Adversarial Generation of Multi-View Tensor Graphs: A Preliminary Model and Encouraging Results

William Shiao
wshia002@ucr.edu
University of California Riverside
Riverside, CA, USA

Benjamin A. Miller
miller.be@northeastern.edu
Northeastern University
Boston, MA, USA

Kevin Chan
kevin.s.chan.civ@mail.mil
U.S. Army Research Laboratory
Adelphi, MD, USA

Paul Yu
paul.l.yu.civ@mail.mil
U.S. Army Research Laboratory
Adelphi, MD, USA

Tina Eliassi-Rad
eliassi@northeastern.edu
Northeastern University
Boston, MA, USA

Evangelos E. Papalexakis
epapalex@cs.ucr.edu
University of California Riverside
Riverside, CA, USA

ABSTRACT

In this work, we explore multi-view graph generation with deep generative models. We discuss some of the challenges associated with multi-view graph generation that make it a more difficult problem than traditional graph generation. We propose 3 different criteria for evaluating the quality of generated graphs: a graph-attribute-based, a classifier-based, and a tensor-based method. We also propose TenGAN, a baseline tensor-decomposition-based GAN to reduce the number of parameters required for multi-view graph generation. We evaluate its performance on 2 datasets—a Twitter football dataset and an EU airlines dataset. We find promising results on the football dataset—multi-view graphs generated by TenGAN are able to fool a classifier 30% of the time—but find that it performs poorly on the airlines dataset. We explore some reasons for this and discuss the limitations of our approach.

KEYWORDS

multi-view graphs, multi-view network generation, generative adversarial networks, tensor decomposition

ACM Reference Format:
William Shiao, Benjamin A. Miller, Kevin Chan, Paul Yu, Tina Eliassi-Rad, and Evangelos E. Papalexakis. 2022. Adversarial Generation of Multi-View Tensor Graphs: A Preliminary Model and Encouraging Results. In *Proceedings of MLoG Workshop at WSDM'22 (MLoG '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Multi-view graph generation can be seen as an extension of the traditional problem of graph generation. The typical problem statement for graph generation is to generate a graph with some desirable properties, often with the goal to structurally emulate another real-world graph without being exactly the same.

Graph generation models have been applied to a wide variety of network types, and have been especially shown to be useful in protein generation [14][26]. These models can be generally split into two types: statistical attribute-based generative models and deep generative models. Statistical generative models like the Erdős-Rényi [5], Barabási–Albert [2], and stochastic block models [13] allow a user to choose certain attributes of the graph. Some of these have also been extended to work for multi-view graphs [16][22][17].

In contrast, many deep generative models can learn directly one or more input graphs. This allows them to mimic attributes of the input dataset without having to explicitly define them. Examples of these include GraphRNN [26], NetGAN [3], and LGGAN [6].

There have also been several models to generate multi-scale graphs. The key difference between a multi-view and multi-scale graph is that a multi-view graph contains the same nodes with different edges in each view, while a multi-scale graph typically contains representations of the same underlying graph at different resolutions (different number of nodes) in each layer. Misc-GAN [27] generates a multi-scale graph before collapsing it into a standard graph, and DMGNN [20] predicts multi-scale graphs from previous ones. However, to the best of our knowledge, there have been no deep generative models for multi-*view* graphs.

This is a especially challenging task because of the higher dimensionality of the data (there can be many views) and the varying relationships between the views. It is also non-trivial to evaluate the quality of the generated multi-view graphs. In this work, we attempt to solve some of these issues and propose TenGAN, a preliminary GAN-based method to generate multi-view graphs.

Our contributions include:

- **Novel method**: We propose a novel GAN-based method to generate multi-view graphs that uses the canonical polyadic decomposition (CPD) to reduce the number of parameters required.
- **Evaluation criteria**: We propose 3 different evaluation metrics for multi-view graph generation.
- **Preliminary experimentation**: We conduct preliminary experiments on two different datasets to evaluate the performance of our method and provide a baseline for future works.

2 BACKGROUND

This section provides the necessary background information. Table 1 contains the description of the symbols used in this paper.

MLoG '22, Feburary 2022, Phoenix, AZ USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

Notation	Meaning
$\mathcal{X}, \mathbf{X}, \mathbf{x}, x$	Tensor, matrix, vector, scalar
$\|\mathbf{X}\|_F$	Frobenius norm
$g^{(i)}$	The i-th view of a multi-view graph g
$\mathbf{X}_i$	The i-th row of the matrix $\mathbf{X}$
$\circ$	Outer product
$\mathcal{T}_{i,j,k}$	The value in the i-th, j-th, and k-th entry along the 1st, 2nd, and 3rd mode of the tensor

Table 1: Table of symbols and their description

2.1 Tensors & Decompositions

Networked data (e.g., a social network) can be represented in many different formats. One of the most common formats is to use an adjacency matrix. In a similar manner, a multi-view graph can be viewed as a third-order tensor where $\mathcal{T}_{i,j,k} = w$ if there is an edge of weight w (or 1 in the case of an unweighted graph) between node i and node j in view k. One advantage of storing multi-view graphs in the tensor format is that it allows us to use tensor decomposition methods.

One of the most common tensor decomposition methods is the CANDECOMP/PARAFAC or Canonical Polyadic Decomposition (CPD). Given an integer r, the CPD decomposes a tensor $\mathcal{T}$ into the sum of r outer products of vectors. While the CPD can be applied to a tensor of any order, we focus on the third-order case in this work. The CPD can be written (for a third-order tensor) as:

$$\mathcal{T} \approx \sum_{i=1}^{r} \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$$

where $\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i$ are the factor matrices. It is convention to write the factors as matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, which consist of the corresponding vectors horizontally stacked. For example, the i-th column of $\mathbf{A}$ would be $\mathbf{a}_i$.

CPD is usually solved by using alternating least squares to minimize the Frobenius norm of the error: $\left\|\mathcal{T} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\right\|_F$. We can then divide this number by $\|\mathcal{T}\|_F$ to calculate a normalized error value, which provides a heuristic for how well a rank r decomposition can approximate a given tensor.

2.2 Generative Adversarial Networks

Goodfellow *et al.* [9] introduced Generative Adversarial Networks (GANs) introduced in 2014. A GAN consists of two components: a generator and a discriminator. The generator takes in some random noise $\mathbf{z}$ as input and outputs data (in this case, a multi-view graph). The discriminator takes in data and outputs the probability that the data was drawn from the dataset, rather than from the generator.

These two networks are then optimized together in unison, with the goal of increasing the quality of generated samples and improving the discriminator's ability to distinguish the generated data from the real data.

3 PROBLEM FORMULATION

We consider the following problem:

> **Given** a set of multi-view graphs G, **generate** a set of graphs G' that are not identical to G, but possess similar graph attributes to and are indistinguishable (to a classifier) from them.

Although this may seem like a simple problem, there are several challenges that we must address in order to solve this problem.

Challenge 1: Sampling from Multi-view Graphs. Many datasets consist of a single graph with multiple views. However, since we are emulating a set of multi-view graphs, we need many smaller multi-view graphs to form a distribution. Therefore, we need a method that samples smaller subgraphs from a larger multi-view graph. We defer a detailed evaluation of different methods to a future work.

Challenge 2: Inadequate Evaluation Criteria. Before we can decide on an appropriate method to accomplish the aforementioned sampling task, we must determine our evaluation criteria. This is challenging because we need to consider not only the graph attributes of each view, but also the relationships between each view. This means that many of the graph evaluation metrics commonly used in graph generation are insufficient for multi-view graph generation. Many existing works in multi-view graph generation rely on statistical properties to mirror

Challenge 3: Large Number of Parameters. If we naively attempt to generate a multi-view graph, the number of parameters required will explode. This is because we need $\mathcal{O}(k \times n^2)$ parameters to generate an adjacency tensor for a multi-view graph with k views and n nodes. We propose a method TenGAN (described in Section 4.2) that generates a compressed representation of the tensor to solve this problem.

4 PROPOSED METHOD

We propose TenGAN, a GAN-based model that first generates factors of the CPD, then uses those to generate the adjacency tensor. To train our model, we first sample sub-multi-view graphs from the dataset, as described in Section 4.1 below. Then, we train our GAN on the sampled multi-view graphs. Finally, we evaluate our model with the metrics described in Section 4.3.

4.1 Sampling

Many generative models require multiple input samples, rather than a single example. For example, LGGAN [6] is trained on 2-hop and 3-hop egonets extracted from the original source graph. We extend this sampling method to work for multi-view graphs by taking all possible egonets across each view and using the induced subgraph across the same nodes in the other views. While there are many other multi-view sampling methods, like those described by Gjoka *et al.* [8] and Interdonato *et al.* [15], we leave a detailed evaluation of them to a future work.

4.2 Architecture

Our model is a GAN and consists of a generator network and a discriminator network. The discriminator uses the max pool of several Graph Convolutional Networks (GCNs) [18] (one per view) followed by a fully-connected layer to predict if a sample is generated or drawn from the original real dataset. A diagram of our architecture is shown in Figure 1.

4.2.1 Generator Architecture. The generator uses a shared feature extractor layer and splits into separate networks, each of which generates a different factor in the CPD. This is similar to the BRGAN-B [24] architecture, but uses the higher-order CPD instead of the SVD. After generating the factors, we calculate the sum of the outer products of vectors from our factor matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$: $\sum_{i=1}^{r} \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$. This helps us reduce the number of parameters needed to generate a given multi-view graph.

If we attempted to generate an adjacency tensor directly, we would have to use $\mathcal{O}(k \times n^2)$ parameters in the final layer. However, if we generate the CPD factors first, we only need $\mathcal{O}(r \times (n+k))$ parameters in the final layer, where r is a hyperparameter that increases the quality of the fit at the cost of more parameters. We show that our models works well for $r \ll n^2$ in Section 5.2.

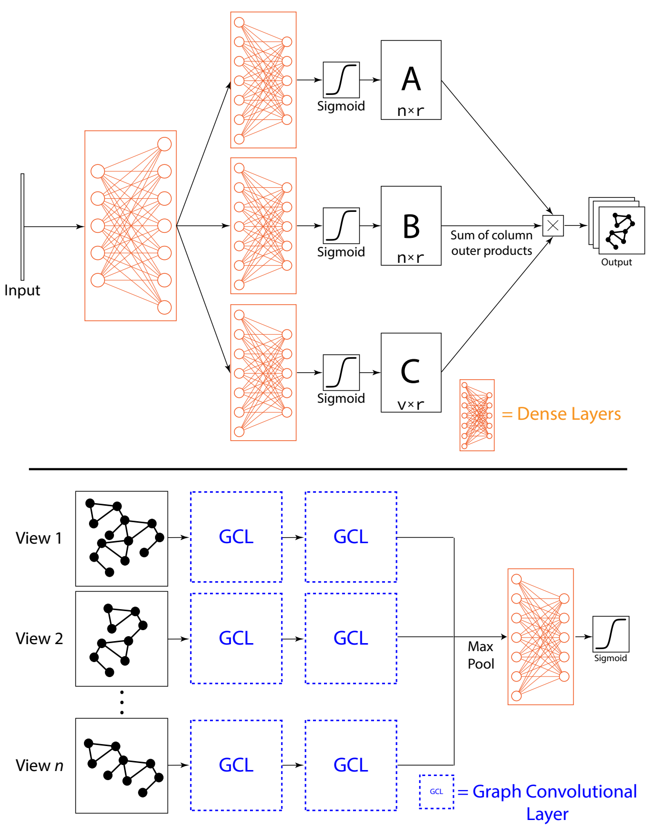

Figure 1: Diagram of the TenGAN generator (top) and discriminator (bottom). We generate the A, B, and C factor matrices before taking the sum of their column outer products to form our output multi-view graph.

4.3 Evaluation Metrics

Another difficult task in multi-view graph generation is evaluating the quality of the generated graphs. Gretton *et al.* [11] found that measuring the Maximum Mean Discrepancy (MMD) between distributions of different graph statistics works well for simple graphs. MMD is also used in other works [26] [6]. Here, we propose three methods for evaluating the structural similarity between the generated and input graphs. These methods are described in detail below.

4.3.1 MMD-Based Evaluation. One method to evaluate the quality of generated multi-view graphs would be to apply the evaluation criteria used for simple graphs to each view. We measure the Mean MMD (M-MMD) score between the distributions of different graph attributes. More concretely, for each graph attribute, we take the mean of the MMD between the i-th view of a generated graph G' and a graph G. We can use the clustering coefficient, degree distribution, and the orbit of the graphs similar to You *et al.* [26].

However, the main downside of this approach is that it does not take the relationship between the views into account. For example, consider the case where we generate multi-view graphs with two views. Let the list of the generated first and second views be $V_1' = g'^{(1)} \forall g' \in G'$ and $V_2' = g'^{(2)} \forall g' \in G'$. Then, suppose the MMD scores of V_1' and V_2' across all the graph attributes are 0. Then, the overall Mean MMD (M-MMD) would be 0. However, if we permute V_1' and V_2', we would get the same M-MMD score.

This is clearly an undesirable behavior in any case where each of the views are correlated with each other. An extreme example of this would be a multi-view graph g where $g^{(1)}$ has an edge iff $g^{(2)}$ does not have an edge. Then, it is possible for a generated graph g' to have $g'^{(1)} = g'^{(2)}$, but still have a perfect M-MMD of 0 in all the graph attributes. To address this issue, we propose a tensor-based evaluation method:

4.3.2 Tensor-Based Evaluation. As described above in Section 2.1, multi-view graphs can be viewed as third-order tensors, and tensor decompositions have been shown to be able to extract structure (like communities) from multi-view graphs [12][7][1][23]. We take advantage of this fact by applying the CPD to each multi-view graph or tensor. The normalized reconstruction error of the decomposition for various values of r provides a heuristic for how much structure there is along the three modes of the tensor. We then compare the errors of the generated and original tensors to see if they are similar in terms of structure.

We randomly sample n generated tensors from the generated and real tensors. We then calculate the sum of the Wasserstein metric (a.k.a. the earth mover's distance: EMD) between all n^2 pairs. The lower this score, the more similar pairs are (on average). While this score works well across a fixed dataset, it is difficult to compare this score across datasets of different sizes. This is because the number of feasible r values changes with the size of the tensor; and a given dataset may naturally have a wider range of pairwise distances.

To solve this issue, we normalize the sum of generated-real distances by the sum of pairwise real-real distances. More formally, given real error matrix $\mathbf{E}$ and generated error matrix $\mathbf{E}'$ (where every row $\mathbf{E}_i$ is a vector of the errors of the i-th sample's decomposition):

$$\textsc{TenScore} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{EMD}(\mathbf{E}_i, \mathbf{E'}_j)}{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{EMD}(\mathbf{E}_i, \mathbf{E}_j)} \quad (1)$$

The lower the TenScore, the more realistic the generated samples are.

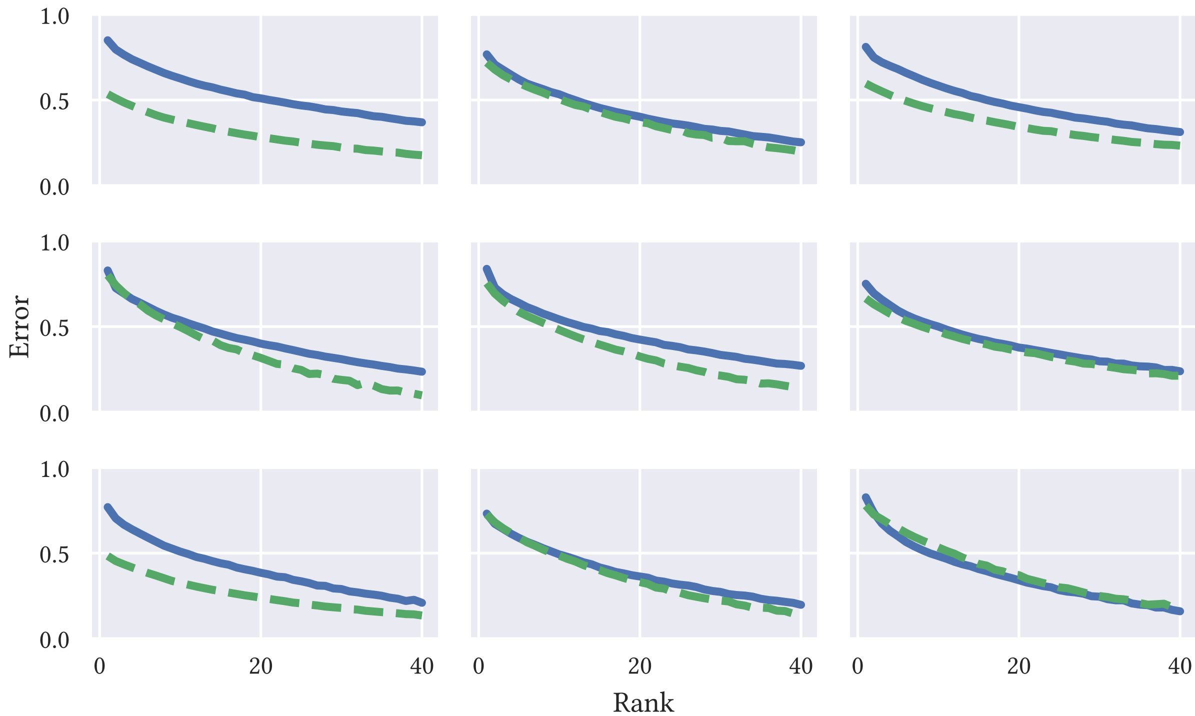

Figure 2: Error vs. rank plot for 9 randomly sampled pairs on the football dataset. The dashed green lines are the generated results, and the solid blue lines are the original results. We can see that the error of generated samples roughly matches those of the original samples. We compute the normalized EMD of all sampled pairs in our tensor-based evaluation method.

4.3.3 Classifier-Based Evaluation. Another solution to evaluating the similarity between the generated and original multi-view graphs is to use a classifier. We train a classifier on generated and original data; then check to see if it correctly predicts the origin of an example. We calculate accuracy and F1 score of the resulting model (the closer to 0.5 or 50%, the better).

In the model, we calculate a graph2vec [21] embedding for each view of the multi-view graph. Then, we split the embeddings into training/test data and train a classier for each view. Finally, we can take the majority vote of the ensemble. These steps are shown in Figure 3. It is worth noting that the embedding method and model can be swapped out for other models depending on

However, this still has the problem that it does take the correlation between views into account. This could be solved by using a multi-view graph embedding method instead, but we leave this to a future work and instead rely on the tensor-based evaluation method to address this case.

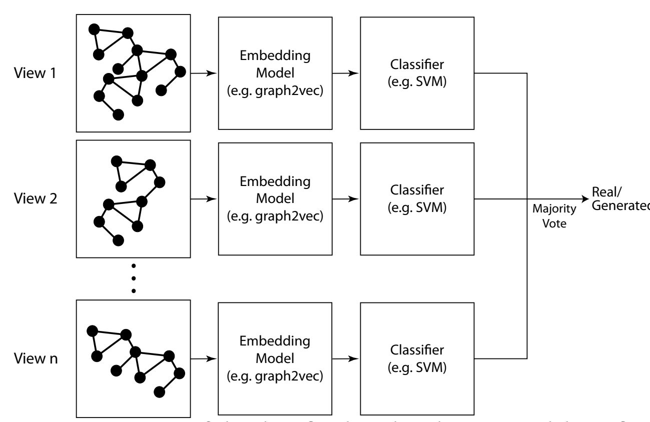

Figure 3: Diagram of the classifier-based evaluation model. We first calculate an embedding and train a classifier on each view before using the majority vote to guess if the result is a real or generated multi-view graph. Note that this is similar to, but different from the discriminator in Figure 1.

5 EXPERIMENTAL EVALUATION

5.1 Datasets

We used two datasets in this work:

(1) **football** [10]: 248 English Premier League football players and clubs on Twitter, where each of the 6 views corresponds to a different interaction between the accounts (follows, followed-by, mentions, mentioned-by, retweets, retweeted-by). Note that 3 of the views are transposes of the other 3.

(2) **EU airlines** [4]: airline routes from 250 European airports, where each of the 37 views corresponds to a route by a different airline.

We sample egonets of size 30–50 from each view, as described above in Section 4.1.

5.2 Results

We evaluated TenGAN on the 2 datasets using the 3 evaluation criteria described above in Section 4.3, and the results are shown in Table 2.

5.2.1 MMD-Based Evaluation. In terms of mean MMD scores, TenGAN on the football dataset performs similarly to BRGAN [24] and LGGAN [6] on the small Citeseer and CORA datasets. However, TenGAN does relatively poorly on the EU airlines dataset, which is also reflected in the other evaluation criteria. We talk about some possible reasons for this in Section 5.2.4. Based on the four runs, we can see that increasing the r generally leads to better mean MMD scores.

5.2.2 Tensor-Based Evaluation. TenGAN does well for $r = 100$ on the football dataset, with an TenScore of less than 1. This indicates that mean distance for all real-generated pairs is lower than the mean distance within the real dataset. However, TenGAN has a very poor TenScore on the football dataset with $r = 50$. This is likely because 50 components are insufficient to express the rank of the football tensor. TenGAN also has a poor TenScore on the EU airlines dataset; some possible reasons for this are discussed in Section 5.2.4.

5.2.3 Classifier-Based Evaluation. TenGAN is able to fool our classifier on the football dataset roughly 24% of the time with $r = 50$ and improves to 30% of the time with $r = 100$. This shows that TenGAN can generate fairly realistic results on the football dataset. However, TenGAN fools the classifier only 2–4% of the time on the EU airlines dataset. We talk about some potential reasons for this below.

5.2.4 Summary. TenGAN shows promising preliminary results on the football dataset, but, TenGAN performs very poorly on the EU airlines dataset. This is likely because the EU airlines dataset also has much less correlation between each view, making the tensor higher rank and therefore much harder to model with TenGAN. It also has a large number of views (which can exceed the number of nodes, depending on the size of the sub-multi-view graph). One potential solution to this would be to increase the number of parameters in the network by increasing r and hidden layer sizes. However, we defer a detailed evaluation of this to a future work.

6 RELATED WORK

There have been several works on the topic of multi-view graph sampling. Interdonado *et al.* [15] mentions that sampling random nodes with their induced subgraphs works on multilayer graphs, but may lead to degree distribution and connectivity issues. A solution to this is to use exploration-based sampling. Methods that work on standard graphs like Metropolis-Hastings random walks, BFS, and forest fire sampling [19] can also be applied to multi-view graphs [15] [16].

To improve random walk sampling on multi-view graphs, Gjoka *et al.* [8] proposes union multigraph sampling—a method that uses the "union multigraph", which consists of all edges across all views in the multi-view graph. Union multigraph sampling then performs a random walk over this multigraph to sample it. While unbiased samples are useful, we sometimes want a biased sample. Khandagi *et al.* [16] proposes using learning automata to perform a biased sample to better sample nodes with special properties.

There has also been some previous work on multi-view graph generation. For example, Nicosia *et al.* [22] proposes a model to grow a multi-view graph based on traditional preferential attachment models like the Barabási-Albert model [2]. Kim *et al.* [17] uses single-layer preferential attachment models and tunes the correlation between layers.

To the best of our knowledge, there have been no other neural-network-based models for multi-view graph generation. However, there has been previous work in graph generation using neural networks. GraphRNN [26] uses a RNN to model graphs as a sequence of nodes of edges. NetGAN [3] uses a LSTM to learn the distribution of biased random walks. GraphVAE [25] uses a variational autoencoder to generate graphs. LGGAN [6] generates the adjacency matrix directly, along with associate labels. Finally, BRGAN [24] generates rank-constrained graphs by first generating factor matrices, much like TenGAN.

7 CONCLUSION

In this work, we discuss some of the issues associated with multi-view graph generation, as well as some potential solutions to these issues. One of these issues is the large number of parameters required to generate a multi-view graph using a neural network. To help solve this, we propose a novel GAN-based method that leverages the CPD to reduce the number of parameters required.

Another issue with multi-view graph generation is a lack of evaluation criteria. We attempt to address this by proposing 3 different evaluation metrics that evaluate the realism of the graph along different aspects. Finally, we run our model on 2 different datasets and evaluate the results according to the metrics we define. This provides a preliminary baseline for future works to build on top of.

8 ACKNOWLEDGEMENTS

We would like to thank Ananthram Swami for his valuable feedback and discussions. Research was supported by the National Science Foundation under CAREER grant no. IIS 2046086. This research was also sponsored by the Combat Capabilities Development Command Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). BAM and TER were also sponsored in part by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory, the United States Air Force, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation here on.

	Football						EU Airlines					
Model Name	Deg	Clust	Orbit	F1	Acc	TenScore	Deg	Clust	Orbit	F1	Acc	TenScore
TenGAN ($r = 50$)	0.18	0.69	**0.09**	0.71	0.76	1.60	0.91	1.31	0.83	0.98	0.98	**1.43**
TenGAN ($r = 100$)	**0.10**	**0.45**	0.10	**0.57**	**0.70**	**0.92**	0.91	**1.27**	**0.81**	**0.97**	**0.96**	1.59

Table 2: Results of TenGAN on the football and EU airlines datasets. Deg, Clust, and Orbit are the mean MMD scores of our MMD-based evaluation method. F1 and Acc are the scores produced by the classifier-based method. TenScore is the score produced by the tensor-based method. Lower is better for all of these metrics. TenGAN does poorly on the EU airlines dataset—some possible reasons are discussed in Section 5.2.4.

REFERENCES

[1] Esraa Al-Sharoa, Mahmood Al-khassaweneh, and Selin Aviyente. 2017. A tensor based framework for community detection in dynamic networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, New Orleans, 2312–2316. https://doi.org/10.1109/ICASSP.2017.7952569

[2] Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512. https://doi.org/10.1126/science.286.5439.509 arXiv:https://www.science.org/doi/pdf/10.1126/science.286.5439.509

[3] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. 2018. NetGAN: Generating Graphs via Random Walks. arXiv:1803.00816 [stat.ML]

[4] Alessio Cardillo, Jesus Gomez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco Del Pozo, and Stefano Boccaletti. 2013. Emergence of network features from multiplexity. *Scientific Reports 2013 3:1* 3, 1 (2 2013), 1–6. https://doi.org/10.1038/srep01344

[5] P. Erdös and A. Rényi. 1959. On Random Graphs I. *Publicationes Mathematicae Debrecen* 6 (1959), 290.

[6] Shuangfei Fan and Bert Huang. 2021. Labeled Graph Generative Adversarial Networks. arXiv:1906.03220 [cs.LG]

[7] Laetitia Gauvin, André Panisson, and Ciro Cattuto. 2014. Detecting the Community Structure and Activity Patterns of Temporal Networks: A Non-Negative Tensor Factorization Approach. *PLoS ONE* 9, 1 (01 2014), e86028. https://doi.org/10.1371/journal.pone.0086028

[8] Minas Gjoka, Carter T. Butts, Maciej Kurant, and Athina Markopoulou. 2011. Multigraph Sampling of Online Social Networks. arXiv:1008.2565 [cs.NI]

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

[10] Derek Greene and Pádraig Cunningham. 2013. Producing a Unified Graph Representation from Multiple Social Network Views. In *Proceedings of the 5th Annual ACM Web Science Conference* (Paris, France) *(WebSci '13)*. Association for Computing Machinery, New York, NY, USA, 118–121. https://doi.org/10.1145/2464464.2464471

[11] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research* 13, Mar (2012), 723–773. http://jmlr.csail.mit.edu/papers/v13/gretton12a.html

[12] Ekta Gujral and Evangelos E. Papalexakis. 2018. *SMACD: Semi-supervised Multi-Aspect Community Detection*. SIAM, San Diego, CA, 702–710. https://doi.org/10.1137/1.9781611975321.79 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611975321.79

[13] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks* 5, 2 (6 1983), 109–137. https://doi.org/10.1016/0378-8733(83)90021-7

[14] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. 2019. Generative Models for Graph-Based Protein Design. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., Vancouver, Canada. https://proceedings.neurips.cc/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf

[15] Roberto Interdonato, Matteo Magnani, Diego Perna, Andrea Tagarelli, and Davide Vega. 2020. Multilayer network simplification: Approaches, models and methods. *Computer Science Review* 36 (2020), 100246. https://doi.org/10.1016/j.cosrev.2020.100246

[16] Ehsan Khadangi, Alireza Bagheri, and Amin Shahmohammadi. 2016. Biased sampling from facebook multilayer activity network using learning automata. *Applied Intelligence* 45, 3 (01 Oct 2016), 829–849. https://doi.org/10.1007/s10489-016-0784-0

[17] Jung Yeol Kim and K.-I. Goh. 2013. Coevolution and Correlated Multiplexity in Multiplex Networks. *Phys. Rev. Lett.* 111 (Jul 2013), 058702. Issue 5. https://doi.org/10.1103/PhysRevLett.111.058702

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, Toulon, France, 1–14. https://openreview.net/forum?id=SJU4ayYgl

[19] Jure Leskovec and Christos Faloutsos. 2006. Sampling from Large Graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Philadelphia, PA, USA) *(KDD '06)*. Association for Computing Machinery, New York, NY, USA, 631–636. https://doi.org/10.1145/1150402.1150479

[20] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Dynamic Multiscale Graph Neural Networks for 3D Skeleton-Based Human Motion Prediction. arXiv:2003.08802 [cs.CV]

[21] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning Distributed Representations of Graphs. arXiv:1707.05005 [cs.AI]

[22] V. Nicosia, G. Bianconi, V. Latora, and M. Barthelemy. 2013. Growing Multiplex Networks. *Phys. Rev. Lett.* 111 (Jul 2013), 058701. Issue 5. https://doi.org/10.1103/PhysRevLett.111.058701

[23] Fatemeh Sheikholeslami and Georgios B. Giannakis. 2018. Identification of Overlapping Communities via Constrained Egonet Tensor Decomposition. *IEEE Transactions on Signal Processing* 66, 21 (2018), 5730–5745. https://doi.org/10.1109/TSP.2018.2871383

[24] William Shiao and Evangelos E. Papalexakis. 2021. Adversarially Generating Rank-Constrained Graphs. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Porto, Portugal, 1–8. https://doi.org/10.1109/DSAA53316.2021.9564202

[25] Martin Simonovsky and Nikos Komodakis. 2018. GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11139 LNCS (2 2018), 412–422. http://arxiv.org/abs/1802.03480

[26] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, Stockholm, Sweden, 5694–5703. http://proceedings.mlr.press/v80/you18a.html

[27] Dawei Zhou, Lecheng Zheng, Jiejun Xu, and Jingrui He. 2019. Misc-GAN: A Multi-scale Generative Model for Graphs. *Frontiers in Big Data* 2 (2019), 3. https://doi.org/10.3389/fdata.2019.00003