Large language model as user daily behavior data generator: balancing population diversity and individual personality

Anonymous ACL submission

Abstract

Predicting human daily behavior is challenging due to the complexity of routine patterns and short-term fluctuations. While data-driven models have improved behavior prediction by leveraging empirical data from various platforms and devices, the reliance on sensitive, large-scale user data raises privacy concerns and limits data availability. Synthetic data generation has emerged as a promising solution, though existing methods are often limited to specific applications. In this work, we introduce BehaviorGen, a framework that uses large language models (LLMs) to generate high-quality synthetic behavior data. By simulating user behavior based on profiles and real events, BehaviorGen supports data augmentation and replacement in behavior prediction models. We evaluate its performance in scenarios such as pertaining augmentation, fine-tuning replacement, and fine-tuning augmentation, achieving significant improvements in human mobility and smartphone usage predictions, with gains of up to 18.9%. Our results demonstrate the potential of BehaviorGen to enhance user behavior modeling through flexible and privacy-preserving synthetic data generation.

1 Introduction

Predicting human behavior is challenging due to a mix of habitual patterns and context-driven fluctuations (Nadkarni, 2016). With the growing availability of user behavior data from web platforms and smart devices, data-driven models have advanced significantly, enabling intelligent systems to support daily activities (Zhang and Dai, 2018; Zhang et al., 2019; Li et al., 2022; Chung and Lee, 2018; Tulshan and Dhage, 2019; Savcisens et al., 2023). However, privacy concerns and difficulties in collecting large-scale, high-quality data have hindered the development of behavior modeling applications. Synthetic data generation has emerged as a promising solution, with deep generative models already employed in varied domains (Shi et al., 2019; Liu et al., 2022; Luo et al., 2022; Feng et al., 2020; Yuan et al., 2024). Despite these advancements, existing approaches lack generalization across diverse scenarios.

Large language models (LLMs) (Zhao et al., 2023; Brown, 2020; Long et al., 2024) offer a new avenue for generating synthetic behavior data, as they have demonstrated capabilities in simulating human behaviors (Shao et al., 2024; Wang et al., 2024; Li et al., 2024). These models not only capture population-level diversity but can also generate highly personalized synthetic data.

Therefore, we introduce the BehaviorGen framework, which prompts LLMs to simulate a specific user's behavior based on a provided profile and a few real behavior events. This approach enables the flexible generation of high-quality synthetic user behavior data.

We evaluate BehaviorGen's data generation capabilities across various usage scenarios, including: (1) *pretraining data augmentation*, enhancing generalist models with diverse behavior data; (2) *fine-tuning data replacement*, generating personalized data to replace real data; (3) *fine-tuning data augmentation*, supplementing limited real data with synthetic personalized data.

Surprisingly, we find that BehaviorGen enables LLMs to generate user behaviors that reflect both population diversity and individual personality. BehaviorGen demonstrates strong performance metrics across all three scenarios.

2 Preliminary

2.1 Behavior Data Generation Problem

Now, we give a formal definition of our research problem: PROBLEM (User behavior generation). The user behavior can be represented as

$$x_i = (d_i, t_i, l_i, b_i, p_i)$$

where b_i denotes a specific behavior occurring at location l_i during time slot t_i on day d_i. Here, d_i, t_i, l_i, and b_i are the weekday, time slot, location, and behavior IDs, respectively. We denote the sets of weekdays, time slots, locations, and behaviors as $\mathcal{D}$, $\mathcal{T}$, $\mathcal{L}$, and $\mathcal{B}$, with sizes N_D, N_T, N_L, and N_B.

Furthermore, p_i represents the user profile, which consists of five key attributes: age, education, gender, consumption, and occupation.

The user behavior sequences can be represented as

$$\boldsymbol{x_i} = [x_1, x_2, x_3, ..., x_I]$$

Our goal is to generate the user behavior sequence, which can be formulated as:

$$[\hat{x_1}, \hat{x_2}, \hat{x_3}, ..., \hat{x_O}] = G([x_1, x_2, x_3, ..., x_I]) \quad (1)$$

By incorporating user profiles $\mathcal{P}$ into the behavior generation process, our method ensures that the generated behavior sequences align with realistic user characteristics, leading to more accurate and personalized synthetic data.

2.2 Behavior Prediction Problem

To demonstrate the effectiveness of the generated sequence, we design the user behavior prediction experiment. User behavior prediction aims to forecast future user behavior based on its past I event series, which can be formed as,

$$\hat{b}_t = f(x_{t-I}, x_{t-I+1}, ..., x_{t-1}) \quad (2)$$

3 BehaviorGen Framework

3.1 Data Generation Procedure

3.1.1 Data generation Process

Role Setting: In this stage, the Large Language Model (LLM) is assigned the role of "Generator." We choose gpt-4o-2024-0806 model as our generator. As shown in Figure 1, by explicitly defining the role, the LLM is better equipped to understand the task structure and objectives, leading to more coherent and contextually appropriate output.

Format Restrictions: In order to ensure that the generated data adheres to a consistent and interpretable structure, we impose strict formatting requirements, where the output is specified as [weekday, timestamp, location, intent]. Additionally, we limit the value and scope of the generated data to ensure the validity of generated data, reducing the subsequent steps in data processing.

Segmented Generation: Given the complexity of generating long sequences of behavioral data, we utilize a segmented approach, where the user's behavior is divided into weekly segments. This reduces the risk of context drift and helps maintain consistency throughout the generation process.We give detailed experiments and explanations in the Appendix A.10. as to why we chose weekly segments.

3.2 Usage Scenarios

3.2.1 Pretraining Augmentation

In real-world scenarios, application service providers can only collect a limited amount of user data, which is insufficient to support the training of a population-level behavior prediction model. Therefore, it is necessary to synthesize additional data to enhance population diversity. This need arises from the challenges of establishing a population-level model capable of capturing common behavioral patterns.

Building upon this foundational training, we incorporate behavioral data generated by the LLM as a means of data augmentation.

3.2.2 Finetuning Replacement

Post pre-training, the fine-tuning phase serves as a pivotal step in enhancing the personalization and accuracy of recommendation systems. However, leveraging real user behavioral data in this phase poses significant privacy and security concerns. To mitigate these risks, we propose using behavioral data generated by the LLM as a replacement for real user data during fine-tuning

This approach enables the fine-tuning of the pretrained model while preserving user privacy.

3.2.3 Finetuning Augmentation

Accurate prediction of long-tail user behavior within recommendation systems poses a significant challenge due to the infrequency of such data and the inherent difficulties in its collection. In response to this challenge, we advocate for a strategy that involves the synthesis of behavioral data using a limited amount of real user behavior data as a base data. By combining LLM-generated user data with this small set of authentic user data, we aim to enrich the training dataset for fine-tuning. This approach enhances the model's capacity to predict long-tail user behaviors, ensuring that even less common patterns can be adequately represented.

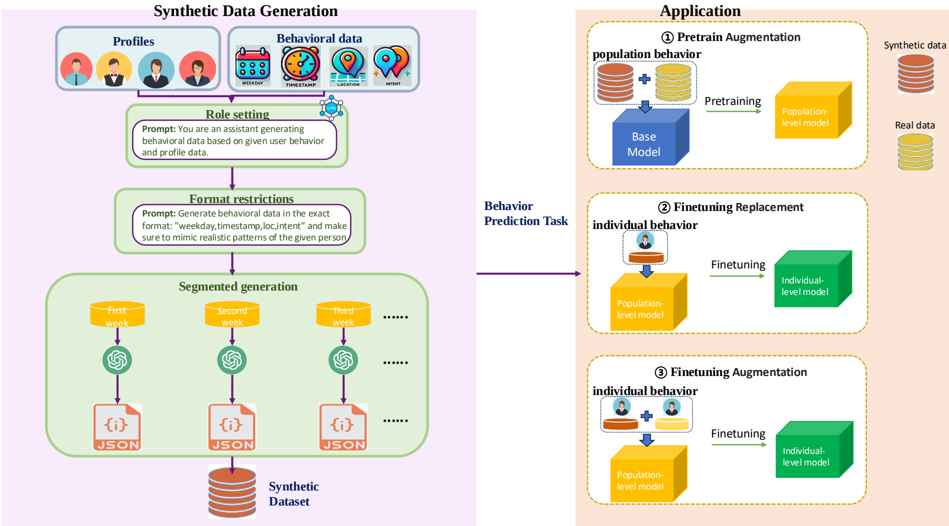


Figure 1: The Framework of BehaviorGen.

Category		Tencent Dataset								Smartphone Dataset							
	Backbone	Bert4Rec				PITuning				Bert4Rec				PITuning			
		Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5
Real Data	Pretrained	0.427	0.466	0.666	0.663	0.418	0.449	**0.667**	**0.661**	0.149	0.280	0.515	0.551	0.123	0.168	0.435	0.468
Real Data + Synthetic Data	SeqGAN	0.417	0.452	0.682	0.676	0.401	0.429	0.638	0.630	0.150	0.259	0.524	0.554	0.134	0.162	0.435	0.463
	DiffuSeq	0.436	0.471	0.684	0.685	0.281	0.366	0.624	0.620	0.167	0.283	0.524	0.557	0.136	0.174	0.433	0.467
	UPC_SDG	0.424	0.457	0.676	0.670	0.384	0.417	0.630	0.632	0.188	0.295	0.528	0.558	0.130	0.169	0.438	0.472
	Ours	**0.447**	**0.480**	**0.702**	**0.694**	**0.426**	**0.450**	0.655	0.659	**0.213**	**0.315**	**0.543**	**0.570**	**0.201**	**0.186**	**0.454**	**0.479**
	Improvement	2.5%	1.9%	2.6%	1.3%	1.9%	0.2%	-1.8%	-0.3%	13.3%	6.8%	2.8%	2.2%	4.8%	6.9%	3.7%	1.5%

Table 1: Overall prediction performance Pretrain Augmentation compared with baselines on Tencent and Smartphone datasets. The improvement here is calculated using the formula: (ours - the best result from pretrained and baseline) / the best result from pretrained and baseline.

4 Experiment

4.1 Experiment Settings

4.1.1 Datasets

We evaluate the performance of our model on two large-scale real-world activity datasets.Details of the two datasets can be found in the appendix A.5.

4.1.2 Metrics

To assess model performance, we employ four widely used metrics: precision (Pre), recall (Rec), and NDCG(N). NDCG gauge classification accuracy and ranking quality, respectively, while Pre and Rec evaluate the average prediction accuracy for each intent, indicating the model's predictive quality across intents. Refer to Appendix A.6 for metric calculations.

4.1.3 Baselines

We carefully select the following three representative methods to compare with our proposed algorithm, which include generative methods for sequence data (SeqGAN (Yu et al., 2017)), diffusion-based sequence generation models (DiffuSeq (Gong et al., 2022)), and a synthetic data generation method (UPC_SDG (Liu et al., 2022)). We provide the details of baselines in Appendix A.7.

4.1.4 Evaluation Backbones.

We choose PITuning (Gong et al., 2024) and Bert4Rec (Sun et al., 2019) as the evaluation backbone.

• **PITuning** PITuning is a Population-Individual Tuning framework that enhances common pattern extraction through dynamic event-to-intent transition modeling and addresses long-tailed preferences via adaptive unlearning strategies.

• **Bert4Rec** Bert4Rec, a bidirectional encoder representation from Transformers, enhances the power of the historical sequence representations by jointly conditioning the left and right context.

4.2 Overall Performance Analysis

We report experiments on three usage scenarios for two prediction applications (Table 1-3). Across all experiments, our framework demonstrates clear superiority over baseline methods, both in terms of performance metrics and its ability to produce

Category	Backbone	Tencent Dataset								Smartphone Dataset							
		Bert4Rec				PITuning				Bert4Rec				PITuning			
		Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5
Real Data	Pretrained	0.447	0.474	0.693	0.691	0.422	0.454	0.684	0.678	0.207	0.340	0.542	0.568	0.126	0.178	0.440	0.478
	Finetuned	0.597	0.614	0.790	0.791	0.583	0.604	0.780	0.774	0.322	0.366	0.594	0.614	0.306	0.355	0.627	0.668
Synthetic Data	SeqGAN	0.185	0.221	0.381	0.375	0.194	0.228	0.394	0.392	0.288	0.309	0.542	0.577	0.227	0.296	0.576	0.616
	DiffuSeq	0.152	0.223	0.409	0.409	0.161	0.234	0.417	0.425	0.233	**0.340**	0.550	0.589	0.228	0.301	0.591	0.628
	UPC_SDG	0.172	0.148	0.229	0.223	0.170	0.159	0.236	0.234	0.280	0.315	0.543	0.569	0.260	0.317	0.562	0.585
	Ours	**0.540**	**0.539**	**0.746**	**0.734**	**0.516**	**0.529**	**0.733**	**0.724**	**0.308**	0.334	**0.568**	**0.593**	**0.270**	**0.333**	**0.602**	**0.643**
	Replacement	62.0%	46.4%	54.4%	43.0%	58.4%	50.0%	51.0%	47.9%	87.8%	-23.1%	50.0%	54.3%	80.0%	87.6%	86.6%	86.8%

Table 2: Overall prediction performance Finetuning Replacement compared with baselines on Tencent and Smartphone datasets. The replacement here is calculated using the formula: (ours - pretrained) / (real data finetuned - pretrained).

Category	Backbone	Tencent Dataset								Smartphone Dataset							
		Bert4Rec				PITuning				Bert4Rec				PITuning			
		Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5	Pre	Rec	N@3	N@5
Limited Real Data	Finetuned	0.495	0.528	**0.709**	**0.715**	0.455	0.493	0.697	0.695	0.322	0.366	0.594	0.614	0.306	0.355	0.627	0.668
Limited Real Data + Synthetic Data	SeqGAN	0.261	0.297	0.494	0.496	0.251	0.287	0.488	0.493	0.331	0.377	0.600	0.621	0.315	0.343	0.624	0.675
	DiffuSeq	0.219	0.298	0.494	0.501	0.207	0.282	0.504	0.503	0.333	0.376	0.596	0.621	0.316	0.356	0.628	0.674
	UPC_SDG	0.309	0.277	0.407	0.419	0.277	0.278	0.435	0.439	0.339	0.378	0.600	0.621	0.308	0.354	0.635	0.672
	Ours	**0.545**	**0.547**	0.708	0.709	**0.541**	**0.541**	**0.728**	**0.722**	**0.345**	**0.398**	**0.612**	**0.635**	**0.328**	**0.364**	**0.643**	**0.682**
	Improvement	10.1%	3.5%	-0.1%	-0.8%	18.9%	9.7%	4.4%	3.9%	1.8%	5.3%	2%	2.3%	3.8%	2.2%	1.3%	1.0%

Table 3: Overall prediction performance Finetuning Augmentation compared with baselines on Tencent and Smartphone datasets. The improvement here is calculated using the formula: (ours - the best result from pretrained and baseline) / the best result from pretrained and baseline.

high-quality synthetic data. The fine balance between diversity and faithfulness achieved by our framework not only leads to enhanced model performance but also offers a scalable solution for privacy-preserving data generation.

• **Our method demonstrates minimal discrepancies compared to fine-tuning with real data.** Specifically, the proposed framework effectively generates personalized synthetic data, crucial for maintaining performance levels that closely resemble those achieved through fine-tuning on real data, all while ensuring user privacy. As evidenced in Table 2, models fine-tuned using synthetic data exhibit a performance gap of merely 5.7% and 1.4% in average precision, achieving scores of 0.540 on the Tencent dataset and 0.308 on the Smartphone dataset, respectively. Furthermore, the average replacement rate of 57.6% highlights the equilibrium our framework achieves between privacy preservation and model efficacy.

• **Population-Level Analysis** In the pre-training phase, as shown in Figure 1, we performed data augmentation using population-level data combined with synthetic data. The emphasis during this phase was on extracting common features across the population. The introduction of synthetic data not only enriched the diversity of user behavior patterns but also maintained a high level of faithfulness to real user data. As presented in Table 1, models pre-trained with a mix of real and synthetic data exhibited significant improvements in accuracy and recall, indicating that synthetic data introduces sufficient variability without compromising the coherence of user trajectories.

• **Individual-Level Analysis** In the fine-tuning phase, we synthesized a personalized dataset derived from individual user data to replace real user data. This approach not only ensures privacy but also faithfully captures individualized behavior patterns critical for intent prediction and behavior modeling tasks. At the individual level, our synthetic data remains faithful to real user behaviors while introducing subtle variations that better capture users' distinct decision-making processes.

As shown in Tables 2 and 3, models fine-tuned with synthetic data significantly outperformed those fine-tuned solely on real data, particularly in metrics such as NDCG@3 and NDCG@5. The higher NDCG scores highlight that synthetic data more effectively mirrors individual users' preferences, improving the model's performance in recommendation tasks.

5 Conclusion

This preliminary study explores the potential of large language models (LLMs) for generating synthetic user behavior data. Experimental results across three synthetic data usage scenarios show promising performance in enhancing two downstream behavior prediction applications. These findings suggest that the generated synthetic behavior data effectively captures both population-level diversity and individual-level specificity, reflecting the complexity of human daily behavioral patterns.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA. Association for Computing Machinery.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Derek Chen, Celine Lee, Yunan Lu, Domenic Rosati, and Zhou Yu. 2023. Mixture of soft prompts for controllable data generation. *arXiv preprint arXiv:2303.01580*.

Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. Autodebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–30.

Hyunji Chung and Sangjin Lee. 2018. Intelligent virtual assistant knows your life. *arXiv preprint arXiv:1803.00466*.

YA DeMontjoye. 2013. Seqgan: Sequence generative adversarial nets with policy gradient. 31(1).

Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to simulate human mobility. In *KDD*, pages 3426–3433.

Jiahui Gong, Jingtao Ding, Fanjin Meng, Guilong Chen, Hong Chen, Shen Zhao, Haisheng Lu, and Yong Li. 2024. A population-to-individual tuning framework for adapting pretrained lm to on-device user intent prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 896–907, New York, NY, USA. Association for Computing Machinery.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Xu Guo and Yiqiang Chen. 2024. Generative ai for synthetic data generation: Methods, challenges and the future. *arXiv preprint arXiv:2403.04190*.

Xu Guo, Boyang Li, and Han Yu. 2022. Improving the sample efficiency of prompt tuning with domain adaptation. *arXiv preprint arXiv:2210.02952*.

Xingwei He, Zhenghao Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. 2023. Annollm: Making large language models to be better crowdsourced annotators. *arXiv preprint arXiv:2303.16854*.

Tong Li, Tong Xia, Huandong Wang, Zhen Tu, Sasu Tarkoma, Zhu Han, and Pan Hui. 2022. Smartphone app usage analysis: datasets, methods, and applications. *IEEE Communications Surveys & Tutorials*, 24(2):937–966.

Xuchuan Li, Fei Huang, Jianrong Lv, Zhixiong Xiao, Guolong Li, and Yang Yue. 2024. Be more real: Travel diary generation using llm agents and individual profiles. *arXiv preprint arXiv:2407.18932*.

Fan Liu, Zhiyong Cheng, Huilin Chen, Yinwei Wei, Liqiang Nie, and Mohan Kankanhalli. 2022. Privacy-preserving synthetic data generation for recommendation systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1379–1389.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

Xufang Luo, Zheng Liu, Shitao Xiao, Xing Xie, and Dongsheng Li. 2022. Mindsim: user simulator for news recommenders. In *WWW*, pages 2067–2077.

Devaki Nadkarni. 2016. Superforecasting: The art & science of forecasting. *Anvesha*, 9(1):61.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Vladimir Provalov, Elizaveta Stavinova, and Petr Chunaev. 2021. Synevarec: A framework for evaluating recommender systems on synthetic data classes. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 55–64.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*, pages 1–7.

Germans Savcisens, Tina Eliassi-Rad, Lars Kai Hansen, Laust Hvas Mortensen, Lau Lilleholt, Anna Rogers, Ingo Zettler, and Sune Lehmann. 2023. Using sequences of life-events to predict human lives. *Nature Computational Science*, pages 1–14.

Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. 2024. Beyond imitation: Generating human mobility from context-aware reasoning with large language models. *arXiv preprint arXiv:2402.09836*.

Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. 2019. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *AAAI*, volume 33, pages 4902–4909.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Amrita S Tulshan and Sudhir Namdeorao Dhage. 2019. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *Advances in Signal Processing and Intelligent Recognition Systems: 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 4*, pages 190–201. Springer.

Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, and Chuan Xiao. 2024. Large language models as urban residents: An llm agent framework for personal mobility generation. *arXiv preprint arXiv:2402.14744*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in neural information processing systems*.

Yuan Yuan, Jingtao Ding, Huandong Wang, and Depeng Jin. 2024. Generating daily activities with need dynamics. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–28.

Hongtao Zhang and Lingcheng Dai. 2018. Mobility prediction: A survey on state-of-the-art schemes and future applications. *IEEE access*, 7:802–822.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Larisa Zherdeva, Denis Zherdev, and Artem Nikonorov. 2021. Prediction of human behavior with synthetic data. In *2021 International Conference on Information Technology and Nanotechnology (ITNT)*, pages 1–6.

A Appendix

A.1 Related Work

A.1.1 Synthetic Data Generation with LLMs

Synthetic data generation has gained significant momentum with the advent of large language models (LLMs) (Guo and Chen, 2024). The data generated by LLMs closely approximates real-world data, making this approach a powerful solution to addressing the challenges of resource scarcity.

Designing an informative prompt is key to effective data generation with LLMs. Yu et al. (2023) explore synthetic data generation using diversely attributed prompts, which have the potential to produce diverse and richly attributed synthetic data. Reynolds and McDonell (2021) propose MetaPrompt, a method where an expanded prompt is first generated by ChatGPT, then used to further prompt LLMs for data generation. Another promising approach for task-specific data generation is to aggregate a few-shot dataset and perform parameter-efficient adaptation on the LLM (Guo et al., 2022). Chen et al. (2023) train a set of soft prompt embeddings on few-shot, task-specific training data to condition the LLM for more effective text generation. He et al. (2023) AnnoLLM, an LLM-powered annotation system. It first prompts LLMs to explain the reasoning behind a ground truth label, then uses these explanations to create a few-shot chain-of-thought prompt for annotating unlabeled data.

However, existing work has not adequately addressed the balance between population diversity and individual preference, a crucial consideration in user behavior generation.

A.1.2 Synthetic Data for User Behavior Modeling

Due to user privacy concerns and the difficulty of data collection, it is difficult to collect a large amount of data for model training in some user behavior domains. synthetic data generation provides a promising way.

Park et al. (2023) instantiate generative agents to populate an interactive sandbox environment inspired by The Sims, where end users can guide the generation of behaviors of agents using natural language. Zherdeva et al. (2021) use the generated synthetic data to train the Mask R-CNN framework, which is used for digital human interaction with the 3D environment. Liu et al. (2022) present UPC-SDG, a User Privacy Controllable Synthetic Data, which generates synthetic interaction data for users based on their privacy preferences to improve the performance of recommendations. Chen et al. (2021) leverage a small set of uniform synthetic data to optimize the debiasing parameters by solving the bi-level optimization problem in recommendations. Provalov et al. (2021) propose a novel method for evaluating and comparing recommender systems using synthetic user and item data and parametric synthetic user-item response functions.

However, current work focuses on a specific domain of user behavior and lacks work on generating user behavior in all scenarios and around the clock.

A.2 Ablation Study

In this study, we conducted a comprehensive evaluation of our proposed method through a series of ablation experiments, which were designed to assess the impact of various components on the quality of the generated behavioral data. The results of these experiments are summarized in Table 4 and include several key performance indicators that are critical for evaluating the efficacy of our approach.

Method	KS_P↑	BLEU↑	BD↓	JSD↓	Pass@1
no_profile	0.231	0.444	0.068	0.053	100%
no_role	0.213	0.449	0.066	0.053	97%
our	0.327	**0.512**	0.050	**0.029**	100%
no_segment	**0.489**	0.492	**0.035**	0.041	22.5%
no_format	nan	nan	nan	nan	0%

Table 4: The table presents the results comparing various methods in data generation based on several evaluation metrics: KS_P, BLEU, BD, JSD, and Pass@1. The highest value for KS_P and $BLEU$ and the lowest value for BD and JSD are highlighted.

We use following metrics: **KS_P** measures the discrepancy between the distributions of generated and real data, with higher values indicating better alignment. **BLEU** assesses the n-gram overlap between generated and reference text, where a higher score signifies greater textual similarity. **BD** quantifies the similarity between two probability distributions, with lower values indicating greater similarity. **JSD** evaluates the similarity between distributions, ranging from 0 to 1, where lower scores denote closer alignment. Finally, **Pass@1** reflects the proportion of instances where the model successfully predicts user behavior.

Profile information: As shown in Table 4, profile information significantly improves model performance, with KS_P increasing from 0.231 to 0.327 and JSD decreasing from 0.053 to 0.029, indicating better distribution alignment and enhanced generation quality.

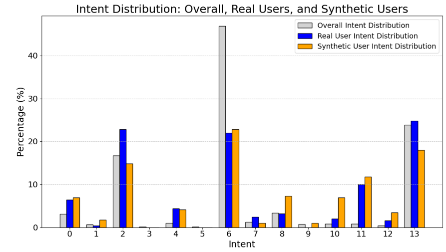
Role setting: The "no_role" method shows moderate performance in KS_P and BLEU, indicating that including role information positively impacts the diversity and coherence of the generated output. The relatively low BD and JSD values suggest that this method produces a more faithful representation of the target distribution. The high Pass@1 score indicates that users could successfully identify correct outputs in 97% of cases, which is commendable.

Format restrictions: The "no_format" method shows NaN values across all metrics, indicating that this setting was unable to produce outputs in the correct format, resulting in a complete loss of data usability. The 0% Pass@1 further emphasizes that the outputs were entirely unusable, underscoring the critical role of format restrictions in generating coherent and interpretable results. This implies that neglecting format considerations severely hampers the model's ability to produce valid outputs.

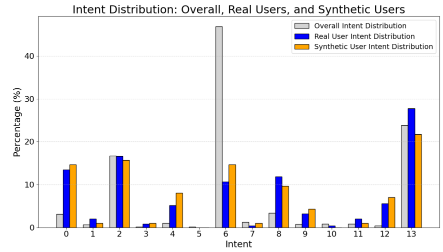
Segmented generation: The "no_segment" method achieves the highest KS_P score and a competitive BLEU score, suggesting that segmenting the data enhances diversity and textual coherence significantly. The low BD and JSD values indicate that this method produces outputs that are closely aligned with the intended data distribution, improving the quality of the generated content. However, the low Pass@1 score (22.5%) implies that while the outputs are diverse and coherent, they may not be entirely aligned with user expectations or specific intents, leading to a lower success rate in identifying correct outputs. Therefore, we adopt a segmented generation approach combined with Role setting and Format restrictions, ensuring the generated data maintains both diversity and fidelity while consistently producing effective and usable outputs.The prompt used in our method can be found in Appendix A.8.

A.3 Case Study: Intent Distribution Analysis

In this case study, we analyze the intent distribution at both the population level and individual user level to demonstrate the necessity and effectiveness of the fine-tuning phase in our model. Specifically, we examine how well the synthetic data captures individual users' intent distributions compared to the population-level distribution.



(a) User A



(b) User B

Figure 2: population and individual intent distribution.

We selected two users, User A and User B, for a comparative analysis of their intent distributions. Figure 2(a) and Figure 2(b) present the real intent distributions of these users alongside the intent distributions generated from synthetic data. And the population level intent distribution is shown in grey. For both User A and User B, the real intent distribution (shown in blue) demonstrates a pronounced deviation from the population-level distribution. In contrast, the synthetic data (shown in orange) reflects a strong alignment with the real intent distribution, validating the hypothesis that synthetic data can faithfully represent individual user behaviors.

The discrepancies between the population-level intent distribution and the individual user intent distributions emphasize the necessity of the fine-tuning phase. By utilizing synthetic data tailored to reflect individual users' intents, we can enhance the model's performance in personalized recommendation tasks. The findings from this analysis confirm that the fidelity of synthetic data is crucial, as it ensures that the model not only generalizes well across the population but also effectively adapts to the unique preferences of individual users.

A.4 Limitations

Ethical Considerations. The ethical implications of using real behavior data in this study are of utmost importance. While the data we used is anonymized and preprocessed by our providers using privacy-preserving techniques, including differential privacy, to prevent any risk of personal identification, it is still necessary to address potential concerns around privacy. The use of differential privacy ensures that individual-level data cannot be reconstructed from aggregated information, further strengthening data security. We have signed non-disclosure agreements (NDAs) with our data providers and work under their supervision to ensure responsible data handling and analysis.

Bias. Since our work uses real user data to prompt LLMs in generating synthetic behavior data, there are two potential sources of bias. The first source is the empirical data provided, which may not equally represent all user groups, potentially leading to biases in how certain behaviors or demographics are modeled. The second source of bias stems from the LLMs themselves, which may exhibit biases based on the composition of their pretraining corpus, reflecting imbalances or stereotypes present in the data they were trained on. To address these concerns, we plan to implement several mitigation strategies. This includes applying fairness-aware techniques during both data preprocessing and model prompting to ensure diverse and equitable representation across user groups.

Future Directions. There are several areas where our work can be further enhanced. First, developing more data-efficient generation methods is crucial, as behavior prediction scenarios typically involve large volumes of training data. Reducing the dependency on massive datasets without compromising model performance would significantly improve scalability and practicality. Second, improving the underlying LLMs to better understand and model human daily activities will be key to generating higher-quality synthetic data.

A.5 Details of Datasets.

- **Tencent Dataset.** The Tencent Dataset consists of anonymous user trajectory data collected from October to the end of December. The dataset includes a total of 667 users and 189,954 behavioral data entries. At the population level, we select 466 users for training, while at the individual level, we use the remaining 201 users. In this dataset, we utilize location categories to represent user activities and intents.
- **Smartphone Dataset.** The Smartphone Dataset is sampled from the usage log of the mobile phones. When a user uses mobile phones, various types of logs are generated, desensitized and reported (with user consent). We selected 114 types of events that are commonly monitored in most mobile applications and classified them into 18 intents, which cover the aspects of news, study, work, entertainment, sports, etc. We sampled two datasets between June 1st and August 22nd, 2023 (the first) and August 22nd and September 10th, 2023 (the second) which in total contain 4,500 and 5,000 anonymous users.

A.6 Details of Metrics

we employ four widely used metrics:precision (Pre), recall (Rec), and NDCG(N). The calculation of each metric is as follows. The formula for Pre :

$$Pre == \frac{1}{|C|} \sum_{c \in C} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c} \tag{3}$$

The formula for Rec :

$$Rec = \frac{1}{|C|} \sum_{c \in C} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \tag{4}$$

Where $|C|$ represents the total number of classes, True Positives (TP_c) denotes the number of samples correctly classified as class c, False Positives (FP_c) represents the number of samples incorrectly classified as class c, and False Negatives (FN_c) stands for the number of samples incorrectly classified as other classes instead of class c. And Precision and Recall respectively refer to the precision and recall of class c.

The formula for $N@k$:

$$N@k = \frac{\sum_{i=1}^{K} \frac{2^{rel_i}-1}{\log_2(i+1)}}{\sum_{j=1}^{|REL_K|} \frac{rel_j - 1}{\log_2(j+1)}} \tag{5}$$

where rel_i means the graded relevance of the result at position i, and $|REL_K|$ means the list of predictions in the result ranking list up to position K.

A.7 Details of Baselines

Here we introduce the details of each baseline.

- **SeqGAN (Yu et al., 2017).** SeqGAN is a sequence generative adversarial network that models sequence data generation as a reinforcement learning task, utilizing a GAN structure to capture the sequential dependencies in data generation.
- **DiffuSeq (Gong et al., 2022).** DiffuSeq is a diffusion-based sequence generation model

that adapts the diffusion process for text and sequence data generation, offering state-of-the-art performance on various generative tasks by leveraging noise-perturbed transitions during generation.

- **UPC_SDG (Liu et al., 2022).** UPC_SDG is a user trajectory synthetic data generation model, which focuses on preserving the statistical characteristics of the original data. It generates plausible user trajectories by maintaining important spatiotemporal relationships and is particularly effective for data privacy scenarios.

A.8 Used Prompts

The prompts we used are shown in 3

```
messages = [
    {
        "role": "system",
        "content": """
You are an assistant generating
    behavioral data based on given
    user behavior and profile data. I
    will provide you with a subset of
    real behavioral data in the format
     [weekday, timestamp, loc, intent
    ].

Your task:
1. Generate behavioral data for one
    month (minimum 90 lines) in the
    exact format: "weekday,timestamp,
    loc,intent".
2. Make sure to mimic realistic
    patterns of the given person, such
     as daily routines, work hours,
    and leisure activities, while
    ensuring diversity in location (
    loc) and intent. Don't have
    repetitive generation.
3. Ensure the weekdays values are
    within the range of 0-6, and
    timestamp values are within the
    range of 0-95.
4. Ensure that generated data has
    more than 100 lines and is in the
    correct format.
    },
    {
        "role": "user",
        "content": f"Profile:\n{json.
            dumps(user_profile)}\
            nBehavior data:\n{
            behavior_part.to_string(
            index=False)}"
    }
]
```

Figure 3: Prompt for generating behavioral data.

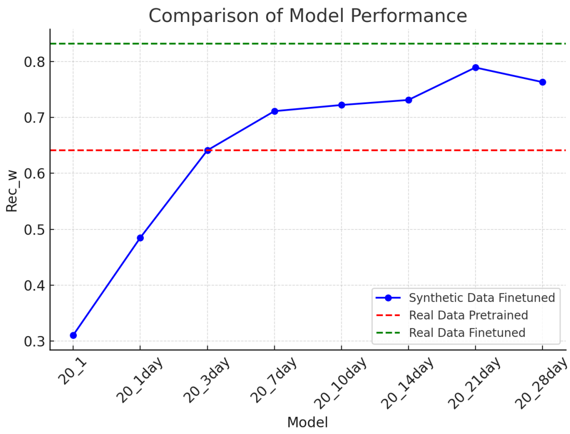


Figure 4: segment study

A.9 Costs of synthetic data

We used the gpt-4o-2024-0806 model to generate synthetic data using the CloseAI(https://www.closeai-asia.com/) API, priced at 1.5 times the official OpenAI pricing. For the Tencent dataset, synthetic data of 200 users were generated, totaling 61,308 data logs, and 80 RMB was spent. For the Smartphone dataset, synthetic data of 1000 users were generated, totaling 565664 data logs, and 638 RMB was spent.

A.10 Study of segment

We did experiments on segments on a small scale before generating synthetic data for all users of the dataset. We randomly select a batch of users (20), and give LLM users' 1 piece, 1 day, 3 days, 7 days, 10 days......of real data and then fine-tuned on the pre-trained model with the generated synthetic data to see how the metrics change, as shown in Figure4.It can be seen that when 7 days of data are provided to LLM, the effect of synthetic data is close to convergence.The line charts of the other metrics except Rec also show this trend. Although more data is provided, weekly segment is considered as the best choice for cost and benefit considerations.

A.11 Study of privacy analysis

To prove that the synthetic data generated by our framework does not leak individual privacy, we perform experiments from three aspects.

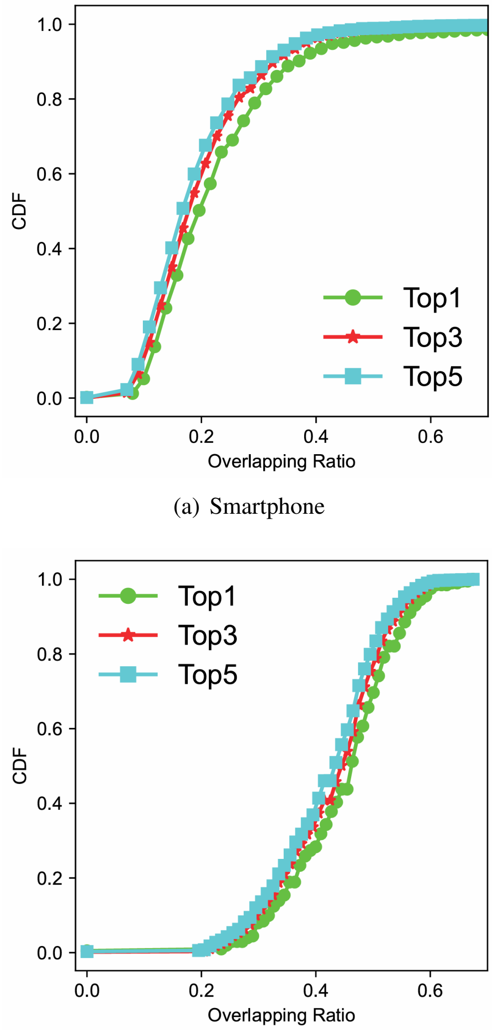
- **Uniqueness testing (DeMontjoye, 2013).** This measure evaluates whether the generated data is completely identical to the original data. It highlights the extent to which the model directly generates copies instead of brand-new data.

To prove that the realistic generated mobility trajectory is not a simple copy of the real trajectory but a brand-new trajectory, we perform a uniqueness testing of it by comparing it with the real data. We randomly select generated trajectories and compare them with all the real trajectories from the training set. The two trajectories are aligned in the time dimension one by one and determine whether the locations at the corresponding time points are exactly the same. The overlapping ratio is defined as the ratio of the number of identical locations to the total trajectory length. Next, we choose The real trajectory that is most similar to the generated one is defined as the one with the highest overlapping ratio. We calculate the overlapping ratio distribution of all the generated trajectories with the most similar real trajectories mentioned before. The results can also be extended by considering more similar trajectories, e.g., the top-3 and top-5 most similar real trajectories.

As shown in Supplementary Figure5, for the Smartphone datasets, more than 80% of the generated mobility trajectories cannot find any real trajectories that have more than a 30% overlapping ratio with them. For the Tencent dataset, more than 80% of the generated mobility trajectories overlap with real trajectories with an overlapping ratio of less than 50%. These results demonstrate that, while capturing mobility patterns, our framework indeed learns to generate brand-new and unique trajectories rather than simply copying.

- **Membership inference attack (Shokri et al., 2017).** If the generated data does not reveal the identities of users from the original data, it should not be possible to use the generated data to reidentify users in the training set.For this purpose,we use the framework of membership inference attack (Shokri et al., 2017). Stronger privacy protection leads to a lower attack successrate.

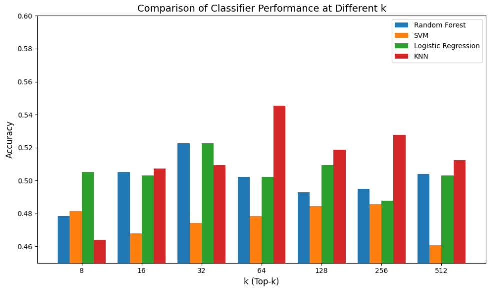
  Given a deep learning model and an individual record, the goal of the attack is to determine whether this record was included in the training set or not. We follow the attack settings as described in (Shokri et al., 2017), where the attacker's access to the deep learning model allows them to obtain the model's output. To improve the attack performance, we estimate individual information leakage using powerful machine learning models trained to predict whether an individual is in the training set. To control the impact of classification methods, we include four commonly used classification algorithms: Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF). The positive samples are those individuals in the training data, while the negative samples are not. The input feature is the overlapping ratio of multiple runs. The evaluation metric is the success rate, defined as the percentage of successful trials in determining whether a sample is in the training set. Stronger privacy protection leads to a lower success rate. As shown in Supplementary Figure6, on the Smartphone datasets the attack success rate is less than 0.55, and the Tencent



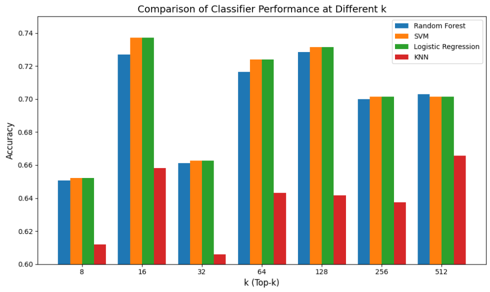
(a) Smartphone

(b) Tencent

Figure 5: Privacy evaluation in terms of uniqueness testing.

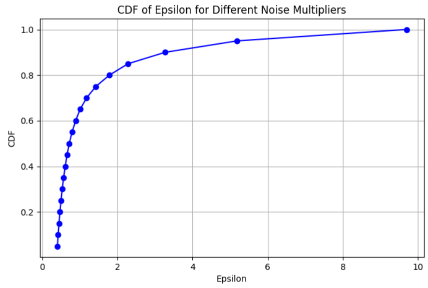


(a) Smartphone

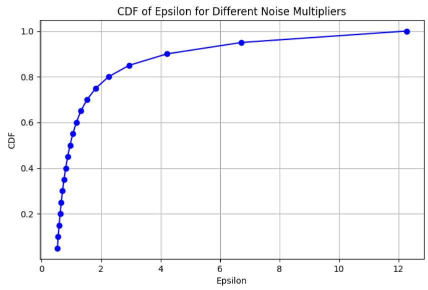


(b) Tencent

Figure 6: Privacy evaluation in terms of Membership inference attack.



(a) Smartphone



(b) Tencent

Figure 7: Privacy evaluation in terms of Differential privacy.

dataset is less than 0.74. This result indicates that attackers can hardly infer whether individuals are in the training set based solely on the information of the generated urban mobility data. Thus, our framework demonstrates robustness against membership inference attacks.

Differential privacy (Abadi et al., 2016). A model is dierentially private if for any pair of training datasets and that differ in the record of a single user, it holds that:$M(z;D) \leq e^{\epsilon}M(z;D') + \delta$ which means one can hardly distinguish whether any individual is included in the original dataset or not by looking at the output. It is a rigorous mathematical definition of privacy

For the output z, $M(z, D)$ denotes the probability distribution of z with the data D as the input. Smaller values of ϵ and δ provide stronger privacy guarantees. In our experiment, we examine the privacy budget of our proposed model from the perspective of changes in the overlapping ratio. Specifically, the overlapping ratio of each individual, under the conditions that this individual is included in the training set or not, is modeled by two Gaussian distributions, which are then regarded as $M(z, D)$ and $M(z, D')$ to calculate the privacy budget ϵ. For each user, we compute ϵ using TensorFlow Privacy (Abadi et al., 2016). The cumulative distribution of ϵ is illustrated in Supplementary Figure7. We observe that, without any additional privacy-preserving mechanism, when CDF is less than 0.9, our model achieves a maximum privacy budget of $\epsilon < 4$, which is typically considered a reasonable operating point for generative models. For example, Apple adopts a privacy budget of $\epsilon = 4.0$. The privacy budget can be further improved by incorporating DP-SGD or DP-GAN.