FEDERATED TIMELINE SYNTHESIS: AN EMPIRICAL STUDY OF GENERATIVE KNOWLEDGE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic patient timelines offer a means of transferring learned information between electronic health record models without pooling raw longitudinal records. We study Federated Timeline Synthesis (FTS), a one-shot protocol in which clients share locally trained autoregressive generators and a server trains a global generator on their synthetic timelines. Building on ETHOS, we evaluate eleven training configurations across nine structured EHR prediction endpoints using three disjoint, harmonized subsets of MIMIC-IV. The experiments distinguish synthetic-only transfer, multi-source pooling, and augmentation of smaller real-data sources. In the smallest-source setting, augmentation with timelines from a larger-source generator increases DRG accuracy from 0.364 to 0.717 and hospital mortality AUROC from 0.835 to 0.893, with higher point estimates on all nine endpoints. In the larger recipient setting, gains are more task dependent, while adding synthetic sources does not consistently improve performance. These results support synthetic timelines as a mechanism for transferring predictive information between generative EHR models, while showing that the benefit depends strongly on the recipient and training configuration. We publish code: `https://anonymous.4open.science/r/fts`.

1 INTRODUCTION

Self-supervised learning has made it possible to train models that reuse information across many prediction tasks. In healthcare, however, the patient histories needed for such models are fragmented across institutional data systems. Pooling them requires access to longitudinal records that organizations may be unable to share. Clinical data also differ in coding practices, measurement frequency, and population composition. These constraints motivate methods for transferring learned information between EHR models while retaining local control of the underlying records. A central question is whether a locally trained model can convey useful clinical information through the patient histories it generates.

Generative EHR models represent longitudinal records as sequences of clinical events and learn distributions over their continuations (Kraljevic et al., 2024; Renc et al., 2024). We build on the ETHOS Patient Health Timeline (PHT) representation, which encodes structured events, numerical measurements, and elapsed time in a shared token sequence conditioned on patient attributes. This lets one autoregressive model represent a history spanning hospital, ICU, and emergency-department care.

The model supports clinical prediction by sampling future timelines and applying endpoint-specific outcome rules. For example, hospital mortality risk can be estimated from the fraction of scorable futures ending in death before discharge. Other endpoints use different prediction points and outcome rules without changing model parameters or fitting separate prediction heads. This capability suggests a second use for generated timelines: as training sequences for another model. If predictive information survives this transfer, the recipient model can support multiple clinical endpoints without receiving the source longitudinal records or being distilled for a single predefined outcome.

We investigate this mechanism through Federated Timeline Synthesis (FTS). Each client trains an ETHOS generator locally and shares it once, together with the representation and conditioning information needed for synthesis. The server generates synthetic PHTs from one or more sources and trains a *global generator* on the resulting corpus. Alternatively, a recipient combines externally

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

generated timelines with its own real records. The shared generator thus connects local learning to subsequent model training through an explicit synthetic corpus. One-shot refers to the generator upload; local training, synthesis, and subsequent training can each require many optimization or generation steps.

Research questions. This arrangement makes the utility of the generated corpus the central empirical question. Agreement in frequent codes need not establish that the sequences preserve information relevant to a particular clinical endpoint. Equally, a synthetic corpus that is an imperfect substitute for real data may still provide useful information when combined with a recipient's own records. Pooling additional sources introduces another uncertainty: it changes generator composition, source proportions, and training exposure together. We therefore separate three questions: how much predictive utility survives synthetic-only transfer; how adding external synthetic data changes a real-data recipient's performance; and how pooling synthetic sources changes the resulting model.

We evaluate these questions using eleven training configurations and nine endpoints. Three disjoint subsets of MIMIC-IV and MIMIC-IV-ED v2.2 (Johnson et al., 2023c;a; Pollard et al., 2026) simulate sources with different amounts of data, using a common tokenization and model architecture. This provides a harmonized testbed for studying synthetic-only transfer, recipient augmentation, and multi-source pooling.

Contributions. *First*, we implement one-shot transfer through synthetic timelines between generative EHR models. *Second*, we evaluate synthetic-only transfer, recipient augmentation, and source pooling through task-level performance and marginal-fidelity diagnostics. *Third*, we show that externally generated timelines can substantially improve prediction in a data-limited recipient, while augmentation effects differ across recipient configurations. We also find that simply pooling more synthetic sources does not necessarily improve performance, and that high marginal fidelity can coexist with downstream transfer gaps.

2 Related Work

BEHRT, Med-BERT, and MOTOR learn patient representations or time-to-event predictions from longitudinal records (Li et al., 2020b; Rasmy et al., 2021; Steinberg et al., 2023). Foresight and ETHOS instead model patient trajectories autoregressively (Kraljevic et al., 2024; Renc et al., 2024). Delphi-2M demonstrates that generated disease trajectories can train a subsequent predictive model (Shmatko et al., 2025). FTS examines this training role in a one-shot source–recipient setting, comparing synthetic-only transfer, augmentation of a disjoint recipient's real data, and pooling of timelines generated by independently trained sources. Synthetic EHR research includes discrete-record generation with medGAN (Choi et al., 2017) and its medWGAN and medBGAN extensions (Baowaly et al., 2019), hierarchical longitudinal generation with HALO (Theodorou et al., 2023), and hierarchy- and semantics-guided generation with HiSGT (Zhou & Barbieri, 2025). FTS builds on these modeling and synthesis approaches to study one-shot transfer through synthetic timelines.

FedAvg repeatedly aggregates locally trained parameters (McMahan et al., 2017); FedProx and SCAFFOLD address aspects of heterogeneous local optimization (Li et al., 2020a; Karimireddy et al., 2020). FedGAN extends parameter aggregation to local generators and discriminators (Rasouli et al., 2020). An alternative is to communicate synthetic examples: Goetz & Tewari (2020) optimize small client-side datasets to approximate local updates and transmit them during federated training. FTS uses neither parameter averaging nor transmission of locally generated cohorts; the server constructs its training corpus from uploaded generators.

FedGen trains a server-side generator from client prediction rules and broadcasts it to guide subsequent local learning (Zhu et al., 2021). DENSE is a closer one-shot alternative: clients upload models, the server learns an auxiliary generator using their ensemble, and a global model is distilled from ensemble predictions on generated examples (Zhang et al., 2022). FTS directly samples the uploaded timeline generators and trains with a next-token objective, without an ensemble-logit distillation loss. These are protocol distinctions; we do not empirically compare their relative performance.

Across healthcare systems, GRASP uses language-model embeddings of medical concepts to improve prediction transfer across countries and coding systems (Kirchler et al., 2026). EHRFL combines text-based EHR modeling with participant selection for federated, institution-specific prediction (Kim

et al., 2026). These address representation compatibility and collaborative prediction; FTS evaluates synthetic-corpus transfer under a shared representation.

Prior work also combines federation with clinical-data synthesis. Weldon et al. (2021) train federated GANs on partitioned binary ICU diagnosis records and evaluate the synthetic records statistically and through clinical review. FedSyn constructs a generator through federated GAN training (Behera et al., 2022). FTS examines ordered patient timelines and their use in training another generative model, extending the evaluation beyond the synthetic records themselves.

Reviews of EHR foundation models emphasize testing the claimed downstream benefit (Wornow et al., 2023), while the federated-synthesis literature distinguishes generation as the final output from generation to support subsequent learning (Little et al., 2023). EHR-Safe evaluates fidelity and downstream utility alongside empirical privacy measures (Yoon et al., 2023); Torfi et al. (2022) explicitly incorporate differential privacy into a convolutional GAN framework for medical-data synthesis. These evaluation dimensions are complementary. Our transfer experiments address predictive utility; keeping source longitudinal records local does not itself establish empirical or formal privacy guarantees.

3 METHODS

FTS transfers information through tokenized Patient Health Timelines (PHTs): local generators learn from real records, synthesize new timelines, and supply the training data for another generative model. We first describe the representation and inference formalism adopted from ETHOS (Renc et al., 2024), then formulate synthetic-only transfer, pooled global training, and recipient augmentation. Figure 1 summarizes the source, synthesis, and recipient-training stages.

Timeline representation. We model patient p by an ordered sequence of clinical events,

$$\mathcal{T}_p = (e_{p,1}, \ldots, e_{p,N_p}), \qquad e_{p,i} = (\tau_{p,i}, a_{p,i}, y_{p,i}), \qquad (\tau_{p,i}, s_{p,i}) <_{\text{lex}} (\tau_{p,i+1}, s_{p,i+1}). \tag{1}$$

Here $\tau_{p,i}$ is a timestamp, $a_{p,i}$ identifies the clinical event or code, and $y_{p,i}$ is its numerical or categorical payload, or is empty for an event without an additional value. The secondary key $s_{p,i}$ records the preprocessing order for simultaneous events. Thus equal timestamps do not imply that event order can be exchanged arbitrarily: within-timestamp ordering determines which information is present at a prediction prefix. The evaluated events comprise diagnoses, medications, procedures, laboratory measurements, vital signs, and hospital, ICU, and emergency-department transitions.

Patient Health Timeline tokenization. An event tokenizer maps each event, together with its preceding time gap $\Delta\tau_{p,i}$, to a short sequence from a shared vocabulary $\mathcal{V}$. Concatenation produces the dynamic PHT:

$$\begin{aligned} T(e_{p,i}; \Delta\tau_{p,i}) &= I(\Delta\tau_{p,i}) \,\|\, C(a_{p,i}, y_{p,i}), \\ \mathbf{x}_p &= \big(\texttt{START}\big) \,\|\, T(e_{p,1}; 0) \,\|\, \cdots \,\|\, T(e_{p,N_p}; \Delta\tau_{p,N_p}) \,\|\, \big(\texttt{END}\big). \end{aligned} \tag{2}$$

The operator $\|$ denotes concatenation; I encodes elapsed time and C encodes the event identity and value. Tokens in $\mathbf{x}_p = (x_{p,1}, \ldots, x_{p,L_p})$ are vocabulary identifiers, and L_p is the stored timeline length. An event may contribute several tokens, so token positions and clinical-event indices are distinct. START and END abbreviate the timeline-boundary tokens. A recorded timeline ending marks the extent of the available record, not necessarily death or completion of care.

Static context. A prefix $\mathbf{c}_p(t)$ supplies patient attributes at a window's starting time, including age, sex, race, and available BMI and marital-status information. Although supplied as static context within a window, some of these attributes change over the patient's history. The prefix is rebuilt when a different window is selected, and inserted context tokens are not prediction targets. Changes in mutable attributes can also appear in the dynamic stream. This distinguishes a token's role in conditioning from its clinical meaning and avoids treating all demographic or contextual information as permanently fixed.

Hierarchical and categorical tokens. Categorical events use code tokens; hierarchical diagnoses and medications are decomposed into ordered components, and procedures can likewise require multiple tokens. These components let the transformer model both shared code structure and specific combinations. A compound event must remain contiguous when a context boundary is adjusted. The

representation uses the coding semantics supplied by preprocessing: a common identifier does not itself reconcile different institutional coding practices.

Interval tokens. The gap $\Delta\tau_{p,i} = \tau_{p,i} - \tau_{p,i-1}$ is represented by predefined interval bins. Gaps below the minimum threshold produce no interval token; sufficiently long gaps may require repeated tokens for the largest interval. During generation, elapsed time is reconstructed using each interval token's estimated mean duration μ_b:

$$\widehat{\Delta t}(x_{1:j}) = \sum_{r=1}^{j} \sum_{b \in \mathcal{B}} \mathbf{1}\{x_r = \texttt{TIME_INTERVAL}_b\}\, \mu_b. \tag{3}$$

This supplies a time scale for interpreting generated futures while retaining a discrete sequence model. It approximates continuous elapsed time; it does not preserve every original timestamp exactly.

Measurement tokens. Numerical measurements are paired with discretized value tokens. For measurement type a, let $b_{a,1} < \cdots < b_{a,R_a}$ be the distinct internal quantile boundaries learned on the training fold. The bin assignment is

$$q_a(v) = 1 + \sum_{r=1}^{R_a} \mathbf{1}\{v > b_{a,r}\}, \qquad R_a \le Q - 1, \quad Q = 10. \tag{4}$$

Repeated quantile boundaries can reduce the number of distinct bins. The event's identity and bin are represented together, schematically as $C(a, v) = (\texttt{MEASUREMENT}_a, \texttt{QUANTILE}_{q_a(v)})$; events containing several measurements can contain several value tokens. This separates the identity of a measurement from its relative magnitude. The first-day SOFA transform is an exception: it uses fixed-width score bins rather than empirical quantiles. All sources use the same 4,503-token vocabulary, value boundaries, and interval estimates fitted on the original training fold before source partitioning.

Embeddings and autoregressive modeling. A trainable lookup $E : \mathcal{V} \to \mathbb{R}^d$ maps each token to an embedding; position embeddings $P_j \in \mathbb{R}^d$ give the transformer input $\mathbf{h}_j^{(0)} = E(x_j) + P_j$. The same mechanism encodes the static prefix. Following the Transformer architecture (Vaswani et al., 2017), causal self-attention conditions each next-token prediction on preceding input positions. We use structured tokens throughout the evaluated system; unstructured modality encoders are not part of these experiments. For a prefix $\mathbf{u}$, the model factorizes a continuation as

$$p_\theta(x_{1:L} \mid \mathbf{u}) = \prod_{j=1}^{L} p_\theta(x_j \mid \mathbf{u}, x_{<j}), \qquad p_{\theta,T}(v \mid \mathbf{u}) = \frac{\exp(z_\theta(v; \mathbf{u})/T)}{\sum_{v' \in \mathcal{V}} \exp(z_\theta(v'; \mathbf{u})/T)}, \tag{5}$$

where z_θ denotes next-token logits and T is decoding temperature. Conditioning histories are limited to the model's context capacity; for long synthetic sequences they are rebuilt by the continuation procedure below.

Training local generators. Assume K clients with disjoint real corpora $D_k = \{\mathbf{x}_p : p \in \mathcal{P}_k\}$. Each client trains a local generator p_{θ_k} by next-token negative log-likelihood. To make the objective reflect window sampling and masking, write $\mathcal{W} \sim \mathcal{S}(D)$ for a minibatch of sampled training windows, $u_{w,j}$ for the input prefix preceding target $x_{w,j}$, and $m_{w,j}$ for the target's loss mask. Then

$$\mathcal{L}_D(\theta) = -\mathbb{E}_{\mathcal{W} \sim \mathcal{S}(D)} \left[\frac{\sum_{w \in \mathcal{W}} \sum_j m_{w,j} \log p_\theta(x_{w,j} \mid u_{w,j})}{\sum_{w \in \mathcal{W}} \sum_j m_{w,j}} \right], \qquad \widehat{\theta}_k = \mathrm{Train}(D_k; \mathcal{L}_{D_k}). \tag{6}$$

The training operator includes optimization and validation-based checkpoint selection, rather than assuming an exact likelihood optimum. Context-prefix and padding labels are masked; genuine timeline-end labels are trainable. Explicitly forced synthetic ends are also masked. Token-sampled windows may cross patient boundaries, with their boundary tokens and associated context. Consequently, the objective is not uniform sampling of whole patients, and longer timelines contribute more possible window starts.

Server-side synthesis. After local training, the server receives the generators and the information needed to use their representation and construct generation prompts: model configuration, vocabulary and numerical decoding metadata, and source conditioning profiles. Let $a_{k,i}$ denote the profile and timeline-start information for source patient i, and let $G(\widehat{\theta}_k, a_{k,i}; T_{\mathrm{syn}}, B)$ denote autoregressive

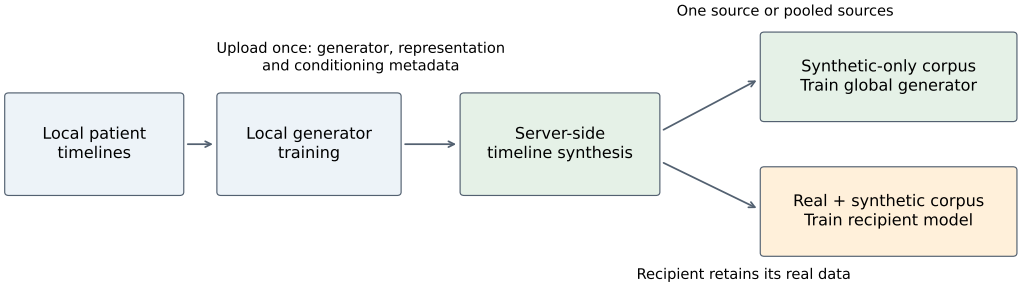

Figure 1: FTS information flow. Sources upload generators and metadata once; the server synthesizes timelines for global training or augmentation of a recipient's real data.

generation with the continuation and stopping policy, at temperature T_{syn} and token budget B. The retained synthetic corpus is

$$\widetilde{\mathbf{x}}_{k,i} \sim G(\widehat{\theta}_k, a_{k,i}; T_{\text{syn}}, B), \qquad \widetilde{D}_k = \{\widetilde{\mathbf{x}}_{k,i} : i \in \mathcal{A}_k\}, \qquad M_k = |\mathcal{A}_k|, \tag{7}$$

where $\mathcal{A}_k$ indexes accepted outputs. Each corpus targets the source's patient count, but M_k can be smaller because short outputs are excluded. The implementation reuses source birth-date and static-code profiles rather than independently sampling demographics. Some attributes use their first recorded value, which may occur after the real timeline starts. This is population-conditioned generation, not a forecast of that patient's history from information available at the start.

Synthesis begins with the static prefix and START at $T_{\text{syn}} = 1.0$, using sliding continuation for histories exceeding the context window. Appendix B describes continuation, stopping, filtering, and retained cohort sizes.

Global generator and recipient training. For a source set $\mathcal{K}$, the server pools the accepted corpora and trains a fresh global generator using the same next-token objective. Recipient augmentation instead includes real data from recipient r:

$$\begin{aligned} \widetilde{D}_{\mathcal{K}} &= \biguplus_{k \in \mathcal{K}} \widetilde{D}_k, & \widehat{\theta}^{\text{syn}}_{\mathcal{K}} &= \text{Train}(\widetilde{D}_{\mathcal{K}}; \mathcal{L}_{\widetilde{D}_{\mathcal{K}}}), \\ D^{\text{aug}}_{r,\mathcal{K}} &= D_r \uplus \widetilde{D}_{\mathcal{K}}, \quad r \notin \mathcal{K}, & \widehat{\theta}^{\text{aug}}_{r,\mathcal{K}} &= \text{Train}(D^{\text{aug}}_{r,\mathcal{K}}; \mathcal{L}_{D^{\text{aug}}_{r,\mathcal{K}}}). \end{aligned} \tag{8}$$

Here $\uplus$ denotes corpus concatenation, retaining all records rather than deduplicating token sequences. Synthetic-only transfer is the special case $\mathcal{K} = \{k\}$, compared with training on real D_k. Multi-source pooling varies $\mathcal{K}$; recipient augmentation compares $D^{\text{aug}}_{r,\mathcal{K}}$ with D_r. These are distinct comparisons even though they share an architecture and loss. Client contributions are unequal: source k supplies $M_k / \sum_{j \in \mathcal{K}} M_j$ of synthetic patients, whereas its token share is

$$\pi^{\text{tok}}_k = \frac{\sum_{\mathbf{x} \in \widetilde{D}_k} |\mathbf{x}|}{\sum_{j \in \mathcal{K}} \sum_{\mathbf{x} \in \widetilde{D}_j} |\mathbf{x}|}. \tag{9}$$

Token sampling is tied to possible window starts, not an equal-source mixture. Validation-tail removal and window boundaries further affect actual training exposure. One-shot describes the one-time generator upload: synthesis and downstream training require substantial server computation but no repeated client–server synchronization or subsequent access to the source longitudinal event records.

Zero-shot inference through future PHTs. Given an observed history ending at a task-defined prediction point, form a context $\mathbf{u}_i$ containing static context and up to 2,048 total input tokens. Sample N future PHTs $\widetilde{\mathbf{x}}_i^{(n)}$ from the model at inference temperature T_{inf}, stopping at task-relevant tokens or the remaining context budget. The same model supports different endpoints by changing the prediction point and the rule that reads the generated trajectory, without task-specific parameter updates or separately trained heads. Outcome labels are still required to evaluate predictions.

For a binary endpoint, let $g_{\mathcal{E}}(\widetilde{\mathbf{x}}_i^{(n)}) \in \{0,1\}$ encode the event and its time-window criterion, and let $v_i^{(n)}$ indicate that the future is scorable. With $N = 100$ requested futures, the predicted probability is

$$N_i^{\text{valid}} = \sum_{n=1}^{N} v_i^{(n)}, \qquad \widehat{P}_i(\mathcal{E}) = \frac{\sum_{n=1}^{N} v_i^{(n)} g_{\mathcal{E}}(\widetilde{\mathbf{x}}_i^{(n)})}{N_i^{\text{valid}}}, \qquad N_i^{\text{valid}} > 0. \tag{10}$$

The outcome rule can depend on event order and elapsed time, not merely the presence of a token. Futures without a resolved outcome are excluded rather than counted as negative. An example with $N_i^{\text{valid}} = 0$ is unscored, so coverage can differ across models. These probabilities summarize generated futures conditional on scoring validity; sampling alone does not establish calibration.

For a numerical endpoint, a decoder $h(\widetilde{\mathbf{x}})$ extracts the value from a valid future and we compute

$$\widehat{y}_i = \frac{\sum_{n:v_i^{(n)}=1} h(\widetilde{\mathbf{x}}_i^{(n)})}{N_i^{\text{valid}}}. \tag{11}$$

For SOFA, h decodes the quantile immediately following the first SOFA marker in the generated ICU stay to the mean integer score in that bin. A malformed pair or a value from a later stay is not substituted for a missing prediction.

For the evaluated multiclass endpoint, DRG, the completed stay is already in the input. The model directly predicts the next code from the allowed DRG set $\mathcal{V}_{\text{DRG}}$, rather than estimating class frequencies across simulated futures:

$$\widehat{P}_i(c) = \frac{\exp z_\theta(c; \mathbf{u}_i)}{\sum_{c' \in \mathcal{V}_{\text{DRG}}} \exp z_\theta(c'; \mathbf{u}_i)}, \qquad \widehat{c}_i = \arg\max_{c \in \mathcal{V}_{\text{DRG}}} \widehat{P}_i(c). \tag{12}$$

This constrained top-1 decision is unchanged by a common positive temperature. Binary tasks use raw AUROC, DRG uses top-1 accuracy, and SOFA uses R^2; Table 2 specifies the prediction point and scoring rule for each endpoint.

4 Experimental Design

Data and configurations. We use MIMIC-IV v2.2 and MIMIC-IV-ED v2.2 (Johnson et al., 2023c;b;a; Pollard et al., 2026). To construct sources with different amounts of available data, we partition the training set into 14 disjoint patient-level shards and define `big`, `small`, and `little` using 10, 3, and 1 shards, respectively. The resulting sources contain 168,223, 50,315, and 16,846 patients, corresponding to 71.47%, 21.38%, and 7.16% of the training set. The full patient-level split contains 235,384 training patients (`orig`), 13,098 study-evaluation patients (`eval`), and 13,108 test patients, approximately 90%/5%/5%. The three source datasets provide harmonized settings with different levels of data availability.

Eleven configurations comprise four real-data models, three synthetic-only models, two synthetic pools (`big_synth+small_synth`, with and without `little_synth`), and two augmented recipients (`small+big_synth`, `little+big_synth`). The centralized `orig` reference includes all three real sources; it is not a matched two-source reference or a presumed upper bound. Table 3 defines every corpus and its size.

Training. All models use the same three-layer GPT-2 architecture with hidden dimension 768, 12 attention heads, and a 4,096-token context. We train on token-sampled windows with cosine learning-rate decay and validation-based checkpoint selection. Appendices C–E provide the training configuration, corpus composition, and computational costs, including per-model measurements in Table 4.

Endpoints and evaluation. We evaluate nine clinically relevant prediction endpoints spanning hospital, ICU, and emergency-department care: hospital readmission, hospital mortality, ICU admission, ICU readmission, first-day SOFA, completed-stay DRG, ED hospitalization, ED critical outcome, and ED re-presentation. Seven binary endpoints are evaluated using raw AUROC, completed-stay DRG using accuracy, and first-day SOFA using R^2. Table 2 specifies the prediction point, prediction horizon, and scoring criteria for each endpoint.

Table 1: Test performance for all eleven training configurations. B, S, and L denote big, small, and little; a tilde denotes the corresponding synthetic corpus. Bold/underline denote best/second-best displayed values. Complete metrics and 95% intervals appear in Table 11.

Training corpus Metric	Readm. AUC	Mort. AUC	ICU adm. AUC	ICU readm. AUC	SOFA R^2	DRG Acc.	ED hosp. AUC	ED crit. AUC	ED re-pres. AUC
Centralized reference									
orig	**0.755**	0.867	**0.889**	**0.701**	<u>0.489</u>	**0.766**	**0.823**	**0.893**	0.629
Synthetic-only transfer: real / synthetic									
B	0.744	0.869	0.877	0.684	**0.494**	<u>0.757</u>	0.810	0.868	<u>0.634</u>
$\widetilde{B}$	0.733	0.861	0.875	<u>0.693</u>	0.461	0.705	0.816	0.884	0.609
S	0.744	<u>0.892</u>	<u>0.880</u>	0.680	0.474	0.582	0.819	<u>0.891</u>	0.622
$\widetilde{S}$	0.735	0.875	0.862	0.644	0.446	0.502	0.813	0.885	0.607
L	0.720	0.835	0.856	0.623	0.422	0.364	0.802	0.868	0.573
$\widetilde{L}$	0.702	0.782	0.803	0.603	0.331	0.252	0.780	0.812	0.583
Recipient augmentation: compare with S and L above									
$S + \widetilde{B}$	0.747	0.879	0.879	0.690	0.483	0.730	0.819	0.890	**0.637**
$L + \widetilde{B}$	<u>0.752</u>	**0.893**	0.879	0.684	0.465	0.717	<u>0.821</u>	0.887	0.626
Synthetic-source pooling									
$\widetilde{B} + \widetilde{S}$	0.743	0.862	0.871	0.673	0.478	0.695	0.817	0.879	0.614
$\widetilde{B} + \widetilde{S} + \widetilde{L}$	0.725	0.783	0.828	0.631	0.409	0.505	0.799	0.800	0.584

Inference and uncertainty estimation. We evaluate the `orig` model on `eval` across inference temperatures from 0.7 to 1.0 to characterize downstream decoding sensitivity and select a common inference setting for subsequent experiments. This analysis is conducted separately from synthetic-data generation temperature. We report 95% bootstrap confidence intervals obtained by resampling scored evaluation examples. Complete task-level results with confidence intervals are provided in Table 11.

5 RESULTS

Table 1 summarizes test performance across all eleven training configurations, while Figure 2 highlights the recipient-augmentation comparisons. Complete task-level confidence intervals and configuration contrasts are provided in Appendix H, Tables 10 and 11. We select inference temperature 1.0 based on the `orig` model's evaluation on `eval` (Appendix G) and use this common setting for all results below.

The configurations distinguish source-trained donors, students trained on their generated timelines, and recipients trained on real–synthetic combinations. These comparisons assess synthetic-only utility relative to the source-trained donor, augmentation relative to the recipient's local-only baseline, and the additional benefit of including local real data relative to synthetic-only training. Table 10 reports the task-level differences for augmentation and source pooling.

Synthetic-only training supports multiple clinical endpoints. A model trained on `big_synth` achieves hospital mortality AUROC of 0.861 and ICU-admission AUROC of 0.875, compared with 0.869 and 0.877 for the corresponding real-data donor model. Its ICU-readmission and ED critical-outcome AUROCs are 0.693 and 0.884, compared with 0.684 and 0.868 for the donor. Completed-stay DRG accuracy is 0.705, compared with 0.757. These results demonstrate that generated patient histories can support training for multiple prediction endpoints, with task-dependent differences from the source-trained model.

The generated corpora also show high agreement with their real-source counterparts on the evaluated marginal diagnostics. Across the three sources, timeline-view unigram R^2 ranges from 0.983 to 0.998 and dimension-wise R^2 from 0.992 to 0.996. Readmission-view values range from 0.988 to 0.998 and from 0.989 to 0.997, respectively (Tables 5–6; Figures 3–4). These measurements characterize

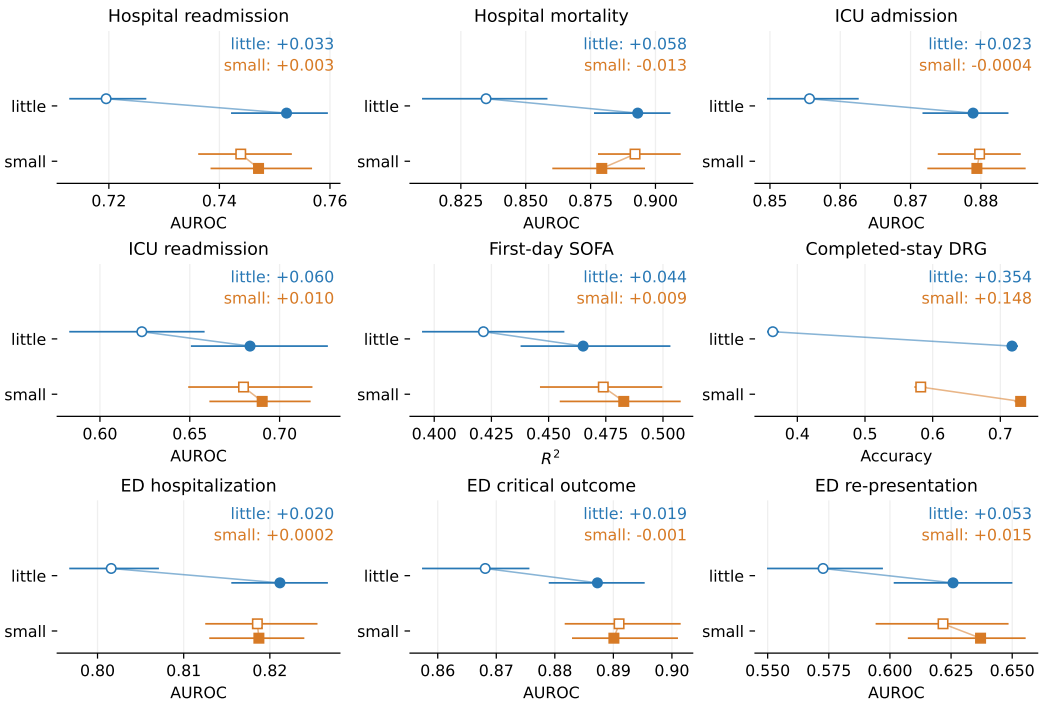

Figure 2: Recipient augmentation on test: real data alone (open) versus adding `big_synth` (filled), for little (blue) and small (orange). Annotations show augmented-minus-real-only differences computed from unrounded scores; bars show 95% intervals. Axes are scaled independently, so compare numerical changes rather than visual displacement.

agreement in the code-frequency distributions of selected patient-bounded contexts, complementing downstream evaluation without establishing entire-record temporal fidelity.

The smallest-source comparison illustrates the distinction between marginal agreement and downstream training utility. Despite timeline-view unigram R^2 of 0.983 and dimension-wise R^2 of 0.992, the synthetic-trained model has lower point estimates than its real-data counterpart on eight of nine endpoints, including DRG accuracy of 0.252 versus 0.364 and SOFA R^2 of 0.331 versus 0.422. ED re-presentation is the exception, with AUROC of 0.583 versus 0.573.

External synthetic timelines support local-data augmentation. Augmenting `little` with `big_synth` raises the point estimate on all nine endpoints (Fig. 2). DRG accuracy increases from 0.364 to 0.717, hospital mortality AUROC from 0.835 to 0.893, and ICU readmission AUROC from 0.623 to 0.684. SOFA R^2 increases from 0.422 to 0.465. Here, the recipient benefits from timelines generated by a model trained on a larger, disjoint source.

For the larger `small` recipient, changes are more heterogeneous. DRG accuracy increases from 0.582 to 0.730 and ED re-presentation AUROC from 0.622 to 0.637. Hospital readmission changes from 0.744 to 0.747 and SOFA R^2 from 0.474 to 0.483. Hospital mortality decreases from 0.892 to 0.879, while ICU-admission and ED critical-outcome AUROCs decrease slightly; ED hospitalization is nearly unchanged (0.819 in both settings). Augmentation thus improves all nine point estimates for little but has mixed effects for small. Recipient size, cohort identity, and synthetic-to-real data ratio differ together, so the contrast cannot be attributed to size alone. The added corpus also increases training exposure.

Both augmented models also achieve higher point estimates than the synthetic-only student trained on `big_synth` on eight of nine endpoints; ICU readmission is the exception. For example, completed-stay DRG accuracy increases from 0.705 for the synthetic-only student to 0.717 and 0.730 for the `little` and `small` augmented recipients, respectively. Hospital mortality AUROC increases from 0.861 to 0.893 and 0.879. Alongside the comparisons with local-only training, these results support

combining generated histories with recipient real data as a useful training configuration. Table 10 presents the differences relative to both constituent-corpus baselines.

As an additional reference, `small+big_synth` also achieves higher point estimates than the unadapted `big` donor on seven of nine endpoints, including hospital mortality AUROC of 0.879 versus 0.869 and ED critical-outcome AUROC of 0.890 versus 0.868. The donor retains higher completed-stay DRG accuracy and SOFA R^2.

Training utility varies across synthetic-source combinations. Adding `small_synth` to `big_synth` produces mixed changes. Hospital readmission AUROC rises from 0.733 to 0.743 and SOFA R^2 from 0.461 to 0.478, whereas DRG accuracy falls from 0.705 to 0.695 and ICU readmission AUROC from 0.693 to 0.673. Adding `little_synth` to this two-source pool lowers all nine point estimates: DRG accuracy becomes 0.505, hospital mortality AUROC 0.783, and ED critical-outcome AUROC 0.800. More synthetic sources do not consistently improve the resulting model.

Together, these comparisons distinguish the training utility of generated histories across synthetic-only learning, recipient augmentation, and source pooling, with the clearest augmentation gains in the `little` configuration.

6 Discussion and conclusion

Synthetic timelines as a training resource. FTS turns generated patient histories into training data for generative EHR models supporting multiple clinical endpoints. Once source generators and the required metadata are uploaded, the server constructs synthetic corpora without further access to source longitudinal records. The same next-token objective supports synthetic-only training, recipient augmentation, and multi-source pooling.

Synthetic histories complement local real data. In the smallest-recipient configuration, real–synthetic training achieves higher point estimates than both the local-only model and the synthetic-only student on eight of nine endpoints. These comparisons support generated histories as useful supplements to local records, even when the synthetic-only student does not match the donor on every endpoint. The more task-dependent pattern for `small` further motivates assessing augmentation separately from standalone synthetic-data utility.

Training utility complements marginal fidelity. The evaluation examines both code-frequency agreement and the predictive performance of models trained on generated histories. High marginal agreement coexists with task-dependent synthetic-only performance and useful recipient augmentation. Together with the pooling comparisons, these results support assessing synthetic corpora through both statistical agreement and their utility in a specified training role.

Limitations and further evaluation. The experiments use disjoint, harmonized subsets of one institutional dataset. Recipient cohort, synthetic-to-real ratio, optimization schedule, training exposure, and validation composition are not independently controlled across configurations. Bootstrap intervals quantify scored-example variability rather than independent repetitions of training and synthesis; the resampling does not cluster episodes by patient, and scoring coverage can differ across models. Marginal diagnostics assess selected contexts rather than complete temporal or clinical validity. Direct donor fine-tuning, teacher ensembles, and alternative federated optimizers remain to be evaluated. Finally, shared generators, downstream models, and reused conditioning profiles remain potential disclosure channels; formal privacy guarantees and empirical resistance to disclosure attacks are not established.

Conclusion. FTS characterizes synthetic patient timelines as standalone training data, supplements to local real records, and components of multi-source training corpora. Across eleven configurations and nine clinical endpoints, the clearest positive result is recipient augmentation: in the smallest-recipient setting, real–synthetic training achieves higher point estimates than either constituent-only model on eight endpoints. These findings support generated timelines as useful complements to local data and distinguish their augmentation value from their utility in synthetic-only training.

Reproducibility Statement

Appendix A specifies the downstream endpoints and scoring rules; Appendix B describes representation and synthesis; Appendices C–E document training, corpus composition, and computational costs; and Appendices F–H provide fidelity diagnostics, inference-temperature and calibration analyses, and complete task-level results. The primary configuration comparisons use the held-out test split, while the inference-temperature and calibration analyses use `eval`. Binary endpoints are evaluated using raw AUROC.

AI Use Disclosure

Generative AI tools were used to assist literature review, experimental planning, code implementation, figure and table preparation, and manuscript drafting and polishing. The synthetic EHR corpora evaluated in this work were generated by the trained ETHOS models described in the methods. The authors are responsible for the final content and verification of the AI-assisted work.

References

Mrinal Kanti Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3):228–241, 2019.

Monik Raj Behera, Sudhir Upadhyay, Suresh Shetty, Sudha Priyadarshini, Palka Patel, and Ker Farn Lee. Fedsyn: Synthetic data generation using federated learning. *arXiv preprint arXiv:2203.05931*, 2022.

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*, pp. 286–305. PMLR, 2017.

Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020. URL `https://arxiv.org/abs/2008.04489`.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and Steven Horng. MIMIC-IV-ED, 2023a. URL `https://physionet.org/content/mimic-iv-ed/2.2/`. Version 2.2.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV, 2023b. URL `https://doi.org/10.13026/6mm1-ek67`. Version 2.2.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023c.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Jiyoun Kim, Junu Kim, Kyunghoon Hur, and Edward Choi. Federated learning for heterogeneous electronic health record systems with cost effective participant selection. *Scientific Reports*, 16: 6876, 2026. doi: 10.1038/s41598-026-38299-9.

Matthias Kirchler, Matteo Ferro, Veronica Lorenzini, Robin P. van de Water, FinnGen, Christoph Lippert, and Andrea Ganna. Large language models improve transferability of electronic health record-based predictions across countries and coding systems. *npj Digital Medicine*, 9:177, 2026. doi: 10.1038/s41746-026-02363-5.

Zeljko Kraljevic, Dan Bean, Anthony Shek, Rebecca Bendayan, Harry Hemingway, Joshua Au Yeung, Alexander Deng, Alfred Balston, Jack Ross, Esther Idowu, et al. Foresight—a generative pretrained transformer for modelling of patient timelines using electronic health records: a retrospective modelling study. *The Lancet Digital Health*, 6(4):e281–e290, 2024.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020a.

Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):7155, 2020b.

Claire Little, Mark Elliot, and Richard Allmendinger. Federated learning for generating synthetic data: a scoping review. *International Journal of Population Data Science*, 8(1):2158, 2023.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Tom Pollard, Benjamin E. Moody, Li-wei Lehman, Brian J. Gow, Chrystinne Fernandes, Chen Xie, Alistair Johnson, Roger G. Mark, and Thomas Heldt. PhysioNet as a global platform for biomedical research. *Nature Health*, 1(8):792–795, 2026. doi: 10.1038/s44360-026-00096-z.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86, 2021.

Mohammad Rasouli, Tao Sun, and Ram Rajagopal. FedGAN: Federated generative adversarial networks for distributed data. *arXiv preprint arXiv:2006.07228*, 2020. URL `https://arxiv.org/abs/2006.07228`.

Pawel Renc, Yugang Jia, Anthony E Samir, Jaroslaw Was, Quanzheng Li, David W Bates, and Arkadiusz Sitek. Zero shot health trajectory prediction using transformer. *NPJ Digital Medicine*, 7 (1):256, 2024.

Artem Shmatko, Alexander Wolfgang Jung, Kumar Gaurav, Søren Brunak, Laust Hvas Mortensen, Ewan Birney, Tom Fitzgerald, and Moritz Gerstung. Learning the natural history of human disease with generative transformers. *Nature*, 647:248–256, 2025. doi: 10.1038/s41586-025-09529-3.

Ethan Steinberg, Jason Fries, Yizhe Xu, and Nigam Shah. Motor: a time-to-event foundation model for structured medical records. *arXiv preprint arXiv:2301.03150*, 2023.

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.

Amirsina Torfi, Edward A Fox, and Chandan K Reddy. Differentially private synthetic medical data generation using convolutional gans. *Information Sciences*, 586:485–500, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

John Weldon, Tomas Ward, and Eoin Brophy. Generation of synthetic electronic health records using a federated gan. *arXiv preprint arXiv:2109.02543*, 2021.

Michael Wornow, Yizhe Xu, Rahul Thapa, Birju Patel, Ethan Steinberg, Scott Fleming, Michael A Pfeffer, Jason Fries, and Nigam H Shah. The shaky foundations of large language models and foundation models for electronic health records. *npj digital medicine*, 6(1):135, 2023.

Jinsung Yoon, Michel Mizrahi, Nahid Farhady Ghalaty, Thomas Jarvinen, Ashwin S Ravi, Peter Brune, Fanyu Kong, Dave Anderson, George Lee, Arie Meir, et al. Ehr-safe: generating high-fidelity and privacy-preserving synthetic electronic health records. *NPJ digital medicine*, 6(1):141, 2023.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/868f2266086530b2c71006ea1908b14a-Abstract.html.

Guanglin Zhou and Sebastiano Barbieri. Generating clinically realistic EHR data via a hierarchy- and semantics-guided transformer. In *ECAI 2025*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2025. doi: 10.3233/FAIA251280.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhu21b.html.