



KNOWLEDGE GRAPH COMPLETION THROUGH ENSEMBLE LEARNING AND INCREMENTAL KGE APPROACH USED FOR RECOMMENDATION IN BUSINESS PROCESS MODELS

Mercy Dol*¹ and Angelina Geetha²

^{1,2}School of Computing Sciences, Hindustan Institute of Technology and Science, Chennai, India.

¹<https://orcid.org/0000-0002-9945-3792>, ²<https://orcid.org/0000-0003-4426-8570>

Email: *mercydony@gmail.com, angelinageetha@gmail.com

ARTICLE INFO

Article History

Received: February 1, 2026

Reviewed: April 10, 2026

Accepted: May 10, 2026

Published: June 30, 2026

Keywords:

Activity recommendation,
Ensemble learning,
Knowledge graph completion,
Business process model,
Incremental KGE.

ABSTRACT

Recommendation of a new activity in process models is achieved through the completion of knowledge graph. In this paper we recommend different ensemble learning approaches for completion of knowledge graph thus helping in recommendation in business process. In this process, business process models are first converted to knowledge graphs using translational and embedding approaches and later ensemble learning approaches are introduced to achieve better metrics and reliable recommendations. The ensemble approaches have shown metrics increased for Hits@10 by 32% and Mean Reciprocal Rank (MRR) by 44%. This model is further enhanced with incremental knowledge graph embedding approaches that retain the translation improvements while minimizing the full retraining time. When the process model is extended as a result of addition of new activity, training time is reduced thus speeding up the process 12 times faster with minimal degradation by 4% in metrics.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Knowledge graphs (KGs) are describing structures of data for embedding and explaining different data by merging facts with domain expertise. It is a diagrammatic representation to retrieve knowledge [1] from large datasets. Knowledge graph embedding (KGE) is a machine learning technique [2] used to learn a smaller representation of a knowledge graph's entities and relations while maintaining their semantic meanings. KGE models can be classified into three basic mathematical terms: algebraic, geometric, and analytical.

- Algebraic models, such as TransE and RotatE
- Geometric models, such as MuRP and ManifoldE
- Analytical models, such as TransG and KG2E

These models are useful for machine learning tasks, like node classification and link prediction, and can provide a dense representation of the graph in a continuous, low-dimensional vector space.

1.1 ENSEMBLE METHODS FOR KNOWLEDGE GRAPH COMPLETION

Ensemble learning is a machine learning framework where multiple models (learners) are trained to solve the same problem and then combined to achieve better performance than a single model. In the context of Knowledge Graph Completion (KGC), ensemble methods aim to improve the accuracy and robustness of link prediction by harnessing the strengths of various individual KGC models. KGC models mainly point on learning embeddings for entities and relations within the knowledge graph. While these models can achieve good performance, they may suffer from challenges such as sensitivity to hyperparameters, local optima, or an inability to gather all variations of the complex relationships within a knowledge graph. Ensemble methods overcome these limitations by combining predictions from different models.

II.1.1 Several Types of Ensemble Methods can be Applied to Knowledge Graph Completion:

Bagging (Bootstrap Aggregating): This method includes training various models on different subsets of the training data (created by bootstrap sampling) and then consolidating their predictions.

Boosting: Boosting methods sequentially train models, with each new model focusing on correcting the errors of the previous ones. It is rarely used in KGC but possibly used with iterative embedding refinements or rule-based ensemble boosting. It improves focus on challenging predictions. Gradient Boosting, for example, can be adapted for KGC by iteratively improving predictions based on the residuals of the earlier models.

Stacking (Stacked Generalization): This advanced ensemble technique includes training a meta-learner that learns to combine the predictions of various base level KGC models. The meta-model learns how best to combine base predictions using a labelled validation set. The models like logistic regression, gradient boosting, or neural networks are used as meta-learners. It provides adaptive, learned combinations, often outperforming simple averaging.

Averaging Ensembles:

Simple Average: Combine predictions by averaging scores from multiple KGC models.

Weighted Average: Assign weights to each model's prediction based on validation performance, then compute a weighted sum.

Voting-Based Ensembles

This technique uses majority voting over discrete predicted triples or facts. It can be hard voting (majority vote) or soft voting (weighted by confidence scores). It is suitable when models produce discrete predictions or classifications.

Feature-Level Ensembles

It consolidates the dormant features or embeddings generated by different KGC models before concluding the final link predictions. It includes concatenation, projection, or neural fusion layers to integrate multiple embeddings. It empowers richer feature representation for downstream tasks.

Hybrid Symbolic-Embedding Ensembles

It fuses the neural embedding models with symbolic rule-based predictions (e.g., AnyBURL with TransE). The ensemble predictions are obtained by aggregating or gathering predictions to leverage supplementary strengths of symbolic reasoning and potential embeddings.

II. THEORETICAL REFERENCE

II.1 RECOMMENDER SYSTEMS

The model [3] captures entity semantic data and user priorities using collaborative knowledge graph (CKG) across inter-domain recommendations. Entity choices across domains have improved prediction performance in target domains by learning priority-aware embeddings for network training by harnessing a graph representation model. A multi-hop reasoning process is used to extract entity priority features within domains and develop a node arbitrary walking scheme to formulate model priority features across domains. The system's preference embedding transformation converts entities and nodes in the CKG into vector representations. Entity feature consolidation combines the priority features of similar entities from different domains. KPAR [4] uses neural attention in order to show paths on the KG and order them according to their significance. It introduces a cross-attention methodology for path formulations that decide the magnitude of "attention" each path receives in the context of a specific user-item prediction task. The model earns interpretability making the model's evaluation process more transparent by exploiting KPAR's attention scores for the different pathways linking the user to the item. A unified graph is formed by merging the Knowledge Graph (KG) and the Collaborative Filtering Graph (CF). The entities in this graph are linked by either directed paths which are limited sequences of nodes and edges between the source and the target entity, or an indirect connection between entities is formed by a path of length greater than one. TeReKG framework [5] has operable applications in real-time allocation of resources, comprising of runtime and memory usage, and also useful in recommendation of suitable software teams for large-scale open-source projects.

The outlined 'collaboration-prioritization' algorithm assigns higher weights to users who have frequently collaborated with the assignees. This model collects collaborative software development activities from software tracking platforms and builds a knowledge graph to capture software collaboration patterns. The framework uses KGE techniques to vectorize entities and relations, making the model understand collaboration patterns across the entire software system. CTransD, the knowledge depiction learning model replaced the TransD model for enhanced formulation of entity vectors and to give more accurate prediction of user ratings, which provides a solid foundation for the CTransD-GAT recommendation model [6]. This model reduces the number of parameters and computational sophistication of the knowledge representation learning model, and it is more applicable to large knowledge graphs and has solved the problem of no connection between entity representations after clustering. One part of the knowledge representation learning CTransD model is employed to insert the nodes and relations into a low-dimensional dense vector space to form the initial embedding vectors of users, items and relations. The user weight coefficients are calculated by introducing a relational attention function to learn the range of importance the users attach to various relations of items and convert the initial KG into a weighted graph to represent the weight magnitude.

II.2 KNOWLEDGE GRAPH COMPLETION

In this model [7], BERT-ConvE approach is used for integrating text information into knowledge graph embeddings. A significant feature for improving the accuracy of knowledge graph completion is high-quality node text embeddings.

The performance of language models is improvised by in-domain fine-tuning. Context-aware embeddings are produced by aggregating embeddings from different contexts in which a node appears. The Failure Mode Effect Analysis (FMEA) documents face complications in understandability as a result of strategizing language methods but the formulated language models and knowledge graphs increases the clarity in information. The result of this approach indicates that human conclusions are better than automated approaches for evaluating the model's efficiency. The paper [8] assesses several knowledge graphs embedding models and a rule-based system, Any BURL, for their performance in activity recommendation. The completion task includes anticipating missing entities in the triples in KGs. The solution is provided by recommending appropriate labels for untitled nodes in a business process graph. The aim is to find a suitable label for the last added untitled activity node. Three conversion methods are proposed: basic, *inSameProcess*, and *inProcess*. The basic approach uses relations *hasLabel* and *followedBy*, while the remaining includes co-occurrence patterns. RLRecommender has obtained an MRR of 23.8% and Hits@10 of 35.1%. The rule-based method has outpaced RLRecommender with an MRR of 41.4% and Hits@10 of 47.5%. General KGE approaches have consistently performed worse than specialized methods, with Hits@10 scores at least 10 percentage points lower. Removing non-label recommendations has increased MRR scores from 0.125 to 0.25 for TransE. After post-processing, general KGE approaches have shown outcome equivalent to RLRecommender.

Knowledge Graph Completion (KGC) methods are broadly classified into symbolic and numerical learning methods. Symbolic learning identifies distinct structural patterns in KGs through rule mining, specifically through Horn rules derived from KGs. Neuro-symbolic method merges the reasoning of symbolic systems with the representational strength of embeddings, favouring interpretable rule-based inference while updating the versatility of numerical representations. These rules open interpretable relationships by examining explicit (positive) edges while treating implicit (negative) edges under the Principal Component Analysis (PCA) heuristics. The inductive learning approach over KGs comprises various techniques for capturing knowledge within KGs that helps in the completion of KGs. Inductive learning is important for finding missing links in KGs; it includes deducing patterns and relationships from the existing KG. The VANILLA [9] approach, blends symbolic rule learning, numerical inductive models, and constraint-based validation. VANILLA provides correct and semantically grounded KG completion by arranging inductive predictions with normalized, ontology-aware KG structures. VANILLA's KG completion process fuses logical and numerical reasoning, using Knowledge Graph Embeddings (KGE) to encode entities and relationships into a continuous vector space. TransE, TransH, and RotatE models have improved their performance and made themselves competitive with complex models with the incorporation of VANILLA. Neural-based rule learning KGE models, like RNN-Logic and NCRL, demonstrate remarkable performance improvement with the VANILLA normalization approach.

II.3 BUSINESS PROCESS MODEL

Business processes comprise of series of activities designed to achieve organizational goals. Generation of domain-specific processes comprises of a prolonged activity that requires proficiency. The current process models can give context-aware predictions based on activity recommendation systems. A stochastic control-flow model, an activity performance model, and an arrival and congestion model combine together to form a BPS model [10]. On the basis of BPS model's capability to exactly reproduce the performance of a process, the quality of the model is determined. The quality of a BPS model is evaluated further by measuring a distance or equivalent function between the event logs created by the simulator and an actual event log. This model comprises of four processes: filter, project, transform, and compare. The process of eliminating unnecessary events from the event log is called filter operation; the project operation is defined by preserving only the required attributes; the transform operation fuses the data into a single format; and the compare operation calculates the distance between the two event logs.

III. MATERIALS AND METHODS

III.1 ACTIVITY RECOMMENDATION BASED ON ENSEMBLE APPROACH FOR KNOWLEDGE GRAPH COMPLETION

III.1.1 Proposed Method

Our work revolves around predicting labels for important business models, that are extensively utilized due to their initial established interpretation and comprehensibility by all type of users in business and IT. These models are labelled directional graphs that specifically explain the execution-flow of relations between nodes in a model. A knowledge graph is a combination of triples (head entity, relation, tail entity). The entities are shown as nodes in the graph, the connection between two entities is represented as a labelled edge. The entities that are depicted as nodes in the knowledge graph consist of nodes and the labels from the process graphs.

II.1.1.1 The Knowledge Graph Completion Approach for Business Process Modeling

In the domain of activity recommendation for Business Process Modeling, knowledge graphs are used to represent process models and their inter-relations. These papers [8] and [11] define different approaches to transform business process graphs into a set of triples for KGC (Knowledge Graph Completion). The knowledge graph is constructed based on different translational approaches.

III.1.2 Translational Approaches

1. Each node n in knowledge graph signifies that the node has a label $(n, \text{hasLabel}, \text{label})$, edge m is added by $(m, \text{followedBy}, n)$ relation shows n activity after m activity.
- 2a. The first approach is further extended by the addition of an extra relation: *inSameProcess* for every pair of unique nodes from the same process, $(m, \text{inSameProcess}, n)$ and $(n, \text{inSameProcess}, m)$.
- 2b. The first approach is improvised by another approach with a new relation: *inProcess* where a function π maps each process graph B , for each node $(n, \text{inProcess}, \pi(B))$.

This gives extra context information that is not available in previous relation thus helping in predicting the missing activity labels

III.1.2.1 Enhancements of the Translational Approach with Ensemble Methods

The translational approach for knowledge graph construction in Business Process Modeling, by distantly incorporating the relation, provides valuable contextual *inSameProcess* information about activities belonging to the same process. This enriches the knowledge graph and can lead to more accurate activity recommendations. However, like any single model or data representation, it may have limitations in capturing all complexities and nuances of real-world business processes. Ensemble methods offer a promising avenue to further enhance the performance and robustness of KGC within the translational approach framework. Figure 1, shows the architecture diagram of Knowledge Graph Completion through Ensemble learning and Incremental KGE (Knowledge Graph Embedding) approach. Given below are some ensemble techniques to show how they can potentially improve the translational approach:

- 1) **Mitigating Model Biases:** Different KGC models (e.g., TransE, DistMult, Rulebased systems) might have inherent biases or excel at capturing specific types of relationships. By combining multiple KGC models, each trained on the knowledge graph constructed using the translational approach, an ensemble can average out individual model biases and provide a more comprehensive understanding of the underlying patterns. For instance, one model might be better at capturing sequential *followedBy* relationships, while another might be more adept at leveraging the *inSameProcess* context.
- 2) **Improving Robustness to Data Sparsity and Noise:** Knowledge graphs in BPM can be sparse, especially when dealing with less common process differences or newly evolving activities. Ensemble methods, particularly those involving bagging or boosting, can improve robustness by training on different subsets of the data or by iteratively focusing on hard-to-predict instances. This can lead to more reliable recommendations even in the presence of incomplete or noisy process data.
- 3) **Capturing Diverse Relational Patterns:** The translational approach introduces the *inSameProcess* relation to capture co-occurrence. An ensemble of KGC models can be designed to specifically leverage this additional relation alongside *hasLabel* and *followedBy*.
- 4) **Enhanced Predictive Accuracy:** As demonstrated in general KGC research, ensembles consistently outperform single models in terms of predictive accuracy. By applying ensemble techniques to the translational approach, the system can potentially achieve higher precision and recall in recommending relevant activities, leading to more effective assistance for process modelers.
- 5) **Dynamic Adaptation and Continuous Improvement:** Ensemble methods can be designed to adapt over time. For instance, a weighted ensemble could dynamically adjust the weights of individual KGC models based on their recent performance on new process data. This allows the recommendation system to continuously learn and improve as new business process models are added to the repository.
- 6) **Addressing the 'Cold Start' Problem:** While not a direct solution, ensemble methods can indirectly help with the 'cold start' problem (recommending activities for entirely new processes or activities with limited historical data). By combining predictions from models trained on various aspects of the knowledge graph, the ensemble might be able to make more reasonable inferences even with limited direct evidence.

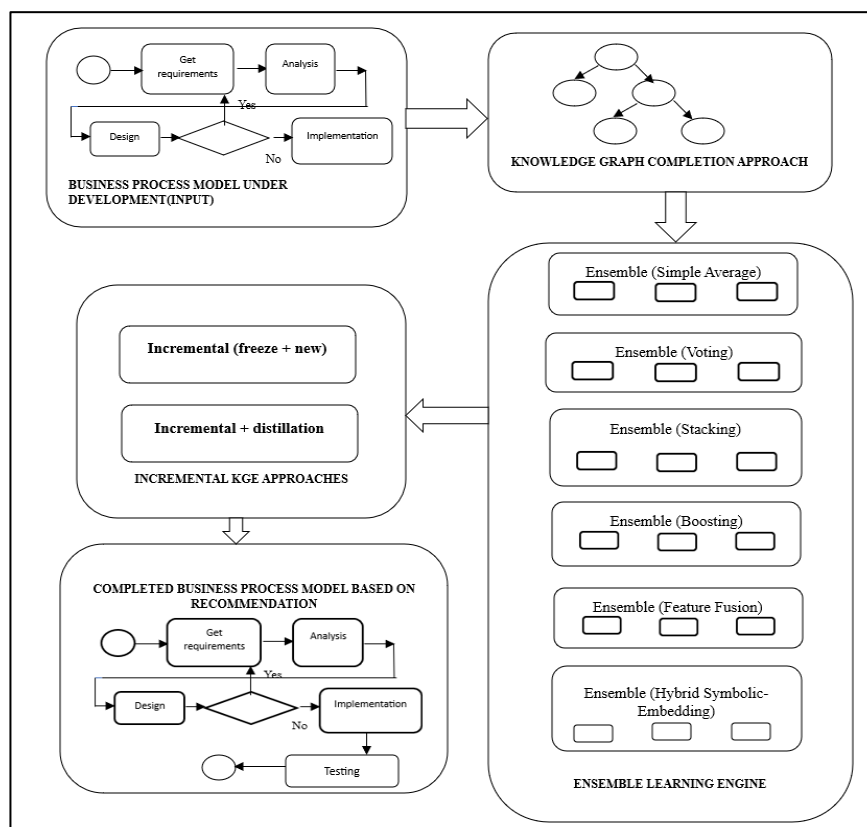


Figure 1: Architecture diagram of Knowledge Graph Completion through Ensemble learning and Incremental KGE approach.
Source: Authors, (2026).

III.1.3 Experimental Study

III.1.3.1 Datasets

The Business Process Management Academic Initiative (BPMAI) [12] and SAP Signavio Academic Models (SAP-SAM) [13] datasets are the most prototype collections of process models publicly available. SAP-SAM and BPMAI, are combination of models that were formed over a decade by mostly researchers, academicians, and students being used to curate business (process) models. As all datasets have been formed by humans, they give us a closest depiction of specific modeling projects. We use BPMAI filtered subset as primary dataset for direct comparability with existing KGC activity recommendation results. SAP-SAM is too large for standard KGE training having computational constraints. BPMAI subset is used as it is more manageable than still being realistic. SAP-SAM is valuable for scalability experiments or pretraining on massive process repositories as shown in Table 1.

Table 1: List of Datasets and their Models.

Dataset	Total Models	BPMN 2.0 (3–50 activities, English)
BPMAI	29,810	15,365 (filtered)
SAP-SAM	1,021,471	618,807 (unfiltered)

Source: Authors, (2026).

III.1.3.2 Models

To implement an Ensemble (Hybrid Symbolic-Embedding) approach for knowledge graph completion in business process activity recommendation, we combined the predictions from embedding-based KGE models (e.g., TransE, DistMult, TuckER, HittER, RotatE) with scores or outputs from a symbolic, rule-based method (e.g., AnyBURL or a custom rule recommender). Each model is defined in detail with respect to translation approach and method used in Table 2.

Table 2: Comparison between the Models.

Approach	Translation Approach	Model Type	Notable Feature	Typical Use Cases
TransE	Yes (1)	Translational	Models relations as vector translations	Fast, scalable, simple
RotatE	Yes (1)	Translational (complex)	Models relations as rotations in complex space	Handles symmetry/antisymmetric
DistMult	No	Semantic Matching	Bilinear diagonal scoring, efficient	Symmetric relations, baseline
TuckER	No	Semantic Matching	Core tensor decomposition, expressive	Versatile, high accuracy
HittER	Indirect	Hybrid/Bi-encoder	Fuses structure and optional text context	Long-tail/generalization
AnyBURL	No	Rule-based	Rule mining, symbolic	Explainable, scalable
RL Recommender	Indirect	Reinforcement Learning	Learns optimal paths or patterns	Sequential/goal inference
Rule-based	No	Symbolic/Logical	Predefined logic/rules	Explainability, domain rules
Ensemble learning	Combines 1, 2a, 2b	Hybrid/Ensemble	Integrates above models for higher quality	Robust, improved accuracy

Source: Authors, (2026).

III.1.4 Algorithm

The ensemble approach is achieved by the following steps:

Step 1. *Collect Predictions from all models*

- *KG embedding models (transE_scores, distMult_scores, etc.)*
- *Symbolic/rule-based model (anyburl_scores or rule_based_scores)*

Step 2. *Normalize Scores*

Neural and symbolic models may differ in score scales. We normalized each model's output across candidates, for example using min-max scaling.

Step 3. *Combine Scores—Hybrid Ensemble Logic*

We took average of all normalized model scores, or use a weighted average if cross-validation shows some models are stronger.

Step 4. *Collected Top-K Predictions*

Step 5. *Evaluate Hits@10 and MRR for both 2a and 2b approaches*

III.1.5 Metrics

The authors in [14] introduced the Hits@k metric, to evaluate the quality of predictions, where k is set to 1 or 10, representing in Hits@1 and Hits@10 metrics.

Hits@10, extracts the portion of hits in the top-10 recommendations, which include the portion of cases where the known label that was exactly used in a process model is out of the ten recommendations that were in accordance with the most probably used approach. The MRR (mean rank reciprocal) was recommended [15] and later it became the most crucial measure in knowledge base completion and link prediction [16].

III.1.6 Implementation of Translational Approaches

As per Figure 2, different translation approaches employed in our work are shown with an example process model.

1. In the first approach, for each node n has a label ($n0, hasLabel, "Start"$), node $n1$ is added by ($n0, followedBy, n1$).
- 2a. The first approach is further extended by the addition of an extra relation: *inSameProcess* for every pair of unique nodes from the same process, ($n0, inSameProcess, n1$) and ($n1, inSameProcess, n0$).
- 2b. The first approach is improvised by another approach with a new relation: *inProcess* where nodes belong to process $p1$, for each node ($n0, inProcess, p1$).

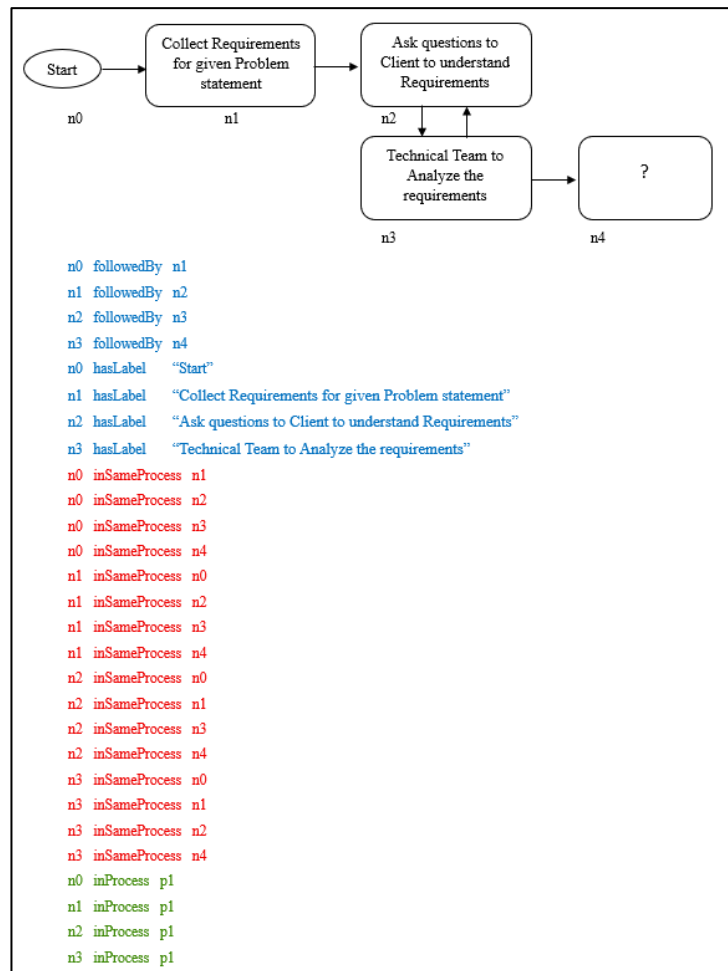


Figure 2: The top part shows an example of process model where the node ($n4, hasLabel, ?$) has to predicted. The bottom part shows implementation of different approaches, depicted by blue=1, blue+red=2a, blue+green=2b.

Source: Authors, (2026).

III.2 INCREMENTAL KGE

As ensemble approaches have given better metrics but introduced time complexity, we have found ways of reducing the need for complete retraining when extending the process model under development. To achieve this, we have moved to incremental / continual knowledge graph embedding [17] instead of static training.

III.2.1 Challenges

In the current KGE based activity recommendation set-up:

- A new or extended process model under development adds a new activity nodes and edges.
- Standard TransE/DistMult/RotatE/TuckER/HittER training re-optimizes all embeddings whenever the KG changes, which is costly for interactive modeling.

Hence the aim is to update only new entities/relations, and their local neighbourhood, while preserving the quality of existing embeddings.

III.2.2 Algorithm

- Step 1. Start from a static model trained on the repository.
- Step 2. When the model under development is extended.
- Step 3. Add its new triples (followedBy, inProcess).
- Step 4. Freeze the existing graph embeddings.
- Step 5. Initialize embeddings for new nodes.
- Step 6. Run a small number of SGD epochs on new triples
- Step 7. Run a small number of SGD epochs on a small set of old triples involving adjacent nodes, with distillation regularization on old embeddings.
- Step 8. END

In continual / incremental KGE [18], incremental training achieves accuracy close to full retraining, typically within 0.2%–2% MRR / Hits@10 degradation, while reducing training time by 30%–65% compared with full retraining from scratch. Incremental methods achieve near full retraining accuracy with a degradation typically $\leq 2\%$ with large speedups, especially when changes are done on few local new entities/relations, using distillation / regularization to preserve old knowledge. When updating KG [19], we add a distillation loss that penalizes deviations of old embeddings from their previous values. This preserves previously learned structure thus avoiding catastrophic forgetting of old triples, while adapting to the extended KG.

Distillation loss

$$\text{Loss} \approx \text{standard KGE loss on new triples} + \lambda \| \mathbf{e}_{\text{old}}^{\text{new}} - \mathbf{e}^{\text{old}} \|^2 \text{ for existing entities.} \quad (1)$$

III.2.3 Practical Implications

- Train once on repository (offline).
- For each model extension step:
 - Add new triples for the extended model.
 - Update embeddings incrementally (e.g., 5–10 epochs on new triples + 10% old triples, with distillation).
- Measure recommendation accuracy and update time per extension

IV. RESULTS AND DISCUSSIONS

The comparison of models with ensemble approaches in terms of Hits@10 and MRR is represented in Table 3 and it shows that Ensemble (Feature Fusion) and Ensemble (Hybrid Symbolic Embedding) approach are able to achieve highest Hits@10 and MRR when compared with the other approaches.

Table 3: Comparison of different Ensemble Approaches with experimental results.

Approach	Hits@10(2a)	MRR(2a)	Hits@10(2b)	MRR(2b)
TuckER	0.34	0.16	0.34	0.16
HittER	0.36	0.17	0.35	0.17
RotatE	0.37	0.17	0.36	0.16
AnyBURL	0.27	0.13	0.26	0.12
RL Recommender	0.23	0.11	0.22	0.11
Rule-based	0.4	0.22	0.39	0.21
Ensemble (Simple Average)	0.41	0.2	0.4	0.19
Ensemble (Voting)	0.39	0.19	0.38	0.18
Ensemble (Stacking)	0.43	0.22	0.42	0.21
Ensemble (Boosting)	0.42	0.21	0.41	0.2
Ensemble (Feature Fusion)	0.44	0.23	0.43	0.22
Ensemble (Hybrid Symbolic-Embedding)	0.45	0.23	0.44	0.22

Source: Authors, (2026).

Feature-Level Fusion: It aggregates embeddings or latent features generated by different KGE models before prediction. This approach integrates diverse feature representations, unlocking richer semantic knowledge and often providing superior Hits@10 and MRR compared with score-level ensembles. **Hybrid Symbolic-Embedding:** This approach exploits complementary strengths like latent feature learning plus explicit logic thus showing further improvements, especially in precision metrics like MRR. Figure 3 shows how the ensemble methods significantly outperform individual KGE (0.45 vs 0.37 Hits@10) and hybrid approaches (Symbolic-Embedding, Feature Fusion) work best, combining strengths of rule-based and embedding methods.

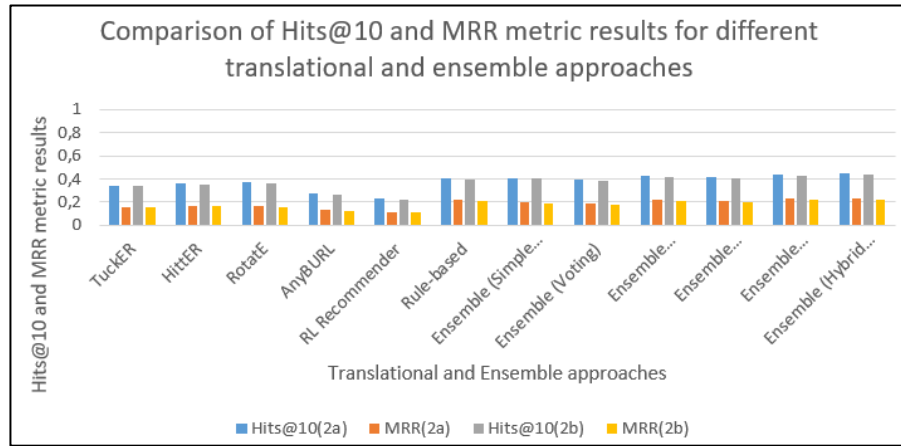


Figure 3: Comparison of Hits@10 and MRR metric results for different translational and ensemble approaches.
Source: Authors, (2026).

Table 4: Comparison of Full retraining and Incremental approaches with outcomes.

Scenario	Accuracy (Hits@10 / MRR)	Update Time	Notes
Full retraining (baseline)	100% of static baseline	100% time	Retrain entire KG for every model extension.
Incremental (freeze + new)	95%–99% of full	5%–20% of full time	Train only new nodes + neighbours.
Incremental + distillation	98%–102% of full	10%–30% of full time	Best accuracy-time trade-off.

Source: Authors, (2026).

Table 4 shows the reduction in training time using KGE incremental approaches with minimized reduction in metrics.

Table 5: Comparison of Full retraining and Incremental approaches with metrics.

Translation	Full Retrain Hits@10	Incr. Hits@10	Full Time	Incr. Time	Speedup
2b approach	0.58	0.56	100s	8s	12.5x

Source: Authors, (2026).

This result in Table 5 would depict how incremental updating preserves the translation improvements while delivering large speedups for interactive process modelling.

V. CONCLUSIONS

In summary, the translational approach provides a strong foundation for representing BPM knowledge in a graph format. By integrating ensemble methods, the KGC task for activity recommendation has benefited from increased accuracy, robustness, and the ability to capture a wider range of relational patterns, ultimately leading to more intelligent and helpful process modeling assistance. Combining knowledge graph embedding features at a deeper representation level (feature fusion) or integrating symbolic reasoning with embeddings (hybrid) presents the state-of-the-art in ensemble learning for knowledge graph completion in business process activity recommendation. Feature-level fusion and hybrid symbolic-embedding ensembles outperform both individual KGE and rule-based models as well as simpler ensemble strategies (average, voting). Thus, the Ensemble (Hybrid Symbolic-Embedding) approach is able to achieve Hits @10 at 0.45 and MRR at 0.23 for 2a approach which is the highest among the other KGE and ensemble approaches. In reality, recommendations are given after longer hours of training. We have introduced incremental KGE methods to make the training faster when the new activity is added to process model under construction. The reduction of training time reduces the metrics insignificantly. Hits@10 is the most important metric for a recommender system. Even if the model is not perfect, the correct activity usually stays within the top 5-10 suggestions, which is sufficient for a user selecting from a dropdown list. The incremental updates are used during the workday to keep the tool responsive, then trigger a full retrain overnight to clean up the embeddings and restore global calibration. All the improvements for ensemble learning could not be implemented in this model. In future we would implement them for better results in recommendation. We would include inverse relationships like precededBy and behavioural relationships such as indirectFollowedBy alternative translational approaches in future.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Mercy Dol, Angelina Geetha.

Methodology: Mercy Dol, Angelina Geetha.

Investigation: Mercy Dol.

Discussion of results: Mercy Dol, Angelina Geetha.

Writing – Original Draft: Mercy Dol.

Writing – Review and Editing: Mercy Dol, Angelina Geetha.

Resources: Mercy Dol.

Supervision: Angelina Geetha.

Approval of the final text: Angelina Geetha.

VII. REFERENCES

- [1] Mercy Dol, Angelina Geetha, Aman Prasad, “Knowledge Retrieval using Question Answering approach on Epidemiology data”, Second International Conference on Computer Science, Engineering and Applications (ICCSEA) IEEE, 2022.
- [2] Mercy Dol, Angelina Geetha, “ A Learning Transition from Machine Learning to Deep Learning: A Survey”, IEEE International Conference on Emerging Techniques in Computational Intelligence (ICETCI),2021.
- [3] Yakun Li , Lei Hou , And Juanzi Li, “Preference-aware Graph Attention Networks for Cross-Domain Recommendations with Collaborative Knowledge Graph”, ACM Transactions on Information Systems, Vol. 41, No. 3, Article 80, February 2023.
- [4] Leigh Eytan, Veronika Bogina, Irad Ben-Gal, Noam Koenigstein, “KPAR: Knowledge-Aware Path-Based Attentive Recommender with Interpretability”, ACM Transactions on Recommendation System, ACM 2770-6699/2024/6-Art, 2024.
- [5] Pisol Ruenin, Morakot Choetkiertikul, Akara Supratak, Suppawong Tuarob, “TeReKG: A Temporal Collaborative Knowledge Graph Framework for Software Team Recommendation”, Knowledge-Based Systems 289, Elsevier B.V., 2024.
- [6] Qing He , Songyan Liu , and Yao Liu, “Optimal Recommendation Models Based on Knowledge Representation Learning and Graph Attention Networks”, IEEE Access, 2023.
- [7] Houssam Razouk , Xing Lan Liu, and Roman Kern, “Improving FMEA Comprehensibility via Common-Sense Knowledge Graph Completion Techniques”, IEEE Access, 2023.
- [8] Keyvan Amiri Elyasi, Diana Sola, Christian Meilicke, Han van der Aa, Heiner Stuckenschmidt, “Knowledge Graph Completion for Activity Recommendation in Business Process Modeling”, Springer KI - Künstliche Intelligenz,2024.
- [9] Disha Purohit, Yashrajsinh Chudasama, María-Esther Vidal, “VANILLA: Validated Knowledge Graph Completion - A Normalization-Based Framework for Integrity, Link Prediction, and Logical Accuracy”, Knowledge-Based Systems-325, Elsevier B.V. 2025.
- [10] David Chapela-Campa, Ismail Benchekroun, Opher Baron, Marlon Dumas, Dmitry Krass, Arik Senderovich, “A Framework for Measuring the Quality of Business Process Simulation Models”, Elsevier, 2024.
- [11] Sola, D., Meilicke, C., van der Aa, H., & Stuckenschmidt, H. “On the Use of Knowledge Graph Completion Methods for Activity Recommendation in Business Process Modeling”, book chapter in Business Process Management Workshops,2021.
- [12] Model collection of the BPM Academic Initiative, [http:// bpmai.org/](http://bpmai.org/)
- [13] Sola D, Warmuth C, Schäfer B, Badakhshan P, Rehse JR, Kampik T, “ Sap signavio academic models: a large process model dataset”, International Conference on Process Mining, Springer, 2022.
- [14] Rossi A, Barbosa D, Firmani D, Matinata A, Meriardo P, “Knowledge graph embedding for link prediction: A comparative analysis”. ACM Trans Knowl Discov Data (TKDD) 15(2):1–49, 2021.
- [15] Toutanova K, Chen D, “Observed versus latent features for knowledge base and text inference”. In: Proceedings of the 3rd workshop on continuous vector space models and their compositionality, pp. 57–66, 2015.
- [16] Bordes A, Usunier N, García-Durán A, Weston J, Yakhnenko O, “Translating embeddings for modeling multi-relational data.” NIPS. pp. 2787–2795, 2013.
- [17] Jiajun Liu, Wenjun Ke, Peng Wang, Jiahao Wang, Jinhua Gao, Ziyu Shang, Guozheng Li, Zijie Xu, Ke Ji and Yining Li., “Fast and Continual Knowledge Graph Embedding via Incremental LoRA”, Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24),2024.
- [18] Roderick van der Weerdt, Victor de Boer, Ronald Siebes and Frank van Harmelen, “Updating Knowledge Graph Embeddings by Intermediate Estimations on Numerical Attributes”, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024),2024.
- [19] Mercy Dol, Angelina Geetha,” Information to Inference –A Process Flow using Knowledge Graph, Journal-Mathematical Statistician and Engineering Applications, ISSN: 2094-0343 2326-9865, Page Number: 188-197, Publication Issue: Vol 71 No. 3s2,2022.