FlowGPT: How Long can LLMs Trace Back and Predict the Trends of Graph Dynamics?

Zijian Zhang, Zonghan Zhang, Zhiqian Chen
Computer Science and Engineering, Mississippi State University
{zz242,zz239}@msstate.edu, zchen@cse.msstate.edu

Abstract

In this paper, we introduce a benchmark specifically for LLMs in the context of dynamic graphs. This benchmark focuses on evaluating LLMs' performance in tasks related to flow, such as predicting the influence of certain nodes (forward prediction) and identifying original sources in diffusion processes (backward prediction). Our objective is to delineate the competencies and limitations of LLMs in the realm of dynamic graph analysis. The development of this benchmark is anticipated to enhance the comprehension and application of LLMs in dynamic graph scenarios, an area critical for predicting and understanding temporally varying complex phenomena. Benchmark datasets is released at `https://zenodo.org/records/10517068`.

1 Introduction

The importance of benchmarks in evaluating Large Language Models (LLMs) like ChatGPT has grown, especially for assessing their effectiveness in various domains. Key studies (Liu et al., 2023; Mao et al., 2023; Hu et al., 2020) highlight benchmarks as crucial in understanding LLMs' capabilities and limitations, especially in processing complex data like graphs. Significant research (Wang et al., 2024) has introduced the NLGraph benchmark with eight graph reasoning tasks, revealing LLMs' reduced effectiveness in complex graph tasks. The NLGraph benchmark, also used by (Guo et al., 2023), points to a performance decline in LLMs for advanced graph issues, suggesting a need for improved methods. In dynamic graphs, benchmarks are scarce. (Zhang et al., 2023a) developed the LLM4DyG benchmark with nine tasks for understanding dynamic graphs' temporal change on graph structure. However, it lacks focus on flow dynamics (node dynamics over graphs), underscoring the need for more comprehensive benchmarks in this area. FlowGPT, our novel framework, focuses on graph flow research, utilizing LLMs for dynamic graph interactions. It includes diverse graph datasets and metrics for graph flow studies.

2 FlowGPT

We generate the datasets of Watts-Strogatz small-world graph with 1000 nodes and SIR diffusion model by our open-source library (Zhang et al., 2023b), which is publicly available at `https://github.com/XGraphing/XFlow`. It should be noted that there is considerable uncertainty due to the inherent randomness in the SIR model, which poses a significant challenge to accurate prediction. Our benchmark includes two tasks: *forward* flow and *backward* flow. In forward flow, we observe the data flow starting from the source node set and predict the target node set forwardly. In backward flow, we start from the target node set and predict to identify the source node set backwardly.

We have developed an experimental framework to evaluate predictive capabilities using a fleet of 14 custom GPTs from ChatGPT. These GPTs are divided into two main categories: Forward Prediction GPTs and Backward Prediction GPTs. Forward Prediction GPTs predict the diffusion process from the source to the target. There are two subgroups within this category: pure interval GPTs and mixed interval GPTs. Pure Interval GPTs: These models, including GPT FW1 (1 step interval, e.g., step 1 →2, 2→3, etc.), FW2 (e.g., 1→3, and 2→4), FW4, and FW8, identify patterns and trends over consistent time intervals. Mixed Interval GPTs: This subgroup tests prediction accuracy across varying time intervals, presenting more complex scenarios. Models like GPT FW1_2, FW1_4, and FW1_8 are trained on data combining 1 and 2 intervals, 1 and 4 intervals, and 1 and 8 intervals, respectively, with testing data structured in the same manner. Backward Prediction GPTs focus on the reverse process, predicting from the target back to the source. These GPTs also include two groups, BW1, BW2, BW4, and BW8 as well as BW1_2, BW1_4, and BW1_8,

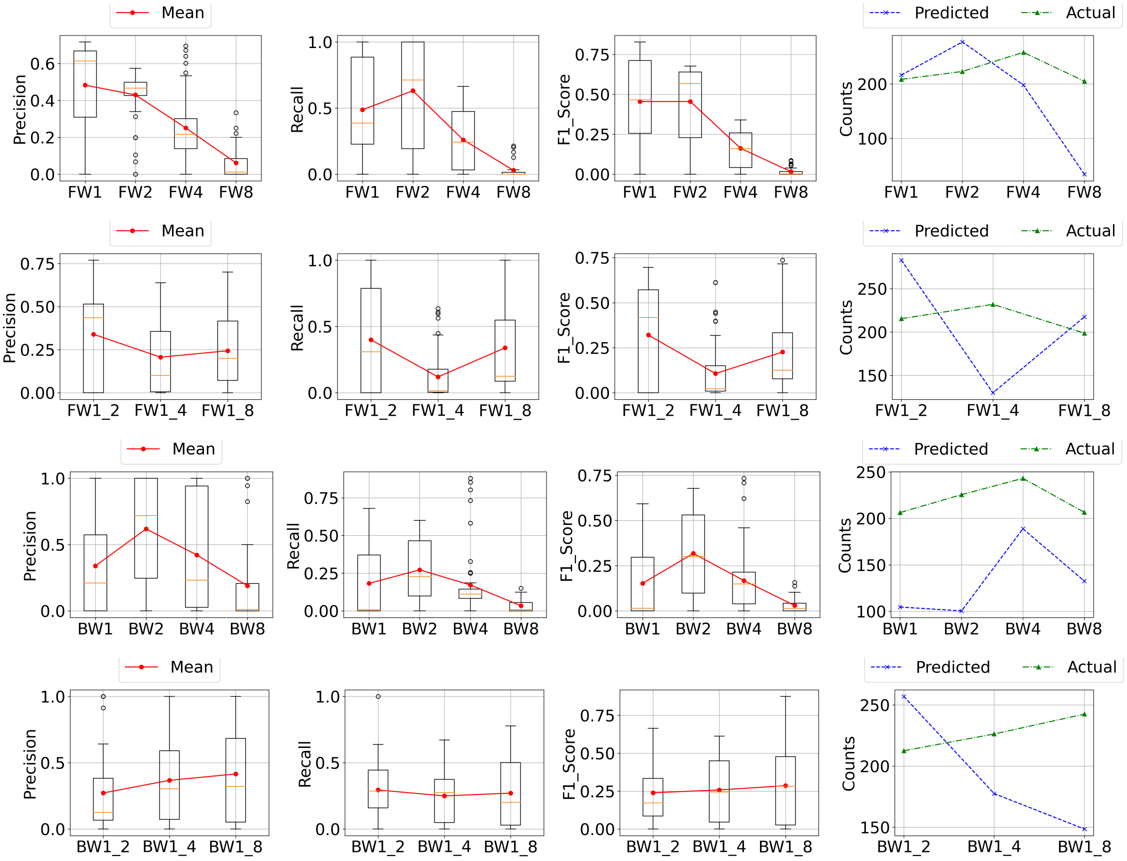

Figure 1: Forward prediction (upper two rows) and Backward prediction (lower two rows).

which mirror the structure and objectives of their Forward Prediction counterparts.

3 Results

In **forward** task, as shown in 1st rows in Fig. 1, there is a lower precision at FW2 and higher precision at FW1. Recall initially rises from FW1 to FW2, then falls to FW8, with the largest variability at early intervals. The F1 score, similarly, decreases notably from FW2 to FW8. The number of predicted nodes (4th subgraph) varies across intervals, with more candidates predicted in earlier intervals (FW1, FW2) than later ones (FW4, FW8). Mixed intervals (2nd row) display different trends: FW1 mixed with FW2 shows the highest median precision, recall, and F1 compared to other combinations, suggesting a balance in performance. However, high recall outliers in FW1 mixed with FW4 indicate potential for high performance under certain conditions. The accuracy benefits from early-late-stage interval combinations, implying the importance of strategic interval selection and the complexities of interval mixing. In **backward** tasks (lower 2 rows), precision, recall and F1 initially increase from BW1 to BW2, then decrease at longer intervals (BW4, BW8). In terms of predicted versus actual nodes, GPTs with pure intervals predicted fewer nodes than actuals, with higher accuracy at earlier intervals (BW1, BW2). The hypothesis suggests better performance at BW2 due to clearer flow patterns. Mixed intervals (4th row) show that combining early (BW1) and late stages (BW8) yields better precision and F1 than closer stages. A trend emerges: accuracy decreases as intervals lengthen, but mixing early and later stages improves outcomes, highlighting the benefits of strategically combining different stages for optimal results.

4 Conclusion

We evaluated LLMs on proposed benchmarks and found that shorter intervals yield better outcomes in forward task. Interval mixing has a dual effect, enhancing or reducing accuracy due to noise variation. Combining early and late intervals proves beneficial in backward tasks. This study highlights the critical roles of interval choice and mixing.

5 Ethics Statement

This study's findings are impactful across various domains, including social media, disease modeling, electrical grids, and biological neural networks, underscoring the need for effective flow disruption management. It delves into strategies like network restructuring and targeted lockdowns. Moreover, this work aims to enlighten the NLP community about the inner workings of LLMs, paving the way for their safer and more informed application.

References

Roy M. Anderson and Robert M. May. 1980. The population dynamics of microparasites and their invertebrate hosts. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 291(1054):451–524.

Wei Chen, Tian Lin, and Cheng Yang. 2016. Real-time topic-aware influence maximization using preprocessing. *Computational social networks*, 3(1):1–19.

Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 12(3):211–223.

Mark Granovetter. 1978. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1443.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. 2023. GPT4Graph: Can Large Language Models Understand Graph Structured Data ? An Empirical Evaluation and Benchmarking. ArXiv:2305.15066 [cs].

Jing Guo, Peng Zhang, Chuan Zhou, Yanan Cao, and Li Guo. 2013. Personalized influence maximization on social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 199–208.

Hans Heesterbeek, Roy M. Anderson, Viggo Andreasen, Shweta Bansal, Daniela De Angelis, Christopher Dye, Ken Eames, W. John Edmunds, Simon Frost, Sebastian Funk, et al. 2005. Concepts and methodological issues in infectious disease modeling. *Epidemiology and Infection*, 133(02):389–403.

Herbert W. Hethcote. 2000. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133. Curran Associates, Inc.

W. O. Kermack and A. G. McKendrick. 1927a. A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721.

W. O. Kermack and A. G. McKendrick. 1927b. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721.

Yuchen Li, Ju Fan, Dongxiang Zhang, and Kian-Lee Tan. 2017. Discovering your selling points: Personalized social influential tags exploration. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 619–634.

Yuchen Li, Dongxiang Zhang, and Kian-Lee Tan. 2015. Real-time targeted influence maximization for online advertisements.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of ChatGPT-Related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. GPTEval: A Survey on Assessments of ChatGPT and GPT-4. ArXiv:2308.12488 [cs].

Hung T Nguyen, Thang N Dinh, and My T Thai. 2016. Cost-aware targeted viral marketing in billion-scale networks. In *IEEE INFOCOM 2016-the 35th annual IEEE international conference on computer communications*, pages 1–9. IEEE.

Shan Tian, Songsong Mo, Liwei Wang, and Zhiyong Peng. 2020. Deep reinforcement learning-based approach to tackle topic-aware influence maximization. *Data Science and Engineering*, 5:1–11.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can Language Models Solve Graph Problems in Natural Language? ArXiv:2305.10037 [cs].

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, Simin Wu, and Wenwu Zhu. 2023a. LLM4DyG: Can Large Language Models Solve Problems on Dynamic Graphs? ArXiv:2310.17110 [cs].

Zijian Zhang, Zonghan Zhang, and Zhiqian Chen. 2023b. Xflow: Benchmarking flow behaviors over graphs. *arXiv preprint arXiv:2308.03819*.

Appendix

The detailed formulations are explained below.

In IM method, it takes input from the graph structure information $G = (V, E, A)$ and a seed budget

$k \in \mathbb{N}^+$, and aim to determine a k-sized source node set Ω that maximizes the expected influence spread $|\sigma(\Omega)|$. The influence spread is defined as the size of the final activated subgraph $\sigma(\Omega)$. Two types of IM techniques exist: simulation-based and proxy-based. Both require a known diffusion model D, such as IC, LT, SI, or SIR. Once the seed budget is met ($|\Omega| = k$), the objective of IM is to find a seed set Ω that maximizes the expected influence spread as follows:

$$\Omega = \arg\max_{\Omega^*} |\sigma(\Omega^*|G, D)|. \tag{1}$$

The Forward Flow task is closely related to IM, but has a wider scope. While IM typically focuses on maximizing the spread of influence, Forward Flow is an influence inference and prediction task that involves understanding and predicting the patterns and extents of influence propagation in a network, regardless of whether the aim is maximization. This task involves utilizing network dynamics to predict how influence flows through the network, which nodes are the key influencers, and how different nodes contribute to the overall spread.

In contrast, when we have a snapshot of a graph, denoted as $G = (V, E, A)$, we can observe the diffusion status at a specific point in time. The activated subgraph, σ, of this snapshot is the result of the propagation of an unknown seed set, Ω. The goal of an SL method is to identify Ω using σ, the graph structure, G, and the diffusion model, D. The most probable seed set that resulted in the observed σ can be found using the following equation:

$$\Omega = \arg\max_{\Omega^*} Pr(\Omega^*|\sigma, G, D) \tag{2}$$

The Backward Flow task is a backward influence inferences task, which is similar to the principles of SL, but with a greater focus on understanding more complex network dynamics. Instead of just localizing the source, the backward influence inference aims to comprehend the pathways and dynamics of the reverse spread. This involves analyzing how information, rumors, or diseases propagate backward through a network to identify potential sources and understand the mechanisms of spread.

Our upcoming project aims to unify influence inference and prediction tasks as well as the backward influence inference and prediction tasks in a single framework that uses the LLM technique instead of conventional simulation and proxy methods. The objective of this undertaking is to obtain a comprehensive understanding of network dynamics by leveraging the strengths of both approaches while taking advantage of the in-context learning capabilities of LLMs. Our goal is to develop a dynamic model that can predict the most effective points of influence in a network and also reveal the origins of diffusion patterns by combining influence inference in both directions. Furthermore, we aim to explore the limits of LLMs to comprehend these complex tasks in dynamic graphs and flow by combining them. This comprehensive approach will significantly enhance our understanding of network behaviors, which will enable us to devise more sophisticated strategies for predicting and analyzing influence spread in a variety of complex networks.

Implementation

Bridging the gap between the known and the unknown in dynamic graph benchmarking presents a unique set of challenges. The primary obstacles faced in our work include computational device limitations, the unpredictability of outputs from LLMs, and managing the context length within LLMs. These challenges posed significant hurdles in developing robust benchmarks for dynamic graphs. Our approach not only confronts these challenges head-on but also paves the way for innovative solutions in the field, thereby filling a critical gap in dynamic graph benchmarking. In order to overcome the challenges we faced, when setting up the experiments, we utilized a series of targeted techniques. Firstly, we made use of online tools such as customized GPTs from ChatGPT to circumvent computational limitations. Secondly, we provided structural knowledge, formatted inputs, and clear instructions to the GPTs fleet we set up. This was crucial in managing the unpredictability of outputs. Lastly, we addressed issues related to context length in LLMs by splitting the data and employing other strategies. These techniques not only aided the development of dynamic graph benchmarks but also significantly contributed to advancing the field.

Diffusion Models. In our approach, we place a significant emphasis on the SIR (Susceptible-Infected-Removed) model, a well-established framework in diffusion modeling. This model, a cornerstone in understanding diffusion processes, is particularly notable for its ability to capture the dynamics of infection spread and recovery in a network

(Kermack and McKendrick, 1927a). While our focus is primarily on the SIR model, it's important to recognize the diversity of diffusion models available in the field. Other progressive models, such as the Independent Cascade (IC) (Goldenberg et al., 2001) and Linear Threshold (LT) (Granovetter, 1978) models, have been widely studied for their unique mechanisms of vertex activation and permanence in an active state. In the realm of non-progressive models, variants of the SIR model offer nuanced perspectives on diffusion processes. These include the SEIR model, which introduces an 'Exposed' state (Kermack and McKendrick, 1927b), the SIRS model that allows recovered individuals to become susceptible again (Anderson and May, 1980), the SIRD model, which accounts for 'Dead' individuals (Hethcote, 2000), and the SEIRS model, a hybrid that incorporates features of both SEIR and SIRS models (Heesterbeek et al., 2005). While most existing diffusion models adhere to a mean-field approach, treating each entity with identical diffusion behavior (Guo et al., 2013; Li et al., 2015; Nguyen et al., 2016; Chen et al., 2016; Li et al., 2017; Tian et al., 2020), our focus on the SIR model aligns with its relevance and applicability in depicting real-life diffusion scenarios. By concentrating on the SIR model, we aim to delve deeper into its intricacies and potential applications in understanding and analyzing diffusion phenomena.

Graph Datasets. Our project is currently prioritizing the exploration and analysis of Watts-Strogatz small-world graphs, while still acknowledging the diverse range of datasets and graph structures that we have previously integrated into our framework. Our implementation is designed to be highly adaptable and compatible with various graph representations, including several NetworkX graph objects like Barabási–Albert (BA), Erdős–Rényi (ER) models, and more, as extensively listed in the Graph Generators section of NetworkX. It also encompasses graph objects from PyTorch Geometric, such as Cora, CiteSeer, PubMed, and co-purchasing networks like Amazon Photo and Computers, along with synthetic graph generators, as seen in PyTorch Geometric. However, our current focus is on Watts-Strogatz small-world graphs, as they uniquely model the small-world phenomenon in network theory. This targeted approach allows for a more specialized and comprehensive examination of these graphs, illuminating their distinct properties and applications in complex network analysis. We generate the datasets of Watts-Strogatz small-world graph with 1000 nodes by our open-source library, which is publicly available at: https://github.com/XGraphing/XFlow.

In order to ensure the reliability of each bot configuration, we are replicating them more than three times with different testing data. During each replication, the fleet will share the same graph data but with varying training and testing data. We utilized our XFlow library to generate the graph data and provided it to the GPTs as background knowledge. To prepare the training and testing data, we ran 10 simulations on that graph with the diffusion model of SIR for 20 intervals. For each interval, the source infected node and the target infected node are recorded. The GPTs will also be provided with the training data containing source intervals, source infected nodes, target intervals, and related target infected nodes as knowledge. During testing, formatted files will be uploaded to the GPTs and processed. The model will be instructed to fill in the missing data and return the updated files. For the forward task, the files will be formatted with data of source interval, source infected nodes, and target intervals, while target infected nodes are deleted. The Forward prediction GPTs are then asked to predict the target infected nodes. For the backward task, the files will be formatted with data of target interval, target infected nodes, and source intervals, while source infected nodes are removed. The Backward prediction GPTs are then asked to predict the source infected nodes. An important aspect of testing is the omission of specific parameters such as the SIR model's infected rate and recovery rate values, challenging the model to infer them from the training data. Future work may involve explicitly providing these parameters to observe the impact on prediction accuracy. This experimental setup aims to rigorously test and validate the predictive capabilities of our framework under a variety of scenarios, from simple to complex, ensuring a comprehensive evaluation of its effectiveness in different network dynamics.

Evaluation Metrics

We evaluated the performance of the chatGPT custom GPTs we had created to deduce the infected nodes in both the Forward and Backward tasks. We utilized three crucial evaluation metrics in classification, namely precision, recall, and F1 score. The responses of the Forward GPTs are stored as the list of the Predicted Target Infected Nodes List, while the responses of the Backward bots are stored as the list of the Predicted Source Infected Nodes List. Both of those predicted results from the GPTs will be considered as the list of predicted positives. In the Forward experiments, the results from the SIR simulation based on the Small World graph are stored as the Actual Target Infected Nodes List. Meanwhile, in the Backward experiments, the results from the SIR simulation based on the Small World graph are stored as the Actual Source Nodes List. We have calculated the overlap between the Actual List and the Predicted List for both the Forward and the Backward tasks.

Precision, is defined as the ratio of correctly predicted positive observations to the total predicted positive ones, including the true positive as well as the false positives, which can be expressed as:

$$Precision = TP/(TP + FP) \tag{3}$$

In our experiments, based on this definition, it will be the number of overlapping nodes divided by the total number of predicted nodes in the predicted list. Precision reflects the GPT's ability to return only relevant instances. A higher precision indicates that the GPT returned significantly more relevant results than irrelevant ones, while a lower precision indicates many false positives. That means the GPT predicted many instances as positive that are actually negative.

Recall, also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all observations in the actual class, including the true positives and false negatives. The formula for Recall is:

$$Recall = TP/(TP + FN) \tag{4}$$

In our case, it will be the number of overlapping nodes over the number of actual nodes. Recall measures the ability of the model to identify all relevant cases within a given dataset. A higher recall indicates that the GPTs returned most of the relevant results, while a lower recall indicates that they missed a significant number of relevant results.

F1 Score, is the harmonic mean of precision and recall. The formula is:

$$F1 = 2*(Precision*Recall)/(Precision+Recall) \tag{5}$$

F1 score reaches its best at 1, which represents perfect precision and recall, and its worst at 0. A higher F1 score suggests a model with a good balance of precision and recall, minimizing both false positives and false negatives. Conversely, a lower F1 score implies that the model has issues with either precision or recall, leading to a higher rate of false positives or false negatives.

Additional Results

In the comparison between pure intervals and mixed intervals shown in Figure 2, we can see that for pure intervals 1 or 2 and mixed intervals 1 and 2, Precision FW1 is larger than FW2. However, when the data of intervals 1 and 2 are combined, the precision becomes even smaller. FW2 has the highest recall score. Moreover, after combining the data of intervals 1 and 2, the recall of F1_2 becomes the lowest. The mixed case results in a lower F1 score compared to pure intervals. Although the F1_2 GPT provides more candidate nodes in the response, the accuracy drops. For pure intervals 1 or 4 and mixed intervals 1 and 4, Precision, Recall, and F1 Score decrease from FW1 to FW4. Combining FW1 and FW4 results in even smaller metrics, a lower mean, and a bigger IQR. The figure also displays that the GPT FW1_4 predicted more nodes than the actual. For pure intervals 1 or 8 and mixed intervals 1 and 8, after merging the intervals, the performance of FW1 blended with FW8 improved. However, it is still inferior to pure FW1, which exhibited significantly larger metrics than pure FW8. The mixed GPT FW1-8 provides more predicted candidates than the pure FW8 one. To conclude, combining early stage intervals FW1 with closer intervals like FW2 and FW4 reduces the performance, while combining them with later stages increases the performance. The reason might be that combining the closer intervals might bring some noise, even though it provides more information. Combining with the later stage, on the other hand, might overcome the noise because the performance of FW8 was low before the time span was long enough. Providing information from the beginning intervals might bring some noise, but the more information it provides, the more it will overcome that noise.

Figure 3 shows the comparison between pure intervals and mixed intervals in the Backward task. The data shows that BW1 has a wide range of Precision values, with the median falling in the middle

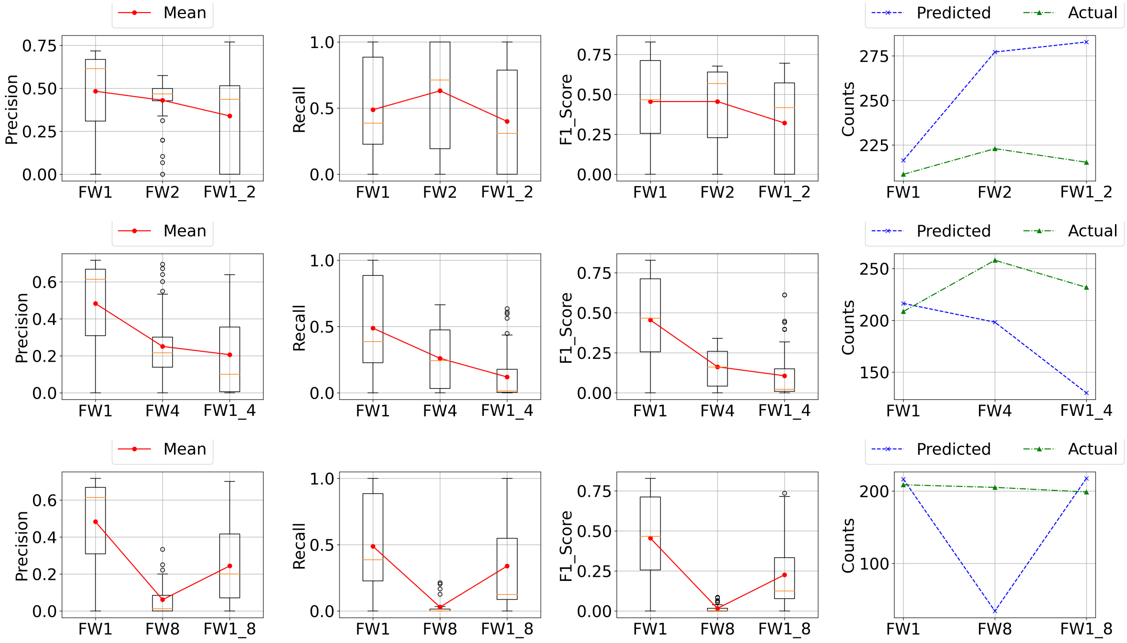

Figure 2: When we combine data from intervals that are closer together, such as FW1 with FW2, the accuracy of our predictions can decrease. However, when we mix data from early and later intervals, such as combining FW1 with FW8, it can actually improve prediction accuracy despite the initial introduction of noise. Based on this data, it seems that finding the right balance between informative data and potential noise can enhance the predictive abilities of GPT models.

of the box and the possibility of outliers on the higher end. On the other hand, BW2 has a larger Precision than BW1, but with a huge IQR, indicating more variability. When both BW1 and BW2 are mixed, the precision and the F1 score are lower than either of the pure BW1 or BW2. However, the Recall slightly increases. When combining BW1 and BW4, the data indicates that BW4 has a higher precision than BW1, but with a large interquartile range (IQR), indicating greater variability. Additionally, BW4 has a low recall. The mixture of BW1 and BW4 outperformed BW1 in terms of precision, recall, and F1 Score. When adding BW1 to BW8, BW8 has the lowest performance in terms of Precision, Recall, and F1 Score. However, when mixed with BW1, it outperforms both BW1 and BW8. In summary, when BW2 is mixed with BW1, it results in a lower score, given pure BW2 already outperforms BW1. On the other hand, when BW4 is mixed with BW1, it leads to an increase in the score to some extent. A closer mix in intervals might not bring some benefit but noise, but a further mix might be beneficial.

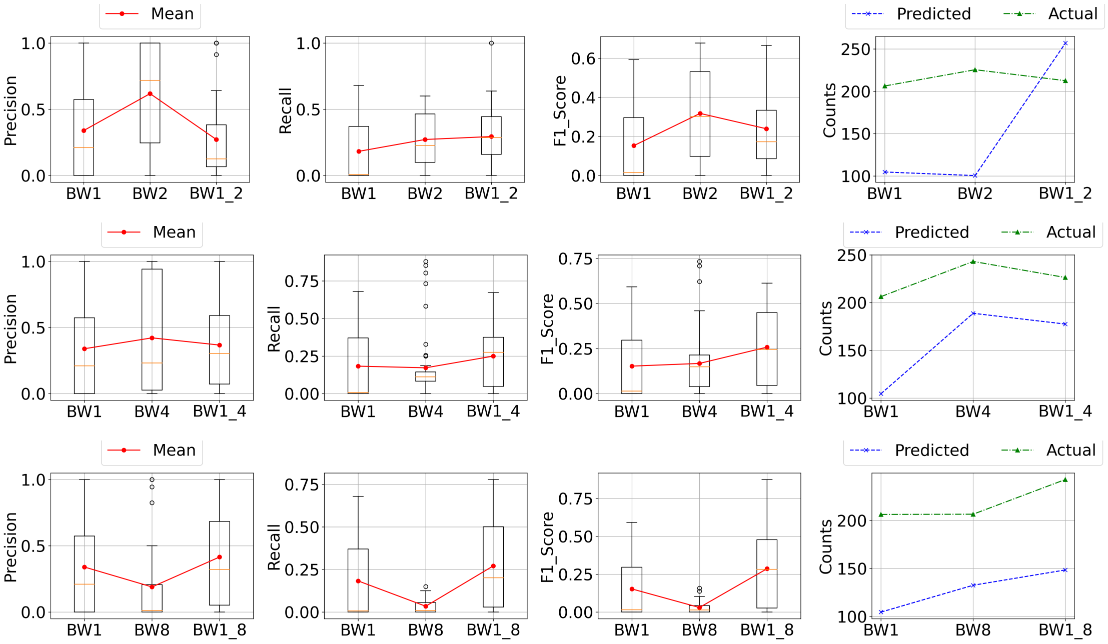

Figure 3: Combining early and later intervals (BW1 with BW4 and BW8) can improve precision and F1 scores in the Backward task. Although there may be a temporary drop in performance when BW1 is merged with the adjacent interval BW2, the overall trend suggests that integrating early-stage data with extended intervals leads to a more accurate predictive outcome. It is important to avoid combining intervals that are too close together as this may introduce noise to the results.