SynEHRgy: Synthesizing Mixed-Type Structured Electronic Health Records using Decoder-Only Transformers

Hojjat Karami
EPFL
hojjat.karami@epfl.ch

David Atienza
EPFL
david.atienza@epfl.ch

Anisoara Ionescu
EPFL
anisoara.ionescu@epfl.ch

Abstract

Generating synthetic Electronic Health Records (EHRs) offers significant potential for data augmentation, privacy-preserving data sharing, and improving machine learning model training. We propose a novel tokenization strategy tailored for structured EHR data, which encompasses diverse data types such as covariates, ICD codes, and irregularly sampled time series. Using a GPT-like decoder-only transformer model, we demonstrate the generation of high-quality synthetic EHRs. Our approach is evaluated using the MIMIC-III dataset, and we benchmark the fidelity, utility, and privacy of the generated data against state-of-the-art models.

1 Introduction

Electronic Health Records (EHRs) are comprehensive digital repositories of patient health information, encompassing a wide range of data types. In in-patient settings, EHRs include structured data such as demographics and International Classification of Diseases (ICD) codes, time series data (e.g., vital signs and lab results), and unstructured data (e.g., clinical notes and radiology images). These records provide an invaluable resource for developing machine learning models, offering rich datasets that can be harnessed to enhance patient care, predict outcomes, and support clinical decision-making.

Access to real-world EHRs is often limited due to privacy regulations such as HIPAA and GDPR, as well as technical challenges and a lack of incentives for data sharing. Even pseudo-anonymized data remains susceptible to re-identification by malicious actors. Techniques like federated learning [1] and differential privacy [2] are being explored to mitigate these risks. Synthetic data provides a promising alternative by replicating the statistical properties of real data without being tied to actual individuals, thereby circumventing privacy regulations. It can be securely shared among stakeholders, facilitating hypothesis generation, AI model pre-training, and data exploration without exposing sensitive information. This approach not only alleviates privacy concerns but also reduces the effort involved in accessing real data, allowing researchers to focus on validating more probable hypotheses before turning to original datasets for confirmation.

Generating synthetic EHR data presents several challenges. EHRs comprise multiple data types with distinct characteristics. For instance, clinical events like ICD codes are high-dimensional and sequential, while time series data are relatively low-dimensional but feature irregular sampling and significant missingness, which is often informative and not random [3, 4]. Additionally, patients may have multiple visits or admissions, with data across visits being correlated. While many studies focus on generating single data types, such as discrete ICD codes ([5, 6, 7, 8]) or time series data ([9, 10, 11]), few addresses the generation of multiple data types across multiple visits simultaneously ([12]).

In the era of Large Language Models (LLMs), many have been adapted to the clinical domain for tasks such as diagnosis prediction [13], text embeddings [14], and medical reasoning [15, 16]. While these models can interpret structured data and numerical values presented in text format, they are not

GenAI for Health Workshop @ NeurIPS 2024, Vancouver.

directly applicable for generating synthetic structured EHR data. This limitation arises because LLMs are typically not trained on large structured EHR datasets, particularly those containing complex numerical data.

In this work, we present **SynEHRgy**, a framework designed to **Syn**thesize mixed-type structured **EHR**s, including covariates, ICD codes, and irregularly-sampled time series across multiple patient visits. We introduce a novel tokenization strategy for structured EHR data, particularly for numerical variables, allowing decoder-only transformer models to capture underlying data patterns within a causal language modeling framework and generate high-quality synthetic EHRs. Using a small GPT model, we demonstrate the efficacy of our approach on the MIMIC-III dataset [17], comparing the generated data's quality against state-of-the-art models in terms of fidelity, utility, and privacy. The key contributions of this work are as follows:

- We propose a novel tokenization strategy for mixed-type structured EHR data (covariates, ICD codes, irregularly-sampled time series) across multiple visits.
- We empirically demonstrate that a GPT-like decoder-only transformer model can generate high-quality structured EHR data.
- We conduct a comprehensive evaluation of the proposed methodology on the MIMIC-III dataset, comparing the generated data's quality to state-of-the-art models in terms of fidelity, utility, and privacy.
- Our framework shows exceptional performance in generating irregularly-sampled time series data, a challenging task due to high missingness and irregular time points.

2 Related Works

Generating clinical events, particularly ICD codes, has garnered significant research interest in recent years. In contrast, generating clinical time series presents a greater challenge due to irregularly-sampled time points and substantial missingness.

ICD Code Generation A substantial body of work focuses on generating ICD codes using various frameworks. Generative Adversarial Networks (GANs) have been employed for this purpose ([18, 6, 19, 20]), as have Variational Autoencoders (VAEs) ([21]) and diffusion models ([22, 23, 24, 25]). Additionally, Large Language Models (LLMs) have been explored due to the sequential nature of ICD codes and the feasibility of straightforward tokenization. For instance, PromptEHR [8] employs an encoder-decoder transformer model with prompt learning to generate ICD codes conditioned on numerical and categorical demographic features. HALO [12] utilizes a transformer encoder coupled with a linear autoregressive module for generating ICD codes at both the visit and code levels. CEHR-GPT [26] leverages GPT to produce sequences of visits based on demographic prompts and visit time intervals.

Time Series Generation Research on time series generation has predominantly focused on regularly-sampled time series, which are more common across various domains. Techniques such as Generative Adversarial Networks (GANs) have been widely applied ([27, 28, 29, 30, 31]), as well as diffusion models ([32, 33, 34]). However, generating irregularly-sampled time series is less explored. Notable efforts in this area include RTSGAN [9], EHR-Safe [31], and TimEHR [11] using GANs, and TS-Diffusion [35] and TimeDiff [10] using diffusion models. Additionally, research on leveraging language models for time series generation remains limited. While pre-trained LLMs have demonstrated effectiveness with numerical values in text format—e.g., LIFT [36] and GReaT [37] excel in tabular data generation—they primarily treat numerical values as text. Some approaches propose defining specific numerical tokens and training LLMs from scratch. For example, CodeAR [38] utilizes a Vector Quantized Autoencoder (VQ-VAE) for codebook creation, followed by an LSTM with next-token prediction loss. Chronos [39] discretizes numerical values into bins for time series forecasting using encoder-decoder transformers, a strategy also employed by HALO [12] for clinical time series and by [40] for financial time series. To the best of our knowledge, no existing work, other than HALO [12], explores the generation of irregularly-sampled time series using LLMs.

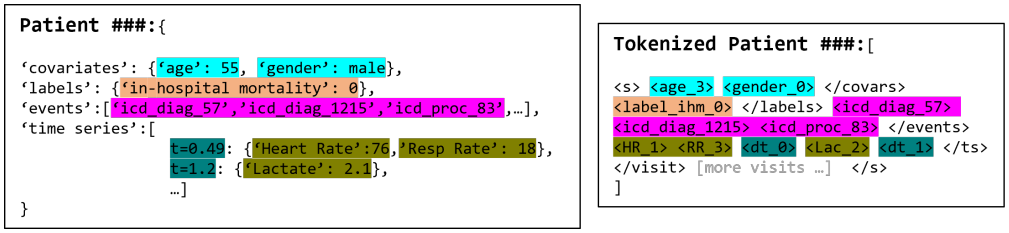

Figure 1: Left: Example of patient's data. Right: Tokenized patient's data

3 Methodology

3.1 Problem Formulation

An EHR dataset can be characterized as $\mathcal{D} = \{P_i\}_{i=1}^{N}$, where N denotes the number of patients and P_i represents the data for the i-th patient. Each patient may have one or more visits, denoted as $P_i = \{V_{ij}\}_{j=1}^{|V_i|}$, with each visit represented as $V_j = \{c_j, y_j, E_j, T_j\}$. Here, c_j denotes the set of covariates (e.g., demographics), y_j represents the set of clinical outcomes (e.g., in-hospital mortality), $E_j = \{e_m\}_{m=1}^{|E_j|}$ is a sequence of clinical events such as ICD codes, and T_j is the time series data, including vital signs and laboratory variables. The time series data for each visit is represented as $T_j = \{t_k, \{(n_{kp}, x_{kp})\}_{p=1}^{|L_k|}\}_{k=1}^{|T_j|}$, where $|T_j|$ denotes the total number of measurement times, and $|L_k|$ indicates the number of available data points per measurement. In this representation, $t_k \in \mathbb{R} \geq 0$ is the timestamp of the k-th measurement, while n_{kp} and x_{kp} are the variable name and value pairs. An example of patient data is illustrated on the left side of Figure 1.

3.2 Tokenization

For numerical values, we employ uniform quantization to discretize numerical variables into equal-sized bins, similar to Chronos [39]. Each bin is assigned a unique token. For instance, the token <HR_0> represents heart rate values in the range of 55-66 beats per minute. For timestamps, we compute the time intervals between consecutive measurements and discretize these intervals into bins. For non-numerical values such as ICD codes and categorical variables, each code or category is assigned a unique token.

We also use special tokens to delineate different sections of the data: <s> and </s> for the start and end of a sequence, respectively; </covars>, </labels>, </ts>, and </visit> to signify the end of covariates, labels, time series, and admissions. The padding token (<PAD>) is employed to ensure that all tokenized sequences are of uniform length. If a patient has multiple admissions, the data for new admissions is appended after the token </visit>. An example of tokenized patient data is illustrated on the right side of Figure 1.

This tokenization strategy offers several advantages: First, it is scalable with respect to the number of variables. For instance, when integrating a new dataset with additional variables, we simply update the tokenization dictionary by incorporating new tokens. Second, numerical tokenization is more efficient than treating numerical values as text. For example, the GPT-2 tokenizer would generate seven tokens for 'Lactate:2.5', whereas our method generates only one token.

3.3 Model Architecture

Decoder-only Transformer models, such as the ones used in language modeling (e.g., GPT [41]), typically employ a causal language modeling (CLM) objective as their loss function. This objective is also known as autoregressive language modeling. The core idea is to predict the next token in a sequence given all previous tokens. The model generates a probability distribution over the vocabulary for the next token, and the loss function measures the discrepancy between the predicted distribution and the actual next token. For an tokenized patient sequence $x = [x_1, x_2, \ldots, x_T]$, the model predicts the probability distribution over the vocabulary for each token in the sequence, conditioned on all previous tokens:

$$p(x_t \mid x_1, x_2, \ldots, x_{t-1})$$

The loss for a single token prediction is typically computed using cross-entropy between the predicted distribution and the one-hot encoded true next token. For the entire sequence, the loss function is:

$$\mathcal{L} = -\sum_{t=1}^{T} \log p(x_t \mid x_1, x_2, \ldots, x_{t-1})$$

3.4 EHR Data Generation

The generation of EHR data follows a process similar to text generation. We initiate the generation with the start token (<s>) and continue generating tokens sequentially until the end token (</s>) is produced. For the de-tokenization of numerical tokens, we apply uniform sampling within the specified bin range to recover the original numerical values. This approach ensures that the generated values are representative of the underlying distribution of the data. The de-tokenization of ICD codes and categorical variables is straightforward, as each token directly maps to a specific code or category.

This method offers several advantages: First, it leverages the well-established text generation framework, which allows for seamless integration with existing models and techniques. Second, uniform sampling for numerical values maintains the integrity of the data distribution, providing a realistic representation of the original values. Lastly, the straightforward de-tokenization of categorical variables ensures simplicity and efficiency in generating accurate categorical data.

4 Experiments

4.1 Dataset

We utilize the MIMIC-III dataset for our experiments [17]. Specifically, we use the preprocessing pipeline provided by [42] to extract data from approximately 42,000 patients with multiple hospital visits. The dataset is divided into training, validation, and test sets using a 70-15-15 split. For each patient, we include age and gender as covariates. Additionally, for each visit, we select 25 phenotype labels along with an in-hospital mortality label. We include ICD diagnosis and procedure codes with a frequency greater than five across the entire dataset, resulting in 4,656 unique ICD codes. We also select 41 time series from vital signs and laboratory variables, including five categorical time series. After tokenization, we have a total of 5,127 unique tokens, with the training split consisting of 29.6 million tokens. Each patient's sequence starts with the <s> token and ends with the </s> token, comprising demographics, ICD codes, and time series tokens, as explained in Section 3.2.

4.2 Evaluation Metrics

We evaluate the synthetic data based on **Fidelity**, **Utility**, and **Privacy**. Since all of our baselines, except HALO, do not generate multiple data types, we assess our metrics separately for each data type. Additionally, we use the test split as a reference for high-quality synthetic data to evaluate these metrics.

Fidelity For ICD codes, similar to [12], we assess fidelity using unigram, bigram, and trigram probabilities within each visit, as well as sequential bigram probabilities between consecutive visits. For instance, the bigram probability of the sequence $[icd_{1053}, icd_{610}]$ is calculated as its frequency divided by the total number of patients. We report Pearson's correlation between the top-1000 n-gram probabilities of the train and test/synthetic data pairs.

For time series data, we create manual embeddings for each patient by computing common statistics (min, max, mean, std) from the first 48 hours of the patient's stay. We then report the precision, recall, density, and coverage (PRDC) metrics [43] to evaluate the fidelity between the embeddings of the train split and the test/synthetic split. Additionally, we calculate the pairwise temporal correlation between all time series variables by concatenating the time series data from all patients and report the mean squared error between the correlation matrices of the train split and the test/synthetic split (MSE_{corr}). We also assess correlation accuracy [30] by discretizing the correlation coefficients into five levels: High negative ($[-1, -0.5)$), Medium negative ($[-0.5, -0.2)$), Low ($[-0.2, 0.2)$), Medium positive ($[0.2, 0.5)$), and High positive ($[0.5, 1)$). The confusion matrix of these correlation levels is then reported. To assess the fidelity of the missingness patterns, we visualize the co-occurrence of

measurements for each pair of time series variables and normalize this by the sum of occurrences for each variable.

Utility We assess the utility of the synthetic data by evaluating its performance in two tasks: *in-hospital mortality* prediction (binary classification) and *phenotypes* classification (multi-label classification). Specifically, we use time series embeddings from the first 48 hours of patient admission, as described earlier, and train a LightGBM model [44]. To examine the impact of data augmentation with synthetic data, we incorporate the entire synthetic dataset into the training data at various ratios (0, 0.1, 0.2, 0.5, 1.0) and evaluate the model on the test split.

Privacy We adopt the Membership Inference Attack (MIA) approach as described in EHR-Safe [31], where the goal is to determine whether specific data points were included in the training dataset. To accomplish this, we fit a K-Nearest Neighbors (KNN) model on the synthetic data and calculate the nearest distances for each patient in both the training and test splits. A significant disparity between the distance distributions in the synthetic-train and synthetic-test sets indicates lower privacy. For ICD code sequences, we use the Hamming distance, whereas for time series embeddings, we employ the Euclidean distance. We then fit Gaussian distributions to these distances and assess the differences between the two distributions using the Wasserstein Distance (WD), Jensen-Shannon Divergence (JSD), and Area Under the Receiver Operating Characteristic (AUROC) metrics.

4.3 Baselines

For ICD code generation, we compare our method with HALO [12], a hierarchical autoregressive language model, and PromptEHR [8], which utilizes an encoder-decoder transformer model. For time series generation, we compare our method with RTSGAN [9] and TimEHR [11], both of which are GAN-based models designed for generating irregularly-sampled time series data. Additionally, we include HALO in the comparison for time series generation, as it employs a similar tokenization strategy for numerical variables.

Ablation study. We further demonstrate the effectiveness of numerical tokenization by removing the discretization step and treating the numerical values as raw text. We refer to this model as SynEHRgy0.

4.4 Training Details

We employ a small GPT-2 model from the Transformers library[1]. Our model consists of 4 layers, 4 attention heads, and 384 embedding dimensions. After testing various context lengths, we found that 1024 tokens offer a good balance between the quality of generated data and training time (notably, context lengths of 256, 512, 1024, and 2048 correspond to 20%, 34%, 51%, and 68% of the original data, respectively). For the ablated model (SynEHRgy0), we need to use a larger context length of size 4098 due to its inefficient tokenization.

We use the Adam optimizer with a learning rate of 3×10^{-4} and train the model for 20 epochs on a system equipped with an *NVIDIA Tesla V100-SXM2-32GB* GPU. The batch size is set to 128, and we employ gradient accumulation with a step of two. All other hyperparameters are set to their default values as per the Transformers library. For generation, we use *top-k sampling* with a temperature of 0.7 and a top-k value of 50. We generate 30,000 synthetic patients which is the same size of training split. Due to computational limitation, we have performed minimal hyperparameter tuning for our model and for the baselines to ensure a fair comparison.

5 Results

5.1 Fidelity

Table 1 shows the correlation values between n-gram probabilities of the training and test/synthetic splits. While HALO achieves the highest performance for unigram probabilities, our method consistently surpasses all baselines, particularly in bigram, trigram, and sequential bigram probabilities. This underscores the ability of our method to generate high-quality synthetic ICD codes.

[1]https://huggingface.co/docs/transformers

Table 1: Fidelity metrics for ICD codes

	single-visit			Sequential bigram
	unigram	bigram	trigram	
Test split	*0.998*	*0.983*	*0.932*	***0.954***
SynEHRgy	0.885	**0.969**	**0.916**	0.856
HALO	**0.978**	0.329	0.102	*0.839*
PromptEHR	0.747	0.246	0.114	NA

Table 2: Fidelity metrics for Time Series. **Bold** and underline values indicate the best and second best results, respectively.

	Precision	Recall	Density	Coverage	MSE_corr
Test split	*0.863 (0.004)*	*0.878 (0.003)*	*0.984 (0.005)*	*0.966 (0.001)*	*0.02*
SynEHRgy	**0.797 (0.005)**	**0.841 (0.004)**	0.673 (0.01)	**0.873 (0.001)**	**0.036**
SynEHRgy0	0.452(.018)	0.227(0.020)	0.345(0.009)	0.150(0.014)	0.133
TimEHR	0.593 (0.008)	0.669 (0.011)	0.385 (0.011)	0.613 (0.003)	0.059
HALO	0.521 (0.021)	0.469 (0.002)	0.387 (0.022)	0.19 (0.003)	0.072
RTSGAN	0.725 (0.026)	0.113 (0.019)	**0.744 (0.054)**	0.108 (0.001)	0.097

Table 2 presents the precision, recall, density, and coverage (PRDC) metrics, as well as the temporal correlation difference (MSE_{corr}) for evaluating the fidelity of time series embeddings. It is important to consider these PRDC metrics collectively. For instance, while RTSGAN achieves the highest density, it performs poorly on recall and coverage metrics. Our method excels in precision, recall, and coverage, and ranks second in density. Additionally, our model demonstrates the lowest MSE_{corr}, indicating strong performance in preserving structural correlations within the data. This is further supported by the confusion matrix of correlation levels shown in Figure 2, where SynEHRgy displays superior performance in capturing correlation levels. Moreover, the co-occurrence analysis in Figure 3 reveals that our method exhibits the highest fidelity in preserving missingness patterns, highlighting its capability to generate realistic irregularly-sampled time series data. The ablated model (SynEHRgy0) exhibits poor fidelity performance, demonstrating the importance of numerical tokenization.

5.2 Utility

Table 3 and Table 4 present the AUROC scores for the 25-phenotype and in-hospital mortality prediction tasks, respectively, across different data augmentation settings. When training exclusively on synthetic data and testing on real data (TSTR), our method surpasses all baseline models and achieves performance closest to that of the Train on Real and Test on Real setting (TRTR). Additionally, our method demonstrates the highest AUROC when synthetic data is incorporated into the training set, although the performance gain diminishes as the proportion of real training data increases. This suggests that while the models leverage the synthetic data effectively, they do not extend beyond the real data distribution to generate new patient profiles that significantly enhance test performance.

Table 3: AUROC for phenotypes prediction. *NA*: Not Applicable.

	Train ratio				
	0 (TSTR)	0.1	0.2	0.5	1
No synthetic	*NA*	*0.884*	*0.892*	*0.9*	*0.906 (TRTR)*
Val split	*0.776*	*0.783*	*0.785*	*0.793*	*0.8*
SynEHRgy	**0.775**	**0.78**	**0.783**	**0.791**	**0.798**
TimEHR	0.731	0.756	0.763	0.771	0.779
HALO	0.665	0.746	0.761	0.78	0.787
RTSGAN	0.545	0.751	0.767	0.781	0.794

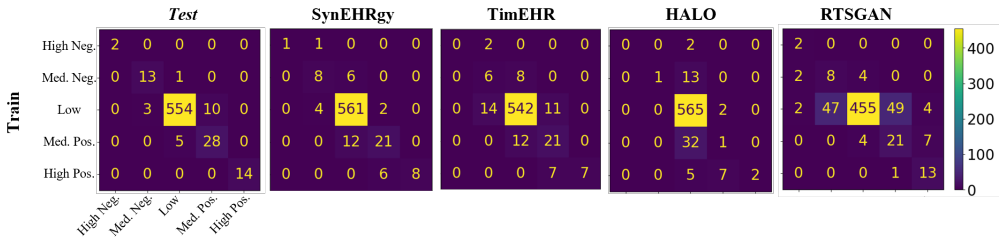

Figure 2: Confusion matrices of correlation levels for time series data

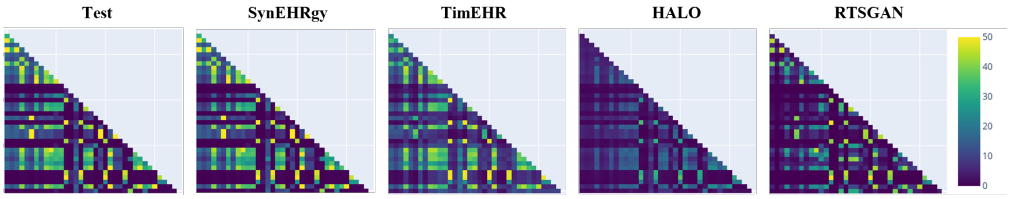

Figure 3: Co-occurrence of measurement for time series data

Furthermore, the validation split shows marginally better performance compared to other synthetic data settings. These utility results highlight the potential of our method for data anonymization (with no exposure of training data) and for data augmentation in scenarios with limited training data.

5.3 Privacy

Membership Inference Attack (MIA) metrics are reported in Table 5. As can be seen, none of methods shows any potential privacy risk. It should be noted that these privacy metrics are computed based on hamming distance of ICD codes and Euclidean distance of time series data, and the results may vary for different distance metrics.

6 Conclusion

In this work, we introduced SynEHRgy, a novel methodology for generating synthetic structured EHR data that incorporates mixed-type data (ICD codes and time series) across multiple visits. A key innovation of our approach is the tokenization strategy for numerical variables, which enhances compatibility with decoder-only transformer models. Despite its advantages, the proposed method has limitations. Specifically, treating each data point (e.g., an ICD code or a heart rate measurement) as a single token may become inefficient for very large sequences due to context length constraints. Future work will explore more efficient tokenization strategies to mitigate this issue. Additionally, integrating high-frequency continuous signals, such as ECG data, is currently not feasible with our tokenization approach. We plan to address this limitation by incorporating patch-based tokenization strategies [45, 39, 46]. Furthermore, we aim to expand our methodology to include other data modalities, such as clinical notes and radiology images, to generate comprehensive multimodal EHR

Table 4: AUROC for in-hospital mortality prediction. *NA*: Not Applicable.

	Train ratio				
	0 (TSTR)	0.1	0.2	0.5	1
No synthetic	*NA*	*0.889*	*0.898*	*0.912*	*0.92 (TRTR)*
Val split	*0.902*	*0.91*	*0.914*	*0.915*	*0.924*
SynEHRgy	**0.894**	**0.902**	**0.902**	**0.905**	**0.915**
TimEHR	0.865	0.887	0.893	0.906	0.912
RTSGAN	0.58	0.878	0.891	0.906	0.919

Table 5: Privacy metrics for ICD codes and Time Series

ICD codes				Time Series			
Model	JSD	WD	AUROC	Model	WD	JSD	AUROC
SynEHRgy	0.012	0.000	0.510	SynEHRgy	0.003	0.001	0.494
HALO	0.013	0.000	0.511	TimEHR	0.003	0.001	0.492
PromptEHR	0.012	0.001	0.455	HALO	0.002	0.001	0.493
				RTSGAN	0.002	0.001	0.494

data. Another promising direction for future research is to enhance the capabilities of clinical LLMs, such as Meditron [16] and Med-PaLM [47], to better understand and generate structured data.

References

[1] Nicola Rieke et al. "The Future of Digital Health with Federated Learning". In: *npj Digital Medicine* 3.1 (Sept. 2020), p. 119. ISSN: 2398-6352. DOI: 10.1038/s41746-020-00323-1. arXiv: 2003.08119 [cs].

[2] Nazish Khalid et al. "Privacy-Preserving Artificial Intelligence in Healthcare: Techniques and Applications". In: *Computers in Biology and Medicine* 158 (May 2023), p. 106848. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2023.106848.

[3] Marzyeh Ghassemi et al. "A Review of Challenges and Opportunities in Machine Learning for Health". In: *AMIA Summits on Translational Science Proceedings* 2020 (May 2020), pp. 191–200. ISSN: 2153-4063.

[4] Amelia L. M. Tan et al. "Informative Missingness: What Can We Learn from Patterns in Missing Laboratory Data in the Electronic Health Record?" In: *Journal of Biomedical Informatics* 139 (Mar. 2023), p. 104306. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104306.

[5] Edward Choi et al. "Generating Multi-label Discrete Patient Records Using Generative Adversarial Networks". In: *arXiv: Learning* (Mar. 2017).

[6] Amirsina Torfi and Edward A. Fox. *CorGAN: Correlation-Capturing Convolutional Generative Adversarial Networks for Generating Synthetic Healthcare Records*. Mar. 2020. DOI: 10.48550/arXiv.2001.09346. arXiv: 2001.09346 [cs, stat].

[7] Siddharth Biswal et al. *EVA: Generating Longitudinal Electronic Health Records Using Conditional Variational Autoencoders*. Dec. 2020. DOI: 10.48550/arXiv.2012.10020. arXiv: 2012.10020 [cs].

[8] Zifeng Wang and Jimeng Sun. *PromptEHR: Conditional Electronic Healthcare Records Generation with Prompt Learning*. Oct. 2022. DOI: 10.48550/arXiv.2211.01761. arXiv: 2211.01761 [cs].

[9] Hengzhi Pei et al. "Towards Generating Real-World Time Series Data". In: (Dec. 2021). DOI: 10.1109/ICDM51629.2021.00058.

[10] Muhang Tian et al. *Fast and Reliable Generation of EHR Time Series via Diffusion Models*. Oct. 2023. DOI: 10.48550/arXiv.2310.15290. arXiv: 2310.15290 [cs].

[11] Hojjat Karami et al. *TimEHR: Image-based Time Series Generation for Electronic Health Records*. Feb. 2024. DOI: 10.48550/arXiv.2402.06318. arXiv: 2402.06318 [cs].

[12] Brandon Theodorou, Cao Xiao, and Jimeng Sun. "Synthesize High-Dimensional Longitudinal Electronic Health Records via Hierarchical Autoregressive Language Model". In: *Nature Communications* 14.1 (Aug. 2023), p. 5305. ISSN: 2041-1723. DOI: 10.1038/s41467-023-41093-0.

[13] Yikuan Li et al. *BEHRT: Transformer for Electronic Health Records*. July 2019. DOI: 10.48550/arXiv.1907.09538. arXiv: 1907.09538 [cs, stat].

[14] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. *ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission*. Nov. 2020. DOI: 10.48550/arXiv.1904.05342. arXiv: 1904.05342 [cs].

[15] Karan Singhal et al. *Towards Expert-Level Medical Question Answering with Large Language Models*. May 2023. DOI: 10.48550/arXiv.2305.09617. arXiv: 2305.09617 [cs].

[16] Zeming Chen et al. *MEDITRON-70B: Scaling Medical Pretraining for Large Language Models*. Nov. 2023. DOI: 10.48550/arXiv.2311.16079. arXiv: 2311.16079 [cs].

[17] Alistair E. W. Johnson et al. "MIMIC-III, a Freely Accessible Critical Care Database". In: *Scientific Data* 3.1 (May 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35.

[18] Mrinal Kanti Baowaly et al. "Synthesizing Electronic Health Records Using Improved Generative Adversarial Networks". In: *Journal of the American Medical Informatics Association* 26.3 (Mar. 2019), pp. 228–241. ISSN: 1067-5027, 1527-974X. DOI: 10.1093/jamia/ocy142.

[19] Ziqi Zhang et al. "SynTEG: A Framework for Temporal Structured Electronic Health Data Simulation". In: *Journal of the American Medical Informatics Association* 28.3 (Mar. 2021), pp. 596–604. ISSN: 1527-974X. DOI: 10.1093/jamia/ocaa262.

[20] Chang Lu et al. "Multi-Label Clinical Time-Series Generation via Conditional GAN". In: *IEEE Transactions on Knowledge and Data Engineering* (2023), pp. 1–13. ISSN: 1558-2191. DOI: 10.1109/TKDE.2023.3310909.

[21] Giannis Nikolentzos et al. "Synthetic Electronic Health Records Generated with Variational Graph Autoencoders". In: *npj Digital Medicine* 6.1 (Apr. 2023), pp. 1–12. ISSN: 2398-6352. DOI: 10.1038/s41746-023-00822-x.

[22] Huan He et al. *MedDiff: Generating Electronic Health Records Using Accelerated Denoising Diffusion Model*. Feb. 2023. DOI: 10.48550/arXiv.2302.04355. arXiv: 2302.04355 [cs].

[23] Taha Ceritli et al. *Synthesizing Mixed-type Electronic Health Records Using Diffusion Models*. Aug. 2023. DOI: 10.48550/arXiv.2302.14679. arXiv: 2302.14679 [cs].

[24] Hongyi Yuan, Songchi Zhou, and Sheng Yu. *EHRDiff: Exploring Realistic EHR Synthesis with Diffusion Models*. Mar. 2024. DOI: 10.48550/arXiv.2303.05656. arXiv: 2303.05656 [cs].

[25] Yuan Zhong et al. *Synthesizing Multimodal Electronic Health Records via Predictive Diffusion Models*. June 2024. DOI: 10.48550/arXiv.2406.13942. arXiv: 2406.13942 [cs].

[26] Chao Pang et al. *CEHR-GPT: Generating Electronic Health Records with Chronological Patient Timelines*. May 2024. DOI: 10.48550/arXiv.2402.04400. arXiv: 2402.04400 [cs].

[27] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. "Real-Valued (Medical) Time Series Generation with Recurrent Conditional GANs". In: *arXiv: Machine Learning* (June 2017).

[28] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. "Time-Series Generative Adversarial Networks". In: (Sept. 2019).

[29] Zinan Lin et al. "Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions". In: *Proceedings of the ACM Internet Measurement Conference*. Virtual Event USA: ACM, Oct. 2020, pp. 464–483. ISBN: 978-1-4503-8138-3. DOI: 10.1145/3419394.3423643.

[30] Jin Li et al. "Generating Synthetic Mixed-Type Longitudinal Electronic Health Records for Artificial Intelligent Applications". In: *npj Digital Medicine* 6.1 (May 2023), pp. 1–18. ISSN: 2398-6352. DOI: 10.1038/s41746-023-00834-7.

[31] Jinsung Yoon et al. "EHR-Safe: Generating High-Fidelity and Privacy-Preserving Synthetic Electronic Health Records". In: *npj Digital Medicine* 6.1 (Aug. 2023), pp. 1–11. ISSN: 2398-6352. DOI: 10.1038/s41746-023-00888-7.

[32] Xinyu Yuan and Yan Qiao. "Diffusion-TS: Interpretable Diffusion for General Time Series Generation". In: *The Twelfth International Conference on Learning Representations*. Oct. 2023.

[33] Jian Qian et al. *TimeLDM: Latent Diffusion Model for Unconditional Time Series Generation*. July 2024. arXiv: 2407.04211 [cs].

[34] Namjoon Suh et al. *TimeAutoDiff: Combining Autoencoder and Diffusion Model for Time Series Tabular Data Synthesizing*. https://arxiv.org/abs/2406.16028v2.

[35] Yangming Li. *TS-Diffusion: Generating Highly Complex Time Series with Diffusion Models*. Nov. 2023. DOI: 10.48550/arXiv.2311.03303. arXiv: 2311.03303 [cs].

[36] Tuan Dinh et al. *LIFT: Language-Interfaced Fine-Tuning for Non-Language Machine Learning Tasks*. Oct. 2022. DOI: 10.48550/arXiv.2206.06565. arXiv: 2206.06565 [cs].

[37] Vadim Borisov et al. *Language Models Are Realistic Tabular Data Generators*. Apr. 2023. DOI: 10.48550/arXiv.2210.06280. arXiv: 2210.06280 [cs].

[38] Yoonhyung Lee, Younhyung Chae, and Kyomin Jung. "Leveraging VQ-VAE Tokenization for Autoregressive Modeling of Medical Time Series". In: *Artificial Intelligence in Medicine* 154 (Aug. 2024), p. 102925. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2024.102925.

[39] Abdul Fatir Ansari et al. *Chronos: Learning the Language of Time Series*. Mar. 2024. arXiv: 2403.07815 [cs].

[40] Alexandru Grigoraș and Florin Leon. "Synthetic Time Series Generation for Decision Intelligence Using Large Language Models". In: *Mathematics* 12.16 (Aug. 2024), p. 2494. ISSN: 2227-7390. DOI: 10.3390/math12162494.

[41] Alec Radford et al. "Language Models Are Unsupervised Multitask Learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[42] Hrayr Harutyunyan et al. "Multitask Learning and Benchmarking with Clinical Time Series Data". In: *Scientific Data* 6.1 (June 2019), p. 96. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0103-9. arXiv: 1703.07771 [cs, stat].

[43] Muhammad Ferjad Naeem et al. *Reliable Fidelity and Diversity Metrics for Generative Models*. June 2020. DOI: 10.48550/arXiv.2002.09797. arXiv: 2002.09797 [cs, stat].

[44] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[45] Gerald Woo et al. *Unified Training of Universal Time Series Forecasting Transformers*. Feb. 2024. DOI: 10.48550/arXiv.2402.02592. arXiv: 2402.02592 [cs].

[46] Yong Liu et al. *Timer: Generative Pre-trained Transformers Are Large Time Series Models*. June 2024. DOI: 10.48550/arXiv.2402.02368. arXiv: 2402.02368 [cs, stat].

[47] Karan Singhal et al. *Large Language Models Encode Clinical Knowledge*. Dec. 2022. DOI: 10.48550/arXiv.2212.13138. arXiv: 2212.13138 [cs].

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our work is mostly empirical rather than theoretical. We have deeply evaluated the quality of synthetic data in terms of utility, fidelity and privacy and compared it with the state-of-the-art models.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: In the conclusion section, we discussed the limitations of our work such as the tokenization strategy might not be efficient for very large sequences due to limited context length. Our work is tested on a real-world dataset and the results may vary for different datasets. We are open to discussing more limitations in the paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our work is mostly empirical and does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The dataset is publicly available for researchers. The model architecture and training details are provided in the main paper. Our work is highly replicable as we use the Transformer pipeline which is widely used in the community.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. Releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The dataset (MIMIC-III) is publicly available. We will release the code and preprocessing pipeline upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments is reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have used a famous preprocessing pipeline for MIMIC-III dataset which splits the data into train, validation and test sets. The labels for downstream tasks are also selected using the same pipeline. We have provided the non-default hyperparameters used for training the model using Huggingface's Transformers library.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: As our training split is large enough (30k), we haven't performed k-fold evaluation because of computational constraints. Yet, there are some ways to report error bars such as running the training procedure multiple times with different random seeds and reporting the mean and standard deviation of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: This information is provided in the training details section of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our dataset is publicly available and we have used a widely used model architecture for our experiments. We have tried our best to preserve the anonymity.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The development of SynEHRgy, while offering significant advances in the generation of synthetic EHR data, comes with broader ethical and societal implications that must be carefully considered. In terms of safety, while the synthetic data generated by our method is not linked to real individuals, there remains the potential risk of misuse in creating misleading datasets that could negatively impact clinical decision-making or research outcomes. Security risks are mitigated by our focus on synthetic data, which reduces the likelihood of exposing real patient information; however, the generated data could still be vulnerable to manipulation or malicious use in healthcare applications if not properly safeguarded. Regarding discrimination, care must be taken to ensure that biases in the original datasets, such as underrepresentation of certain demographic groups, do not carry over into synthetic data and perpetuate disparities in healthcare outcomes. Furthermore, while SynEHRgy does not involve surveillance directly, we recognize that synthetic data might be used to develop surveillance systems or predictive tools that could target vulnerable populations. We strongly advocate against any use of our method that could lead to deceptive practices, including fraud or impersonation, and emphasize the need for transparency in how the data is applied. From an environmental perspective, training large-scale models like transformers requires substantial computational resources, and future work should explore ways to reduce the environmental footprint of these models. Finally, to address fairness and bias, we recommend rigorous evaluation of the synthetic data to identify and correct any imbalances or biases inherited from the original dataset, ensuring equitable outcomes in downstream applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.