Effect of Retraining Graph Generative Models with Generated Graphs

1st Takeru Inada
Graduate School of Engineering
Nagaoka University of Technology
Nagaoka, Niigata, Japan
s213107@stn.nagaokaut.ac.jp

2nd Sho Tsugawa
Institute of Systems and Information Engineering
University of Tsukuba
Tsukuba, Ibaraki, Japan
s-tugawa@cs.tsukuba.ac.jp

3rd Akiko Manada
Department of Electrical, Electronics and Information Engineering
Nagaoka University of Technology
Nagaoka, Niigata, Japan
amanada@vos.nagaokaut.ac.jp

4th Kohei Watabe
Graduate School of Science and Engineering
Saitama University
Saitama, Saitama, Japan
kwatabe@mail.saitama-u.ac.jp

***Abstract*—In recent years, there has been a growing demand for techniques to artificially generate graphs. Various proposals have been made for graph generation models using machine learning. Among these models, GraphTune is a model that allows to specify the features of the generated graphs. GraphTune has not achieved sufficient accuracy when specified values are in ranges where there are few samples in the training dataset. Therefore, in this paper, we propose a method to improve the accuracy of GraphTune by retraining it using graphs generated by the model itself. Through experiments using real-world graphs, we demonstrate that the higher accuracy can be achieved compared to the conventional method.**

***Index Terms*—Graph generation, Generative model, Conditional VAE, Graph feature, Model retraining.**

I. Introduction

The demand for artificial graph generation techniques allowing the specification of graph features is increasing in various fields. For example, generated graphs can be used to simulate information diffusion in distinctive user relationship networks. Graph features refer to indicators representing the characteristics of graph structures, such as the average shortest path length and the clustering coefficient. Various models have been proposed for generating graphs reproducing features of graphs in the training dataset through machine learning. Among these models, GraphTune, which was proposed by Watabe *et al*. [1], allows for specifying features while reproducing the other features of graphs in the dataset. However, GraphTune's accuracy is low for features specified in ranges where there are few samples in the training dataset.

In this paper, we explore a method for retraining GraphTune with the generated graphs, aiming to improve the accuracy. More precisely, we construct new training datasets with different feature distributions by incorporating the graphs generated by GraphTune itself. By retraining and generating again using the constructed datasets, we observe that the accuracy can be significantly improved for features in ranges that originally had low accuracy.

This work was partly supported by JSPS KAKENHI JP23H03379.

II. GraphTune

GraphTune is a generative model capable of learning from input graph datasets and generating graphs that reproduce their features while satisfying specified feature values. GraphTune converts graphs into reversible sequences and learns to reconstruct input sequences using Conditional Variational Auto-Encoder (CVAE) [2].

While GraphTune can accurately specify features that are frequently represented in the training dataset, it faces challenges in improving the accuracy in ranges where there are few samples in the training dataset. Indeed, since the capability of specifying the features strongly depends on the training dataset, the accuracy decreases when specified values are in ranges where there are few samples in the training dataset. This becomes more pronounced as feature values get further far from the peaks.

III. Proposed Method

To address this issue, we propose a method of adding graphs generated by GraphTune to the dataset and retraining. By gradually sliding the values of specified features and repeating the retraining process, we can generate graphs while maintaining high accuracy. Fig. 1 illustrates the overview of the proposed method.

During retraining, the dataset provided as input to GraphTune is constructed using the following procedure. First, specify values of features closer to the final goal (i.e., the feature values that we want to specify) than the peaks of the feature distribution in the training data, and then conduct the learning and generation process with GraphTune. Second, randomly extract samples from both the original training dataset and the generated graphs to construct a new training dataset. It is expected that the peaks of the feature distribution in the constructed training dataset will be closer to the goal than

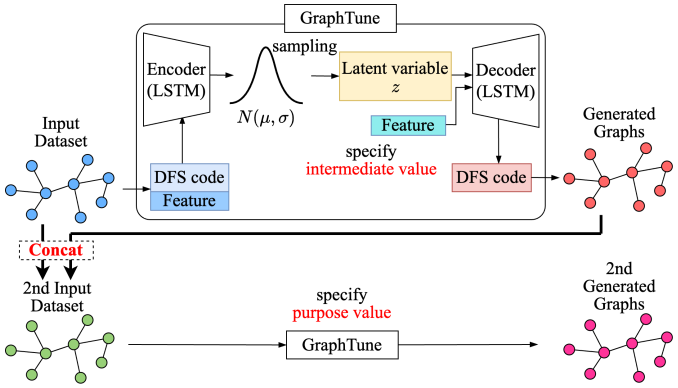


Fig. 1. The overview of proposed method.

Algorithm 1 Retraining process in the proposed method.

Require: G_0 : Training dataset (2000 graphs), $\{v_1, ..., v_N\}$: specified feature values
  $G_{\text{in},1} \leftarrow G_0$
  for $i = 1$ to N **do**
    $G_{\text{out},i} \leftarrow$ 2000 Graphs GraphTune$(G_{\text{in},i}, v_i)$ outputted
    if $i < N$ **then**
      $G_{\text{in},i} \leftarrow$ Sampling randomly 1000 from $G_{\text{in},i}$
      $G_{\text{out},i} \leftarrow$ Sampling randomly 1000 from $G_{\text{out},i}$
      $G_{\text{in},i+1} \leftarrow G_{\text{in},i} \cup G_{\text{out},i}$
    end if
  end for
  return $G_{\text{out},N}$

those of the original training dataset. By repeating the process of training the model, generating graphs, and constructing the dataset, we can specify values of the goal features with higher accuracy than before. Algorithm 1 illustrates the retraining processes in the proposed method.

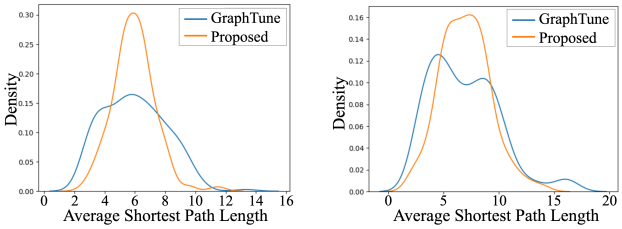
IV. Experiments

To validate the effect of improving the accuracy in specific ranges of features using the proposed method, we conduct learning and generation using real-world graphs. The training dataset is constructed with 2000 connected subgraphs sampled from the Twitter who-follows-whom network, extracted from the Higgs Twitter Dataset [3]. The specified feature is the average shortest path length and we denote by v_s the specified value of the feature. We use Root Mean Square Error (RMSE) to evaluate the performance of the proposed method. RMSE can help consider the error and variance between the specified value and the feature of the output graphs.

The parameters of GraphTune in the proposed method and the conventional method are set according to the reference [1]. In the proposed method, the specified value v_s of the average shortest path length is set to two patterns: 5.5 for the first iteration and 6.0 for the second iteration, and 6.0 for the first and 7.0 for the second. We then compare the proposed method with the conventional method by setting $v_s = 6.0$ or 7.0. For each v_s value, 2000 graphs were generated.

TABLE I
RMSEs of the average shortest path length for proposed method and conventional GraphTune

Average shortest path length	Proposed RMSE	Conventional RMSE
6.0 (Proposed: 5.5 → 6.0)	**1.38**	2.05
7.0 (Proposed: 6.0 → 7.0)	**2.24**	3.09



(a) Kernel density estimation when $v_s = 6.0$ (b) Kernel density estimation when $v_s = 7.0$

Fig. 2. Kernel density estimation plot of the distribution of average shortest path length for generated graphs.

Table I presents the RSMEs for the proposed and conventional methods. Fig. 2 illustrates the kernel density estimation of the average shortest path length in the generated graphs. These results indicate that the proposed method achieves higher accuracy compared to the original GraphTune. For example, the RMSE of the proposed method is smaller than that of GraphTune, indicating a smaller error between the output graph's average shortest path length and the specified value. Additionally, the distribution of the average shortest path length in the generated graphs exhibits higher density around the specified value compared to GraphTune.

V. Conclusion

We proposed a method to improve the accuracy of the graph generation model GraphTune, which allows for the specification of features, by retraining it with generated graphs. Constructing the dataset based on the generated graphs and gradually shifting specified values during training, we confirmed higher accuracy in ranges where there are few samples in the training dataset.

To further improve accuracy and broaden the range of generatable features, we plan to explore superior methods for constructing datasets and shifting specified feature values.

References

[1] K. Watabe, S. Nakazawa, Y. Sato, S. Tsugawa, and K. Nakagawa, "GraphTune: A Learning-Based Graph Generative Model with Tunable Structural Features," IEEE Transactions on Network Science and Engineering, vol. 10, no. 4, pp. 2226-2238, 2023.
[2] K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation using Deep Conditional Generative Models," NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2, pp. 3483-3491, 2015.
[3] M.D. Domenico, A. Lima, P. Mougel, and M. Musolesi, "The Anatomy of a Scientific Rumor," Scientific Reports, vol.3, no.2980, 2013.