89±0.04	.0113±.0033	.0437±.0044	.0239±.0036	4.09±1.41×10^{3}	3.77±0.21
REGAL	.0025±.0009	.0198±.0034	.0105±.0006	2.57±0.19	0.82±0.00	.0042±.0026	.0340±.0062	.0177±.0023	1.78±0.06×10^{1}	1.65±0.16
CrossMNA	.0187±.0044	.3241±.0054	.1172±.0049	6.34±0.24×10^{1}	0.78±0.01	.0061±.0008	.0091±.0030	.0078±.0011	3.18±3.40×10^{1}	2.46±0.01
NetTrans	.2030±.0020	.6018±.0020	.3291±.0015	1.54±0.44	1.92±0.40	.0052±.0012	.0204±.0018	.0115±.0009	2.75±2.01×10^{1}	2.27±1.15×10^{1}
WAlign	.1480±.0018	.2381±.0032	.1834±.0019	0.20±0.06	1.31±0.11	.0094±.0011	.0423±.0021	.0290±.0010	0.55±0.03	4.51±0.08
BRIGHT	.1202±.0007	.4361±.0031	.2218±.0013	2.64±0.07	1.14±0.01	.0259±.0015	.0492±.0037	.0357±.0010	5.44±0.14×10^{1}	6.14±0.04
NeXtAlign	.2154±.0062	.5701±.0144	.3305±.0084	1.86±0.08	1.66±0.00	.0260±.0030	.0541±.0075	.0375±.0023	6.11±0.29×10^{1}	6.85±0.07
WLAlign	.2028±.0021	.3505±.0019	.2517±.0015	6.58±0.41×10^{2}	0.95±0.02	.0080±.0027	.0224±.0012	.0141±.0018	5.37±3.57×10^{2}	3.21±0.02
PARROT	.3469±.0000	.6832±.0000	.4563±.0000	2.73±0.71	1.31±0.02	.0276±.0000	.0608±.0000	.0417±.0000	2.51±0.08×10^{2}	1.51±0.02×10^{1}
SLOTAlign	.0000±.0000	.0078±.0000	.0048±.0000	7.01±0.32	1.40±0.03	.0041±.0000	.0235±.0000	.0145±.0000	8.44±4.05	1.54±0.04×10^{1}
HOT	.1509±.0059	.4600±.0131	.1545±.0026	3.95±1.54×10^{1}	1.37±0.08	.0091±.0037	.0157±.0030	.0063±.0015	1.01±0.02×10^{3}	7.97±0.18
JOENA	.4457±.0038	.8091±.0026	.5657±.0022	0.41±0.06	1.19±0.03	.0290±.0000	.0994±.0062	.0558±.0023	2.02±0.07×10^{2}	1.41±0.02×10^{1}
Dataset	**Flickr-MySpace**					**Arenas**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
IsoRank	.0047±.0000	.0374±.0000	.0225±.0002	2.39±2.35	3.47±0.08	.3981±.0000	.6751±.0000	.4940±.0000	0.09±0.00	1.02±0.63
FINAL	.0000±.0000	.0093±.0000	.0048±.0000	6.74±0.28	5.21±0.01	.3849±.0000	.8811±.0000	.5523±.0000	0.11±0.01	1.13±0.63
IONE	.0056±.0027	.0266±.0058	.0170±.0023	1.19±0.64×10^{3}	1.88±0.07	.9130±.0069	.9763±.0042	.9393±.0060	1.39±0.14×10^{4}	1.25±0.63
REGAL	.0061±.0013	.0322±.0042	.0173±.0017	9.36±0.10	1.06±0.14	.9053±.0052	.9803±.0008	.9354±.0034	1.04±0.02	0.94±0.11
CrossMNA	.0070±.0037	.0107±.0021	.0090±.0032	2.93±3.03×10^{1}	1.36±0.01	.4261±.0175	.8492±.0120	.5812±.0158	5.61±0.10×10^{1}	1.00±0.00
NetTrans	.0000±.0000	.0009±.0013	.0010±.0002	5.42±3.96	7.38±3.86	.8827±.0013	.9983±.0004	.9348±.0010	0.74±0.05	2.22±0.43
WAlign	.0056±.0013	.0444±.0016	.0240±.0008	0.40±0.05	2.49±0.04	.9723±.0003	.9974±.0003	.9837±.0002	0.12±0.01	1.34±0.15
BRIGHT	.0037±.0027	.0154±.0043	.0089±.0012	2.08±0.04×10^{1}	3.32±0.07	.9700±.0005	.9954±.0003	.9817±.0003	0.38±0.01	1.29±0.03
NeXtAlign	.0047±.0017	.0140±.0024	.0089±.0014	2.16±0.06×10^{1}	4.20±0.01	.5347±.0012	.7269±.0031	.6034±.0089	0.28±0.02	1.11±0.08
WLAlign	.0000±.0000	.0070±.0055	.0043±.0021	5.67±3.01×10^{2}	1.51±0.02	.6051±.0022	.7152±.0010	.6491±.0009	6.47±0.89×10^{2}	0.89±0.07
PARROT	.0093±.0000	.0164±.0000	.0114±.0000	3.73±0.08×10^{1}	6.04±0.06	.9780±.0000	.9999±.0000	.9886±.0000	0.88±0.05	1.10±0.63
SLOTAlign	.0021±.0000	.0135±.0000	.0075±.0000	9.78±0.13	6.31±0.04	.9785±.0000	.9999±.0000	.9889±.0000	0.12±0.00	0.86±0.01
HOT	.0014±.0013	.0033±.0039	.0005±.0005	4.46±0.44×10^{2}	4.37±0.23	.9653±.0048	.9958±.0020	.4901±.0020	8.78±1.28	8.19±1.36
JOENA	.0071±.0000	.0476±.0035	.0261±.0006	4.49±0.07	1.79±0.11	.9795±.0006	.9999±.0000	.9894±.0003	0.22±0.35	1.01±0.00

Table 8: Detailed effectiveness results (Part II of II) on plain networks with a training ratio of 20%. The 1st/2nd/3rd best results are marked in red/blue/green respectively. Time denotes the inference time and Mem. denotes the peak memory usage.

Dataset	Cora					ArXiv				
Metrics	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)
IsoRank	.1793±.0000	.5877±.0000	.3127±.0001	0.98±0.30	1.17±0.15	.2247±.0001	.5537±.0000	.3281±.0000	1.26±0.17×10^2	1.52±0.06×10^1
FINAL	.2065±.0000	.6442±.0000	.3488±.0000	0.25±0.01	1.38±0.15	.2551±.0000	.7250±.0000	.4102±.0000	1.61±0.02×10^1	2.22±0.06×10^1
IONE	.3415±.0080	.6611±.0079	.4536±.0078	1.34±0.04×10^4	1.08±0.15	.2688±.0086	.5195±.0093	.3517±.0091	1.66±0.17×10^4	8.79±0.57
REGAL	.0158±.0019	.0789±.0051	.0383±.0032	2.41±0.07	0.86±0.02	.0026±.0003	.0185±.0012	.0098±.0005	3.30±0.22×10^1	6.79±0.10
CrossMNA	.0358±.0045	.4574±.0080	.1740±.0052	5.80±0.17×10^1	0.89±0.01	.2875±.0015	.6742±.0007	.4115±.0012	2.87±0.06×10^3	6.68±0.06
NetTrans	.3703±.0023	.7238±.0017	.4891±.0020	1.38±0.28	1.88±0.26	.4359±.0015	.7831±.0006	.5503±.0010	1.43±0.11×10^3	8.84±0.09
WAlign	.4176±.0041	.5650±.0048	.4753±.0039	0.23±0.04	1.85±0.06	.2308±.0041	.3684±.0073	.2785±.0049	2.52±0.08×10^1	2.29±0.01×10^1
BRIGHT	.3839±.0019	.6966±.0025	.4934±.0013	3.61±0.05	1.88±0.02	.4216±.0006	.7263±.0010	.5270±.0007	3.06±0.02×10^2	1.13±0.00×10^1
NeXtAlign	.4096±.0106	.7212±.0087	.5192±.0087	3.07±0.12	1.67±0.00	.4189±.0067	.7461±.0098	.5291±.0023	2.91±0.07×10^2	1.78±0.02×10^1
WLAlign	.2754±.0011	.4398±.0010	.3349±.0002	7.02±0.43×10^2	1.25±0.03	.4873±.0007	.6593±.0010	.5425±.0006	5.98±0.08×10^3	2.77±0.00×10^1
PARROT	.6961±.0000	.8639±.0000	.7599±.0000	6.02±0.99	1.73±0.12	.7259±.0000	.9169±.0000	.7948±.0000	7.60±0.22×10^2	2.55±0.06×10^1
SLOTAlign	.6654±.0000	.7621±.0000	.7044±.0000	1.79±0.01	1.96±0.10	.3642±.3152	.4853±.4175	.4068±.3501	7.01±5.21×10^2	2.51±0.09×10^1
HOT	.4173±.0071	.6413±.0084	.2481±.0036	3.83±0.07×10^1	2.36±0.58	.3994±.0014	.6475±.0022	.2417±.0009	1.90±0.18×10^3	1.59±0.18×10^1
JOENA	.8238±.0033	.9212±.0005	.8646±.0021	0.50±0.01	1.00±0.04	.7578±.0011	.9370±.0006	.8271±.0004	5.63±0.50×10^1	4.19±0.00
Dataset	PPI					GGI				
Metrics	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)
IsoRank	.3622±.0000	.6175±.0000	.4462±.0000	2.19±0.25	1.18±0.02	.2512±.0000	.5064±.0000	.3372±.0000	3.59±0.09×10^1	5.33±0.18
FINAL	.4479±.0000	.8191±.0000	.5743±.0000	0.50±0.01	1.50±0.00	.1920±.0000	.6396±.0000	.3402±.0000	4.61±0.33	7.42±0.11
IONE	.8350±.0047	.9218±.0036	.8658±.0042	1.68±0.07×10^4	1.17±0.02	.5021±.0101	.6829±.0079	.5652±.0094	1.72±0.06×10^4	3.32±0.18
REGAL	.0110±.0024	.0724±.0047	.0330±.0029	4.83±0.12	0.94±0.00	.0171±.0011	.0780±.0038	.0387±.0017	1.36±0.01×10^1	2.52±0.04
CrossMNA	.8045±.0029	.9179±.0046	.8445±.0028	6.55±0.11×10^2	1.09±0.02	.3961±.0030	.6964±.0041	.5017±.0028	7.65±0.14×10^2	2.67±0.00
NetTrans	.5714±.0011	.8012±.0021	.6459±.0008	3.63±0.18×10^1	2.16±0.11	.4024±.0014	.6705±.0008	.4918±.0011	2.76±0.57×10^2	7.34±4.13
WAlign	.2912±.0037	.3877±.0041	.3264±.0038	1.78±0.06	2.37±0.03	.2978±.0090	.3791±.0092	.3282±.0088	5.33±0.24	7.32±0.07
BRIGHT	.4828±.0023	.6354±.0037	.5387±.0018	4.95±0.04	2.37±0.05	.3960±.0009	.5870±.0023	.4642±.0008	9.11±0.35×10^1	5.74±0.03
NeXtAlign	.4576±.0012	.6098±.0019	.5248±.0031	4.65±0.07	2.54±0.01	.2899±.0067	.5263±.0025	.3717±.0056	5.19±0.03×10^1	6.87±0.04
WLAlign	.7466±.0005	.8126±.0016	.7682±.0005	1.04±0.07×10^3	1.74±0.04	.4354±.0006	.5440±.0012	.4710±.0005	1.82±0.03×10^3	9.40±0.02
PARROT	.9619±.0000	.9926±.0000	.9731±.0000	1.80±0.04×10^1	1.58±0.03	.8203±.0000	.9373±.0000	.8621±.0000	2.30±0.09×10^2	8.47±0.03
SLOTAlign	.7398±.0000	.8219±.0000	.7684±.0000	3.98±0.29	1.64±0.02	.7170±.0003	.8097±.0017	.7500±.0011	1.98±0.04×10^2	8.15±0.03
HOT	.3822±.0024	.4667±.0031	.2052±.0011	5.66±0.77×10^1	6.70±0.00	.3466±.0020	.4724±.0023	.1931±.0011	8.67±4.82×10^2	5.13±0.44
JOENA	.9856±.0000	.9995±.0002	.9907±.0000	0.54±0.02	1.08±0.02	.8665±.0002	.9583±.0001	.9016±.0000	1.01±0.00×10^1	1.99±0.00
Dataset	DBP15K_JA-EN					DBP15K_FR-EN				
Metrics	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)
IsoRank	.1285±.0000	.5095±.0001	.2465±.0001	1.51±0.07×10^2	1.60±0.00×10^1	.1072±.0000	.5024±.0001	.2281±.0000	1.50±0.05×10^2	1.59±0.03×10^1
FINAL	.1437±.0000	.5580±.0000	.2784±.0000	1.82±0.03×10^1	2.46±0.00×10^1	.1414±.0000	.5701±.0000	.2781±.0000	1.81±0.01×10^1	2.45±0.03×10^1
IONE	.0405±.0037	.1864±.0108	.0903±.0058	1.55±0.02×10^4	8.83±0.00	.0366±.0010	.1719±.0066	.0836±.0027	1.46±0.01×10^4	8.72±0.32
REGAL	.0133±.0060	.0453±.0082	.0250±.0067	3.09±0.04×10^1	6.06±0.11	.0065±.0000	.0247±.0007	.0140±.0003	3.16±0.02×10^1	5.92±0.02
CrossMNA	.0179±.0013	.2932±.0042	.1008±.0020	1.01±0.03×10^3	6.18±0.04	.0321±.0010	.2888±.0032	.1121±.0013	1.32±0.04×10^3	6.23±0.04
NetTrans	.3044±.0011	.6373±.0015	.4103±.0010	2.96±0.70×10^2	2.14±0.56×10^1	.2975±.0003	.6457±.0007	.4080±.0004	2.98±0.82×10^2	1.43±0.45×10^1
WAlign	.2334±.0030	.3207±.0029	.2673±.0031	9.46±0.14	1.52±0.00×10^1	.1638±.0053	.2432±.0077	.1946±.0059	1.16±0.04×10^1	1.60±0.00×10^1
BRIGHT	.3264±.0019	.6255±.0078	.4267±.0011	3.91±0.84×10^2	1.17±0.00×10^1	.3143±.0010	.6313±.0018	.4203±.0004	4.31±0.29×10^2	1.19±0.01×10^1
NeXtAlign	.2866±.0402	.6001±.0340	.3920±.0387	2.66±0.80×10^3	1.43±nan×10^1	.2695±.0983	.5981±.0953	.3790±.1004	2.60±0.03×10^3	1.34±0.07×10^1
WLAlign	.2661±.0005	.4378±.0009	.3212±.0005	3.19±0.14×10^3	2.81±0.00×10^1	.2764±.0006	.4769±.0008	.3401±.0004	3.63±0.29×10^3	2.99±0.01×10^1
PARROT	.6453±.0000	.8600±.0000	.7164±.0000	9.13±0.09×10^2	2.87±0.00×10^1	.6999±.0000	.9038±.0000	.7697±.0000	9.01±0.22×10^2	2.87±0.03×10^1
SLOTAlign	.0063±.0015	.0294±.0054	.0157±.0029	2.17±0.12×10^2	2.84±nan×10^1	.0188±.0000	.0725±.0000	.0381±.0000	1.71±0.08×10^2	2.84±0.03×10^1
HOT	.3512±.0034	.6174±.0045	.2199±.0017	2.08±0.04×10^3	1.91±0.10×10^1	.3504±.0029	.6352±.0041	.2210±.0016	2.07±0.06×10^3	1.62±0.06×10^1
JOENA	.6278±.0029	.8631±.0047	.6989±.0036	6.42±0.33×10^2	2.67±0.00×10^1	.7133±.0053	.9210±.0033	.7739±.0041	6.72±0.33×10^2	2.67±0.03×10^1
Dataset	Airport					PeMS08				
Metrics	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)	Hits@1	Hits@10	MRR	Time(s)	Mem.(GB)
IsoRank	.1308±.0000	.3755±.0000	.2074±.0000	0.11±0.00	0.82±0.18	.2537±.0000	.7353±.0000	.4197±.0000	0.23±0.24	0.55±0.02
FINAL	.2117±.0000	.6318±.0000	.3520±.0000	0.16±0.02	0.92±0.18	.1985±.0000	.7574±.0000	.3917±.0000	0.01±0.00	0.64±0.00
IONE	.4809±.0054	.7073±.0078	.5575±.0063	1.46±0.10×10^4	1.03±0.19	.3824±.0252	.6728±.0351	.4833±.0202	7.43±4.40×10^3	0.84±0.02
REGAL	.0302±.0036	.1477±.0054	.0698±.0031	1.17±0.02	0.81±0.00	.0493±.0115	.2272±.0126	.1167±.0087	0.14±0.00	0.81±0.00
CrossMNA	.4304±.0061	.7186±.0178	.5241±.0081	1.40±0.02×10^2	0.98±0.00	.0066±.0048	.0640±.0178	.0336±.0082	0.10±0.00	0.98±0.00
NetTrans	.4293±.0034	.6817±.0013	.5154±.0029	1.80±0.51×10^1	1.47±0.29	.3985±.0141	.8794±.0066	.5621±.0077	0.77±0.44	1.26±0.00
WAlign	.2270±.0113	.4163±.0126	.2924±.0106	0.13±0.00	2.03±0.46	.5846±.0074	.8441±.0042	.6790±.0065	0.13±0.00	1.16±0.34
BRIGHT	.3495±.0026	.5667±.0089	.4282±.0028	0.47±0.02	1.37±0.03	.4566±.0060	.8882±.0056	.6123±.0046	0.03±0.00	1.04±0.00
NeXtAlign	.2946±.0078	.5273±.0012	.3748±.0039	0.71±0.02	1.64±0.03	.4596±.0054	.8676±.0123	.6139±.0087	0.04±0.00	1.17±0.04
WLAlign	.4761±.0018	.6167±.0011	.5252±.0014	7.06±0.71×10^2	1.33±0.00	.3676±.0069	.5125±.0072	.4203±.0029	3.28±0.55×10^2	1.33±0.00
PARROT	.6891±.0000	.8687±.0000	.7488±.0000	1.32±0.15	0.91±0.18	.7647±.0000	.9228±.0000	.8209±.0000	0.07±0.00	0.82±0.00
SLOTAlign	.6691±.0008	.7873±.0010	.7098±.0012	9.89±0.32	1.12±0.02	.9118±.0000	.9743±.0000	.9280±.0000	1.54±0.18	0.96±0.00
HOT	.2889±.0124	.4648±.0118	.1737±.0059	1.54±0.19×10^1	2.17±0.17	.6103±.0355	.8566±.0045	.3529±.0134	0.44±0.10	2.10±0.00
JOENA	.8459±.0026	.9637±.0011	.8887±.0012	0.07±0.00	1.03±0.00	.9390±.0134	.9941±.0020	.9568±.0101	0.03±0.00	1.04±0.00

Table 9: Detailed effectiveness results on attributed networks with a training ratio of 20%. The 1st/2nd/3rd best results are marked in red/blue/green respectively. Time denotes the inference time and Mem. denotes the peak memory usage.

Dataset	Douban					Flickr-LastFM				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
FINAL	.5397±.0012	.9838±.0013	.6999±.0032	0.90±0.01	0.87±0.02	.0152±.0002	.1022±.0004	.0422±.0000	3.66±0.80	3.70±0.00
REGAL	.0352±0003	.1525±.0012	.0758±.0006	3.20±0.08	2.55±0.03	.0086±.0020	.0580±.0026	.0283±.0013	$2.63\pm0.22\times10^{1}$	2.70±0.27
NetTrans	.3274±.0006	.6145±.0002	.4226±.0004	1.51±0.06	1.23±0.00	.0041±.0001	.0401±.0012	.0201±.0021	$3.15\pm0.51\times10^{1}$	$2.41\pm0.15\times10^{1}$
WAlign	.2855±.0012	.5698±.0027	.3798±.0023	0.03±0.00	1.25±0.01	.0169±.0006	.0710±.0029	.0416±.0008	2.30±0.05	4.62±0.05
BRIGHT	.2813±.0075	.6095±.0119	.3966±.0083	1.52±0.45	1.45±0.04	.0345±.0031	.1019±.0059	.0602±.0022	$6.15\pm0.57\times10^{1}$	3.80±0.00
NeXtAlign	.1879±.0045	.4918±.0012	.2756±.0022	1.48±0.01	1.47±0.00	.0083±.0038	.0384±.0141	.0214±.0070	2.04±0.26	2.00±0.10
PARROT	.6413±.0000	.9408±.0000	.7481±.0000	0.25±0.02	1.34±0.02	.0442±.0000	.1064±.0000	.0701±.0000	$5.13\pm0.02\times10^{1}$	1.86±0.00
SLOTAlign	.4397±.0000	.7207±.0000	.5394±.0000	0.01±0.00	1.31±0.01	.0055±.0000	.0387±.0000	.0205±.0000	$2.58\pm0.02\times10^{1}$	$1.63\pm0.02\times10^{1}$
HOT	.3223±.0032	.6391±.0174	.2578±.0064	8.76±0.21	1.41±0.00	.0152±.0013	.0525±.0003	.0141±.0021	$4.63\pm0.17\times10^{2}$	8.12±0.19
JOENA	.6542±.0474	.9173±.0205	.7525±.0446	0.24±0.02	1.37±0.00	.0345±.0011	.1064±.0034	.0600±.0003	$1.89\pm0.21\times10^{1}$	2.63±0.02
Dataset	**Flickr-MySpace**					**Arenas**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
FINAL	.0023±.0000	.0234±.0000	.0113±.0000	1.57±0.31	2.33±0.00	.4284±.0000	.9069±.0000	.5928±.0000	0.61±0.24	0.88±0.01
REGAL	.0065±.0030	.0355±.0061	.0197±.0040	$1.45\pm0.12\times10^{1}$	1.65±0.07	.8961±.0255	.9815±.0030	.9281±.0179	2.02±0.55	1.32±0.12
NetTrans	.0075±.0019	.0374±.0052	.0201±.0014	2.58±0.09	1.32±0.00	.9581±.0036	.9879±.0013	.9688±.0026	1.03±0.05	1.17±0.00
WAlign	.0112±.0026	.0505±.0054	.0316±.0021	1.96±0.58	2.65±0.07	.9808±.0003	.9978±.0000	.9886±.0002	1.75±1.05	2.19±0.07
BRIGHT	.0061±.0021	.0332±.0019	.0196±.0012	$3.50\pm0.30\times10^{1}$	2.38±0.00	.9794±.0005	.9950±.0006	.9863±.0004	0.51±0.21	1.05±0.00
NeXtAlign	.0037±.0027	.0243±.0173	.0162±.0075	0.91±0.04	1.48±0.03	.6684±.2300	.8230±.1401	.7244±.1974	0.47±0.33	1.09±0.04
PARROT	.0070±.0000	.0397±.0000	.0223±.0000	$1.01\pm0.02\times10^{1}$	1.40±0.00	.9879±.0000	.9999±.0000	.9936±.0000	$2.56\pm1.51\times10^{1}$	0.77±0.01
SLOTAlign	.0023±.0000	.0257±.0000	.0163±.0000	$1.03\pm0.01\times10^{1}$	7.51±0.10	.9891±.0010	.9999±.0000	.9942±.0005	0.13±0.00	0.88±0.01
HOT	.0000±.0000	.0257±.0000	.0024±.0000	$3.26\pm0.05\times10^{2}$	2.71±0.00	.9714±.0026	.9927±.0015	.4909±.0009	4.20±0.62	1.15±0.00
JOENA	.0117±.0001	.0584±.0000	.0345±.0000	4.15±0.02	1.71±0.01	.9873±.0002	.9999±.0000	.9929±.0000	0.34±0.00	1.02±0.04
Dataset	**ACM-DBLP**					**Cora**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
FINAL	.4054±.0000	.7980±.0000	.5366±.0000	2.56±0.64	1.19±0.06	.7891±.0000	.9026±.0000	.8332±.0000	1.48±0.13	1.04±0.01
REGAL	.4159±.0035	.6373±.0064	.4890±.0040	$1.95\pm0.12\times10^{1}$	$1.78\pm0.05\times10^{1}$	.3576±.0278	.4705±.0271	.3986±.0242	5.15±1.27	6.84±0.21
NetTrans	.6874±.0015	.9300±.0019	.7716±.0012	$2.64\pm2.18\times10^{2}$	1.48±0.02	.7907±.4417	.8005±.4454	.7955±.4428	1.47±0.07	1.32±0.00
WAlign	.6675±.0024	.9109±.0012	.7524±.0016	6.82±1.00	4.85±0.03	.9551±.0012	.9724±.0010	.9621±.0011	3.38±1.28	2.36±0.05
BRIGHT	.4858±.0025	.8740±.0020	.6163±.0021	$1.22\pm0.53\times10^{2}$	2.35±0.00	.7989±.0051	.9902±.0012	.8813±.0029	6.07±3.03	1.32±0.04
NeXtAlign	.3512±.0966	.7633±.0872	.4851±.0981	$1.45\pm0.64\times10^{1}$	1.47±0.15	.3336±.0418	.6629±.0291	.4444±.0376	1.56±0.22	1.15±0.02
PARROT	.6867±.0000	.9437±.0000	.7770±.0000	$1.70\pm0.40\times10^{1}$	1.27±0.00	.9654±.0000	.9684±.0000	.9667±.0000	1.13±0.56	1.05±0.02
SLOTAlign	.6673±.0011	.8720±.0003	.7409±.0009	$8.34\pm0.04\times10^{1}$	7.45±0.16	.9949±.0000	.9999±.0000	.9974±.0000	1.92±0.13	2.01±0.10
HOT	.3893±.0050	.6180±.0055	.2350±.0017	$4.28\pm0.13\times10^{2}$	2.85±0.01	.7493±.0040	.7549±.0040	.3762±.0021	$2.17\pm0.59\times10^{1}$	3.29±0.28
JOENA	.7859±.0053	.9847±.0101	.8569±.0120	9.78±0.27	1.59±0.11	.9947±.0002	.9999±.0000	.9966±.0001	0.38±0.03	1.14±0.02
Dataset	**PPI**					**DBP15K_FR-EN**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
FINAL	.0316±.0000	.1564±.0000	.0760±.0000	1.87±0.28	0.93±0.00	.1954±.0000	.4381±.0000	.2800±.0000	$4.36\pm0.10\times10^{2}$	$2.37\pm0.00\times10^{1}$
REGAL	.1515±.0080	.2910±.0072	.2012±.0078	9.30±0.42	3.89±0.12	.0027±.0004	.0038±.0004	.0035±.0004	$5.30\pm0.18\times10^{1}$	$9.33\pm0.03\times10^{1}$
NetTrans	.7356±.0027	.8581±.0024	.7783±.0027	$2.39\pm0.50\times10^{1}$	1.22±0.00	.4308±.0012	.7789±.0081	.5412±.0109	$1.38\pm1.03\times10^{1}$	$1.03\pm0.01\times10^{1}$
WAlign	.7657±.0021	.8733±.0007	.8041±.0009	6.84±1.54	3.15±0.01	.5077±.0034	.6492±.0046	.5572±.0037	$2.03\pm0.26\times10^{1}$	$1.67\pm0.00\times10^{1}$
BRIGHT	.7156±.0030	.8623±.0036	.7691±.0026	8.76±3.53	1.29±0.01	.4393±.0029	.8020±.0023	.5652±.0029	$2.48\pm0.05\times10^{2}$	9.55±0.03
NeXtAlign	.0306±.0142	.1059±.0517	.0579±.0266	$1.41\pm0.84\times10^{1}$	1.25±0.10	.4781±.0023	.8780±.0012	.6109±.0032	$2.78\pm0.08\times10^{2}$	9.87±0.01
PARROT	.9916±.0001	.9977±.0000	.9943±.0001	$5.55\pm3.14\times10^{1}$	1.58±0.02	.8737±.0000	.9550±.0000	.9040±.0000	$1.60\pm0.22\times10^{3}$	$2.76\pm0.00\times10^{1}$
SLOTAlign	.9531±.0000	.9777±.0000	.9618±.0000	3.70±0.01	1.71±0.00	.7619±.0012	.8721±.0029	.8091±.0012	$1.70\pm0.12\times10^{2}$	$2.18\pm0.00\times10^{1}$
HOT	.6705±.0032	.7154±.0054	.3439±.0018	$3.84\pm0.99\times10^{1}$	1.30±0.00	.6172±.0134	.7980±.0012	.6512±.0012	$1.70\pm0.12\times10^{3}$	$2.17\pm0.00\times10^{1}$
JOENA	.9804±.0012	.9943±.0041	.9857±.0024	0.73±0.12	1.10±0.00	.9804±.0012	.9943±.0041	.9857±.0024	0.73±0.12	1.10±0.00
Dataset	**Airport**					**PeMS08**				
Metrics	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**	**Hits@1**	**Hits@10**	**MRR**	**Time(s)**	**Mem.(GB)**
FINAL	.3897±.0000	.8267±.0000	.5414±.0000	0.47±0.15	0.89±0.00	.2132±.0000	.7610±.0000	.4051±.0000	0.02±0.02	0.90±0.00
REGAL	.0516±.0037	.2096±.0125	.1063±.0059	2.04±0.48	1.38±0.10	.3617±.0321	.5655±.0360	.4283±.0338	0.22±0.09	1.20±0.00
NetTrans	.2430±.0038	.4667±.0034	.3192±.0028	2.32±0.33	1.18±0.00	.6353±.0048	.8963±.0048	.7127±.0022	0.07±0.04	1.18±0.00
WAlign	.3041±.0054	.5803±.0049	.3987±.0059	1.50±0.98	2.33±0.04	.6963±.0118	.9066±.0076	.7689±.0070	2.15±0.41	2.20±0.01
BRIGHT	.3343±.0062	.5671±.0028	.4153±.0044	1.29±0.56	1.06±0.00	.5280±.0166	.8765±.0159	.6524±.0074	0.53±1.04	1.06±0.00
NeXtAlign	.0792±.0561	.2551±.1263	.1398±.0785	0.46±0.10	1.02±0.00	.2992±.0573	.7162±.1283	.4373±.0657	0.04±0.04	0.99±0.00
PARROT	.8582±.0000	.9622±.0000	.8971±.0000	$2.28\pm1.91\times10^{1}$	0.92±0.02	.8787±.0000	.9596±.0000	.9071±.0000	8.31±4.67	0.85±0.00
SLOTAlign	.8918±.0000	.9816±.0000	.9256±.0000	0.15±0.01	0.98±0.01	.9853±.0000	.9999±.0000	.9926±.0000	0.01±0.00	0.90±0.00
HOT	.4318±.0052	.5770±.0116	.2413±.0027	5.33±0.24	1.16±0.00	.7324±.0175	.8853±.0120	.3921±.0063	0.35±0.04	1.16±0.00
JOENA	.8734±.0034	.9748±.0104	.9117±.0030	0.10±0.00	1.02±0.00	.9999±.0000	.9999±.0000	.9999±.0000	0.05±0.00	1.02±0.01