SMVKGC: A Runtime Plug-in for Streaming Knowledge Graph Construction via Inductive Multi-View Clustering

Jui-Chien Lin[1,*], Mohammad Mohammadi Amiri[1] and Oshani Seneviratne[1]

[1]Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Knowledge graph (KG) construction from streaming data poses significant challenges, particularly in efficiently integrating incoming entities that often arrive without known connections to existing nodes. This dynamic setting complicates critical graph maintenance tasks such as entity resolution and community detection, as new entities must be appropriately placed within the existing graph structure. Most current methods are designed for static graphs and rely on complete graph structure, requiring full model retraining when new entities arrive and resulting in prohibitive computational costs for real-time applications. To address these limitations, we propose Streaming Multi-View Knowledge Graph Clustering (SMVKGC), a novel framework designed as a runtime plug-in for KG construction pipelines that leverages multi-view graph representations to efficiently assign streaming entities to existing clusters without requiring graph structure information or model retraining. Our approach employs view-specific Graph Neural Networks (GNNs) to capture local neighborhood structures within each view, where a view is defined as a distinct relation type between nodes, and integrates these representations using a Transformer-based encoder with contrastive learning objectives to produce discriminative embeddings. Crucially, a lightweight projector network approximates the full GNN-encoder pipeline using only node features, enabling rapid inference for streaming entities. Once embeddings are generated, the system either assigns incoming entities to existing clusters or performs lightweight re-clustering over the expanded node set, achieving substantial runtime savings since clustering is computationally negligible relative to model retraining. We evaluate SMVKGC on six benchmark datasets (ACM, DBLP, IMDB, Texas, Chameleon, and Wisconsin), where it achieves competitive clustering performance across standard metrics (NMI, ARI, ACC, F1) while reducing inference time by orders of magnitude compared to retraining-based baselines.

Keywords

Knowledge Graph Construction, Streaming Data, Multi-View Graph Clustering, Entity Assignment

1. Introduction

Clustering is a fundamental unsupervised learning technique for discovering inherent structures in unlabeled data, where labeled examples are often scarce or costly to obtain [1]. While traditional methods such as K-means are effective for vectorized data, they fail to exploit the complex structural dependencies encoded in graph-structured data [2, 3, 4]. Graph clustering addresses this by grouping nodes based on both topological connectivity and node attributes, a paradigm that has proven valuable in applications such as social network analysis [5] and knowledge graph construction and maintenance [6].

Knowledge graph (KG) construction pipelines convert raw data from multiple sources into semantically rich graphs through sequential steps such as entity extraction, relation mining, schema inference, and quality control [7, 8]. Recent integration of large language models (LLMs) has further streamlined entity recognition and relation classification [9, 10]. Two core maintenance tasks within these pipelines are entity resolution, which identifies and consolidates duplicate entities through similarity-based matching, and community detection, which groups entities into meaningful clusters to support link prediction and improve graph density [11, 12].

Extending KG construction to streaming scenarios shifts the focus to incremental processing that preserves operational continuity for querying and publishing [13, 14, 15, 16]. However, static GNN-based embedding models dominate these pipelines yet introduce critical inefficiencies: their transductive nature assumes a complete graph during training, preventing direct embedding of new entities since

KGC@ESWC 2026 — Seventh International Workshop on Knowledge Graph Construction
*Corresponding author.
linj26@rpi.edu (J. Lin); mamiri@rpi.edu (M. M. Amiri); senevo@rpi.edu (O. Seneviratne)

GNN propagation relies on existing neighborhoods [17]. This forces either full model retraining or fallback to feature-only similarity metrics that discard relational context, both of which scale poorly and create bottlenecks in high-volume streaming settings.

Clustering serves as a valuable strategy in streaming KG pipelines by enabling incremental entity grouping that narrows entity resolution to within-cluster duplicate searches and confines edge prediction to detected communities [18, 17]. Once a new entity is assigned to a cluster, its community membership is identified, enabling subsequent edge connections with existing nodes. However, static clustering methods require model updates before matching, whereas inductive alternatives support rapid assignment without retraining. Our proposed approach, Streaming Multi-View Knowledge Graph Clustering (SMVKGC), targets this gap by enabling efficient multi-view entity assignment in streams without retraining.

Several recent works have advanced the state of the art in Multi-View Knowledge Graph Clustering (MVKGC). O2MAC [19] learns embeddings from a primary view and reconstructs auxiliary views. MCGC [2] constructs a consensus graph via smoothing and graph-level contrastive loss. NGCE [3] balances ego-node and neighborhood features for robust clustering across homophilous and heterophilous graphs. Shen et al. [20] dynamically balances view contributions through a self-supervised mechanism, while Khan and Kleinsteuber [21] integrates clustering priors into heterogeneous network representations. MVGRL [22] maximizes mutual information across structural views, and Kang et al. [23] extends multi-view subspace clustering to large-scale settings via structured bipartite graphs. Despite these advances, all such methods assume static graphs where nodes and edges remain fixed during training [2, 3, 19, 20, 21, 22, 24, 23, 25]. Since existing MVKGC methods are predominantly transductive, they cannot embed unseen nodes without full model retraining, introducing latency that is unsuitable for real-time entity resolution and community detection in streaming knowledge graphs [5, 26].

To enable efficient entity assignment and community detection in streaming knowledge graphs, we propose **Streaming Multi-View Knowledge Graph Clustering (SMVKGC)**, a unified framework that learns cluster representations from multi-view training data and assigns newly arriving entities to existing clusters *without* retraining or graph reconstruction. In this work, we adopt a graph-structural definition of *view*: each view corresponds to a distinct relation type between nodes in the knowledge graph (e.g., co-authorship links and citation relations in a scholarly KG). This differs from the modality-based definition commonly used in the broader multi-view learning literature, where views refer to heterogeneous data sources such as text, images, or tabular metadata. The framework operates exclusively on textual node features. SMVKGC employs view-specific Graph Neural Networks (GNNs) for local feature aggregation, a Transformer encoder for cross-view fusion, and contrastive learning to produce robust entity embeddings. A lightweight projector network then enables inductive assignment of streaming entities based solely on their features.

Motivating Scenario. Consider a scholarly knowledge graph that continuously ingests publications from a venue such as the ACM Digital Library, where papers and subject areas are connected through multiple relation types such as co-authorship and citation links, each constituting a distinct view. Newly submitted papers arrive without established graph connections, making transductive GNN-based clustering methods inapplicable at ingestion time, as their message passing mechanism requires a fixed and complete neighborhood structure during training. Forcing full model retraining upon each arrival introduces unacceptable latency in high-throughput settings.

SMVKGC addresses this challenge by training a lightweight projector network that maps raw textual features directly to the unified embedding space, thereby enabling immediate cluster assignment for streaming entities without requiring graph connectivity. Designed as a runtime plug-in for knowledge graph construction pipelines, SMVKGC supports entity resolution and community detection under streaming conditions: once the projector generates embeddings for an incoming entity, the system either assigns it to the nearest existing cluster or performs lightweight re-clustering over the expanded node set. Since clustering is computationally negligible compared to model retraining, this design delivers substantial runtime savings without sacrificing integration quality. As demonstrated in Section 4.3, the projector achieves inference times several orders of magnitude faster than retraining-based alternatives, providing a solid foundation for downstream tasks in dynamic knowledge graph pipelines.

The main contributions of this work are as follows:

- A unified multi-view clustering framework with contrastive regularization and cross-view alignment, equipped with a projector network for inductive assignment of newly arriving entities.
- A lightweight runtime plug-in for knowledge graph construction pipelines that supports streaming entity resolution and community detection without model retraining or graph reconstruction.
- Comprehensive evaluations on six benchmark datasets (ACM [2], DBLP [2], IMDB [2], Texas [3], Chameleon [3], and Wisconsin [3]), demonstrating the robustness and efficiency of SMVKGC and its practical value for maintaining knowledge graph quality in streaming pipelines.

2. Related Work

MVKGC integrates information from multiple relational perspectives to produce more accurate node groupings than single-view approaches. Methods are broadly categorized into subspace-based techniques, which identify a shared low-dimensional space across views [27], and graph-based methods, which construct consensus graphs or align view-specific structures [28]. Early work such as MNE [29] and PMNE [30] learned unified representations across graph layers but did not incorporate node attributes.

Recent GNN-based methods have made significant progress on challenging issues such as noise, heterophily, and incomplete views. DMAC [31] employs learnable anchors for efficient deep clustering, IMGCGGR [7] applies global graph refinement to resolve cross-view inconsistencies, AMMGC [32] imputes missing attributes iteratively, RSEA-MVGNN [11] enhances noise-resistant aggregation, and DWGF [33] dynamically reweights views during fusion. Despite these advances, the majority of methods assume static graphs where all nodes and edges are available during training. Due to their transductive nature, new nodes cannot be directly embedded without model retraining, creating significant inefficiency in streaming scenarios. Our work addresses this gap by enabling inductive entity assignment in multi-view streams without retraining or graph reconstruction.

3. Methodology

This section details the SMVKGC framework, covering how view-specific information is aggregated and fused into unified embeddings, how newly arriving entities are inductively embedded without graph structure access, and how the framework integrates into streaming KG construction pipelines.

3.1. Notation

We define a multi-view graph consisting of V distinct views. Each view v has its own adjacency matrix $\boldsymbol{A}^{(v)} \in \mathbb{R}^{N\times N}$, where N denotes the number of nodes (with $|\mathscr{V}| = N$, and $\mathscr{V}$ representing the node set). All views share a common node feature matrix $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, where D denotes the feature dimension.

3.2. View-Specific Embedding Generation

For each view v, we employ a GNN, denoted as $\mathrm{GNN}^{(v)}$, to aggregate information from neighboring nodes, producing view-specific node embeddings. Formally, for node i with features $\boldsymbol{x}_i$ and neighbor set $\mathscr{N}_i^{(v)}$ under view v, the view-specific embedding is computed as:

$$\boldsymbol{h}_i^{(v)} = \mathrm{GNN}^{(v)}\Big(\boldsymbol{x}_i,\ \Big\{\boldsymbol{x}_j \mid j \in \mathscr{N}_i^{(v)}\Big\}\Big). \tag{1}$$

This aggregation step captures local structural information by incorporating features from neighboring nodes, where the edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ indicates a view-specific correlation. This process enriches the representation of node $\boldsymbol{x}_i$ with complementary contextual information from its local neighborhood, as illustrated in Figure 1.

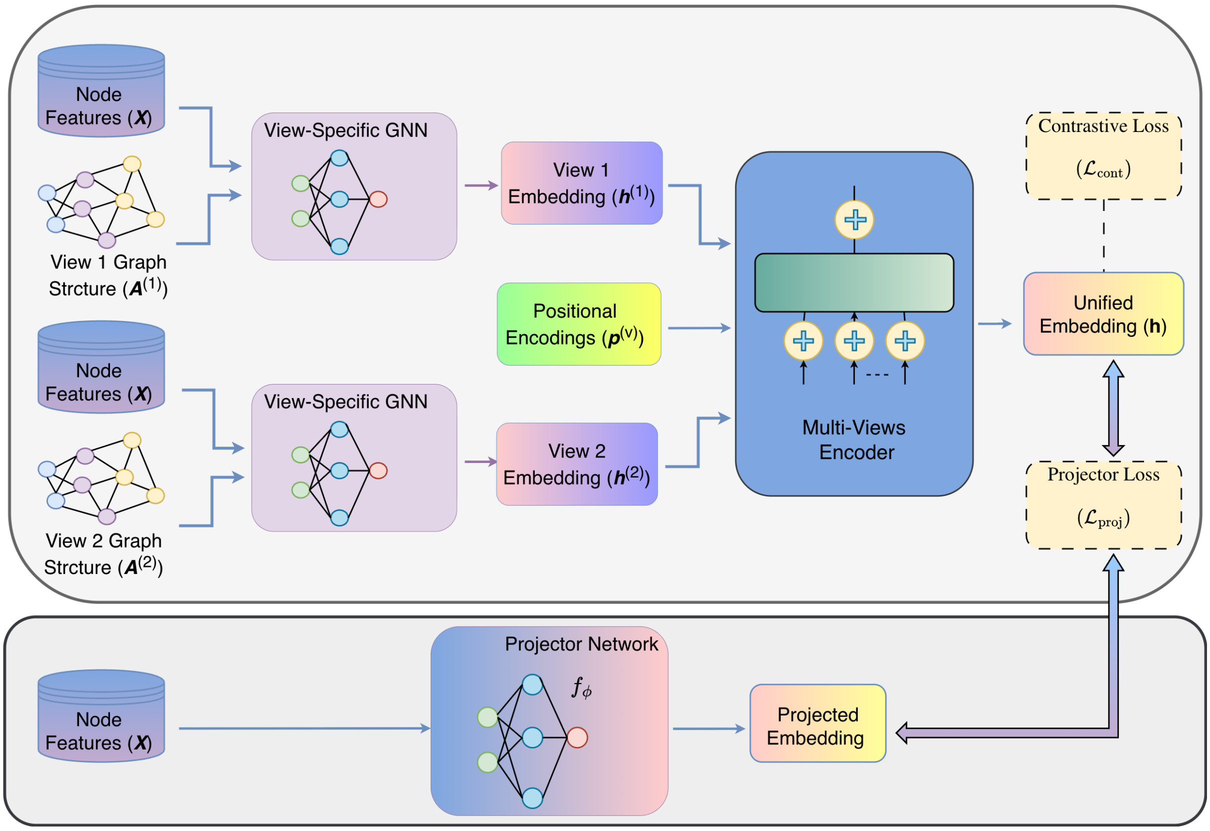

Figure 1: Overview of SMVKGC framework. The projector network is trained to simulate the combined functionality of the view-specific GNNs and multi-view encoder, enabling efficient embedding generation for streaming nodes. For clarity, only two views are shown in this illustration.

3.3. Multi-View Aggregation via Encoder

Following the view-specific graph learning phase, the next step is to aggregate the view-specific embeddings into a unified representation. We employ a transformer-based encoder architecture inspired by BERT [34], where the encoder takes as input the view-specific embeddings, each augmented with a learnable positional encoding $\boldsymbol{p}^{(v)}$ to distinguish between different views, along with a special learnable aggregation token $\boldsymbol{z}_{\text{agg}}$:

$$\boldsymbol{h}_i = \text{Encoder}\big[\boldsymbol{z}_{\text{agg}}, \boldsymbol{h}_i^{(1)} + \boldsymbol{p}^{(1)}, \boldsymbol{h}_i^{(2)} + \boldsymbol{p}^{(2)}, \dots, \boldsymbol{h}_i^{(V)} + \boldsymbol{p}^{(V)}\big]_{[0]}, \tag{2}$$

where $[\cdot]_{[0]}$ denotes the output at the position of the aggregation token. Since the transformer's self-attention mechanism allows every token in the sequence to attend to all others, $\boldsymbol{z}_{\text{agg}}$ aggregates information from all views and serves as the unified embedding $\boldsymbol{h}_i$, providing a comprehensive representation of node i.

3.4. Contrastive Learning

We adopt contrastive learning to encourage embeddings that promote cluster separability. We define positive samples as the union of all neighbors across all views. Negative samples are all remaining nodes excluding node i itself and its positive samples:

$$\begin{aligned} \mathcal{N}_i^+ &= \bigcup_{v=1}^{V} \mathcal{N}_i^{(v)}, \\ \mathcal{N}_i^- &= \mathcal{V} \setminus \big(\{i\} \cup \mathcal{N}_i^+\big). \end{aligned} \tag{3}$$

The contrastive loss function aims to minimize the distance between node embeddings and their positive samples while maximizing the distance from negative samples. For each node i, we define the node-level loss using a temperature-scaled similarity metric:

$$\mathscr{L}_i = -\log \frac{\sum_{j \in \mathcal{N}_i^+} \exp(\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j)/\tau)}{\sum_{k \neq i} \exp(\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_k)/\tau)}, \tag{4}$$

where $\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j)$ denotes cosine similarity, and τ represents the temperature parameter that controls the concentration level of the contrastive objective. The overall contrastive loss $\mathscr{L}_{\mathrm{cont}}$ is obtained by:

$$\mathscr{L}_{\mathrm{cont}} = \frac{1}{|\mathscr{V}|} \sum_{i \in \mathscr{V}} \mathscr{L}_i, \tag{5}$$

3.5. Projector Network

To enable inductive clustering for streaming entities that lack neighborhood information, we introduce a projector network that learns to reconstruct the fused embeddings $\boldsymbol{h}_i$ from raw features $\boldsymbol{x}_i$ alone. The projector network f_ϕ is a multi-layer perceptron (MLP) that minimizes the squared L2 reconstruction loss:

$$\mathscr{L}_{\mathrm{proj}} = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{h}_i - f_\phi(\boldsymbol{x}_i)\|_2^2. \tag{6}$$

This projector network approximates the combined functionality of the view-specific GNNs and the multi-view encoder using only the original node features, thereby enabling the generation of high-quality embeddings for streaming nodes without requiring graph structure information or model retraining, as illustrated in Figure 1.

3.6. Joint Optimization

To jointly optimize the framework, we combine the contrastive and projector losses:

$$\mathscr{L}_{\mathrm{total}} = \mathscr{L}_{\mathrm{cont}} + \lambda \mathscr{L}_{\mathrm{proj}}, \tag{7}$$

where λ is a hyperparameter that controls the trade-off between the contrastive learning objective and the projector reconstruction objective. This joint optimization ensures that: (i) the view-specific GNNs effectively capture neighborhood structure, (ii) the encoder successfully integrates embeddings from all views into a unified representation, and (iii) the projector network accurately approximates the combined GNN-encoder pipeline. Through this unified framework, our model achieves the dual objectives of learning discriminative multi-view graph embeddings while maintaining the capability to efficiently generate embeddings for streaming nodes without requiring explicit graph connectivity information. The complete SMVKGC architecture is illustrated in Figure 1.

3.7. Streaming Inference and KG Construction Integration

SMVKGC acts as a lightweight runtime plug-in trained on the initial multi-view KG, where each view corresponds to a distinct relation type. During inference, a new entity with feature vector $\boldsymbol{x}_{\mathrm{new}}$ is mapped to an embedding $\boldsymbol{h}_{\mathrm{new}} = f_\phi(\boldsymbol{x}_{\mathrm{new}})$ via the projector network, and assigned to the nearest cluster centroid or integrated through fast re-clustering over the expanded node set, all without retraining or graph reconstruction.

This design directly supports two core KG construction tasks on streaming entities: (i) **entity resolution**, by restricting duplicate checks to intra-cluster candidates via embedding similarity, and (ii) **community detection**, by placing nodes into coherent groups for improved link prediction and graph densification.

4. Experiments

In this section, we evaluate the effectiveness and efficiency of SMVKGC. We first compare SMVKGC against state-of-the-art static multi-view graph clustering baselines on standard clustering metrics. Finally, we conduct a streaming experiment to demonstrate the advantage of our projector-based inductive inference for dynamic knowledge graph construction, where new entities arrive continuously without graph connectivity.

4.1. Datasets and Setup

We evaluate our method on six multi-view graph datasets: **ACM**, **DBLP**, **IMDB**, **Texas**, **Chameleon**, and **Wisconsin**. Table 1 summarizes the statistics of these datasets, including the number of nodes, feature dimensionality, number of views (i.e., the number of adjacency matrices $\boldsymbol{A}^{(v)}$), and the number of classes. The class labels serve as ground truth for evaluation, enabling us to assess the quality of the learned embeddings through clustering performance metrics.

Table 1
The statistics information of datasets.

Dataset	Nodes	Features	Views	Clusters
ACM	3025	1870	2	3
DBLP	4057	334	3	4
IMDB	4780	1232	3	3
Texas	183	1703	2	5
Chameleon	2277	3132	2	5
Wisconsin	251	1703	2	5

4.2. Benchmark

We compare SMVKGC against several static, transductive baselines to demonstrate that it maintains competitive clustering performance. Table 2 presents results across all datasets. While SMVKGC does not achieve state-of-the-art on every benchmark, it offers a critical advantage that all baselines lack: the ability to efficiently incorporate streaming nodes without graph structure information or model retraining, which is illustrated in Section 4.3.

Our method achieves competitive performance on ACM and DBLP, even outperforming several existing baselines. On the remaining datasets, our method exhibits moderate performance; however, these datasets are inherently challenging, as evidenced by the relatively modest performance of most existing methods on these benchmarks. We further note that many baseline methods [35, 36, 20, 37, 38, 19, 39] incorporate pseudo cluster labels into their objective functions during training, effectively injecting prior knowledge about the cluster structure that would not be available in realistic scenarios. In contrast, our approach does not rely on such supervision, making it more applicable to real-world settings where ground-truth cluster assignments are unknown. Despite this stricter setting, SMVKGC maintains consistent performance while uniquely supporting streaming inference, a capability not addressed by any of the baselines.

4.3. Clustering for Streaming Data

In this experiment, we evaluate whether the projector module can effectively generate high-quality embeddings for newly arriving streaming nodes. We conduct this evaluation on **ACM**, **DBLP**, **IMDB**, and **Chameleon**, as these datasets contain substantially more samples than **Texas** and **Wisconsin**. Specifically, we use 98% of the nodes from each dataset to train SMVKGC, reserving the remaining 2% as the test set to simulate streaming arrivals. We partition the test set into batches of 20 nodes each and sequentially feed these batches to the model, simulating a realistic streaming data scenario.

Table 2
Comparison with existing approaches.

Method	ACM				DBLP				IMDB			
	NMI (%)	ARI (%)	ACC (%)	F1 (%)	NMI (%)	ARI (%)	ACC (%)	F1 (%)	NMI (%)	ARI (%)	ACC (%)	F1 (%)
AGCN [35]	68.38	74.20	90.06	90.58	39.68	42.29	73.26	72.80	0.29	0.66	54.73	25.14
AGE [36]	69.47	75.47	91.11	91.08	37.22	38.49	69.88	68.59	7.08	13.85	56.32	45.44
BMGC [20]	75.87	81.25	93.37	93.40	78.21	83.53	93.17	92.72	8.62	6.57	46.62	42.01
DCRN [37]	70.44	76.00	91.27	91.23	45.56	48.28	76.51	76.32	0.3	1.0	54.35	26.55
DuaLGR [38]	73.47	78.76	92.40	92.44	75.78	81.85	92.46	91.91	5.08	4.51	45.36	44.20
O2MAC [19]	62.40	66.24	87.07	87.25	70.38	75.10	89.62	89.01	0.2	10.00	43.10	34.33
VGMGC [39]	68.48	71.38	89.12	88.90	77.71	83.06	92.93	92.34	6.65	12.86	52.80	43.34
SMVKGC (ours)	69.44	75.02	91.01	91.06	70.64	76.57	90.12	89.58	0.04	0.05	50.31	28.92

Method	Texas				Chameleon				Wisconsin			
	NMI (%)	ARI (%)	ACC (%)	F1 (%)	NMI (%)	ARI (%)	ACC (%)	F1 (%)	NMI (%)	ARI (%)	ACC (%)	F1 (%)
AGCN [35]	15.4	18.1	68.85	32.9	10.72	6.91	39.26	28.52	6.4	6.8	49.3	24.9
AGE [36]	12.67	12.16	46.99	35.92	14.39	7.01	39.39	32.58	20.16	13.30	47.01	36.43
BMGC [20]	7.88	7.07	36.50	27.15	10.64	5.94	31.86	31.77	14.99	9.95	42.48	36.88
DCRN [37]	14.63	24.59	56.83	30.53	7.45	2.0	35.70	23.43	12.02	8.08	46.61	32.97
DuaLGR [38]	32.13	23.29	55.74	41.05	18.22	12.31	41.24	40.51	42.07	35.25	58.57	48.11
O2MAC [19]	4.10	5.29	49.18	21.27	8.95	1.54	32.15	23.41	5.67	3.02	45.82	28.24
VGMGC [39]	23.68	14.60	49.73	39.74	17.35	10.87	39.22	33.87	44.82	37.98	58.17	50.31
SMVKGC (ours)	7.61	10.00	40.44	27.62	9.83	4.19	35.22	32.45	12.95	8.65	41.43	31.90

Figure 2 presents the overall clustering performance (aggregated across both training and test nodes) after incorporating each new batch of streaming data, where the solid lines (streaming) represent the projector-based streaming setting and the dashed lines represent full model retraining. The curves remain stable across batches, indicating that newly arriving nodes are correctly assigned to existing clusters without degrading overall performance. Table 3 reports the per-batch clustering performance on newly arriving nodes only, along with a runtime comparison between streaming inference and full retraining. Note that we employ early stopping during retraining, which explains the non-linear scaling of retraining time.

SMVKGC achieves strong streaming performance on **ACM**, **DBLP**, and **IMDB**, with inference times several orders of magnitude faster than retraining. On **Chameleon**, clustering quality remains limited; however, this is consistent with the difficulty of this dataset, as most existing static methods also struggle on this benchmark.

5. Conclusion

We proposed SMVKGC, a streaming multi-view graph clustering framework designed as a runtime plug-in for KG construction pipelines. By combining view-specific GNN aggregation and Transformer-based fusion with a lightweight projector network, SMVKGC enables inductive entity assignment without graph structure access, model retraining, or graph reconstruction. Evaluations on six benchmarks demonstrate competitive static clustering performance alongside inference speedups of several orders of magnitude over retraining-based baselines, making it suitable for high-throughput streaming scenarios. The cluster assignments directly support entity resolution and community detection, two core KG maintenance tasks. Limitations include the assumption of feature availability for incoming entities and reduced structural expressiveness in highly heterophilic graphs. Future work will explore lightweight edge prediction, heterogeneous node types, and incremental model update strategies. The code is publicly available[1].

Declaration on Generative AI

The authors have used Generative AI tools for language polishing and grammar checking. All content has been reviewed and verified by the authors.

[1] https://anonymous.4open.science/r/smvkgc-743D/

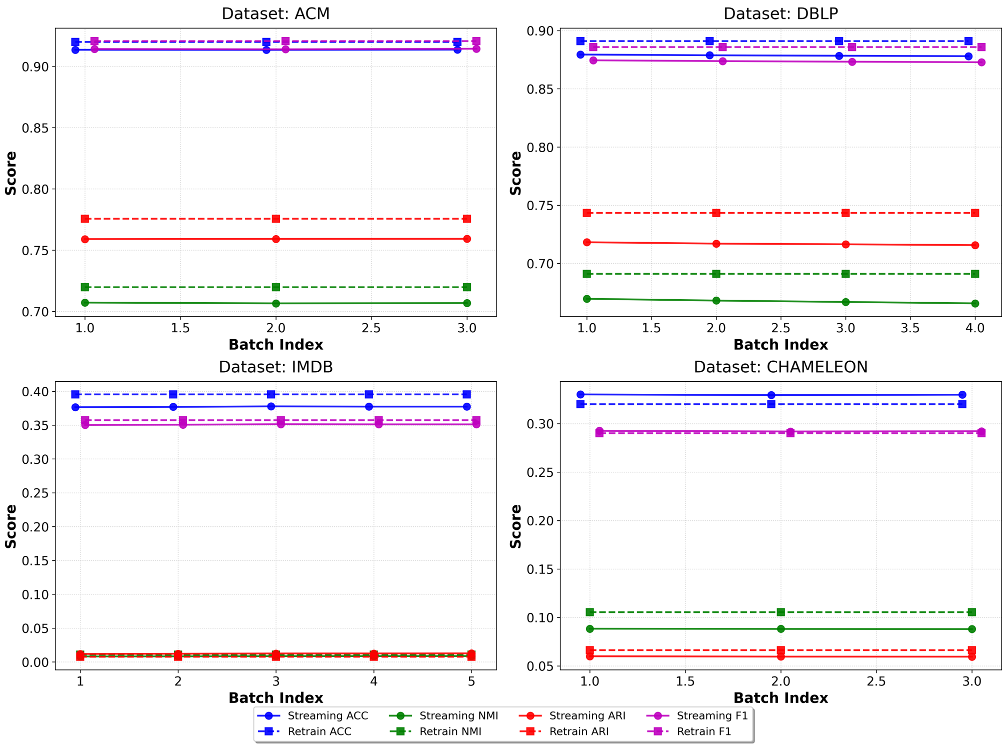

Figure 2: Overall clustering performance on the complete dataset after sequentially incorporating batches of streaming nodes. The projector-based approach achieves competitive clustering quality across all metrics (ACC, NMI, ARI, F1), demonstrating its effectiveness for dynamic graph scenarios.

Table 3
Clustering performance and runtime comparison on streaming nodes. Results show clustering metrics for each sequential batch of newly arriving nodes, along with cumulative runtime in seconds. The projector-based streaming approach achieves competitive clustering performance while requiring substantially less computation time compared to full model retraining.

Dataset	ACM			IMDB				
Batch id	1	2	3	1	2	3	4	5
Size	20	20	20	20	20	20	20	16
NMI (%)	84.25	74.15	75.7	24.12	8.56	14.95	10.83	9.59
ARI (%)	83.29	79.13	80.37	5.01	8.79	15.82	10.33	9.34
ACC (%)	95	92.3	93.33	55	52.5	53.33	47.5	44.79
F1 (%)	95.60	90.5	93.33	50.12	45.45	47.41	43.42	40.49
Streaming Time (s)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Retrain Time (s)	56.134	69.131	66.883	325	339	329	353	320

Dataset	Chameleon			DBLP			
Batch id	1	2	3	1	2	3	4
Size	20	20	6	20	20	20	20
NMI (%)	33.68	20.01	15.86	55.70	49.32	49.54	50.04
ARI (%)	2.07	1.51	1.16	41.93	46.65	51	52.09
ACC (%)	40	35	34.78	75	75	76.69	77.5
F1 (%)	38.11	31.95	32.44	73.38	73.29	74.01	74.95
Streaming Time (s)	0.002	0.002	0.001	0.002	0.002	0.002	0.002
Retrain Time (s)	81	67	335	1545	1020	785	875

References

[1] G. Gan, C. Ma, J. Wu, Data clustering: theory, algorithms, and applications, SIAM, 2020.

[2] E. Pan, Z. Kang, Multi-view contrastive graph clustering, Advances in neural information processing systems 34 (2021) 2148–2159.

[3] Y. Ren, J. Ke, Z. Wen, T. Wu, Y. Yang, X. Pu, L. He, Multi-view graph clustering via node-guided contrastive encoding, in: Forty-second International Conference on Machine Learning, 2025.

[4] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic web 8 (2016) 489–508.

[5] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, et al., A comprehensive survey on community detection with deep learning, IEEE transactions on neural networks and learning systems 35 (2022) 4682–4702.

[6] Y. Yang, H. Wang, Multi-view clustering: A survey, Big data mining and analytics 1 (2018) 83–107.

[7] L. Zeng, S. Yao, Y. Huang, Y. Cheng, Y. Qian, Improved multi-view graph clustering with global graph refinement, Remote Sensing 17 (2025) 3217.

[8] H. Alani, D. Dupplaw, J. Sheridan, K. O'Hara, J. Darlington, N. Shadbolt, C. Tullo, Unlocking the potential of public sector information with semantic web technology, in: International Semantic Web Conference, Springer, 2007, pp. 708–721.

[9] J. Bai, W. Fan, Q. Hu, Q. Zong, C. Li, H. T. Tsang, H. Luo, Y. Yim, H. Huang, X. Zhou, et al., Autoschemakg: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora, arXiv preprint arXiv:2505.23628 (2025).

[10] H. Bian, Llm-empowered knowledge graph construction: A survey, arXiv preprint arXiv:2510.20345 (2025).

[11] J. Chen, L. Shi, B. Chen, Rsea-mvgnn: Multi-view graph neural network with reliable structural enhancement and aggregation, Information Fusion 121 (2025) 103143.

[12] S. Fortunato, Community detection in graphs, Physics reports 486 (2010) 75–174.

[13] G. Weikum, X. L. Dong, S. Razniewski, F. Suchanek, Machine knowledge: Creation and curation of comprehensive knowledge bases, Foundations and Trends in Databases 10 (2021) 108–490.

[14] G. A. Gesese, R. Biswas, M. Alam, H. Sack, A survey on knowledge graph embeddings with literals: Which model links better literal-ly?, Semantic Web 12 (2021) 617–647.

[15] C. Min, S. Bansal, J. Pan, A. Keshavarzi, R. Mathew, A. V. Kannan, Towards practical graphrag: Efficient knowledge graph construction and hybrid retrieval at scale, arXiv preprint arXiv:2507.03226 (2025).

[16] M. Sankaradas, R. K. Rajendran, S. T. Chakradhar, Streamingrag: Real-time contextual retrieval and generation framework, arXiv preprint arXiv:2501.14101 (2025).

[17] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Advances in neural information processing systems 30 (2017).

[18] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of statistical mechanics: theory and experiment 2008 (2008) P10008.

[19] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, B. Wang, One2multi graph autoencoder for multi-view graph clustering, in: proceedings of the web conference 2020, 2020, pp. 3070–3076.

[20] Z. Shen, H. He, Z. Kang, Balanced multi-relational graph clustering, in: Proceedings of the 32nd ACM International Conference on Multimedia, 2024, pp. 4120–4128.

[21] R. A. Khan, M. Kleinsteuber, Cluster-aware heterogeneous information network embedding, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 476–486.

[22] K. Hassani, A. H. Khasahmadi, Contrastive multi-view representation learning on graphs, in: International conference on machine learning, PMLR, 2020, pp. 4116–4126.

[23] Z. Kang, Z. Lin, X. Zhu, W. Xu, Structured graph learning for scalable subspace clustering: From single view to multiview, IEEE Transactions on Cybernetics 52 (2021) 8976–8986.

[24] A. Tsitsulin, J. Palowitch, B. Perozzi, E. Müller, Graph clustering with graph neural networks, Journal of Machine Learning Research 24 (2023) 1–21.

[25] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, X. Peng, Completer: Incomplete multi-view clustering via contrastive prediction, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 11174–11183.

[26] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, P. S. Yu, Deep learning for community detection: progress, challenges and opportunities, arXiv preprint arXiv:2005.08225 (2020).

[27] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 586–594.

[28] F. Nie, J. Li, X. Li, et al., Self-weighted multiview clustering with multiple graphs., in: Ijcai, 2017, pp. 2564–2570.

[29] H. Zhang, L. Qiu, L. Yi, Y. Song, Scalable multiplex network embedding., in: IJCAI, volume 18, 2018, pp. 3082–3088.

[30] W. Liu, P.-Y. Chen, S. Yeung, T. Suzumura, L. Chen, Principled multilayer network embedding, in: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2017, pp. 134–141.

[31] B. Wang, C. Zeng, M. Chen, X. Li, Towards learnable anchor for deep multi-view clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, 2025, pp. 21044–21052.

[32] B. Zhao, Q. Wang, Z. Ding, Q. Gao, Attribute-missing multi-view graph clustering, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 25832–25841.

[33] Y. Ren, J. Pu, C. Cui, Y. Zheng, X. Chen, X. Pu, L. He, Dynamic weighted graph fusion for deep multi-view clustering, IJCAI, 2024.

[34] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[35] Z. Peng, H. Liu, Y. Jia, J. Hou, Attention-driven graph clustering network, in: Proceedings of the 29th ACM international conference on multimedia, 2021, pp. 935–943.

[36] G. Cui, J. Zhou, C. Yang, Z. Liu, Adaptive graph encoder for attributed graph embedding, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 976–985.

[37] Y. Liu, W. Tu, S. Zhou, X. Liu, L. Song, X. Yang, E. Zhu, Deep graph clustering via dual correlation reduction, in: Proceedings of the AAAI conference on artificial intelligence, volume 36, 2022, pp. 7603–7611.

[38] Y. Ling, J. Chen, Y. Ren, X. Pu, J. Xu, X. Zhu, L. He, Dual label-guided graph refinement for multi-view graph clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 8791–8798.

[39] J. Chen, Y. Ling, J. Xu, Y. Ren, S. Huang, X. Pu, Z. Hao, P. S. Yu, L. He, Variational graph generator for multiview graph clustering, IEEE Transactions on Neural Networks and Learning Systems 36 (2025) 11078–11091.