Table 7: Comparison of learning-based GED computation approaches.

Approach	Graph type	Semantic-aware	Time complexity
SIMGNN [5]	Node-labeling	×	$O(n+m)$
TaGSim [4]	Node & edge-labeling	×	$O(n+m)$
GEDGNN [58]	Node-labeling	×	$O(n^2)$
GPN [87]	Node-labeling	×	$O(n+m)$
GEDIOT [13]	Node-labeling	×	$O(n+m)$
SEABED (**Ours**)	Knowledge graph	✓	$O(n+m)$

★ Note: n and m are the numbers of vertices and edges in the graph, respectively.

A WORKFLOW OF SEABED

Algorithm 1: The workflow of SEABED

input : A KG pair $(\mathcal{G}_1, \mathcal{G}_2)$
output: The prediction GED over $\mathcal{G}_1$ and $\mathcal{G}_2$
// ❶ initial embedding extraction
$X_{\mathcal{G}_1}, X_{\mathcal{G}_2} \leftarrow \texttt{InitialEmb}(\mathcal{G}_1, \mathcal{G}_2)$;
// ❷ local semantic aligment
// 2.1 Semantics-driven graph partition
$\mathcal{S}_1, \cdots, \mathcal{S}_k \leftarrow \texttt{GraphPartition}(X_{\mathcal{G}_1}, X_{\mathcal{G}_2}, \mathcal{G}_1, \mathcal{G}_2)$;
// 2.2 Adaptive local alignment
$\mathcal{I}_{\mathcal{G}} \leftarrow \texttt{LocalAlignment}(\mathcal{S}_1, \cdots, \mathcal{S}_k)$;
// ❸ Global-semantic estimator
$\widehat{d}(\mathcal{G}_1, \mathcal{G}_2) \leftarrow \texttt{SemanticEstimator}(\mathcal{I}_{\mathcal{G}}, X_{\mathcal{G}_1}, X_{\mathcal{G}_2})$;
return $\widehat{d}(\mathcal{G}_1, \mathcal{G}_2)$

In this section, we provide the workflow of SEABED, and compare the differences of learning-based GED estimation methods in Table 7. We can see that a notable feature of SEABED is its semantic-awareness: it captures both local and global semantic information, as well as structural differences, making it truly KG-native.

A.1 Additional proof

Lemma 4.3. For two KGs, $\mathcal{G}_1$=($\mathcal{V}_1, \mathcal{E}_1$) and $\mathcal{G}_2$=($\mathcal{V}_2, \mathcal{E}_2$), and the GED between the two KGs is 0. For any motif $\mathcal{M}$, let $\Psi_{\mathcal{M}}(u, \mathcal{G})$ denote the number of motifs in $\mathcal{G}$ containing u. Then, if $u \in \mathcal{V}_1$ and $v \in \mathcal{V}_2$ are in the same orbit, we have $\Psi_{\mathcal{M}}(u, \mathcal{G}_1) = \Psi_{\mathcal{M}}(v, \mathcal{G}_2)$.

Proof. Suppose, for the sake of contradiction, that $\Psi_{\mathcal{M}}(u, \mathcal{G}_1) \neq \Psi_{\mathcal{M}}(v, \mathcal{G}_2)$, even though $u \in \mathcal{V}_1$ and $v \in \mathcal{V}_2$ are in the same graph orbit and the GED between $\mathcal{G}_1$ and $\mathcal{G}_2$ is 0. Since the GED is 0, $\mathcal{G}_1$ and $\mathcal{G}_2$ are graph isomorphic, meaning there exists a bijection $\Pi : \mathcal{V}_1 \to \mathcal{V}_2$ such that for every $u \in \mathcal{V}_1$, $\Pi(u) = v$. We note that graph isomorphism preserves all structural properties, including the number of motifs of any type containing a given vertex. Therefore, we must have $\Psi_{\mathcal{M}}(u, \mathcal{G}_1) = \Psi_{\mathcal{M}}(\Pi(u), \mathcal{G}_2) = \Psi_{\mathcal{M}}(v, \mathcal{G}_2)$. If $\Psi_{\mathcal{M}}(u, \mathcal{G}_1) \neq \Psi_{\mathcal{M}}(v, \mathcal{G}_2)$, this would violate the structural invariance required by isomorphism, and thus such an isomorphism cannot exist. Thus, the lemma holds. □

B MORE DISCUSSIONS

B.1 P-value

In this subsection, we present how to compute the *p-value* using a well-known statistical test to compare the performance of two methods, denoted as **Method A** and **Method B**. We adopt the *Welch's two-sample t-test*, which does not assume equal variances between the two groups.

The test statistic t is computed as:

$$t = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}},$$

where $\bar{X}_A$ and $\bar{X}_B$ are the sample means of Method A and Method B, s_A^2 and s_B^2 are their sample variances, and n_A and n_B are the numbers of independent runs for each method, respectively.

Afterwards, the degrees of freedom ν are approximated using the **Satterthwaite formula**:

$$\nu = \frac{\left(\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}\right)^2}{\frac{(s_A^2/n_A)^2}{n_A - 1} + \frac{(s_B^2/n_B)^2}{n_B - 1}}.$$

Given the statistic t and degrees of freedom ν, the *two-tailed p-value* is obtained as:

$$p = 2 \cdot P(T_\nu \geq |t|),$$

where T_ν denotes a t-distribution with ν degrees of freedom.

We adopt the standard significance threshold $\alpha = 0.05$ to determine whether the performance difference between the two methods is statistically significant.

B.2 Failure Cases of SEABED

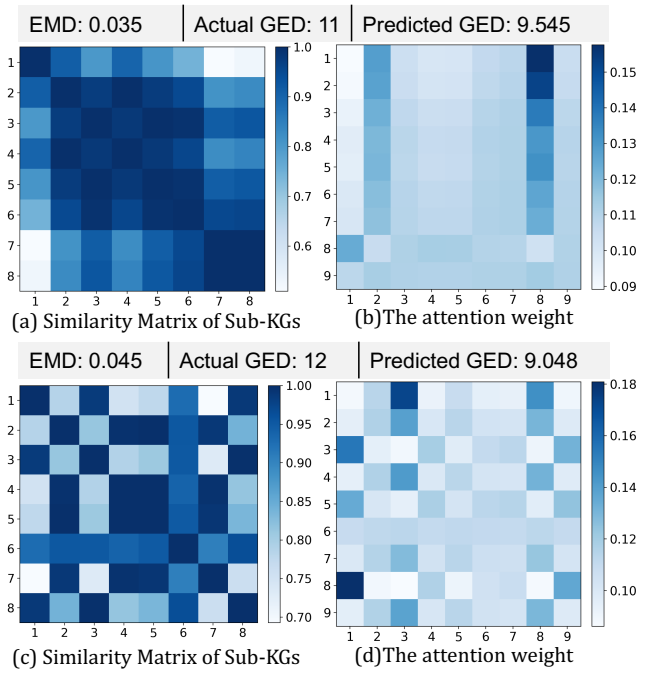


Figure 8: Two failure cases of SEABED.

Here, we present two failure cases in which the KG pairs exhibit high global semantic similarity (i.e., very small EMD values). The first case is shown in Figures 8(a)–(b), and the second case in Figures 8(c)–(d). In both examples, the two KGs share strong global semantic similarity, but their local sub-KG similarities differ: the first pair shows high local similarity, whereas the second pair

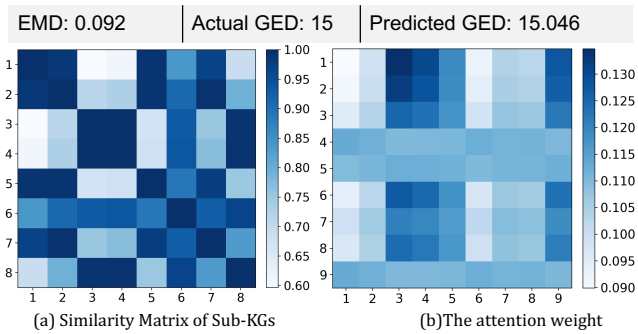
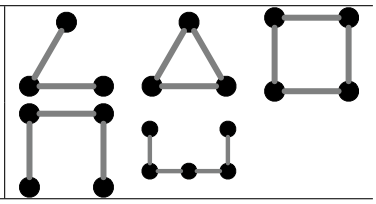


Figure 9: The attention weight matrix produced by SEABED when it performs well on estimating the GED between two KGs.

Table 8: Structures of used motifs.

Motif Structure	

shows clear local semantic differences. We note that for all cases SEABED underestimates the true GED in both cases. More specifically, in the first case, the similarity matrix across all sub-KG pairs is highly consistent. Consequently, the attention weights produced by the local semantic alignment module in SEABED become almost identical across sub-KG pairs. That is, the model is difficult to identify meaningful local distinctions and leads SEABED to predict an overly small GED. In the second case, although the local semantic structures differ substantially (as shown in Figure 8(c)) and these discrepancies are correctly captured by SEABED (see the attention weights in Figure 8(d)), the model still underestimates the true GED. This occurs because the global semantic signal dominates the final prediction. The above cases highlight a key limitation: when two KGs exhibit strong global similarity, the global signal may dominate the final prediction, preventing the local semantic differences from fully influencing the estimated GED and causing SEABED to underestimate the true distance.

B.3 Integrate SEABED into the graph query processing

We would like to emphasize that the GED over KGs is a fundamental metric for measuring structural similarity, which plays a critical role in graph query processing. Specifically, given a query graph G_q, the graph query task aims to retrieve all graphs in the dataset whose GED with respect to G_q is smaller than a specified threshold k. Our proposed method, SEABED, can be integrated into existing KG processing systems by replacing their original GED estimation component. In other words, the system needs to substitute its built-in GED computation module with the GED values estimated by SEABED. In practice, SEABED requires an offline stage to collect training data and train the model; once trained, it can be directly deployed in the online query phase to provide accurate and efficient GED estimation during KG retrieval.

B.4 The structure of motif

In Table 8, we present the structure of motifs used in SEABED.

B.5 Attention weights

In Figure 9, we illustrate how the self-attention mechanism works. Figure 9(a) shows the similarity matrix across all sub-KG pairs, where the first four sub-KGs belong to one KG and the remaining four belong to the other when computing their GED. Figure 9(b) presents the corresponding attention weights learned by SEABED. We observe that for two KGs with large global differences (i.e., high EMD values), the attention scores vary significantly across sub-KG pairs, indicating that the model selectively focuses on locally mismatched regions. This demonstrates that SEABED effectively captures local semantic discrepancies between the two KGs, which contributes to its strong overall performance.

C ADDITIONAL EXPERIMENTS

C.1 Setup

C.1.1 Dataset. SWDF [51] contains a smaller number of triples, and has a high number of interconnections between the terms. LUBM is a widely used RDF benchmark [16, 24], for which we use a scaling factor of 20 by following the existing works [16, 65]. YAGO is a large KG with general knowledge about people, cities, countries, movies, and organizations that are widely used in the existing works [57, 65], and we utilize the dataset from G-CARE [57] in our paper. WIKIDATA is a large cross-domain KGs, provided by Wikipedia, and this dataset is also sourced from GNCE [65].

C.2 Comparison with the Existing Methods

C.2.1 Hyperparameter settings. In this section, we present the detailed selection of hyperparameters in SEABED. Specifically, we use a 2-layer GNN as the backend for the network, where the initial embedding dimensions for both entities and predicates are set to 100. The number of subgraphs is set to 8, and the graph-level embedding dimension is set to 32. For model optimization, we use the Adam optimizer [37] with a learning rate of 0.001 and a weight decay of 1×10^{-4}. The control parameter λ for the EMD loss is set to 0.01. All machine learning models are trained for 15 epochs.

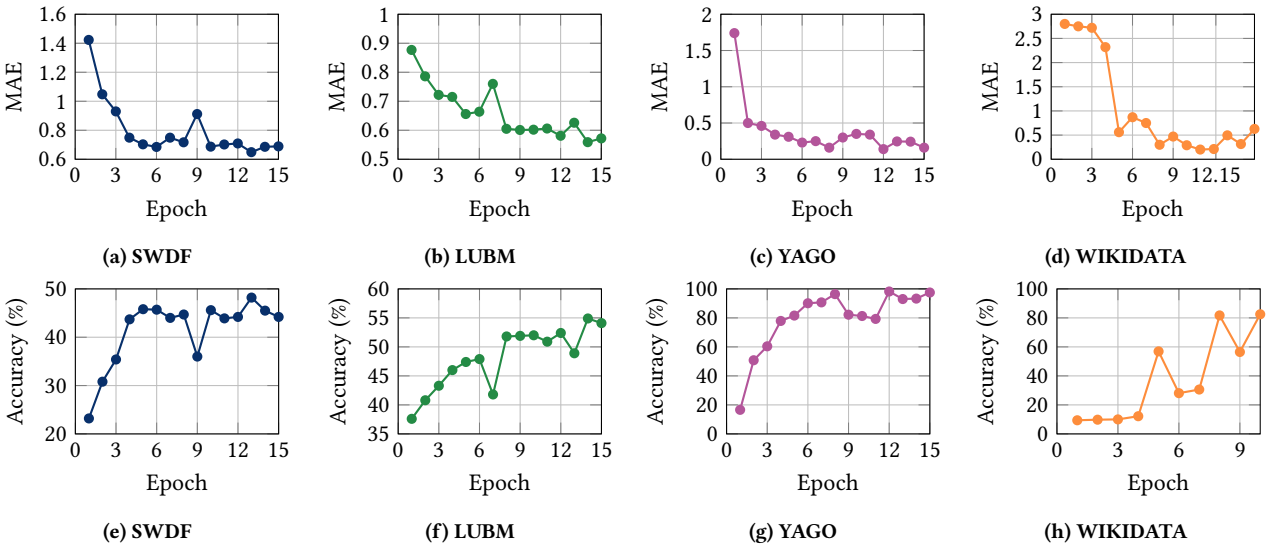
C.2.2 Competitors. In the literature, the existing methods for GED computing can be divided into two categories: learning-based and traditional. For the former type, we mainly compare our method with the five approaches:

- SIMGNN [5]: the first method that uses neural networks to predict the GED.
- TAGSIM [4]: the GNN-based method that employs type-aware auxiliary information to predict GED.
- GPN [87]: a learning-based method that trains a path network to predict GED.
- GEDGNN [58] a learning-based GED estimation method that utilizes the k-best matching to improve the prediction accuracy.
- GEDIOT [13] The state-of-the-art learning-based GED estimation method integrates neural networks with optimal transport theory to enhance prediction performance.

Besides, we also compare our method with the two representative traditional methods:

Table 9: p-value analysis comparing GEDIOT and SEABED.

Dataset	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
SWDF	0.0443	0.0024	0.0243	0.0453	0.5132	0.2660	0.2607	0.0975
LUBM	1.03×10^{-5}	1.70×10^{-4}	1.09×10^{-8}	5.35×10^{-9}	5.03×10^{-7}	1.62×10^{-7}	3.28×10^{-8}	2.87×10^{-7}
YAGO	0.0109	0.0039	0.1512	0.1488	0.0985	0.1237	0.2035	0.1536
WIKIDATA	0.0282	0.0041	0.1265	0.1270	0.1082	0.1062	0.1600	0.1550



(a) SWDF (b) LUBM (c) YAGO (d) WIKIDATA

(e) SWDF (f) LUBM (g) YAGO (h) WIKIDATA

Figure 10: Effect of training epochs in terms of MAE and Accuracy.

Table 10: Statistics of different scale KG sets.

Name	# KGs	$\lvert\overline{V}\rvert$	$\lvert\overline{E}\rvert$	$\lvert V\rvert_{\max}$	$\lvert E\rvert_{\max}$	# of pairs
SWDF-1	1000	7	8	10	20	50000
SWDF-2	1000	20	23	30	61	50000
SWDF-3	1000	40	50	50	92	50000
SWDF-4	1000	60	75	70	123	50000
SWDF-5	1000	75	95	90	136	50000

- Greedy [39]: a widely used method applies the Hungarian algorithm in a greedy manner to calculate the GED values.
- Noah [87]: a optimized version of the traditional A* algorithm that employs GPN to guild A*-beam search.

1. Time efficiency. As shown in Table 3, we report the testing and post-processing time for each KG pair (in milliseconds). All learning-based models are more efficient than traditional methods, and they often achieve comparable performance. Besides, we can see that Noah is the slowest algorithm, since it employs the A*-beam search algorithm with exponential time complexity.

2. The p-values. We compute the p-values [83] (a well-known statistical significance measure) between GEDIOT and SEABED to quantify the performance advantage of our method, as shown in Table 9. For both MAE and Accuracy, the p-values are below 0.05 across all datasets, indicating that the improvements achieved by SEABED are statistically significant. Moreover, for Accuracy, the p-values are below 0.01, suggesting a highly significant improvement. For the remaining metrics, the improvements appear smaller or comparable. We would like to highlight that these metrics are computed as follows: given a query graph G_q, all candidate graphs are first sorted by their predicted GEDs with respect to G_q, and this predicted ordering is then compared with the ground-truth ordering derived from the true GED values. In other words, these metrics depend solely on relative order, rather than the absolute accuracy of GED predictions. That is, even noticeable improvements in the predicted GED values may lead to only minimal changes in the ranking. As a result, these metrics are inherently less sensitive, which explains why the observed differences remain modest. Nevertheless, we emphasize that MAE and Accuracy are the primary and most informative metrics for GED estimation, and both consistently demonstrate the superior performance of SEABED.

3. Results on YAGO-Real. The YAGO dataset used in our paper is actually a temporal KG, where each relationship is associated with a timestamp. For example, a fact such as ⟨Barack Obama, host a visit, *China*, 2014-12-20 ⟩ records a time-specific event. To better evaluate GED estimation under real-world KG evolution, we construct a new dataset called YAGO-Real, whose scale remains the same as the original YAGO. In YAGO-Real, each KG pair is formed from two temporal snapshots of the same KG. For example, as shown in Figure 11, the snapshots at 2014-12-23 and 2014-12-25 differ by one new event, ⟨*China, Consult, Japan, 2014-12-25*⟩, which appears only in the later snapshot. This single temporal update results in a GED of 1 between the two KGs. By constructing YAGO-Real, we simulate realistic KG evolution through temporal edits, which allows us to validate the performance of SEABED under real-world conditions. The performance of our method and the strongest baseline GEDIOT

Table 11: Comparison of SEABED and GEDIOT on the YAGO-Real dataset.

YAGO-Real	GED		Ranking					
	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
GEDIOT	0.624±0.083	44.94±5.15	0.984±0.007	0.932±0.019	85.63±1.86	92.62±1.04	95.02±0.41	96.27±0.68
SEABED	**0.195±0.055**	**94.07±4.03**	**0.991±0.000**	**0.951±0.000**	**87.71±0.25**	**93.34±0.33**	**95.95±0.16**	**97.06±0.12**

Table 12: Comparative results of SEABED and GEDIOT on different scale datasets.

SWDF-1	GED		Ranking					
	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
GEDIOT	0.771±0.068	40.57±2.29	**0.906±0.014**	**0.792±0.019**	**53.19±1.35**	**68.04±1.58**	**78.08±1.65**	**83.58±1.16**
SEABED	**0.675±0.007**	**47.40±0.38**	0.886±0.002	0.769±0.003	52.81±0.89	67.32±0.49	77.27±0.50	82.52±0.40
SWDF-2	GED		Ranking					
	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
GEDIOT	0.973±0.139	30.51±3.71	0.963±0.004	0.871±0.008	73.75±0.75	82.03±0.23	87.64±0.63	90.94±0.50
SEABED	**0.569±0.031**	**50.51±2.83**	**0.976±0.001**	**0.900±0.002**	**75.73±0.43**	**84.41±0.23**	**89.50±0.23**	**92.49±0.10**
SWDF-3	GED		Ranking					
	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
GEDIOT	2.289±0.068	12.91±0.34	0.969±0.001	0.881±0.001	73.89±0.10	85.23±0.18	90.20±0.07	92.67±0.18
SEABED	**0.579±0.009**	**49.01±0.18**	**0.986±0.000**	**0.928±0.001**	**79.39±0.10**	**89.00±0.70**	**93.07±0.14**	**94.43±0.16**
SWDF-4	GED		Ranking					
	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
GEDIOT	2.444±0.172	11.65±0.31	0.944±0.002	0.825±0.004	65.19±0.40	79.90±0.45	86.83±0.30	89.95±0.23
SEABED	**1.780±0.741**	**21.19±11.25**	**0.952±0.018**	**0.845±0.039**	**69.32±4.46**	**81.73±3.30**	**88.05±2.27**	**90.54±1.81**
SWDF-5	GED		Ranking					
	MAE	Accuracy(%)	γ	τ	p@5(%)	p@10(%)	p@15(%)	p@20(%)
GEDIOT	2.863±0.224	10.06±0.57	**0.971±0.007**	**0.880±0.017**	**76.22±2.23**	86.18±1.34	**91.75±0.84**	**92.99±0.73**
SEABED	**2.078±0.676**	**17.46±9.55**	0.969±0.017	0.880±0.042	75.29±6.66	**86.51±4.62**	90.89±2.94	92.87±2.13

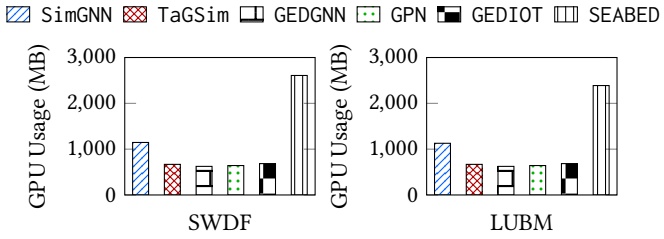


Figure 12: Comparison of GPU usage.

on YAGO-Real is reported in Table 11. We can see that SEABED still outperforms GEDIOT, demonstrating its effectiveness even on more realistic KG evolution scenarios.

C.3 Detailed analysis of SEABED

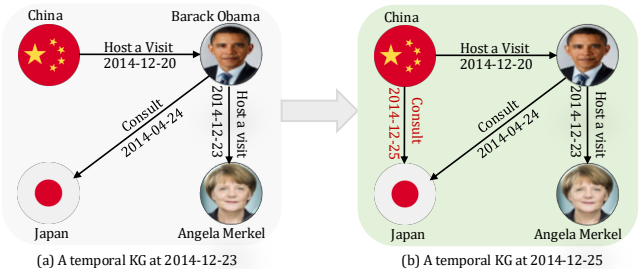


Figure 11: An example pair of temporal KGs with GED=1.

1. Effect of training epochs. Figure 10 shows the changes in MAE and Accuracy in the four datasets with different training epochs. We can see that as the number of training epochs increases, the MAE generally decreases while the accuracy increases, which aligns with our expectations. Additionally, on large KG datasets like YAGO and WIKIDATA, our SEABED achieves good results even with fewer training rounds, highlighting that the SEABED converges quickly and learns efficiently from the data. This demonstrates its ability to capture essential patterns without extensive training.

2. Memory requirements. The design of SEABED incurs slightly higher GPU usage compared to existing methods. As shown in Figure 12, we can see that SEABED requires about 2× the GPU memory of baseline models. However, this overhead is very modest in practice; the total memory consumption remains under 3GB, which is easily supported by mainstream GPUs (e.g., RTX 3090/4090) as well as common personal GPUs (e.g., RTX 4060 Ti 8GB). Given the significant performance gains achieved by SEABED, we believe this small GPU overhead is unlikely to pose a practical limitation in real-world applications.

3. Scalability Study. Based on the SWDF dataset, we constructed four expanded versions by progressively increasing the sizes of the KG pairs in each dataset. Here, SWDF-1 denotes the original SWDF dataset, while SWDF-5 represents the largest version. In general, a larger index X indicates that SWDF-X contains proportionally larger KGs. The statistics of the five datasets are shown in Table 10. Afterwards, we evaluate SEABED and compare it with the strongest baseline GEDIOT across all five datasets. As

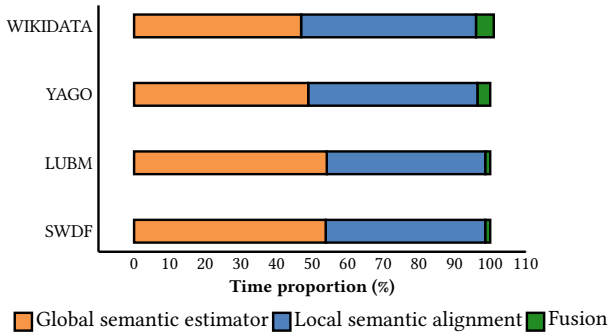


Figure 13: Proportion of time cost of each step in SEABED.

shown in Table 12, SEABED consistently outperforms GEDIOT in both MAE and Accuracy across all scales, demonstrating the strong scalability of our method.

4. Time cost of different steps in SEABED. Recall that SEABED executes the following three steps sequentially: (1) Global semantic estimator, (2) Local semantic alignment, and (3) Multi-scale semantic fusion. Figure 13 shows the time cost of these three steps across four datasets. We observe that in all datasets, the first two steps take up the majority of the execution time, with similar time costs, while the time cost of semantic fusion prediction is relatively small.