TrialSynth: Generation of Synthetic Sequential Clinical Trial Data

Chufan Gao
University of Illinois Urbana Champaign
chufan2@illinois.edu

Mandis Beigi
Medidata Solutions
mandis.beigi@3ds.com

Afrah Shafquat
Medidata Solutions
afrah.shafquat@3ds.com

Jacob Aptekar
Medidata Solutions
jacob.aptekar@3ds.com

Jimeng Sun
University of Illinois Urbana Champaign
Carle Illinois College of Medicine
jimeng@illinois.edu

Abstract

Analyzing data from past clinical trials is part of the ongoing effort to optimize the design, implementation, and execution of new clinical trials and more efficiently bring life-saving interventions to market. While there have been recent advances in the generation of static context synthetic clinical trial data, due to both limited patient availability and constraints imposed by patient privacy needs, the generation of fine-grained synthetic time-sequential clinical trial data has been challenging. Given that patient trajectories over an entire clinical trial are of high importance for optimizing trial design and efforts to prevent harmful adverse events, there is a significant need for the generation of high-fidelity time-sequence clinical trial data. Here we introduce TrialSynth, a Variational Autoencoder (VAE) designed to address the specific challenges of generating synthetic time-sequence clinical trial data. Distinct from related clinical data VAE methods, the core of our method leverages Hawkes Processes (HP), which are particularly well-suited for modeling event-type and time gap prediction needed to capture the structure of sequential clinical trial data. Our experiments demonstrate that TrialSynth surpasses the performance of other comparable methods that can generate sequential clinical trial data at varying levels of fidelity / privacy tradeoff, enabling the generation of highly accurate event sequences across multiple real-world sequential event datasets with small patient source populations. Notably, our empirical findings highlight that TrialSynth not only outperforms existing clinical sequence-generating methods but also produces data with superior utility while empirically preserving patient privacy.

1 Introduction

The data generated from past clinical trials represent a valuable resource for informing drug development [17, 8, 42, 9] and increasing the speed at which vital life-saving drugs arrive to market [15, 13] While the potential value of clinical trial data is high, these data are

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

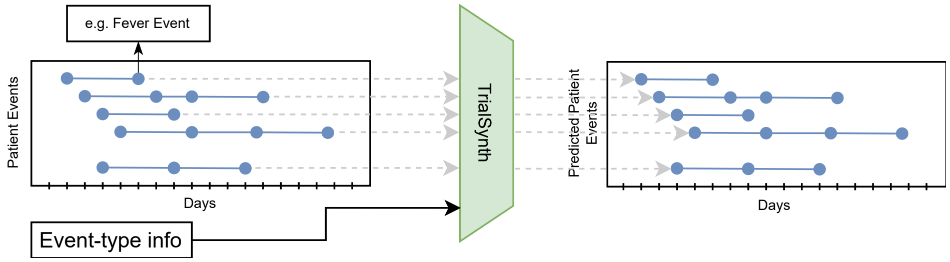

Figure 1: Visualization of data input and synthetic data generation of `TrialSynth`, the model input is the real patient events and their timestamps, and we wish to generate synthetic patient events and their timestamps. This is a particularly challenging task due to the small amount of patient data. `TrialSynth` also explicitly supports adding the event type information in the form of specifying the specific event types to generate.

often inaccessible due to patient privacy concerns and legal constraints [35, 27, 22]. The generation of high-quality synthetic clinical trial data that captures the properties of real data while simultaneously protecting patient privacy is increasingly being seen as a strategy for sharing and applying these data in drug development applications [23].

Though proposed methods for generating synthetic clinical trial data have focused on static context information for each subject (e.g., demographics) [21], many of the highest value applications, including control arm augmentation [37] require generating synthetic time-sequential event data that has high fidelity [4, 46]. However, developing a high-quality model for sequential trial data can be more complicated than data-rich tasks in computer vision or natural language processing, due to the small sample size of training datasets available, which is less common in other applications of generative models.

To address these challenges, we propose `TrialSynth`. This method makes use of Hawkes processes, which are statistical models that are specialized for event-type and time gap prediction [20, 51], as well as Variational Autoencoders (VAEs) [24], a proven generative framework that has worked well for static clinical trial data synthesis [12]. We empirically demonstrate that combining these two classical approaches leads to an algorithm that is capable of generating sequential event synthetic data even on small amounts of clinical trial data.

To summarize our contributions:

1. We introduce `TrialSynth`–a model that combines Variational Autoencoder + Hawkes Process that is both able to generate sequential event clinical trial data and supports a high level of control, allowing users to specify specific event types and variance levels to generate (`https://github.com/chufangao/TrialSynth`).
2. We demonstrate from the analysis of 7 real-world clinical trial datasets that `TrialSynth` outperforms alternative approaches designed for tabular data generation.
3. We also demonstrate `TrialSynth` achieves high performance versus privacy trade-off with two key metrics: ML Inference Score, which shows that synthetic event sequences are hard to distinguish from the original sequences, and Distance to Closest Record (DCR), which shows that synthetic sequences are not copies of the original data.

The rest of this paper is organized as follows: In Section 2, we review the related work. In Section 3, we dive into the proposed `TrialSynth` in detail. In section 4, we compare datasets and baselines, demonstrating the superiority of `TrialSynth`. Finally, in Section 5, we provide a discussion and conclude our findings.

2 Related Work

Synthetic Data Generation as a research area has been quickly garnering attention from the research community, with examples such as CTGAN [45], CTabGan [50], TabDDPM

[25], TWIN-GPT [38], the Synthetic Data Vault[1] [33], and more. However, most of these models, such as TabDDPM and CTGAN, are focused on explicitly generating tabular data with no time component; or, in the case of SDV's ParSynthesizer [48], it is relatively simple and may be approximated with a GRU or LSTM model.

Trial Patient Generation is a research area that has become popular. In Electronic Healthcare Record (EHR) generation [10, 7, 43, 12, 42, 28, 36, 38], the model usually only focuses on generating the *order* at which certain clinical events happen (i.e., the diagnosis code of next patient visit), as opposed to generating the specific times of the visits as well. For example, [12, 38] generates a digital twin of an input patient event sequence via a VAE and a cross-modality model, but cannot handle event timestamp generation. `TrialSynth` extends this line of previous work to include the specific timestamps on which these events occur, as well as the order. [36] created a strong patient EHR generation baseline, but relies on a high amount of training data (929,268 and 46,520 patients in outpatient and inpatient datasets respectively). However, in a single clinical trial, all of our datasets contain less than 1000 patients, which makes HALO difficult to run. `TrialSynth` is designed for and performs well on small clinical trial datasets, particularly if the event types are known.

Hawkes Processes combined with VAEs is an area of research that is particularly appealing for our scenario. We employ the Transformer Hawkes Process [51] for our data generation modeling. To the best of our knowledge, `TrialSynth` is the first to extend Hawkes models to full patient event generation from a single embedding. Unlike the Hawkes Process, it relaxes the assumption that past events can never lower the probability of future events, and performs much better on real world data.

This inherent capability of modeling events and their time occurrences makes Hawkes Processes highly suitable for event prediction. Previous work explores variational Hawkes processes in the context of event prediction for (disease progression [6] and social events sequences [32], but they rely on the context of previous ground truth observations as well as the hidden state. Another work [26] explores using variational approaches to disentangle multivariate Hawkes Process for event type prediction, but it also relies on knowing the ground truth to predict the next timestep. This limitation is a major roadblock in a full synthetic data generation setting. Because of this, there is leaking of information from ground truth event occurrences. This information leakage is not permitted in our task, which is a fully generative setting from the embedding space.

3 TrialSynth

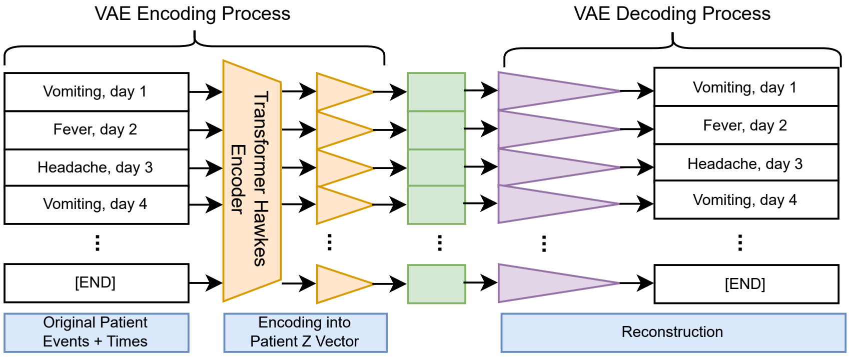

Figure 2: Diagram of the `TrialSynth` Encoder-Decoder structure. Here, the model input is the real patient event sequence + time, which trains a VAE model to the same output time + event sequence. The event sequence length for each event is also predicted. The transformer encoder processes each input timestep, then output embeddings are individually transformed to the z-latent space via a neural network. Sampling and decoding occur from each timestep-specific z-latent representation.

[1] `https://docs.sdv.dev/sdv/`

`TrialSynth` is created to solve the highly specific task of synthetic sequential clinical trial patient generation. As shown in Fig. 1 and Fig. 3, a patient contains many sequences of event types and their timestamps. This essentially creates a high-vocabulary, sequential token (event types) generation problem with a regression component (event times). First, we formulate the components that compose `TrialSynth`. Then, we explain key details of `TrialSynth`, including the ability to input type information in the form of known types to generate (which is common in the trial generation space when up-sampling patient data). Finally, we conclude with experiments on ML utility (usefulness of synthetic data) and inference privacy (important for patient privacy) and a discussion of the results.

3.1 Encoding and Decoding Hawkes Processes

Neural Hawkes Process [30] was proposed to generalize the traditional Hawkes Process. Let us describe the $\lambda(t)$, the intensity function of any event occurring at time t.

$$\lambda(t) := \sum_{k=1}^{K} \lambda_k(t) := \sum_{k=1}^{K} f_k(\boldsymbol{W}_k^\top \boldsymbol{h}(t)) = \sum_{k=1}^{K} \beta_k \log\left(1 + e^{\frac{\boldsymbol{W}_k^T \boldsymbol{h}(t)}{\beta_k}}\right),$$

$\lambda_k(t)$ is the intensity function for the event $k \in \mathcal{K}$ occurring, $K = |\mathcal{K}|$ is the total number of event types, $h(t)$ are the hidden states of the event sequence obtained by a Transformer encoder, and $\boldsymbol{W}_k^\top$ are learned weights that calculate the significance of each event type at time t. $f_k(c) = \beta_k \log(1 + e^{\frac{x}{\beta_k}})$ is the softplus function with parameter β_k. The output of $f_k(x)$ is always positive. Note that the positive intensity does not mean that the influence is always positive, as the influence of previous events is calculated through $\boldsymbol{W}_k^\top \boldsymbol{h}(t)$. If there is an event occurring at time t, then the probability of event k is $P(k_t = k) = \frac{\lambda_k(t)}{\lambda(t)}$. Furthermore, the log-likelihood is: $\ln P_\theta(\{(t_1, k_1), \ldots, (t_L, k_L)\}|\boldsymbol{z}) = \sum_{j=1}^{L} \log(\lambda_\theta(t_j|\mathcal{H}_{t_j,z})) - \int_{t_1}^{t_L} \lambda_\theta(t|\mathcal{H}_{t,z})dt$.

Encoder: The encoder model $E_{\texttt{TrialSynth}}(\mathcal{H}_i) \to \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}$ takes in the original event types and times, and predicts the mean and standard deviation to sample hidden state vector at time $\boldsymbol{z_t}$ at each timestep t. These $\boldsymbol{z_t}$ are concatenated to form $\boldsymbol{z} \sim Normal(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$. $\boldsymbol{z}$ is trained to be close to the $Normal(\mathbf{0}, \mathbf{1})$ via ELBO.

Decoder: We train the decoder to maximize the likelihood of the input Hawkes Process. I.e. the input is the ground truth event type and time-step sequence, and the autoencoder reconstructs it from $\boldsymbol{z}$. For our purposes, we adapt a decoding scheme similar to HALO [36].

At *training time*, the input to the decoder

$$D_{\texttt{TrialSynth}}(\boldsymbol{z}, (t_1, k_1), \ldots (t_i, k_i)) \to (\hat{t}_{i+1}, \hat{k}_{i+1}, \lambda)$$

is a hidden vector $\boldsymbol{z}$ and a sequence of *ground truth* event types and event times. It is tasked with predicting the next type of event $\hat{k}$, the next event time $\hat{t}$, and the intensity function λ that measures the probability of an event occurring. λ is necessary to compute the likelihood $P_\theta((t_1, k_1), \ldots, (t_i, k_i), |\boldsymbol{z})$. [2] Furthermore, we follow Transformer Hawkes Process's approach of also adding mean squared error losses to the time: $time\_loss = \|t - \hat{t}\|^2$ and cross-entropy loss of the predicted $type\_loss = -\sum_{c=1}^{|\mathcal{K}|} k \log(p_k)$

At *inference time*, the input to the decoder is only z, and we decode the predicted event types and times. To predict next time and event tuple $(\hat{t}_i, \hat{k}_i)$, the input is the previously predicted times and events $\{(\hat{t}_1, \hat{k}_1), \ldots, (\hat{t}_{i-1}, \hat{k}_{i-1}))\})$. (each predicted time and event is repeatedly appended to the input).

Finally, we note that we can control for the generation of events that are similar to the original patient by first encoding the original patient and then sampling around it, a benefit of the probabilistic nature of the VAE latent space z. Otherwise, it would be impossible to correspond the original labels to the synthetic data. For all experiments in this work, we take a random sample of the latent vector $\boldsymbol{z}$ to reconstruct our patient. Otherwise, our task would collapse down to a straightforward autoencoder task.

[2] Please see Appendix for details.

3.2 Final Loss Terms

Finally, we write the final loss as

$$L = L_{hawkes} + L_{elbo} + L_{length}$$

The L_{hawkes} is the log-likelihood of the sequence given the Hawkes process above. The L_{elbo} is the VAE loss of the hidden vector KL divergence from a standard Gaussian, the mean-squared error reconstruction loss of the event times, and the cross-entropy loss of the event types. Finally, we additionally add L_{length} to ensure the model learns proper sequence lengths (described in section 3.4).

Numerical Values Note that we do not discretize the time in terms of the time gap. Rather, we pad out each event sequence to the number of the most occurrences, which is usually around 100-200. Each event is considered to be categorical, and numerical events such as wbc (white blood cell count in Figure 3) is discretized based on their unique values in real-world data.

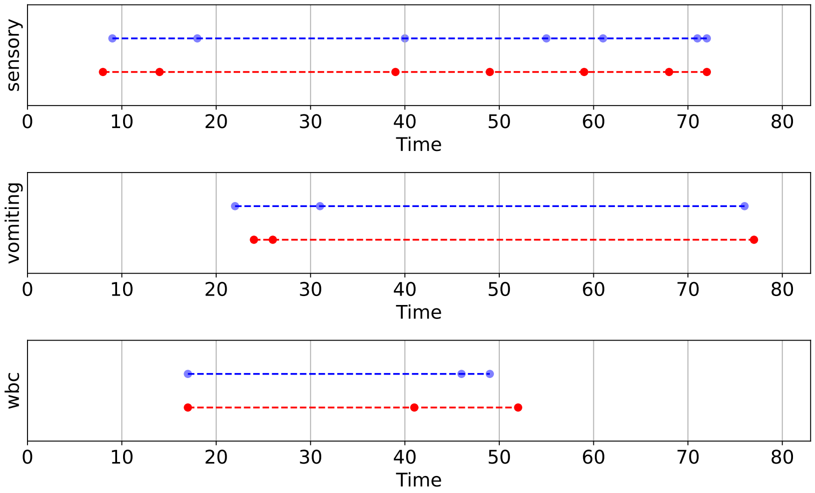

Figure 3: Example of a generated sequence from `TrialSynth` from NCT00003299 plotted by the individual events. Red dots and lines denote ground truth event occurrence and time between events respectively. In this case, the time is in Days. The blue dots and lines are the predicted events. Numerical events such as wbc (white blood cell count) are discretized based on their unique values in the real data. This will be corrected in the new version. Each prediction is linked with dashed lines for clarity.

3.3 Event Type Information

We also propose 2 variants of `TrialSynth`. In some applications, such as clinical trial patient modeling [40, 17, 8, 9, 5, 12], we may be interested in an event sequence *with known event types*, that is, the model only needs to generate the timestamps at which events occur. This is to address the concern of subject fidelity, that is, the generated subject must be significantly similar to the original subject in order for the generated data to be useful; therefore, knowing which events occur in a subject to generate a similar subject would not be unreasonable. Along with the "Events Unknown" model that has no assumptions, we also propose the "Events Known" model was created to enforce ONLY simulating specific events, without considering all events (which may be too numerous and irrelevant to the current patient).

To accommodate `TrialSynth` (Events Known), we use the exact same model as `TrialSynth` (Events Unknown), but restrict the event type prediction module to only valid patient input event types at *inference time*. We retain the same training process for both models, since we do not want to restrict learning event type information at training.

Table 1: A description of all the real-world datasets used in the evaluation. All trial data was obtained from Project Data Sphere [19]. *Num Rows* refers to the raw number of data points in the trial. *Num Subj* refers to the total number of patients. *Num Events* denotes the total number of *unique* events. *Events / Subj* denotes the average number of events that a patient experiences. *Positive Label Proportion* denotes the percentage of patients that did not experience the death event.

Dataset	Description	# Rows	# Subjects	# Events	Events / Subject	Positive Label Proportion
NCT00003299 (LC1)	Small Cell Lung Cancer	20210	548	34	36.880	0.951
NCT00041119 (BC1)	Breast Cancer	2983	425	150	7.019	0.134
NCT00079274 (CC)	Colon Cancer	316	70	18	4.514	0.184
NCT00174655 (BC2)	Breast Cancer	7002	953	21	7.347	0.019
NCT00312208 (BC3)	Breast Cancer	2193	378	182	5.802	0.184
NCT00694382 (VTE)	Venous Thromboembolism in Cancer Patients	7853	803	746	9.780	0.456
NCT03041311 (LC2)	Small Cell Lung Cancer	1043	47	207	22.192	0.622

3.4 Sequence Length Prediction

We generate event sequences $\{(t_j, k_j); j = 1, \ldots, L; k_j \in \mathcal{K}'\}$, where length L is also generated by `TrialSynth`. Taking inspiration from HALO [36], our generation process automatically appends an `[END]` event at the end of each of the patient events. Furthermore, in addition to the event loss from before, we add a cross-entropy loss term on specifically the [END] event.

4 Experiments

4.1 Datasets

We evaluated our models on 7 real-world clinical trial outcome datasets obtained from Project Data Sphere[3] [19, 16, 5, 8, 9]. Specifically, we chose the trials as outlined in Table 1. These datasets have shown to be effective evaluation datasets for tabular prediction [41, 39] and digital twin generation [12, 38]. Specifically, we use LC1 [31], BC1 [3], CC [2], BC2 [14], BC3 [44], VTE [1], LC2 [11, 18, 47]. A full description of the data is shown in Table 1. Each dataset contains events and the times at which they occur, e.g., medications and procedures, as well as some adverse events like vomiting etc. We use these datasets to predict if the subject experiences the death event, which is an external label. Note that `TrialSynth` does not require a fixed patient event sequence length.

4.2 Baseline Methods

One surprising challenge we found was that *existing EHR methods and synthetic patient generation methods are not applicable to our specific task and dataset due to dataset size and lack of support for timestamp generation*; therefore, we primarily compare against general sequential data generation methods.

We compared the following 7 models: First, the **LSTM VAE** is the same as our proposed model, except with an LSTM instead of a Transformer encoder. **PARSynthesizer** from is SDV, based on a conditional probabilistic auto-regressive (CPAR) model, is specifically tailored for synthesizing sequential event data and stands out due to its unique focus and accessible codebase. **TabDDPM** is a state-of-the-art tabular synthesizer using diffusion models, enhanced by adding time as a numerical column for our purposes. Despite not being explicitly designed for sequential data, it surpasses previous models like **CTGAN** in synthetic tabular data generation. Lastly, **HALO**, a hierarchical autoregressive language model, excels in synthesizing Electronic Health Records (EHR) but struggles with clinical

[3] `https://data.projectdatasphere.org/projectdatasphere/html/access`

trial datasets due to the limited size of the training data, highlighting the challenges in this domain.

`TrialSynth` **(Events Unknown)** is the VAE + Multivariate Hawkes Process that is trained without any assumptions. At training time, the task is to predict a patient's events and timesteps given the latent vector. `TrialSynth` **(Events Known)** assumes that one knows which specific events occur for the Hawkes Model. This essentially just restricts the number of valid events in the prediction phase by patient's unique events.

4.3 Utility Evaluation

Table 2: Utility Evaluation: Binary Classification ROCAUCs (↑ higher the better, ± standard deviation) of a downstream LSTM trained on data generated from the `TrialSynth` models as well as the original data and baselines. Note that the LSTM and the `TrialSynth` models estimate their own sequence length. `TrialSynth` (Events Known) is put in a separate category due to its requirement of event type information, with results **underlined**. **Bolded** indicates original data ROC is within 1 standard deviation of synthetic data ROC

Dataset	Original Data	LSTM VAE	PAR	CTGAN	TabDDPM	HALO	`TrialSynth` (Events Unknown)	`TrialSynth` (Events Known)
LC1	0.689$_{\pm 0.105}$	0.563$_{\pm 0.053}$	0.504$_{\pm 0.066}$	0.508$_{\pm 0.122}$	0.557$_{\pm 0.055}$	0.457$_{\pm 0.079}$	**0.672**$_{\pm 0.061}$	**0.709**$_{\pm 0.049}$
BC1	0.678$_{\pm 0.078}$	0.617$_{\pm 0.036}$	0.573$_{\pm 0.043}$	0.550$_{\pm 0.046}$	**0.630**$_{\pm 0.045}$	0.461$_{\pm 0.184}$	**0.651**$_{\pm 0.046}$	**0.665**$_{\pm 0.045}$
CC	0.657$_{\pm 0.140}$	0.481$_{\pm 0.092}$	0.567$_{\pm 0.096}$	0.448$_{\pm 0.023}$	**0.583**$_{\pm 0.098}$	0.446$_{\pm 0.02}$	**0.652**$_{\pm 0.015}$	**0.653**$_{\pm 0.019}$
BC2	0.660$_{\pm 0.128}$	0.535$_{\pm 0.073}$	0.523$_{\pm 0.074}$	0.523$_{\pm 0.11}$	0.513$_{\pm 0.078}$	0.503$_{\pm 0.075}$	**0.599**$_{\pm 0.042}$	**0.594**$_{\pm 0.068}$
BC3	0.632$_{\pm 0.072}$	0.454$_{\pm 0.039}$	0.463$_{\pm 0.039}$	0.493$_{\pm 0.013}$	0.503$_{\pm 0.043}$	0.535$_{\pm 0.183}$	**0.620**$_{\pm 0.038}$	**0.634**$_{\pm 0.032}$
VTE	0.640$_{\pm 0.038}$	0.490$_{\pm 0.019}$	0.549$_{\pm 0.022}$	0.508$_{\pm 0.113}$	0.531$_{\pm 0.021}$	0.485$_{\pm 0.066}$	**0.618**$_{\pm 0.024}$	**0.625**$_{\pm 0.020}$
LC2	0.738$_{\pm 0.149}$	0.563$_{\pm 0.097}$	0.507$_{\pm 0.087}$	0.573$_{\pm 0.118}$	0.574$_{\pm 0.096}$	0.534$_{\pm 0.078}$	**0.729**$_{\pm 0.044}$	**0.755**$_{\pm 0.059}$

Downstream Classification ROCAUC: It is vital that synthetic data perform similarly to real-world data; therefore, we evaluate the utility (ROCAUC) of the generated synthetic data by performing binary classification of death events in all 7 clinical trials. We choose ROCAUC since it has been used for similar tasks in the past[12]. Additionally, ROC AUC is sensitive to class imbalance in the sense that when there is a minority class, one typically defines this as the positive class and it will have a strong impact on the AUC value. This is desirable behavior and is what we look to evaluate in our application.

The standard deviation of each ROCAUC score is calculated via bootstrapping (100x bootstrapped test data points). Training is performed completely on synthetic data by matching each generated patient to its ground truth death event label. Testing is performed on the original held-out ground truth split. For the Original Data baseline, we performed 5 cross-validations on 80/20 train test splits of the real data. The main results are shown in Table 2.

We see that synthetic data generated by `TrialSynth` variants generally perform the best in terms of downstream death event classification performance, where `TrialSynth` (Events Unknown) outperforms the next best model (in 4/7 datasets and is within 1 standard deviation with the rest of the datasets). Furthermore, `TrialSynth` (Events Known) significantly outperforms other baselines, due to the additional input information. Still, `TrialSynth` (Events Unknown) also performs admirably, being on par but slightly less performant than `TrialSynth` (Events Known).

Occasionally, synthetic data is able to support better performance than the original dataset on downstream tasks (this behavior is also seen in TabDDPM). We believe that this is due to the synthetic model generating examples that are more easily separable and/or more diverse than real data. However, this is only a hypothesis and should be investigated further in future research, but we are encouraged to see that our proposed method captures this interesting synthetic data behavior.

4.4 Privacy evaluations

ML Inference Score: This can also be thought of as an adversarial Model Attack [36]. Another main concern is the privacy of the synthetic data, to prevent any data or information

Table 3: Results of ML Inference Score: LSTM binary classification of real vs synthetic (*the closer to 0.5 the score is, the better*). The standard deviation calculated via bootstrapping is shown via ±. AUCROC scores are shown. Bolded indicates the best result or within 1 standard deviation of the best result.

Dataset	LSTM VAE	PAR	CTGAN	TabDDPM	HALO	TrialSynth (Events Unknown)	TrialSynth (Events Known)
LC1	$1.000_{\pm 0.000}$	$0.968_{\pm 0.010}$	$0.952_{\pm 0.056}$	$0.762_{\pm 0.024}$	$1.000_{\pm 0.004}$	$\mathbf{0.613}_{\pm 0.024}$	$0.689_{\pm 0.020}$
BC1	$0.932_{\pm 0.017}$	$0.998_{\pm 0.002}$	$0.973_{\pm 0.082}$	$0.926_{\pm 0.017}$	$1.000_{\pm 0.001}$	$\mathbf{0.616}_{\pm 0.025}$	$0.768_{\pm 0.021}$
CC	$1.000_{\pm 0.000}$	$0.807_{\pm 0.082}$	$0.935_{\pm 0.056}$	$0.894_{\pm 0.050}$	$0.998_{\pm 0.005}$	$0.711_{\pm 0.051}$	$\mathbf{0.701}_{\pm 0.054}$
BC2	$1.000_{\pm 0.000}$	$0.999_{\pm 0.001}$	$0.998_{\pm 0.075}$	$0.998_{\pm 0.001}$	$0.999_{\pm 0.001}$	$\mathbf{0.605}_{\pm 0.048}$	$\mathbf{0.593}_{\pm 0.023}$
BC3	$0.994_{\pm 0.007}$	$0.874_{\pm 0.026}$	$0.895_{\pm 0.098}$	$0.729_{\pm 0.035}$	$0.992_{\pm 0.008}$	$\mathbf{0.689}_{\pm 0.023}$	$\mathbf{0.693}_{\pm 0.038}$
VTE	$1.000_{\pm 0.000}$	$0.923_{\pm 0.012}$	$0.879_{\pm 0.119}$	$0.992_{\pm 0.005}$	$0.000_{\pm 0.004}$	$\mathbf{0.871}_{\pm 0.014}$	$\mathbf{0.856}_{\pm 0.016}$
LC2	$1.000_{\pm 0.000}$	$0.651_{\pm 0.112}$	$0.982_{\pm 0.038}$	$0.374_{\pm 0.021}$	$0.000_{\pm 0.003}$	$\mathbf{0.573}_{\pm 0.111}$	$\mathbf{0.477}_{\pm 0.127}$

leakage. To address this, we calculate the performance of predicting whether a generated sequence is real vs synthetic via an LSTM binary classification [33] (similar to an adversarial model). The real subjects are labeled with "0" and the synthetic subjects are labelled with "1". Results are shown in Table 3, and we see that TrialSynth variants perform closest to the optimal 0.5 ROCAUC ideal score. One thing to note is that a perfect copy of the original data would result in a 0.5 score, so we have the following metric to measure the opposite scenario. Furthermore, we see a continued trend of both forms of TrialSynth generally outperforming other baseline methods, illustrating the importance of giving the model more information in this data-scarce setting.

Distance to Closest Record (DCR) Score: Second, we follow the evaluation metrics per TabDDPM [25]. That is, we compare the feature vectors of the real vs synthetic data and measure how far the synthetic data is from the original. The higher this distance is, the more different the generated data is from the original data, and thus the more private it is. A completely different version of the data would obtain the highest distance but could result in bad performance in the downstream LSTM classification performance or a high ML Inference score (close to 1). We calculate this by featurizing the event time predictions in terms of (count, means, and standard deviations). Then, we normalize and obtain the L2 distance between a generated subject and the closest real subject. Table 4 shows this result. Notice that TrialSynth variants generally obtain quite low scores on this metric. TabDDPM and PAR also generate data closer to the original data compared to LSTM VAE. We note the privacy-fidelity trade-off, as LSTM VAE generates data that is further away from the original, but yields worse utility (Table 2).

Table 4: Distance to Closest Record (DCR) Score. Note that this score only tells part of the picture. The higher this score is, the larger the difference between the synthetic data and the original data. The lower the score, the more similar the synthetic data is to the original data.

Dataset	LSTM VAE	PAR	TabDDPM	TrialSynth (Events Unknown)	TrialSynth (Events Known)
LC1	3.700	2.647	1.426	1.217	1.138
BC1	4.677	4.633	1.007	0.624	0.612
CC	2.732	1.977	1.346	1.519	1.675
BC2	32.185	56.915	3.581	1.452	1.215
BC3	87.015	2.348	1.207	0.515	0.745
VTE	17.946	35.362	1.059	0.983	0.971
LC2	36.740	37.723	4.662	5.015	4.922

Table 5: Dataset Inference attack: *(the closer to .5 the better)*. This is calculated as the percent where the closest record of a training sample is a real vs synthetic sample.

Dataset	LSTM VAE	PAR	CTGAN	HALO	TabDDPM	TrialSynth (Events Unknown)	TrialSynth (Events Known)
LC1	1.00	0.99	0.97	0.98	0.71	0.62	0.59
BC1	1.00	0.92	0.84	1.00	0.71	0.61	0.52
CC	0.97	0.87	0.81	1.00	0.42	0.62	0.38
BC2	1.00	0.98	0.97	0.99	0.99	0.73	0.62
BC3	0.99	0.77	0.60	1.00	0.35	0.40	0.44
VTE	1.00	0.89	0.65	1.00	0.86	0.87	0.37
LC2	1.00	1.00	0.91	1.00	0.27	0.62	0.25

Dataset Attack We evaluate a Dataset Attack scenario as per HALO [36], where we label the real records with the lowest distance (computed by featuring event times into mean, std, counts) to the closest record in the synthetic dataset as 1. It tests the ability of the synthetic dataset to prevent an attacker from inferring whether a real record was used in the training dataset. On real training data, we compare if the closest record is a real record from the training set or a synthetic record. Ideally, we also want this accuracy to be 0.5. From Table 5, we see that `TrialSynth` generally performs the best, even beating out HALO and TabDDPM.

4.5 Utility / Privacy Trade-off

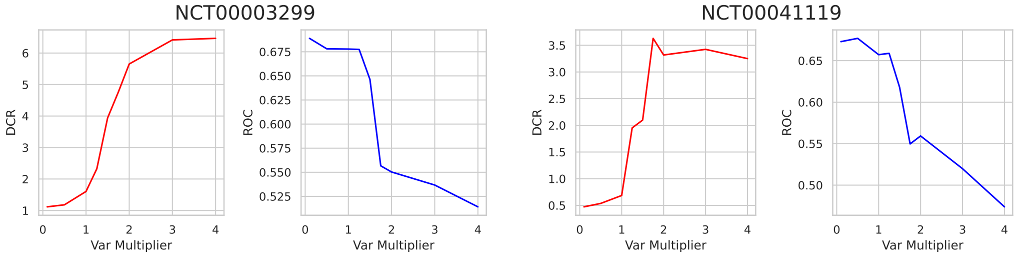

Figure 4: 2 Privacy-Utility Tradeoff examples in `TrialSynth`: Performance of distance to closest record (DCR) (red) and downstream ROC (blue) metrics at varying levels of VAE sampling variance (from 0.1 to 4), represented as the "Var Multiplier."

In `TrialSynth`, the privacy-utility tradeoff is governed by the variance applied to the VAE sampling process (Figure 4 and Figure 7). Increasing the variance in VAE sampling introduces more diversity into the synthetic data, enhancing privacy by making it harder to trace back to original data points. However, as the Var Multiplier rises, the quality of utility metrics such as downstream ROC tends to decrease, reflecting a drop in predictive accuracy and utility for downstream tasks. Conversely, metrics like DCR may rise, indicating a more extensive departure from the original dataset. A unique advantage of `TrialSynth` is its capacity to provide direct control over the tradeoff between fidelity and privacy through the adjustment of VAE sampling variance. By tuning this "Var Multiplier," researchers can precisely regulate how closely the synthetic data resembles the original dataset. Lower variance settings yield data with higher fidelity, making it more useful for predictive analyses and downstream clinical tasks, while higher variance introduces greater diversity, enhancing privacy protections by reducing the likelihood of re-identifying individual patients.

5 Discussion

The study presents `TrialSynth`, an innovative model that combines Variational Autoencoders (VAE) with Hawkes Processes (HP) to generate realistic synthetic sequential clinical trial data. Designed to address the challenges of small patient populations and the need for detailed time-event sequences, `TrialSynth` effectively captures both the timing and type of clinical events with high fidelity. Compared to existing methods, it outperforms in preserving data utility for downstream tasks while maintaining robust privacy protections, making it difficult to distinguish synthetic data from real data. Specifically, we demonstrate that `TrialSynth` outperforms existing methods in terms of data utility, enabling the generation of highly authentic event sequences across multiple real-world sequential event datasets. Empirical experiments indicate that providing the model with additional information, such as event index (Events Known) or event length, leads to significant improvements in the synthetic data quality. Finally, we believe that a sweet spot is reached by allowing the model to know the event index–as it provides a significant downstream classification boost while maintaining a low ML inference score, and is a common assumption when generating specific patients. We note that relaxing this assumption still yields competitive performance. Overall, `TrialSynth` offers a powerful solution for synthetic data generation in healthcare, balancing patient privacy with data authenticity, and shows promise for broader applications in clinical trial design and other healthcare domains that demand high-quality, secure synthetic datasets.

References

[1] Giancarlo Agnelli, Daniel J George, Ajay K Kakkar, William Fisher, Michael R Lassen, Patrick Mismetti, Patrick Mouret, Umesh Chaudhari, Francesca Lawson, and Alexander GG Turpie. Semuloparin for thromboprophylaxis in patients receiving chemotherapy for cancer. *New England Journal of Medicine*, 366(7):601–609, 2012.

[2] Steven R Alberts, Daniel J Sargent, Suresh Nair, Michelle R Mahoney, Margaret Mooney, Stephen N Thibodeau, Thomas C Smyrk, Frank A Sinicrope, Emily Chan, Sharlene Gill, et al. Effect of oxaliplatin, fluorouracil, and leucovorin with or without cetuximab on survival among patients with resected stage iii colon cancer: a randomized trial. *Jama*, 307(13):1383–1393, 2012.

[3] R Michael Baldwin, Kouros Owzar, Hitoshi Zembutsu, Aparna Chhibber, Michiaki Kubo, Chen Jiang, Dorothy Watson, Rachel J Eclov, Joel Mefford, Howard L McLeod, et al. A genome-wide association study identifies novel loci for paclitaxel-induced sensory peripheral neuropathy in calgb 40101. *Clinical Cancer Research*, 18(18):5099–5109, 2012.

[4] Mandis Beigi, Afrah Shafquat, Jason Mezey, and Jacob Aptekar. Simulants: Synthetic clinical trial data via subject-level privacy-preserving synthesis. In *AMIA Annual Symposium Proceedings*, volume 2022, page 231. American Medical Informatics Association, 2022.

[5] Yi-Tan Chang, Eric P Hoffman, Guoqiang Yu, David M Herrington, Robert Clarke, Chiung-Ting Wu, Lulu Chen, and Yue Wang. Integrated identification of disease specific pathways using multi-omics data. *bioRxiv*, page 666065, 2019.

[6] Jintai Chen, Yaojun Hu, Yue Wang, Yingzhou Lu, Xu Cao, Miao Lin, Hongxia Xu, Jian Wu, Cao Xiao, Jimeng Sun, et al. Trialbench: Multi-modal artificial intelligence-ready clinical trial datasets. *arXiv preprint arXiv:2407.00631*, 2024.

[7] Lulu Chen, Chiung-Ting Wu, Robert Clarke, Guoqiang Yu, Jennifer E Van Eyk, David M Herrington, and Yue Wang. Data-driven detection of subtype-specific differentially expressed genes. *Scientific reports*, 11(1):332, 2021.

[8] Tianyi Chen, Nan Hao, Yingzhou Lu, and Capucine Van Rechem. Uncertainty quantification on clinical trial outcome prediction. *arXiv preprint arXiv:2401.03482*, 2024.

[9] Tianyi Chen, Nan Hao, Capucine Van Rechem, Jintai Chen, and Tianfan Fu. Uncertainty quantification and interpretability for clinical trial approval prediction. *Health Data Science*, 4:0126, 2024.

[10] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pages 286–305. PMLR, 2017.

[11] Davey Daniel, Vladimer Kuchava, Igor Bondarenko, Oleksandr Ivashchuk, Sreekanth Reddy, Jana Jaal, Iveta Kudaba, Lowell Hart, Amiran Matitashvili, Yili Pritchett, et al. Trilaciclib prior to chemotherapy and atezolizumab in patients with newly diagnosed extensive-stage small cell lung cancer: a multicentre, randomised, double-blind, placebo-controlled phase ii trial. *International journal of cancer*, 148(10):2557–2570, 2021.

[12] Trisha Das, Zifeng Wang, and Jimeng Sun. Twin: Personalized clinical trial digital twin generation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 402–413, 2023.

[13] Nicholas S Downing, Jenerius A Aminawung, Nilay D Shah, Harlan M Krumholz, and Joseph S Ross. Clinical trial evidence supporting fda approval of novel therapeutic agents, 2005-2012. *Jama*, 311(4):368–377, 2014.

[14] Lynnette Fernández-Cuesta, Catherine Oakman, Priscila Falagan-Lotsch, Ke-seay Smoth, Emmanuel Quinaux, Marc Buyse, M Stella Dolci, Evandro De Azambuja, Pierre Hainaut, Patrizia Dell'Orto, et al. Prognostic and predictive value of tp53mutations in node-positive breast cancer patients treated with anthracycline-or anthracycline/taxane-based adjuvant therapy: results from the big 02-98 phase iii trial. *Breast Cancer Research*, 14(3):1–13, 2012.

[15] Lawrence M Friedman, Curt D Furberg, David L DeMets, David M Reboussin, and Christopher B Granger. *Fundamentals of clinical trials*. Springer, 2015.

[16] Tianfan Fu, Kexin Huang, and Jimeng Sun. Automated prediction of clinical trial outcome, February 2 2023. US Patent App. 17/749,065.

[17] Tianfan Fu, Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Hint: Hierarchical interaction network for clinical-trial-outcome predictions. *Patterns*, 3(4), 2022.

[18] Yi Fu, Yingzhou Liu, Yizhi Wang, Bai Zhang, Zhen Zhang, Guoqiang Yu, Chunyu Liu, Robert Clarke, David M Herrington, and Yue Wang. Ddn3. 0: Determining significant rewiring of biological network structure with differential dependency networks. *Bioinformatics*, page btae376, 2024.

[19] Angela K Green, Katherine E Reeder-Hayes, Robert W Corty, Ethan Basch, Mathew I Milowsky, Stacie B Dusetzina, Antonia V Bennett, and William A Wood. The project data sphere initiative: accelerating cancer research by sharing data. *The oncologist*, 20(5):464–e20, 2015.

[20] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[21] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

[22] Yaojun Hu and Chenhao Li. Drugclip: Contrastive drug-disease interaction for drug repurposing. *arXiv preprint arXiv:2407.02265*, 2024.

[23] Stefanie James, Chris Harbron, Janice Branson, and Mimmi Sundler. Synthetic data use: exploring use cases to optimise data utility. *Discover Artificial Intelligence*, 1(1):15, 2021.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

[26] Xixun Lin, Jiangxia Cao, Peng Zhang, Chuan Zhou, Zhao Li, Jia Wu, and Bin Wang. Disentangled deep multivariate hawkes process for learning event sequences. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 360–369. IEEE, 2021.

[27] Yingzhou Lu. *Multi-omics Data Integration for Identifying Disease Specific Biological Pathways*. PhD thesis, Virginia Tech, 2018.

[28] Yingzhou Lu, Huazheng Wang, and Wenqi Wei. Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062*, 2023.

[29] JR Mackey, T Pieńkowski, J Crown, S Sadeghi, M Martin, Arlene Chan, M Saleh, S Sehdev, L Provencher, V Semiglazov, et al. Long-term outcomes after adjuvant treatment of sequential versus combination docetaxel with doxorubicin and cyclophosphamide in node-positive breast cancer: Bcirg-005 randomized trial. *Annals of Oncology*, 27(6):1041–1047, 2016.

[30] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

[31] Harvey B Niell, James E Herndon, Antonius A Miller, Dorothy M Watson, Alan B Sandler, Karen Kelly, Randolph S Marks, Micheal C Perry, Rafat H Ansari, Grefory Otterson, et al. Randomized phase iii intergroup trial of etoposide and cisplatin with or without paclitaxel and granulocyte colony-stimulating factor in patients with extensive-stage small-cell lung cancer: Cancer and leukemia group b trial 9732. *Journal of Clinical Oncology*, 23(16):3752–3759, 2005.

[32] Zhen Pan, Zhenya Huang, Defu Lian, and Enhong Chen. A variational point process model for social event sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 173–180, 2020.

[33] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, Oct 2016.

[34] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[35] Afrah Shafquat, Jason Mezey, Mandis Beigi, Jimeng Sun, Andy Gao, and Jacob W Aptekar. An interpretable data augmentation framework for improving generative modeling of synthetic clinical trial data. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023.

[36] Brandon Theodorou, Cao Xiao, and Jimeng Sun. Synthesize high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. *Nature Communications*, 14(1):5305, 2023.

[37] Kristian Thorlund, Louis Dron, Jay JH Park, and Edward J Mills. Synthetic and external controls in clinical trials–a primer for researchers. *Clinical epidemiology*, pages 457–467, 2020.

[38] Yue Wang, Yingzhou Lu, Yinlong Xu, Zihan Ma, Hongxia Xu, Bang Du, Honghao Gao, and Jian Wu. TWIN-GPT: Digital twins for clinical trials via large language model. *arXiv preprint arXiv:2404.01273*, 2024.

[39] Zifeng Wang, Chufan Gao, Cao Xiao, and Jimeng Sun. Anypredict: Foundation model for tabular prediction. *arXiv preprint arXiv:2305.12081*, 2023.

[40] Zifeng Wang and Jimeng Sun. Promptehr: Conditional electronic healthcare records generation with prompt learning. *arXiv preprint arXiv:2211.01761*, 2022.

[41] Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915, 2022.

[42] Zifeng Wang, Brandon Theodorou, Tianfan Fu, Cao Xiao, and Jimeng Sun. Pytrial: A comprehensive platform for artificial intelligence for drug development. *arXiv preprint arXiv:2306.04018*, 2023.

[43] Chiung-Ting Wu, Sarah J Parker, Zuolin Cheng, Georgia Saylor, Jennifer E Van Eyk, Guoqiang Yu, Robert Clarke, David M Herrington, and Yue Wang. Cot: an efficient and accurate method for detecting marker genes among many subtypes. *Bioinformatics Advances*, 2(1):vbac037, 2022.

[44] YUE WU, BENJAMIN W EHLERT, DALIA PERELMAN, HEYJUN PARK, AHMED A METWALLY, YINGZHOU LU, ALESSANDRA CELLI, CAROLINE BEJIKIAN, TRACEY MCLAUGHLIN, and MICHAEL SNYDER. 1596-p: Personalized glycemic response to carbohydrates and associated physiological signatures in multiomics. *Diabetes*, 73(Supplement_1), 2024.

[45] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, 2019.

[46] Steven Yi, Adam Yee, John Harmon, Frank Meng, and Saurabh Hinduja. Enhance wound healing monitoring through a thermal imaging based smartphone app. In *Medical imaging 2018: Imaging informatics for healthcare, research, and applications*, volume 10579, pages 438–441. SPIE, 2018.

[47] Bai Zhang, Yi Fu, Yingzhou Lu, Zhen Zhang, Robert Clarke, Jennifer E Van Eyk, David M Herrington, and Yue Wang. DDN2.0: R and python packages for differential dependency network analysis of biological systems. *bioRxiv*, pages 2021–04, 2021.

[48] Kevin Zhang, Neha Patki, and Kalyan Veeramachaneni. Sequential models in the synthetic data vault. *arXiv preprint arXiv:2207.14406*, 2022.

[49] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pages 97–112. PMLR, 2021.

[50] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan+: Enhancing tabular data synthesis. *arXiv preprint arXiv:2204.00401*, 2022.

[51] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.

Contents

A Appendix

A.1 Limitations

The paper presents a promising method for generating synthetic time-sequential clinical trial data, but there are several limitations to consider. First, the generalizability of `TrialSynth` may be restricted, as its performance is demonstrated on small patient populations, leaving its effectiveness on larger, more diverse datasets uncertain. Additionally, while the use of Hawkes Processes (HP) helps model event-type and time gap prediction, this approach may struggle with more complex or non-linear temporal dynamics seen in real-world clinical data. Another limitation lies in the interpretability of the model. As a Variational Autoencoder (VAE), `TrialSynth` can be challenging to interpret compared to more traditional models, which is a crucial aspect when applying the method to clinical scenarios.

While the paper asserts that `TrialSynth` empirically preserves patient privacy, it lacks a comprehensive assessment of potential re-identification risks, leaving questions about the robustness of its privacy-preserving capabilities. Moreover, while the utility of the generated data is demonstrated in specific contexts, the broader applicability of the synthetic data, such as in clinical trials or regulatory processes, remains underexplored, (but this is a problem endemic to the field as a whole.)

A.2 Societal Impact

The societal impact of the proposed method for generating synthetic time-sequential clinical trial data has several promising positive aspects, with a few notable challenges. On the positive side, the ability to generate high-fidelity synthetic clinical data can significantly accelerate the pace of medical research and the development of new treatments. By simulating patient trajectories, researchers can optimize trial designs, potentially reducing the time and cost required to bring life-saving interventions to market. This could lead to faster availability of new drugs and treatments, especially for rare diseases or conditions where patient recruitment for trials is challenging. Additionally, synthetic data can alleviate privacy concerns, as it reduces the reliance on real patient data, thereby protecting sensitive personal information while still enabling valuable research. This would empower institutions to collaborate and share data more freely, further advancing innovation.

Another significant societal benefit lies in improving equity in healthcare research. Many populations are underrepresented in clinical trials due to geographic, socio-economic, or logistical barriers. Synthetic data generation can help address this imbalance by allowing researchers to simulate the effects of treatments on diverse populations, leading to more inclusive healthcare solutions. This could help mitigate health disparities by ensuring new treatments are designed with a broader range of patient needs in mind.

However, there are some societal challenges to consider. One potential negative impact is the over-reliance on synthetic data, which, despite its fidelity, is not a perfect substitute for real-world clinical data. There is a risk that inaccuracies in the synthetic data could lead to suboptimal clinical decisions if the limitations are not adequately understood. Additionally, while synthetic data can protect patient privacy, concerns about data security and the potential for misuse of generated data still remain. Mismanagement of synthetic data could undermine trust in medical research, particularly if stakeholders perceive it as less reliable than traditional methods.

A.3 `TrialSynth` Details

Neural Hawkes Processes are formulated as follows. We are given a set of L observations of the form (time t_j, event_type k_j). $S = \{(t_1, k_1), \ldots, (t_j, k_j), \ldots, (t_L, k_L)\}$ Each time $t_j \in \mathbb{R}^+ \bigcup \{0\}$ and is sorted such that $t_j < t_{j+1}$. Each event $k_j \in \{1, \ldots, K\}$. The traditional Hawkes Process assumption that events only have a positive, decaying influence on future events is not realistic in practice, as there exist examples where an occurrence of an event lowers the probability of a future event (e.g., medication reduces the probability of adverse events). Therefore, the Neural Hawkes Process [30] was proposed to generalize the traditional Hawkes Process. The following derivations follow [51].

$$\lambda(t) := \sum_{k=1}^{K} \lambda_k(t) := \sum_{k=1}^{K} f_k(\boldsymbol{W}_k^\top \boldsymbol{h}(t)) = \sum_{k=1}^{K} \beta_k \log\left(1 + e^{\frac{\boldsymbol{W}_k^T \boldsymbol{h}(t)}{\beta_k}}\right),$$

where $\lambda(t)$ is the intensity function for *any* event occurring, $\lambda_k(t)$ is the intensity function for the event $k \in \mathcal{K}$ occurring, $K = |\mathcal{K}|$ is the total number of event types, $h(t)$ are the hidden states of the event sequence obtained by a Transformer encoder, and $\boldsymbol{W}_k^\top$ are learned weights that calculate the significance of each event type at time t.

$f_k(c) = \beta_k \log(1 + e^{\frac{x}{\beta_k}})$ is the softplus function with parameter β_k. The output of $f_k(x)$ is always positive. Note that the positive intensity does not mean that the influence is always positive, as the influence of previous events are calculated through $\boldsymbol{W}_k^\top \boldsymbol{h}(t)$. If there is an event occurring at time t, then the probability of event k is $P(k_t = k) = \frac{\lambda_k(t)}{\lambda(t)}$.

Let the history of all events before t be represented by $\mathcal{H}_t = \{(t_j, k_j), t_j < t\}$. The continuous time intensity for prediction is defined as

$$\lambda(t|\mathcal{H}_t) := \sum_{k=1}^{K} \lambda_k(t|\mathcal{H}_t) := \sum_{k=1}^{K} f_k\left(\alpha_k \frac{t - t_j}{t_j} + \boldsymbol{W}_k^\top \boldsymbol{h}(t_j) + \mu_k\right),$$

where time is defined on interval $[t_j, t_{j+1})$, f_k is the softplus function as before, α_k is a learned importance of the interpolation between the two observed timesteps t_j and t_{j+1}. Note that when $t = t_j$, α_k does not matter as the influence is 0 (intuitively, this is because we know that this event exists, so there is no need to estimate anything). The history of all previous events up to time t is represented by $\boldsymbol{t}_j$. $\boldsymbol{W}_k^\top$ are weights that convert this history to a scalar. μ_k is the base intensity of event k. Therefore, the probability of $p(t|\mathcal{H}_{t_j})$ is the intensity at $t \in [t_j, t_{j+1})$ given the history $\mathcal{H}_t$ and the probability that no other events occur from the interval (t_j, t)

$$p(t|\mathcal{H}_{t_j}) = \lambda(t|\mathcal{H}_t) \exp\left(-\int_{t_j}^{t} \lambda(t'|\mathcal{H}_{t'}) dt'\right).$$

Note that if t_j is the last observation, then $t_{j+1} = \infty$. Finally, the next time value $\hat{t}_{j+1}$ and event prediction $\hat{k}_{j+1}$ is given as

$$\hat{t}_{j+1} = \int_{t_j}^{\infty} t \cdot p(t|\mathcal{H}_t) dt, \quad \hat{k}_{j+1} = \text{argmax}_k \frac{\lambda_k(t_{j+1}|\mathcal{H}_{t_{j+1}})}{\lambda(t_{j+1}|\mathcal{H}_{t_{j+1}})}$$

For training, we want to maximize the likelihood of the observed sequence $\{(t_1, k_1), \ldots, (t_L, k_L)\}$. The log-likelihood function is given by[4]

$$\ell(\{(t_1, k_1), \ldots, (t_L, k_L)\}) = \sum_{j=1}^{L} \log(\lambda(t_j|\mathcal{H}_{t_j})) - \int_{t_1}^{t_L} \lambda(t|\mathcal{H}_t)dt.$$

Finally, since the gradient of the log-likelihood function has an intractable integral, one may obtain an unbiased estimate by performing Monte Carlo sampling [34].

$$\nabla \left[\int_{t_1}^{t_L} \lambda(t|\mathcal{H}_t)dt\right]_{MC} = \sum_{j=2}^{L}(t_j - t_{j-1})(\frac{1}{N}\sum_{i=1}^{N} \nabla\lambda(u_i))$$

With $u_i \sim Uniform(t_{j-1}, t_j)$. $\nabla\lambda(u_i)$ is fully differentiable with respect to u_i.

Figure 2 shows an example of the proposed model with all optional structural constraints (allowing the model to access the true event knowledge, such as type and event length information). To combine the VAE and the Hawkes process, we realize that the log-likelihood can be modeled as the log-likelihood of a Hawkes process if we assume that the event times t and event types k are generated from a Multinomial Gaussian, i.e., the combined loss may be written as the following.

Sample event sequence $S_z \sim P_\theta(S|z)$ where

$$S_z = \{(t_1, k_1), \ldots, (t_L, k_L)\}$$

Then $H_{t,z}$ denotes the history up to time t in S_z.

$$\lambda_\theta(t|\mathcal{H}_{t,z}) := \sum_{k=1}^{K} \lambda_{\theta,k}(t|\mathcal{H}_{t,z}) = \sum_{k=1}^{K} f_k\left(\alpha_k \frac{t - t_j}{t_j} + \boldsymbol{W}_{\theta,k}^\top \boldsymbol{h}_\theta(t_j) + \mu_{\theta,k}\right)$$

Where $t \in [t_j, t_{j+1})$. That is, t lies between the jth and $j + 1$th observation in S_z (if t_j is the last observation, then $t_{j+1} = \infty$). $\lambda_{\theta,k}$, $\boldsymbol{W}_{\theta,k}^\top$, and $\boldsymbol{h}_\theta^\top$ are the same as the Neural Hawkes process, only parameterized by θ.

The log-likelihood is:

$$\ln P_\theta(S_z|z) = \sum_{j=1}^{L} \log(\lambda_\theta(t_j|\mathcal{H}_{t_j,z})) - \int_{t_1}^{t_L} \lambda_\theta(t|\mathcal{H}_{t,z})dt.$$

For the VAE loss, we want to minimize the Kullback–Leibler divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$, which in practice leads to maximizing the evidence lower bound (ELBO) for training along with the likelihood of x [24].

$$L_{\theta,\phi} = \mathbb{E}_{z\sim q_\phi(\cdot|x)}[\ln P_\theta(x|z)] - D_{KL}(q_\phi(\cdot|x)||P_\theta(\cdot)).$$

Adding the VAE ELBO loss, the combined `TrialSynth` loss is:

$$L_{\theta,\phi} = \mathbb{E}_{z\sim q_\phi(\cdot|S_z)}\left[\ln P_\theta(S_z|z)\right] - D_{KL}(q_\phi(\cdot|S_z)||P_\theta(\cdot|S_z)).$$

A.4 Ethics and Reproducibility

Transformer Hawkes [51] is open source and can be found at `https://github.com/SimiaoZuo/Transformer-Hawkes-Process`. Training on an NVIDIA GeForce RTX 3090 takes around 12 hrs to run the full model. The code will be made public and open source on GitHub. for the camera-ready version. All datasets were obtained from Project Data Sphere [19] with permission via a research data access request form. The links are as follows:

[4] The proof is shown in [30]

1. NCT00003299 [31]: A Randomized Phase III Study Comparing Etoposide and Cisplatin With Etoposide, Cisplatin and Paclitaxel in Patients With Extensive Small Cell Lung Cancer. Available at https://data.projectdatasphere.org/projectdatasphere/html/content/261
2. NCT00041119 [3]: Cyclophosphamide And Doxorubicin (CA) (4 VS 6 Cycles) Versus Paclitaxel (4 VS 6 Cycles) As Adjuvant Therapy For Breast Cancer in Women With 0-3 Positive Axillary Lymph Nodes:A 2X2 Factorial Phase III Randomized Study. Available at https://data.projectdatasphere.org/projectdatasphere/html/content/486
3. NCT00079274 [2]: A Randomized Phase III Trial of Oxaliplatin (OXAL) Plus 5-Fluorouracil (5-FU)/Leucovorin (CF) With or Without Cetuximab (C225) After Curative Resection for Patients With Stage III Colon Cancer. Available at https://data.projectdatasphere.org/projectdatasphere/html/content/407
4. NCT00174655 [14]: An Intergroup Phase III Trial to Evaluate the Activity of Docetaxel, Given Either Sequentially or in Combination With Doxorubicin, Followed by CMF, in Comparison to Doxorubicin Alone or in Combination With Cyclophosphamide, Followed by CMF, in the Adjuvant Treatment of Node-positive Breast Cancer Patients. Available at https://data.projectdatasphere.org/projectdatasphere/html/content/127
5. NCT00312208 [29]: A Multicenter Phase III Randomized Trial Comparing Docetaxel in Combination With Doxorubicin and Cyclophosphamide Versus Doxorubicin and Cyclophosphamide Followed by Docetaxel as Adjuvant Treatment of Operable Breast Cancer HER2neu Negative Patients With Positive Axillary Lymph Nodes. Available at https://data.projectdatasphere.org/projectdatasphere/html/content/118
6. NCT00694382 [1]: A Multinational, Randomized, Double-Blind, Placebo-controlled Study to Evaluate the Efficacy and Safety of AVE5026 in the Prevention of Venous Thromboembolism (VTE) in Cancer Patients at High Risk for VTE and Who Are Undergoing Chemotherapy. Available at https://data.projectdatasphere.org/projectdatasphere/html/content/119
7. NCT03041311 [11]: Phase 2 Study of Carboplatin, Etoposide, and Atezolizumab With or Without Trilaciclib in Patients With Untreated Extensive-Stage Small Cell Lung Cancer (SCLC). Available at https://data.projectdatasphere.org/projectdatasphere/html/content/435

A.5 Baselines

We describe our baselines in this section.

LSTM VAE: To compare against a VAE baseline, we manually implement our own LSTM VAE, which predicts the event type as a categorical classification task and the timestamp as a regression task at each event prediction.

PARSynthesizer from SDV [48, 33] since it is the most relevant model for synthesizing sequential event data, based on a conditional probabilistic auto-regressive (CPAR) model. To the best of our knowledge, no other models specifically handle sequential event data generation from scratch with easily accessible code.

TabDDPM [25] is a recently proposed state-of-the-art general tabular synthesizer based on diffusion models. Although it is not explicitly built for sequential data, we are able to enhance it by adding time as a numerical column. This model also outperforms **CTGAN** models [45, 49, 50], the previous go-to for synthetic tabular data generation. We believe that this is a strong, representative baseline of general tabular synthetic data generation.

HALO [36] is state-of-the art hierarchical autoregressive language model that has achieved state-of-the-art performance for Electronic Health Record (EHR) synthesis. Still, it does not perform well on the clinical trial evaluation datasets, primarily due to the small size of training data, demonstrating the difficulty of this task.

A.6 TrialSynth Hyperparameters

For PARSyntheizer, default hyper-parameters were used. For TabDDPM, we followed the GitHub example for churn2 `https://github.com/yandex-research/tab-ddpm`, but trained for 10,000 steps for each dataset. Total running time took around 5-6 days on a NVIDIA 2080 GPU.

Table 6: Hyperparameters Considered for TrialSynth

Parameter	Space
`embedding_size`	[32,64,128]
`patient_embedding_size`	[64,128,256,512,1024]
`num_transformer_layers` (Encoder)	[1,2,3,4,5,6,7,8]
`num_heads` (Encoder)	[2,4,8]
`num_transformer_layers` (Decoder)	[1,2,3,4,5,6,7,8]
`num_heads` (Decoder)	[2,4,8]
`lr`	[1e-3, 1e-4]

Table 7: Hyperparameters Considered for LSTM VAE

Parameter	Space
`embedding_size`	[32,64,128]
`patient_embedding_size`	[64,128,256]
`num_lstm_layers` (Encoder)	[1,2]
`hidden_size` (Encoder)	[32,64,128]
`num_lstm_layers` (Decoder)	[1,2]
`hidden_size` (Decoder)	[32,64,128]
`lr`	[1e-3, 1e-4]

A.7 ML Utility Calculation Hyperparameters

This section outlines hyperparameters explored for the downstream model for downstream ML Utility.

Table 8: Hyperparameters Considered for LSTM Predictor Models

Parameter	Space
`embedding_size`	[32,64,128]
`num_lstm_layers` (Encoder)	[1,2]
`hidden_size` (Encoder)	[32,64,128]
`lr`	[1e-3, 1e-4]

A.8 Examples

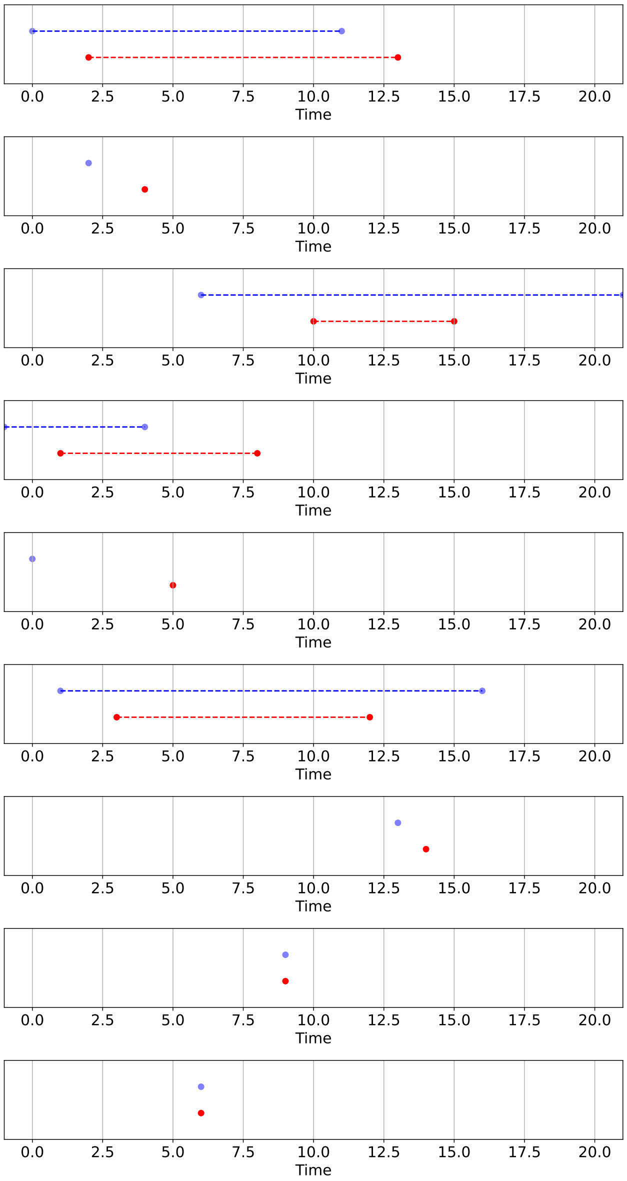

Figure 5: Example of another generated sequence from `TrialSynth` (Events Known) from NCT00003299. The blue dots denoting the specific event timestamp prediction. The red dots are the ground truth timestamps and the ground truth predictions. Each prediction is also linked with dashed lines for clarity

Figure 5 and Figure 6 show some examples of reconstructed subjects as generated by the best-performing model (`TrialSynth` (Events Known)). Intuitively, it visually reveals that the generated data generally matches the original data.

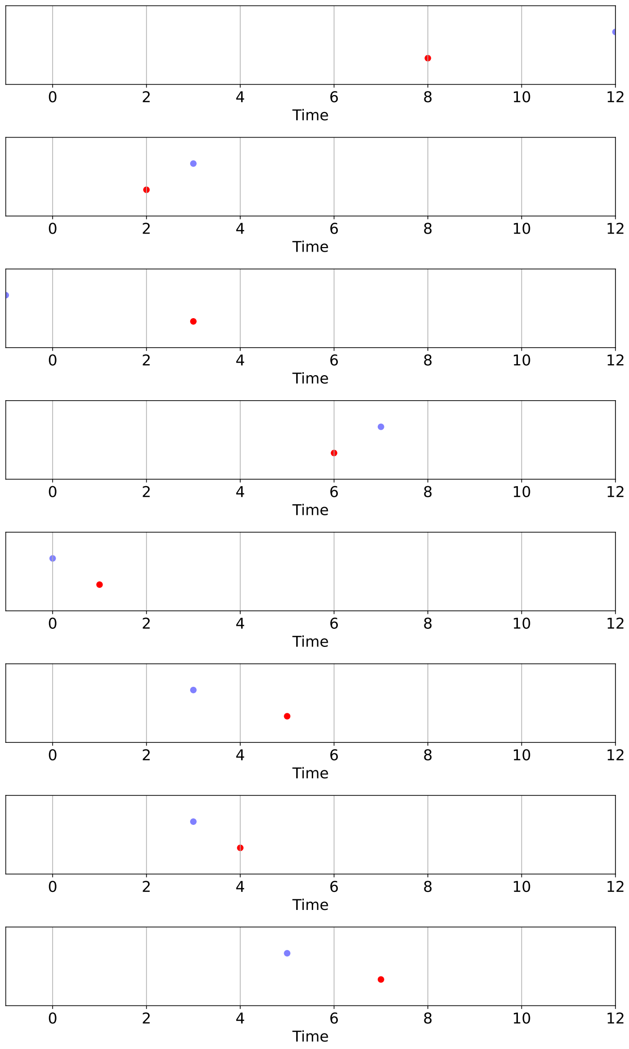

Figure 6: Example of another regenerated (encoded and decoded) sequence from `TrialSynth` (Events Known) from NCT00003299. The blue dots denoting the specific event timestamp prediction. The red dots are the ground truth timestamps and the ground truth predictions. Each prediction is also linked with dashed lines for clarity

A.9 Ablations

In this section, we include additional ablations on varying the multiplier on the standard deviation predicted by `TrialSynth` (Events Unknown), shown in Figure 7.

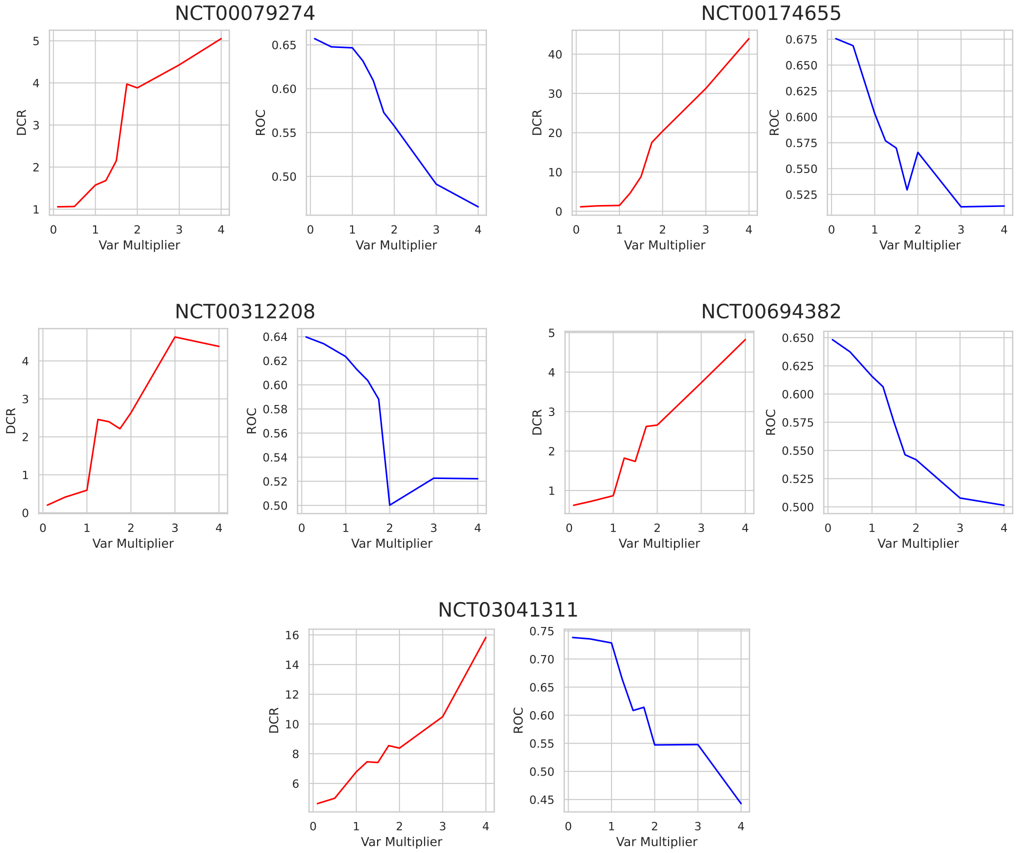

Figure 7: Additional Privacy / Utility tradeoffs examples in `TrialSynth`: Performance of distance to closest record (DCR) (red) and downstream ROC (blue) metrics at varying levels of VAE sampling variance (from 0.1 to 4), represented as the "Var Multiplier."

A.10 Utility / Privacy Spider Plots

Here, we visualize the utility/privacy trade-off that is inherent to any synthetic data generation task. Each metric is normalized for ease of visualization so that the maximum achieved metric is set as the tip of the triangle by dividing by the max. For ML Inference Privacy (where 0.5 is the ideal value), we first take the absolute value of the difference (i.e. $x = |x - 0.5|$), and then divide by the max as before.

The results are shown in Figure 8. We see a clear trade-off, as the best-performing Distance to Closest Record model, usually VAE LSTM or PAR, performs worse on the downstream ROCAC metric. This is because the generated sequences are of poorer quality, being too different from the original. The best-performing Downstream ROCAUC models also generally have good ML Inference Privacy, which is to be expected as those models generate data that is similar to the original, which would allow for (1) better performance on the held-out test set for ROCAUC and (2) being harder to distinguish from original data.

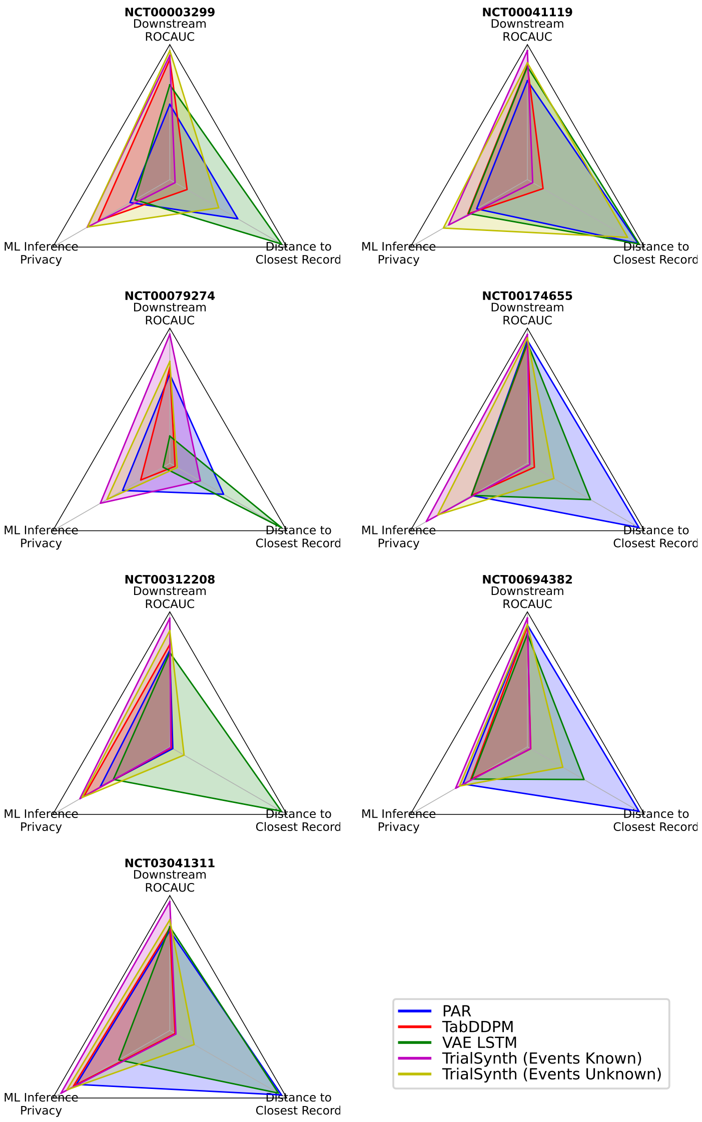

Figure 8: Spider Plots of all Models over all datasets.

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).

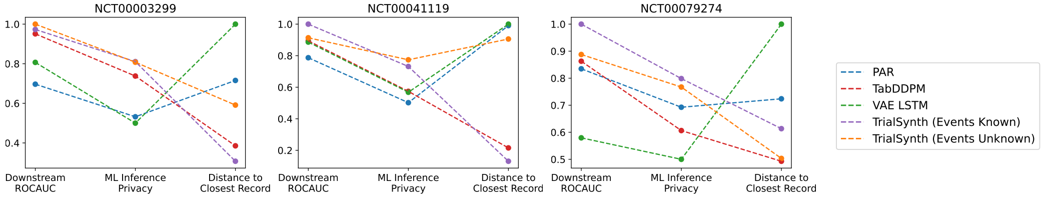

Figure 9: Line plots of all Models over 3 datasets. All metrics are normalized to scale between 0 and 1. The ROCAUC results are the fidelity results on downstream utility (ML classification of binary patient death/survival). Additional graphs are in the appendix in Figure 8.

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: See conclusion and experiments

   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: See limitations

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: [Yes]

   Justification: We point to the proof in the original paper and describe it in the appendix

   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: [No]

Justification: Code will be cleaned and anonymised first, before release

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: Code will be cleaned and anonymised for release

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 are 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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters are described, but code will be released soon after.

   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: [Yes]

   Justification: Yes, reported in table captions

   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: See appendix (hyperparameters)

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: Reviewed

   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: See limitations

    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 the responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Datasets are described, and code will be released

    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 the 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: [Yes]

    Justification: Everything should be correctly cited

    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: Code will be released soon, datasets are described in the appendix

    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: We did not collect human subjects data

    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: We did not collect human subjects data

    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.