Table 1: Comparison of test results when using Hits at 10 (H@10) vs. MRR metric for *TKG Link Prediction* task.

Method	tkgl-smallpedia		tkgl-polecat		tkgl-wikidata	
Test Score	MRR	H@10	MRR	H@10	MRR	H@10
$\text{EdgeBank}_{\text{tw}}$	0.353	0.566	0.056	0.119	0.535	0.596
EdgeBank_{∞}	0.333	0.562	0.045	0.094	0.535	0.596
$\text{RecurrencyBaseline}_{\text{train}}$	**0.655**	**0.723**	0.198	0.317	-	-
$\text{RecurrencyBaseline}_{\text{default}}$	0.570	0.713	0.167	0.264	-	-
RE-GCN	0.594	0.687	0.175	0.292	-	-
CEN	0.612	0.705	0.184	0.323	-	-
TLogic	0.595	0.707	**0.228**	**0.378**	-	-

Additional Evaluation Details and Results

In addition to the MRR, in Table 1, we report the Hits at 10 (H@10), the proportion of test queries for which at least one correct item is among the top 10 ranked results. The results show that the ranking of methods in Hits@10 and MRR metric are closely matched.

Details on Evaluation Protocol

We compute the time-aware *Mean Reciprocal Rank*. Specifically, for each test edge $(s_{test}, r_{test}, o_{test}, t_{test})$, we evaluate the models prediction by removing the object o_{test} from the test query, $(s_{test}, r_{test}, ?, t_{test})$. The model then assigns scores to all possible entities for the object position, which are subsequently ranked in descending order. This is repeated for subject prediction. The MRR is calculated as the mean of the reciprocal of these ranks across all test queries. In applying the *time-aware filter setting*, we filter out quadruples that have the same timestamp as the test query. For example, for a test query (France, wins, Basketball Match, 2025-05-19) a model's prediction (France, wins, Soccer Match, 2025-05-19) would be excluded if (France, wins, Soccer Match, 2025-05-19) is present in the test set. However, it would not be filtered out if (France, wins, Soccer Match, 2025-03-09) is present in the test set. We evaluate all methods in *single-step prediction*. This means that the model always forecasts the next timestep, and then ground truth facts are fed before predicting the subsequent timestep. If there are multiple triples in the candidate set with the same score from the model, our protocol assigns the average rank, i.e. the average between the worst (pessimistic) and best (optimistic) score, to the ground truth triple. The scripts for generating the negative samples can be found here under the folder for each individual dataset (the scripts all have suffix of `ns_gen.py`).

Data Processing Details

Dataset Format. For all TKG datasets, we include the temporal links in `edgelist.csv`. The validation and test negative samples are included in `val_ns.pkl`and `test_ns.pkl` files respectively. Additionally for `tkgl-smallpedia` and `tkgl-wikidata`, we also include the static links related to the wiki entities in `static_edgelist.csv`. For all THG datasets, we include the temporal links in `edgelist.csv`. The validation and test negative samples are included in `val_ns.pkl` and `test_ns.pkl` files respectively. We also include `nodemapping.csv` and `edgemapping.csv` to provide the name of each node relation and edge relation respectively. Lastly, `nodetype.csv` provides the description for node type information.

`tkgl-wikidata` (scripts). This TKG dataset is extracted from the Wikidata KG from 2024 February 20th. We extract the relations from the first 32 million Wikidata entities (by the entity ID). We then retain the relations with the temporal qualifier with relation IDs: `P585`,`P580`, `P582`, `P577`, `P574`. We also retain the static relations with these wiki entities, filtering out less informative relations include `P31` and `P279`. We then keep only links with both start and end date as well as point in time relations. We retain links starting from 0 BC. We also remove any links that show up as both static and temporal link.

`tkgl-smallpedia` (scripts). This is a subset of the `tkgl-wikidata` where links from the first 1 million entities are retained and then filtered by their year. Only links from 1900 to 2024 are kept. We also extract the static relations associated with these first 1 million entities. Further, we remove any links that show up as both static and temporal link.

`tkgl-polecat` (scripts). We extract the raw files from the POLECAT dataset source, we organize the monthly edgelist chronologically and then merge them into a single file. Then, we remove any links with missing source or destination.

`tkgl-icews` (scripts). We extract the raw files from ICEWS Coded Event Data source. We first combine the monthly files into a single one, removing any links missing a source or destination.

`thgl-github` (scripts). The raw data is extracted from GH Arxiv, containing the data from March 2024. We then extract 14 relations based on the common activities on Github. We also removed the `issue comment` and `pr review comment` node types as they are often one time nodes that rarely repeats and kept nodes that has at least two edges in the dataset.

`thgl-software` (scripts). The raw data is extracted from GH Arxiv, containing the data from January 2024. We kept nodes with at least 10 interactions while removing the `issue comment` and `pr review comment` node types.

`thgl-forum` (scripts). The raw data is downloaded from here. We first merge edge files and attribute files by edge ID. Next, we filter out nodes with less than 100 links from the dataset. The node types are user nodes and subreddit nodes. The edge types are user replying to user and user posting on subreddit.

`thgl-myket` (scripts). This is an original dataset provided by the Myket Corporation. The data indicates if an edge is a software upgrade or a first time install which forms the two edge types in this data. The node types are users and apps.