IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 979-8-3503-8447-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOMWKSHPS61880.2024.10620716

GenG: An LLM-based Generic Time Series Data Generation Approach for Edge Intelligence via Cross-domain Collaboration

Xiaomao Zhou[1], Qingmin Jia[1], Yujiao Hu[1], Renchao Xie[1,2], Tao Huang[1,2], and F. Richard Yu[1,3]
[1] Future Network Research Center, Purple Mountain Laboratories, Nanjing, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] Carleton University, ON, Canada
{zhouxiaomao, jiaqingmin, huyujiao}@pmlabs.com.cn

***Abstract*—In this paper, we propose GenG, a generic time series data generation approach for edge intelligence that incorporates knowledge from different domains to synthesize high-fidelity and controllable time series data resembling to different IoT devices. Specifically, GenG decomposes the time series data generation task into two subtasks, the first subtask is to finetune a Large Language Model (LLM) in a self-training method to harness its outstanding knowledge and reasoning capacities for explainable data generation, solving the problem of what to generate. The second one focuses on generating high-quality and controllable time series data conditioning on the output of the finetuned LLM, solving the problem of how to generate. Furthermore, a two-stage generation process is proposed to increase the quality of the generation results by introducing both the abstract and detailed guidance signals, which also enables flexible control over the generation results and ensures synthesized data with consistent features. During deployment, GenG can be arranged in a cloud-edge collaboration way, where the cumbersome LLM and light-weight generation model are placed on the cloud and edge, respectively, fitting well with the resource-constrained edge intelligence. Experimental results in different generation tasks demonstrate GenG's efficiency in reasoning about the generation task and synthesizing high-fidelity time series data with controllable features.**

***Index Terms*—time series data generation, Large Language Model, Diffusion Model, knowledge transfer**

I. Introduction

Edge intelligence has emerged as a promising solution being able to meet the ever-increasing demands of AI applications in Edge Computing, with the scale and complexity of data generated by various machines and devices increasing dramatically. This data, especially the time series data, offers great potential to speed up the production process and improve decision-making, e.g. predicting failure risks [1], monitoring real-time situations [2], and generating expert-level policies [3]. However, collecting high-quality data from the real world is always costly and limited, so synthesizing high-fidelity time series data is becoming ubiquitously important.

A critical challenge in synthesizing time series data is to capture the temporal dependence in which observations in the subsequent time steps are influenced by the previous observations. This becomes even more complicated when generating data with different features. While previous works focus on solving this problem by employing large model structures with recurrent connections [4] and attention mechanisms [5] to learn from labeled time series data, such a learning paradigm, trained only in the time series data domain, drastically impairs the performance and generalization of the generative models due to limited training data and biased knowledge of a single domain. In many cases, these generative models can only generate certain types of time series data, and are not able to reason about the generation tasks.

The recent proliferation of large language models (LLMs) has endowed deep learning models with remarkable cognitive capabilities, including logical reasoning, common sense, and emotional understanding, which provides great promising in boosting the time series data generation. By incorporating LLMs into data generation, the complex temporal dependencies and long-time relationships can be easily captured by the powerful modelling capabilities of LLMs [2]. Furthermore, domain-specific LLMs are capable of reasoning about the generation tasks, providing prior knowledge about the generation process and results, which can mitigate the problem of lack of training data and increase the model's generalization ability. However, since LLMs can only be trained on text data, it is a challenge to apply the knowledge learned from the text data domain to the generation tasks in the time series data domain, where knowledge transfer and alignment between different domains should be considered.

In this paper, we propose a generic time series data synthesis approach for edge intelligence, which decomposes the task of time series data generation into two subtasks. The first one is to finetune an LLM (Cog-LLM) in a self-training method to generate an abstract understanding of the generation task, solving the problem of what to generate, and the second one (Gen-LM) utilizes a transformer-based model to generate time series data conditioning on the output of Cog-LLM, solving the problem of how to generate. Specifically, Cog-LLM starts from a base language model, e.g. LLaMa, and adopts an iterative self-training method by leveraging the model to self-

This paper is supported by National Natural Science Foundation of China (No. 92367104 and No. 92267301).

Corresponding author: Renchao Xie, Renchao_xie@bupt.edu.cn

augment and self-select training data from large amounts of unlabeled documents. The Cog-LLM is finetuned to understand the physical properties, operating mechanisms, and behavioral characteristics of different IoT devices, thus being able to abstractly describe the characteristics of the output data during the time series data generation. Conditioning on Cog-LLM's text descriptions, Gen-LM then generates the time series data with controllable features. Since the Cog-LLM and Gen-LM originate from the text data domain and the time series data domain, respectively, in order to align and transfer knowledge between these two domains, we propose a classifier-free DM [6] (Diffusion Model) based approach that utilizes off-the-shelf domain-specific expert models to facilitate learning relationship and knowledge transfer between different domains. In addition, a two-stage generation process is proposed to increase the quality of the generated time series data. In stage one, the text descriptions of the Cog-LLM are used as a coarse-grained guidance to ensure that the synthesized data have the expected abstract features. In stage two, the time series data examples or previously generated data sequences are used as the fine-grained guidance to fit the generated data with detailed statistics. During deployment, Cog-LLM and Gen-LM can be deployed on the cloud and edge, respectively, facilitating an efficient data synthesis solution for the resource-constrained edge intelligence.

II. Related Work

The generation of time series data has been a subject of extensive research in recent years, with various statistical methods and machine learning techniques being proposed, e.g. flow-based models [6], variational autoencoder based models [7], generative adversarial network based models [8], and DM based models [9]. For high-quality data generation, solutions, e.g. utilizing large model structures with recurrent connections [4], introducing auxiliary supervisions on the generation process [10], and fusion generation results from multiple models [11], are widely adopted. Despite producing promising results in resemblance to real data, most generative models are purely data-driven, whose generation results lack of controllable adjustments and generalization guarantees, failing to deal with complicated large-scale generation tasks.

The advent of LLMs presents new and significant potential in boosting performance of the time series data generation. LLMs, such as GPT3, and LLaMA, have demonstrated remarkable capacities of reasoning about and understanding complex correlations and long-time dependencies in different types of data. When applying LLMs into non-text domains, to deal with the domain difference between different domains, previous works have proposed various approaches, including converting time series data into symbolic representations interpretable by LLMs [12], proposing knowledge transfer models between different domains [13]. However, applying LLMs in time series data generation is still under exploring, and this paper aims to contribute to this emerging field.

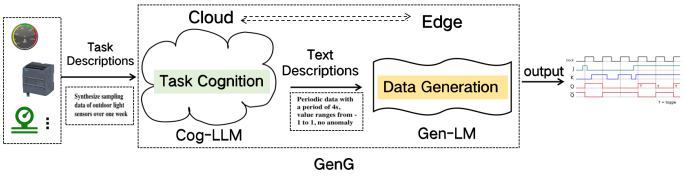

Fig. 1: Illustration of the proposed GenG.

III. Proposed Method

As shown in Figure.1, GenG consists of two modules: Cog-LLM and Gen-LM, which are responsible for solving the two problems of what to generate and how to generate, respectively. For a data synthesis task, e.g. "Synthesize the sampling data of the temperature sensor in a family home for one week ", Cog-LLM first reasons about the generation task, including the machine or sensor, application scenarios, and result expectations, etc., and generates a sequence of text outputs describing the features that the time series data to be generated should present. Conditioning on these text descriptions, Gen-LM then performs the goal-oriented time series data generation by incorporating Cog-LLM's knowledge into the generation process, where knowledge in the text data domain is transferred to the time series data domain.

A. Self-training for Cog-LLM

Cog-LLM is a domain-specific LLM that has an expert-level understanding of the time series data generation, i.e., knowing what features the time series data of one machine presents in different application scenarios. Inspired by previous work [16], we adopt an iterative self-training method to finetune LLMs to have domain-specific knowledge, where the model is used to both augment and select high-quality training examples to improve its own performance. Specifically, Cog-LLM starts from a base language model, e.g., LLaMA, a small amount of seed examples of (machine/device, text descriptions) pairs, where these texts describe the features, including trend, periodicity, stationarity, etc., of the time series of the machine, and a collection of unlabeled data. After being finetuned on the seed data, Cog-LLM possesses the initial knowledge of the behavioral characteristics of different machines. To augment the training dataset, for each unlabeled document, Cog-LLM is used to perform abstract summary and generate data pairs similar to the seed examples, which are considered as candidate augmented training data. Meanwhile, Cog-LLM is instructed to score each candidate pair (machine/device, text descriptions) to derive a quality score, based on which to select candidate pairs with high scores. During training, to further improve the quality of the augmented training dataset, the augmented data from the previous iteration is combined with the seed data to finetune an improved Cog-LLM, which is in turn used to rescore the augmented data at the current iteration. Through such an iterative self-training method, Cog-LLM is able to self-augment and self-select high-quality training examples to improve its domain-specific knowledge.

B. Conditional DM based Knowledge Transfer

To incorporate Cog-LLM's knowledge, which is learned from the text data domain, into the time series data generation process, we propose a conditional DM based knowledge transfer method to enable efficient and flexible domain transfer using off-the-shelf domain-specific expert models. Given the two models, Cog-LLM and Gen-LM, the goal is to control Gen-LM's generation results conditioning on Cog-LLM's text descriptions. As shown in Figure. 2, D_x and D_y represent the text data domain and time series data domain, respectively. Given the Cog-LLM's text description $x \in D_x$, we use an expert model $f()$ to perform the forward inference, where a latent representation Z_x is learned in the intermediate layers. $f()$ represents a pre-trained expert model that has been widely adopted in many tasks such as text classification, text translation. In total, $f(x) = D_x(E_x(x))$, where D_x and E_x denote the front and later part of $f()$, respectively, and the latent representation is represented as $Z_x = E_x(x)$. Similarly, in the time series data domain D_y, the latent representation is learned by the expert model $g()$. In this work, we adopt a BERT based classification model as $f()$ and an auto-encoder based time series data generation model as $g()$. Normally, Z_x and Z_y are nonisomorphic and misaligned. To enable the mapping between Z_x and Z_y, we transform the domain transfer task into the generation task, where Z_y is generated by a DM conditioning on Z_x.

During the forward diffusion process, given a data point x_0 sampled from a data distribution $x_0 \sim q(x)$, through sequentially adding Gaussian noise to the sample over T timesteps, it is gradually converted to a whitened latent x_t.

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{1}$$

Where β_t is constant that controls the noise step size.

During the backward process, the origin data x_0 is generated by a reversed version of the forward process.

$$p_{\theta,\xi}(x_{t-1}|x_t) = Z p_\theta(x_{t-1}|x_t) p_\xi(Z_x|x_{t-1}) \tag{2}$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t), \delta_\theta^2(x_t) I) \tag{3}$$

where, Z is a normalizing constant, the transition probability $p_\theta(x_{t-1}|x_t)$ is parameterized as $N(x_{t-1}; \mu_\theta(x_t, t), \delta_\theta^2(x_t, t)I)$, which are represented by a neural network with parameter θ. In conditional DMs, denotes the probability of the indeterminate generation belongs to the guidance signal Z_x.

Compared to classifier guidance DMs [14] that require training an extra classifier on noisy data, Classifier-free DMs [15] can perform the guidance by mixing the conditional and unconditional score estimates. Based on the Bayes Formula, the score estimate of Classifier-free DM can be updated:

$$\begin{aligned}\overline{\epsilon_\theta}(x_t, t, Z_x) =& \epsilon_\theta(x_t, t, Z_x) - \sqrt{1-\alpha}\omega \bigtriangledown_{x_t} logp(Z_x|x_t) \\ =& \epsilon_\theta(x_t, t, Z_x) + \omega(\epsilon_\theta(x_t, t, Z_x) - \epsilon_\theta(x_t, t)) \\ =& (\omega+1)\epsilon_\theta(x_t, t, Z_x) - \omega\epsilon_\theta(x_t, t)\end{aligned} \tag{4}$$

where w is a parameter denoting the strength of the guidance.

As for the training data, we use the existing time series data analysis tools to build the (time series sequence, text descriptions) pairs. Given a sequence of time series data, we first use statistical methods, deep learning based algorithms, etc., to perform data analysis, including periodicity discovery, amplitude calculation, and anomaly detection, and then arrange these results in natural language with a predefined format. During training, we use a single model to parameterize both the conditional and unconditional models and jointly train them by randomly setting the guidance signal of the unconditional model as null.

After training, knowledge transfer between D_x and D_y can be achieved by the following steps: (1) obtaining the text description x from Cog-LLM, (2) encoding x into the latent representation Z_x with $f(x)$, (3) conditionally generating the latent representation Z_y with the classifier-free DM, (4) decoding Z_y into y with the later part of $g()$. In such a way, Cog-LLM can effectively control the abstract features of the synthesized time series data.

C. Two-stage Data Generation Process

While Cog-LLM's text descriptions only encode the abstract features of the time series data, to further increase the quality of the generation results, we propose a two-stage data generation process. Briefly, in stage one, Cog-LLM's text descriptions are used as the first guidance to ensure the generated time series data to have the desired abstract features, such as trend, periodicity, enabling coarse-grained guidance on the generation results. In stage two, the example samples or historically generated time series sequences are used as the second guidance to continually optimize the generation results from stage one. This enables the generated data to have controllable details, such as numerical value and local variations, achieving fine-grained guidance of the generation process. In the generation process of long-term times series, which usually adopts an iteration generation mechanism, the data generated from the previous step can be used as guidance for the subsequent generation iteration, which guarantees the generated data with consistent statistical characteristics, such as amplitude, mean, and variance.

IV. Experimental Results

We first evaluate the performance of Cog-LLM and Gen-LM separately, and then we test the overall generation results of the proposed GenG in different generation tasks.

A. Cog-LLM Performance Evaluation

Setup. To test Cog-LLM's performance in understanding and reasoning about the time series data generation tasks, we train a domain-specific model based on LLaMA finetuned with data from a mixture of self-constructed (machine/device, text descriptions) pairs and human expert-written examples. To obtain the seed data for the initial model, we select about 300 application scenarios utilizing various devices to generate time series data ranging from weather records, equipment measurements, patience health metrics, financial indicators,

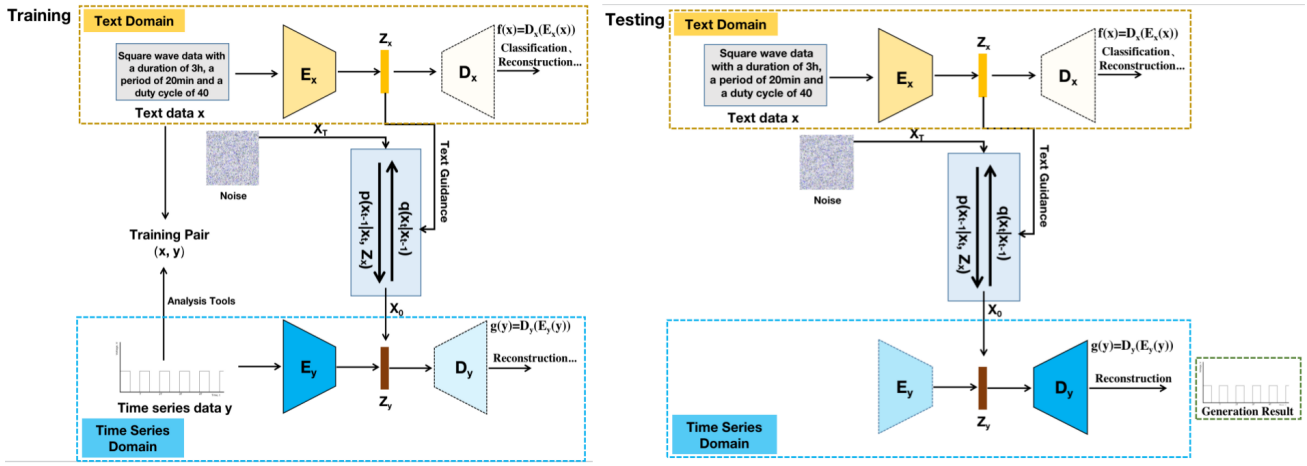

Fig. 2: Illustration of the conditional DM based knowledge transfer between the text domain and time series domain. For latent representations $Z_x = E_x(x)$ and $Z_y = E_y(y)$, a conditional DM learns to generate Z_y conditioning on Z_x, enabling efficient transfer between them using off-the-shelf domain models.

etc, and then we manually or use ChatGPT to summarize the features of the time series data generated by different devices, generating about 1000 (device, text descriptions) pairs as the seed data to train the model. For the unlabeled data, we use the English portion of the Clueweb corpus [17] and sample about 34K segments related to the topic of time series data generation. During training, we use the pretrained LLaMA model with 7B parameters as the base model for finetuning. Following the training strategy in [16], we set the initial learning rate to $1e^{-5}$ which linearly decays to $9e^{-6}$ at the end of training. The batch size is set to 32. Additionally, we compare our model with several baselines.

Results. We evaluate Cog-LLM by comparing its generation results against human-written answers, where a score is assigned based on GPT judgment and human preference evaluation. Specifically, GPT is prompted to assign a score out of ten to Cog-LLM's responses and provide an explanation, where the human-written answers are used as references. Besides, 3 human participants are asked to rate Cog-LLM's responses on a scale of one to ten, resulting in one averaged human score. The final score is calculated by weighting these two scores, where the weight ratio of human to GPT is 7:3. As shown in Table. I, with the proposed training approach, Cog-LLM obviously outperforms the pre-trained LLaMA in generating reasonable text descriptions of the time series data. Meanwhile, the model trained on the data in the second selection iteration obtains a score 26.38% higher than the model trained on the data without selection, but its score is 27.95% lower than the model trained on the data in the fifth selection iteration, demonstrating that Cog-LLM's performance can be increased by the iterative selecting procedure. Additionally, Figure. 3 presents the comparison results of finetuning on training data of different quality and quantity. As shown, the model's performance is slightly improved by increasing the size of the training data, while can be significantly improved by improving the quality of the training data.

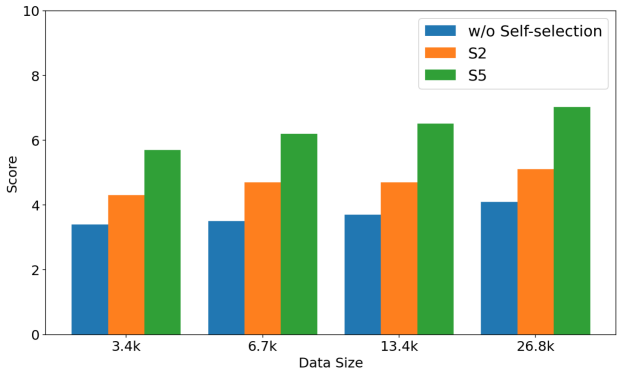

Fig. 3: Evaluating self-selected data of different quality and quantity. We compare three augmentation datasets: without self-selection, S2 data in the second selection iteration and S5 data in the second selection iteration.

TABLE I: Scores of the finetuned models using different training data.

Models	Data	#Examples	GPT S.	Hu. S.	Score
LLaMA	-	-	4.1	4.3	4.24
Cog-LLM$_s$	Seed	1000	4.0	4.7	4.49
Cog-LLM$_5$	Aug, S_5	26.8k	6.7	7.2	**7.05**
Cog-LLM$_2$	Aug, S_2	83.7k	5.3	5.6	5.51
Cog-LLM$_{all}$	Aug, all	134K	4.5	4.3	4.36

B. Domain Transfer Result Evaluation

Setup. We next evaluate the capacity of the DM-based approach in enabling knowledge transfer between the text data domain and the time series data domain. Specifically, we adopt a BERT based classification model and an auto-encoder based time series data generation model to extract the latent representations of the text data and time series

data, respectively. Our DM model is based on Transformer architecture with 80M parameters, with a sequence length of 64, a diffusion step of 500, and a noise schedule. The embedding dimension of the time series data is 64. For the training data, we select time series data from several open-source datasets and use the existing time series data analysis methods to build about 96k (time series sequence, text descriptions) pairs, where features of the time series data, e.g., trend, periodicity, amplitude, anomaly, are analyzed and arranged in natural language with predefined format. During training, we follow the classifier-free DM training strategy [15] and train our model using the Adam optimizer with a batch size of 16 and learning rate 2×10^{-4}. We conduct control experiments to select text descriptions as testing examples, for example, "Periodic data with a period of 6s, value ranges from -5 to 5, no anomaly", where different types of time series data are intentionally generated in 4 categories of generation tasks. For each text description, we generate 20 time series sequences. Then, we evaluate the generation results in terms of trend, periodicity, value range, and anomaly.

Results We present qualitative and quantitative results of the generated time series data conditioning on the text descriptions. As shown in Figure. 4, the generated time series sequences are visually consistent with the text descriptions, which demonstrates the effectiveness of the proposed knowledge transfer approach. However, since text descriptions are abstract and ambiguous, there may be many time series data that corresponds to the same text descriptions. The generated time series sequences vary widely in detailed features despite sharing similar abstract features. Statistical results in Table. II show that these generated time series sequences achieve excellent performance in conforming to abstract features, e.g, trend, periodicity, anomaly, but showing weak performance in satisfying detailed features, e.g, period size (P-Size), value range (V-Range), mean value(M-Value). As presented, despite the high accuracy in abstract features, the generated data still requires considerable improvement in statistics.

TABLE II: Quantitative results of the generated time series data in terms of abstract features and detailed features. Task1-4 is designed to generate trend data, periodic data, anomaly data, and mixed data, respectively.

	Abstract Features			**Detailed Features**		
	Trend	**Per.**	**Ano.**	**P-Size**	**V-Range**	**M-value**
Task1	91.2%	-	-	-	7.8%	6.2%
Task2	-	96.2%	-	7.5%	7.2%	5.2%
Task3	-	-	94.7%	-	13.4%	7.8%
Task4	93.1%	92.2%	92.7%	8.1%	7.4%	6.5%

C. Overall Performance Evaluation

Setup. Finally, we evaluate the overall performance of the proposed GenG. Given abstract task descriptions, GenG needs to understand the data generation task and synthesize high-fidelity time series data. We compare GenG with conditional GAN-based and DM-based models, which use the pretrained

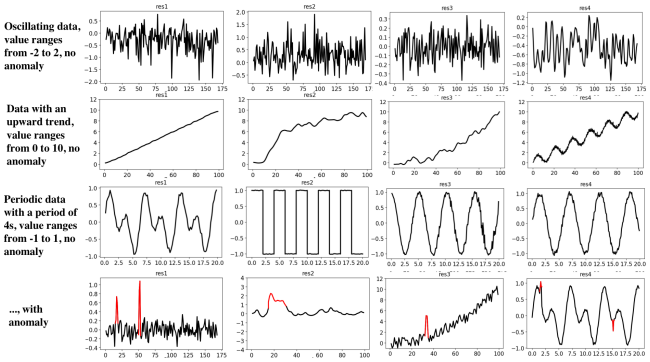

Fig. 4: Qualitative results of the synthesized time series data in different generation tasks. For each text description (Left), we present 4 synthesized data sequences (presented in one row). The red curves represented the anomalies in the data sequences.

LLaMA (without finetuning) to generate conditioning signals based on the task descriptions. We use the same training data to train GenG and the baseline models (C-GAN and C-DM). For the result evaluation, we use different evaluation metrics to assess the generated data in terms of: (1) fidelity that measures the visual and statistical similarity of the synthesized data to the real data. Except for the statistics-based evaluation metrics, e.g. trend, periodicity, and value, we also adopt the Fréchet Inception Distance (FID) and Inception Score (IS) to measure the quality and diversity of generated data. Additionally, we train a 2-layer LSTM based classification model to distinguish between time series sequences from the real and synthesized datasets, where the classification error is used to calculate the discrimination score (DS); (2) usefulness that tests whether the synthesized data is as useful as the real date when used for anomaly detection tasks. (i.e. train-on-synthetic, test-on-real, TSTR)). We train an anomaly detection model based on the synthesized time series data and then test the model on the real data, where the mean absolute error (MAE) is used for performance evaluation. For comparisons, we report the accuracy of training on real and testing on real (TRTR), training on real and testing on synthetic (TRTS) results in the same experiments.

Additionally, we also conduct ablation studies to verify the importance of the key components of GenG. For this aim, we design several models including: 1) **LLaMa and two-stage generation process (L-two)** that adopts a pretrained LLaMA with the proposed two-stage generation process; 2) **Human and two-stage generation process (H-two)** that includes a human to replace the Cog-LLM, with the proposed two-stage generation process; 3) **Cog-LLM and one-stage generation process (Cog-one)** that adopts the finetuned Cog-LLM with only one-stage generation process; 4) **Cog-LLM and two-stage generation process (GenG)** that adopts the finetuned Cog-LLM with the proposed two-stage generation process.

Results. Figure. 5 presents the synthesized time series data in different generation tasks. We can see that GenG

TABLE III: Quantitative comparison with other approaches in terms of Cognition, Fidelity, and Usefulness.

Model	Cognition			Fidelity			Usefulness		
	GPT Score	Human Score	Score	FID (↓)	IS (↑)	DS (↓)	TRTR	TRTS	TSTR
C-GAN	4.1	4.3	4.24	4.72	3.92	0.562	0.79	0.73	0.61
C-DM	3.8	4.1	4.01	5.16	3.87	0.487	0.83	**0.76**	0.57
GenG	6.8	7.1	**7.01**	**3.24**	**5.63**	**0.313**	0.77	0.75	**0.72**

can correctly understand abstract language commands and synthesize high-fidelity time series data with controllable features. In contrast, although C-GAN and C-DM can solve simple generation tasks (Task1 and Task2), they either generate random data (Task3) or simply replicate the sample (Task4) when the task descriptions become complex. Detailed results in Table. III show that GenG outperforms the baseline models in both reasoning about the generation task and generation capacities. Additionally, ablation results in Table. IV demonstrate that both the fine-tuned Cog-LLM and the two-stage generation process have a significant impact on the quality of the generated time series data.

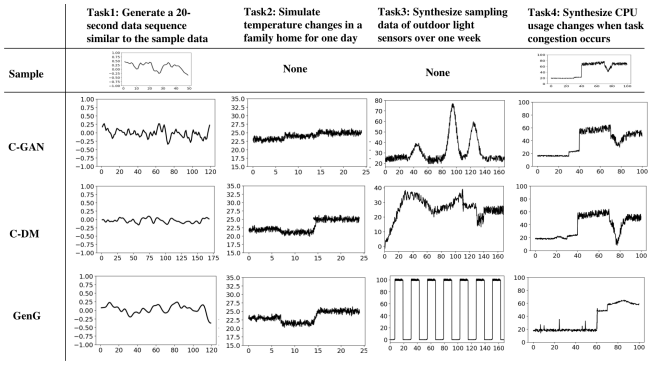

Fig. 5: Qualitative comparison with other approaches in different generation tasks.

TABLE IV: Comparison results of the ablation studies.

Model	Fidelity			Usefulness		
	FID (↓)	IS (↑)	DS (↓)	TRTR	TRTS	TSTR
L-two	4.81	3.87	0.612	0.83	0.65	0.63
H-two	**3.11**	5.47	0.375	0.84	0.71	0.70
Cog-one	3.63	4.13	0.523	0.81	0.61	0.55
GenG	3.24	**5.63**	**0.313**	0.80	**0.73**	**0.72**

V. Conclusion

In this paper, we propose a novel yet efficient approach for generalized time series data generation, which incorporates knowledge learned from the text data domain and time series data domain to generate time series data with controllable features. GenG decomposes the time series data generation task into two sub-tasks: the task of what to generate and the task of how to generate. It adopts a self-training method to finetune LLMs to reason about the generation task, and then generates time series data conditioning on the finetuned LLM's text descriptions, where a classifier-free DM based approach is proposed to facilitate knowledge transfer between different domains. Additionally, a two-stage generation process is proposed to increase the quality of the generated data. Experiment results demonstrate the promise of the proposed approach in generating complex time series data in different generation tasks and achieving great generalization capacity.

References

[1] D. N. Wang, L. Li, D. Zhao,"Corporate finance risk prediction based on LightGBM," Information Sciences, vol. 602, pp. 259-268, 2022.
[2] Y. Hu, Q. Jia, Y. Yao, Y. Lee, C. Wang, X. Zhou, R. Xie, F. R. Yu, "Industrial Internet of Things Intelligence Empowering Smart Manufacturing: A Literature Review," arXiv preprint arXiv:2312.16174, 2023.
[3] Y. Peng, G. Tan, H. Si, "HKGAIL: Policy shaping via integrating human knowledge with generative adversarial imitation learning," IET Intelligent Transport Systems, 17(7), pp. 1302-1311, 2023.
[4] M. Liu, L. He, B. Hu, S. Li, "Recurrent neural network with noise rejection for cyclic motion generation of robotic manipulators," Neural Networks, vol. 138, pp. 164-178, 2021.
[5] Q. Hou, D. Zhou, J. Feng, "Coordinate attention for efficient mobile network design," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, (pp. 13713-13722, 2021.
[6] X. B. Jin, W. T. Gong, J. L. Kong, Y. T. Bai, T. L. Su, "PFVAE: a planar flow-based variational auto-encoder prediction model for time series data," Mathematics, vol. 10, p.610-616, 2022
[7] M. Dogariu, L. D. Ştefan, B. A. Boteanu, C. Lamba, B. Kim, B. Ionescu,"Generation of realistic synthetic financial time-series," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 18, pp. 1-27, 2022.
[8] J. Yoon, D. Jarrett, M. Van der Schaar, "Time-series generative adversarial networks," Advances in neural information processing systems, vol. 32, 2019.
[9] Y. Tashiro, J. Song, Y. Song, S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," Advances in Neural Information Processing Systems, vol. 34, pp. 24804-24816, 2021.
[10] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, S. Fidler, "Atiss: Autoregressive transformers for indoor scene synthesis," Advances in Neural Information Processing Systems, vol. 34, pp.12013-12026, 2021.
[11] T. Zhou, H. Fu, G. Chen, J. Shen, L. Shao, "Hi-net: hybrid-fusion network for multi-modal MR image synthesis," IEEE transactions on medical imaging, vol. 39, pp.2772-2781, 2020.
[12] X. Yu, Z. Chen, Y. Ling, S. Dong, Z. Liu, Y. Lu, "Temporal Data Meets LLM–Explainable Financial Time Series Forecasting," arXiv preprint arXiv:2306.11025, 2023.
[13] R. Rombach, P. Esser, B. Ommer, "Network-to-network translation with conditional invertible neural networks," Advances in Neural Information Processing Systems, vol. 33, pp. 2784-2797, 2020.
[14] H. Kim, S. Kim, S. Yoon, "Guided-tts: A diffusion model for text-to-speech via classifier guidance," In International Conference on Machine Learning, pp. 11119-11133, 2022.
[15] J. Ho, T. Salimans, "Classifier-free diffusion guidance," arXiv preprint arXiv:2207.12598, 2022.
[16] X. Li, P. Yu, C. Zhou, T. Schick, L. Zettlemoyer, O. Levy, J. Weston, M. Lewis, "Self-alignment with instruction backtranslation," arXiv preprint arXiv:2308.06259, 2023.
[17] A. Overwijk, C. Xiong, J. Callan, "ClueWeb22: 10 billion web documents with rich information," In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3360-3362, 2022.