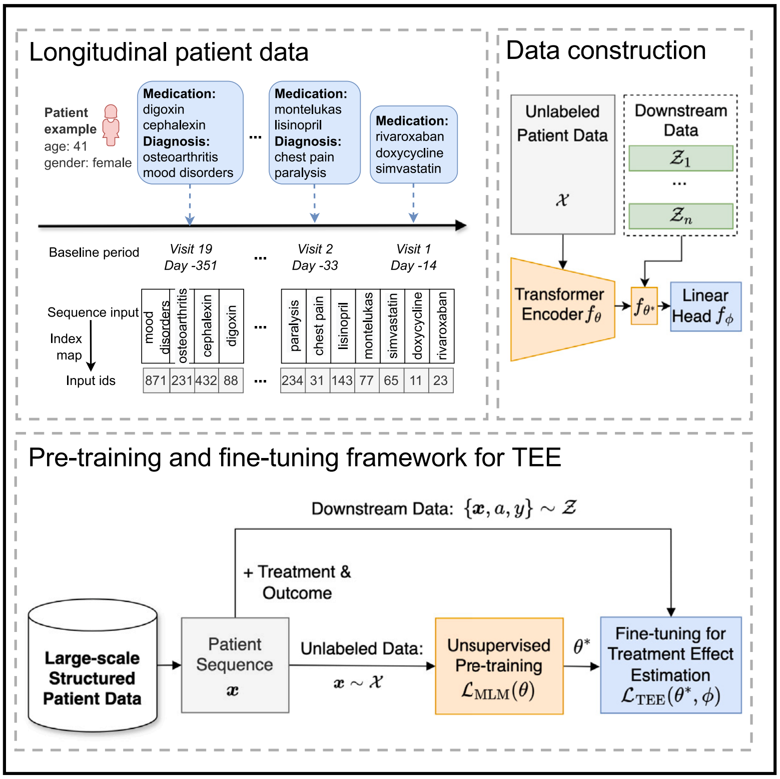
Article

Patterns

CURE: A deep learning framework pre-trained on large-scale patient data for treatment effect estimation

Graphical abstract


Longitudinal patient data
Patient example
age: 41
gender: female
Medication: digoxin cephalexin
Diagnosis: osteoarthritis mood disorders
...
Medication: montelukas lisinopril
Diagnosis: chest pain paralysis
Medication: rivaroxaban doxycycline simvastatin
Baseline period
Visit 19 Day -351
...
Visit 2 Day -33
Visit 1 Day -14
Sequence input
Index map
Input ids
mood disorders
osteoarthritis
cephalexin
digoxin
...
paralysis
chest pain
lisinopril
montelukas
simvastatin
doxycycline
rivaroxaban
871 231 432 88 ... 234 31 143 77 65 11 23
Data construction
Unlabeled Patient Data
$\mathcal{X}$
Downstream Data
$\mathcal{Z}_1$
...
$\mathcal{Z}_n$
Transformer Encoder f_θ
f_{θ^*}
Linear Head f_ϕ
Pre-training and fine-tuning framework for TEE
Downstream Data: $\{\boldsymbol{x}, a, y\} \sim \mathcal{Z}$
+ Treatment & Outcome
Large-scale Structured Patient Data
Patient Sequence $\boldsymbol{x}$
Unlabeled Data: $\boldsymbol{x} \sim \mathcal{X}$
Unsupervised Pre-training $\mathcal{L}_{\mathrm{MLM}}(\theta)$
θ^*
Fine-tuning for Treatment Effect Estimation $\mathcal{L}_{\mathrm{TEE}}(\theta^*, \phi)$

Highlights

- A new pre-training framework enhancing treatment effect estimation
- Pre-training on 3 million patient records, enriching data representation
- Superior performance in treatment effect estimation against existing methods
- Case studies demonstrating the model's efficacy in supplementing RCTs

Authors

Ruoqi Liu, Pin-Yu Chen, Ping Zhang

Correspondence

zhang.10631@osu.edu

In brief

Current treatment effect estimation (TEE) methods are limited by small-scale labeled data reliance. The study proposes CURE (causal treatment effect estimation), a novel transformer-based framework for TEE, leveraging real-world patient data. CURE is pre-trained on large-scale unlabeled patient data to learn representative contextual patient representations and fine-tuned on labeled patient data to enhance TEE. The results highlight CURE's superior performance over existing approaches and its effectiveness in generating clinical hypotheses to supplement standard randomized clinical trials.

Liu et al., 2024, Patterns *5*, 100973
June 14, 2024 © 2024 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.patter.2024.100973

CellPress

Patterns

CellPress OPEN ACCESS

Article

CURE: A deep learning framework pre-trained on large-scale patient data for treatment effect estimation

Ruoqi Liu,[1] Pin-Yu Chen,[2] and Ping Zhang[1,3,4,5,*]

[1]Department of Computer Science and Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210, USA
[2]IBM Research, Yorktown Heights, NY 10598, USA
[3]Department of Biomedical Informatics, The Ohio State University, 1800 Cannon Drive, Columbus, OH 43210, USA
[4]Translational Data Analytics Institute, The Ohio State University, 1760 Neil Avenue, Columbus, OH 43210, USA
[5]Lead contact
*Correspondence: zhang.10631@osu.edu

https://doi.org/10.1016/j.patter.2024.100973

THE BIGGER PICTURE Treatment effect estimation (TEE) is essential to understand the causal impact of medical interventions on patient outcomes, especially when using real-world data (RWDs). The inherent complexity of RWDs, however, complicates the accurate modeling of treatment effects. Traditional machine learning approaches, often hampered by the scarcity of labeled data, fall short of adequately addressing the confounding bias and capturing the complex interplay between treatments, patient characteristics, and outcomes. The advent of foundation models pre-trained on extensive datasets presents a promising opportunity to transcend these limitations. Therefore, developing methods that fully investigate the capabilities of large-scale patient data is important to learn enhanced patient representations and improve the efficacy of TEE.

SUMMARY

Treatment effect estimation (TEE) aims to identify the causal effects of treatments on important outcomes. Current machine-learning-based methods, mainly trained on labeled data for specific treatments or outcomes, can be sub-optimal with limited labeled data. In this article, we propose a new pre-training and fine-tuning framework, CURE (causal treatment effect estimation), for TEE from observational data. CURE is pre-trained on large-scale unlabeled patient data to learn representative contextual patient representations and fine-tuned on labeled patient data for TEE. We present a new sequence encoding approach for longitudinal patient data embedding both structure and time. Evaluated on four downstream TEE tasks, CURE outperforms the state-of-the-art methods, marking a 7% increase in area under the precision-recall curve and an 8% rise in the influence-function-based precision of estimating heterogeneous effects. Validation with four randomized clinical trials confirms its efficacy in producing trial conclusions, highlighting CURE's capacity to supplement traditional clinical trials.

INTRODUCTION

Treatment effect estimation (TEE) is used to evaluate the causal effects of treatment strategies on some important outcomes, which is a crucial problem in healthcare.[1] Randomized clinical trials (RCTs) are the *de facto* gold standard for identifying causal effects through randomizing the treatment assignment and comparing the responses in different treatment groups. However, conducting RCTs is time consuming, expensive, and sometimes unethical.[2,3] Observational data such as medical claims provide a promising opportunity for TEE as a complement to RCTs.[4]

Recently, many works have been proposed to adopt neural networks (NNs) for TEE from observational data.[5–11] Compared to classical TEE methods such as regression trees[12] or random forests,[13] NN-based methods achieve better performance in handling the relationships among covariates, treatment, and outcome. However, there are still some common limitations of existing TEE methods: (1) most model designs are task specific or data specific, so it is hard to adapt the model to a more generalized setting, and (2) existing labeled datasets often have small-scale data size, whereas training neural models requires large and high-quality labeled

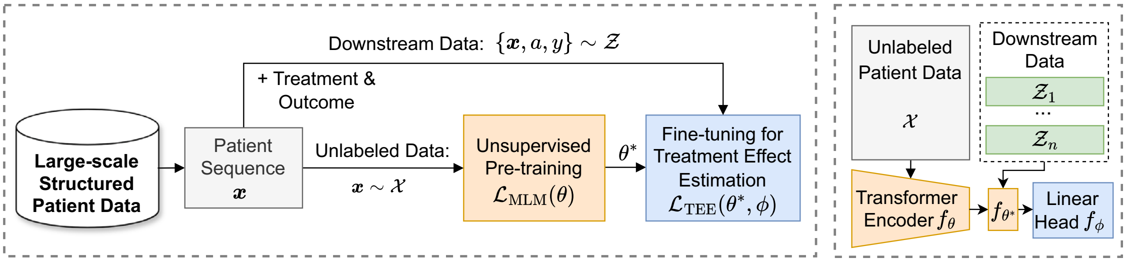

Downstream Data: $\{\boldsymbol{x}, a, y\} \sim \mathcal{Z}$
+ Treatment & Outcome
Large-scale Structured Patient Data
Patient Sequence $\boldsymbol{x}$
Unlabeled Data: $\boldsymbol{x} \sim \mathcal{X}$
Unsupervised Pre-training $\mathcal{L}_{\mathrm{MLM}}(\theta)$
θ^*
Fine-tuning for Treatment Effect Estimation $\mathcal{L}_{\mathrm{TEE}}(\theta^*, \phi)$
Unlabeled Patient Data $\mathcal{X}$
Downstream Data
$\mathcal{Z}_1$
...
$\mathcal{Z}_n$
Transformer Encoder f_θ
f_{θ^*}
Linear Head f_ϕ

Figure 1. The overall pipeline of CURE

It mainly consists of three parts: (1) data encoding of longitudinal patient data, (2) unsupervised pre-training on unlabeled data, and (3) fine-tuning on downstream labeled data for treatment effect estimation (TEE). In TEE, labels mean the studied treatment *a* and outcome *y* of the patient sequence $\boldsymbol{x}$.

data for capturing inherent complex relationships of the input data.

Recently, Transformer[14] has been widely adopted as a critical and unified building block in the pre-training and fine-tuning paradigm across data modalities. Pre-trained Transformer-based models (PTMs) have become the model of choice in many deep learning domains such as natural language processing (NLP)[15–19] and computer vision.[20–22] The dominant approach is to pre-train on a large-scale dataset with unsupervised or self-supervised learning and then fine-tune on a smaller task-specific dataset. Nonetheless, applying this pre-training and fine-tuning paradigm to TEE problems faces the following three major challenges: (1) encoding structured longitudinal observational patient data into sequence input, (2) lack of well-curated large-scale pre-training dataset, and (3) lack of real-world downstream TEE tasks to benchmark baselines.

In this article, we propose a new pre-training and fine-tuning framework for estimating the causal effect of a treatment: causal treatment effect estimation (CURE). As shown in Figure 1, the large-scale structured patient data are extracted from real-world medical claims data (MarketScan Research Databases[23]). We first encode the structured data as sequential input by chronologically flattening and aligning all observed covariates. We obtain around 3 M processed unlabeled patient sequences for pre-training. The downstream datasets with labeled treatment and outcome are created according to specific TEE tasks from established RCTs. Based on the retrospective study design and domain knowledge, we obtain four downstream tasks, each of them containing 10,000–20,000 patient samples. The task is to evaluate the comparative effectiveness of two treatment effects in reducing the risk of stroke for patients with coronary artery disease (CAD). Second, we pre-train a Transformer-based model on the unlabeled data with an unsupervised learning objective to generate contextualized patient representations. To accommodate the issues of complex hierarchical structure (i.e., each record encompasses multiple visits, with each visit comprising various types of medications or diagnoses) and irregularity of the observational patient data, we propose a comprehensive embedding method to incorporate the structure and time information. Finally, we fine-tune the pre-trained model on various downstream TEE tasks.

We are the first study to demonstrate the success of adopting the pre-training and fine-tuning framework to representation learning of patient data for TEE, together with necessary but minimal changes on the Transformer architecture design, and real-world case studies on RCTs. We summarize our main contributions as follows.

- We propose CURE, a new Transformer-based pre-training and fine-tuning framework for TEE. We present a new patient data encoding method to encode structured observational patient data and incorporate covariate type and time into patient embeddings.
- We obtain and pre-process large-scale patient data from real-world medical claims data as our pre-training resource. We derive four downstream TEE tasks according to study designs and domain knowledge from established RCTs for model evaluation.
- We conduct thorough experiments and show that CURE yields superior performance on all downstream tasks compared to state-of-the-art TEE methods. We achieve, on average, 4% and 7% absolute improvement in area under the receiver operating characteristic curve (AUC) and area under the precision-recall curve (AUPR), respectively, for outcome prediction and 8% absolute improvement in influence-function-based precision of estimating heterogeneous effects (IF-PEHE) for TEE over the best baseline among 4 tasks. We also verify the estimated treatment effects with the conclusion of corresponding RCTs.
- We further explore the effectiveness of CURE in several ablation studies including the proposed patient embedding, the influence of pre-training data size on downstream tasks, and the generalizability of low-resource fine-tuning data.

RESULTS

In this section, we evaluate the proposed CURE from three aspects: (1) quantitative analysis comparing the performance of the proposed method with state-of-the-art TEE methods on four downstream tasks, (2) qualitative analysis including the validation of the estimated treatment effects with corresponding RCTs and self-attention feature weights visualization, and (3) ablation studies including proposed feature embedding, pre-training data size, and generalizability of low-resource fine-tuning data.

Pre-training data

We obtain the pre-training data from MarketScan Commercial Claims and Encounters (CCAE)[23] from 2012 to 2017, which

Table 1. The statistics of four downstream datasets

Target vs. compared	Rivaroxaban vs. aspirin	Valsartan vs. ramipril	Ticagrelor vs. aspirin	Apixaban vs. warfarin
No. of patients (target; compared)	26,340 (9,569; 16,771)	12,850 (7,306; 5,544)	29,248 (12,477; 16,771)	18,187 (6,701; 11,486)
Female (%)	30.4	32.4	27.1	31.8
Age (group) on index date	55–64	55–64	55–64	55–64
Patients with stroke (%)	13.7	11.9	18.9	16.7
Average no. of visits per patient	83.4	74.0	70.7	97.1
Average no. of codes per patient	182.3	157.2	152.0	215.9

contains individual-level, de-identified healthcare claims information from employers, health plans, and hospitals. In this article, we include patients who have ever been diagnosed with CAD as our disease cohort. The definition of CAD is in Table S1. After conducting data pre-processing and study design, we obtain 2,955,399 patient sequences for pre-training. We obtain 9,435 medical codes including 282 diagnosis codes (i.e., we map the original ICD-9/10 billing codes into Clinical Classifications Software[24]) and 9,153 medication codes (i.e., we map medications based on generic names from RED BOOK[25]).

Downstream tasks

Given the absence of ground truth treatment effects in observational data, we employ RCTs as the gold standard to validate our findings. We focus on CAD-related RCTs that study the comparative effectiveness of two treatments for reducing the risk of stroke after CAD. We first collect all available phase 2 and phase 3 RCTs with CAD as the disease name and stroke as the disease outcome from https://clinicaltrials.gov/. Stroke is selected because it is commonly used as the primary outcome measurement in various CAD studies and is well-defined in observational data. The definition of stroke is provided in Table S2. Then, we select completed RCTs that study the treatment effect of two drugs with published results. Finally, we end up with 4 RCTs that meet all of the above criteria. We derive downstream tasks from our data to emulate the outcomes of the corresponding RCTs. The statistics of the downstream dataset are provided in Table 1. More details of screening RCTs are in Figure S3. Additional experimental results on a semi-synthetic dataset are provided in Table S10.

Baselines

We compare CURE with 8 NN models for TEE, including state-of-the-art methods. For models designed for continuous outcomes with mean-squared error (MSE) as a training objective, we change the objective function to binary cross-entropy (BCE) for consistency. All the baselines are only trained on downstream data and are summarized below.

- TARNet[5] estimates potential outcomes by generating balanced representations between treated and control groups, thus minimizing biases inherent in observational data.
- DragonNet[6] jointly optimizes treatment prediction and potential outcome prediction. By learning shared representations from input data, DragonNet employs a three-headed NN architecture to accomplish its predictive tasks: one head is dedicated to predicting treatment assignment, while the other two are tasked with estimating the potential outcomes.
- DR-CFR[7] learns disentangled representations for TEE. It operates under the premise that observed covariates can be disentangled into three distinct components: those exclusively influencing treatment assignment, those solely affecting outcome prediction, and those contributing to both.
- TNet[8] functions as a NN-based T-learner, aligning with the meta-learner framework[26] that breaks down TEE into dual sub-regression tasks. It utilizes two distinct neural models to estimate the potential outcomes.
- SNet[8] learns disentangled representations, positing that observed covariates can be segmented into five distinct components when accounting for the two potential outcomes independently.
- FlexTENet[27] integrates inductive bias regarding the shared structure of the two potential outcomes into TEE, flexibly determining the elements to be shared between the potential outcome functions.
- TransTEE[11] introduces a Transformer-based approach for TEE where covariates and treatments are encoded using a Transformer architecture and a cross-attention mechanism for adjusting confounding bias.
- Base Model directly trains on the downstream datasets using the same architecture as CURE**.**

Metrics

We evaluate the performance of factual predictions employing the metrics of the AUC and AUPR. For counterfactual prediction performance, we utilize the IF-PEHE.[28] This metric facilitates the benchmarking of TEE methods in scenarios where the ground truth effects are unavailable. Unlike the conventional PEHE, which quantifies the MSE between estimated and true treatment effects, the IF-PEHE assesses the MSE between estimated treatment effects and their approximated true counterparts. A numeric output is produced by the IF-PEHE metric, with lower values indicating superior performance. A comprehensive discussion on this metric is presented in supplemental experimental procedures 4.

Implementation details

Our pre-training uses the BERT$_{base}$ architecture[15] with 768 hidden sizes, 12 attention heads, a 12 layer Transformer, and 3,072 intermediate sizes (see Figure S2 for detailed model configuration). The maximum input sequence length is 256.

Table 2. Overall performance comparison with baseline TEE methods across four datasets

Method	Rivaroxaban vs. aspirin			Valsartan vs. ramipril		
	AUC ↑	AUPR ↑	IF-PEHE ↓	AUC ↑	AUPR ↑	IF-PEHE ↓
TARNet	0.719 ± 0.015	0.327 ± 0.023	0.286 ± 0.044	0.683 ± 0.028	0.263 ± 0.029	0.284 ± 0.061
DragonNet	0.757 ± 0.013	0.381 ± 0.023	0.275 ± 0.052	0.683 ± 0.026	0.263 ± 0.027	0.278 ± 0.049
DR-CFR	0.759 ± 0.015	0.381 ± 0.026	0.253 ± 0.038	0.751 ± 0.019	0.333 ± 0.032	0.275 ± 0.035
TNet	0.715 ± 0.016	0.318 ± 0.028	0.290 ± 0.059	0.673 ± 0.021	0.256 ± 0.024	0.294 ± 0.065
SNet	0.756 ± 0.014	0.380 ± 0.028	0.247 ± 0.054	0.752 ± 0.022	0.333 ± 0.033	0.269 ± 0.057
FlexTENet	0.717 ± 0.014	0.319 ± 0.022	0.265 ± 0.054	0.662 ± 0.028	0.241 ± 0.027	0.282 ± 0.038
TransTEE	0.750 ± 0.011	0.379 ± 0.021	0.257 ± 0.042	0.773 ± 0.013	0.380 ± 0.025	0.261 ± 0.049
Base Model	0.758 ± 0.029	0.406 ± 0.037	0.180 ± 0.038	0.780 ± 0.050	0.365 ± 0.066	0.190 ± 0.065
CURE	0.803 ± 0.011	0.469 ± 0.023	0.173 ± 0.038	0.811 ± 0.018	0.428 ± 0.041	0.158 ± 0.062
Method	Ticagrelor vs. aspirin			Apixaban vs. warfarin		
	AUC ↑	AUPR ↑	IF-PEHE ↓	AUC ↑	AUPR ↑	IF-PEHE ↓
TARNet	0.714 ± 0.008	0.359 ± 0.016	0.311 ± 0.048	0.748 ± 0.012	0.447 ± 0.030	0.320 ± 0.043
DragonNet	0.741 ± 0.009	0.397 ± 0.020	0.309 ± 0.056	0.792 ± 0.018	0.519 ± 0.035	0.318 ± 0.055
DR-CFR	0.745 ± 0.007	0.403 ± 0.021	0.305 ± 0.048	0.798 ± 0.015	0.531 ± 0.032	0.311 ± 0.043
TNet	0.709 ± 0.009	0.360 ± 0.020	0.315 ± 0.061	0.741 ± 0.015	0.432 ± 0.032	0.322 ± 0.039
SNet	0.742 ± 0.008	0.400 ± 0.020	0.298 ± 0.053	0.795 ± 0.014	0.525 ± 0.034	0.309 ± 0.054
FlexTENet	0.710 ± 0.010	0.351 ± 0.015	0.312 ± 0.046	0.735 ± 0.012	0.413 ± 0.033	0.328 ± 0.030
TransTEE	0.747 ± 0.022	0.385 ± 0.015	0.288 ± 0.021	0.799 ± 0.011	0.517 ± 0.031	0.305 ± 0.059
Base Model	0.751 ± 0.025	0.425 ± 0.04	0.246 ± 0.031	0.791 ± 0.029	0.539 ± 0.039	0.251 ± 0.045
CURE	0.793 ± 0.008	0.489 ± 0.024	0.198 ± 0.068	0.826 ± 0.014	0.588 ± 0.024	0.224 ± 0.066

The results are the average and standard deviation over 20 runs. CURE outperforms all other TEE methods.

The pre-training phase is executed on three NVIDIA GeForce RTX 2080 Ti 11GB GPUs utilizing a batch size of 96. We train our model using the adaptive moment estimation (Adam) optimizer with an initial learning rate of $1e-4$ and learning rate warmup in the first 10% training steps. During the fine-tuning, the learning rate is $5e-5$ without the learning rate warmup. Fine-tuning is carried out across each task for 2 epochs. The data for downstream tasks are randomly divided into training, validation, and test sets with allocations of 90%, 5%, and 5%, respectively. Performance metrics have been exclusively reported based on the test sets. The optimal hyperparameters of pre-training and fine-tuning are shown in Tables S3 and S4. The comparison of the trade-off between the efficacy and efficiency of all methods is in Table S5.

Quantitative analysis

Comparison with state-of-the-art methods

Table 2 shows the performance of factual outcome prediction (measured by AUC and AUPR) and TEE (measured by IF-PEHE) on four different downstream tasks. We compare CURE with the state-of-the-art TEE methods and report the results under 20 random runs. We observe that the proposed CURE has more than 4%, 7%, and 8% respective average AUC, AUPR, and IF-PEHE improvement over the best baseline on these tasks. The results illustrate the promise and effectiveness of our proposed pre-training and fine-tuning methodology for TEE. Notably, even without pre-training, the base model of CURE attains a similar performance to the best baseline, which suggests the effectiveness of our architecture and data encoding designs. The results of the training and validation sets are in Table S6. The results of additional evaluation metrics are in Table S7. The model performance of addressing the treatment selection bias is demonstrated in Figure S7 and Table S8.

Qualitative analysis

Validate with RCT conclusion

As the ground truth treatment effects are not available in observational data, we further evaluate the estimated treatment effects with corresponding ground truth RCTs. In Table 3, we show the confidence intervals of estimated effects under 20 runs and RCT conclusions of each downstream task.

We use the direct difference to estimate the treatment effects.[33] The results can be interpreted as two potential conclusions: (1) the target treatment is significantly more effective than the compared treatment in reducing the risk of the outcome if the upper bound of the confidence interval is lower than zero. (2) The target is not significantly more effective than the compared treatment if the confidence interval covers zero (i.e., no significant difference) or the lower bound is higher than zero (i.e., the compared treatment is more effective than the target treatment). As we can see, our estimated treatment effects are mostly consistent with each corresponding RCT conclusion. Though the generated hypothesis and RCT conclusion are not exactly the same for the third pair (ticagrelor vs. aspirin), they both indicate that there is no significant reduced treatment effect of the target treatment over the compared treatment. The results demonstrate that our proposed CURE successfully identifies correct treatment effects using only observational patient data.

We also verify the estimated treatment effects of all the baselines with RCT conclusions. As shown in Table 4, our method

Table 3. Evaluation of CURE's estimated effects vs. ground truth conclusions from RCTs

Target vs. compared	Estimated effect (CI)	*p* value	Generated hypothesis	RCT conclusion
Rivaroxaban vs. aspirin	[−0.009, 0.006]	0.452	no significant difference	no significant difference[29]
Valsartan vs. ramipril	[−0.003, 0.014]	0.103	no significant difference	no significant difference[30]
Ticagrelor vs. aspirin	[0.022, 0.040]	6e−14	ticagrelor is less effective than aspirin	no significant difference[31]
Apixaban vs. warfarin	[−0.039, −0.002]	4e−4	apixaban is more effective than warfarin	apixaban is more effective than warfarin[32]

Estimated effects are presented within 95% confidence intervals (CI) derived from 20 bootstrap iterations. Conclusions from RCTs are referenced from peer-reviewed publications.

correctly generates 3 (out of 4) RCT conclusions that match the ground truth RCT conclusions, while the best baselines only identify 2 (out of 4) RCT conclusions. Our study demonstrates that pre-training is more effective in identifying trial conclusions than the baselines. Additional comparison results under 100 random bootstraps are provided in Table S9.

Self-attention visualization

The self-attention mechanism of the Transformer enables the exploration of interaction among input covariates and provides a potential interpretation of the prediction results. We use a Transformer visualization tool called bertvi[34] to help visualize learned attention weights. We show the attention weights of a patient from the apixaban treatment group of apixaban vs. warfarin study in Figure 2. Different colors denote the attention heads, and there are 12 heads in total. The medications and diagnosis codes highlighted in the figure are the most related features to the outcome prediction and TEE. For example, amiodarone is an antiarrhythmic medication used to treat and prevent a number of types of cardiac dysrhythmias including atrial fibrillation.[35] A study[36] shows that apixaban is superior to warfarin in preventing stroke in patients with atrial fibrillation. Those attention weights could be used to analyze the treatment effects in some sub-groups that are characterized by the attended feature set.

Ablation studies

Effect of embedding layer

We evaluate the effect of proposed time embedding and type embedding, respectively. As shown in Figure 3, the model with both time and type embedding generally performs better than the other two ablations. Especially, incorporating time embedding yields larger performance improvement than the type embedding. This indicates that the proposed embedding method is better than the standard embedding method and that time information plays a more important role in TEE than the type information. A more fine-grained analysis of the effect of time embeddings is shown in Figure S4.

Effect of downstream data size

We demonstrate the model's effectiveness on the limited downstream data in Figure 4. The plots show the model performance with different fractions of labeled downstream data. Generally,

Table 4. Comparison of the estimated effects by all methods against ground truth from RCTs

	Rivaroxaban vs. aspirin			Valsartan vs. ramipril		
Method	Estimated effect (CI)	*p* value	Match RCT conclusion?	Estimated effect (CI)	*p* value	Match RCT conclusion?
TARNet	[0.066, 0.095]	5.678e−10	no	[−0.037, −0.003]	0.026	no
DragonNet	[0.18, 0.236]	5.979e−12	no	[0.03, 0.07]	4.681e−5	no
DR-CFR	[0.13, 0.183]	2.783e−10	no	[0.002, 0.04]	0.033	no
TNet	[0.041, 0.07]	2.509e−7	no	[−0.038, −0.001]	0.039	no
SNet	[−0.002, 0.008]	0.231	yes	[−0.051, −0.026]	3.168e−6	no
FlexTENet	[0.064, 0.108]	1.529e−7	no	[−0.079, −0.035]	3.184e−5	no
TransTEE	[−0.013, −0.002]	0.018	no	[−0.019, 0.034]	0.420	yes
CURE	[−0.009, 0.006]	0.452	yes	[−0.003, 0.014]	0.103	yes
Method	Ticagrelor vs. aspirin			Apixaban vs. warfarin		
	Estimated effect (CI)	*p* value	Match RCT conclusion?	Estimated effect (CI)	*p* value	Match RCT conclusion?
TARNet	[0.064, 0.101]	2.861e−8	no	[−0.006, 0.028]	0.207	no
DragonNet	[−0.013, 0.01]	0.821	yes	[0.018, 0.056]	6.284e−4	no
DR-CFR	[−0.068, −0.029]	4.915e−5	no	[−0.026, −0.002]	0.047	yes
TNet	[0.046, 0.069]	6.474e−9	no	[0.009, 0.023]	2.329e−4	no
SNet	[0.005, 0.016]	4.398e−4	no	[−0.046, −0.017]	2.112e−4	yes
FlexTENet	[0.045, 0.068]	5.243e−9	no	[0.012, 0.042]	0.001	no
TransTEE	[−0.014, −0.009]	0.0216	no	[−0.027, −0.002]	0.027	yes
CURE	[0.022, 0.040]	6e−14	no	[−0.039, −0.002]	4e−4	yes

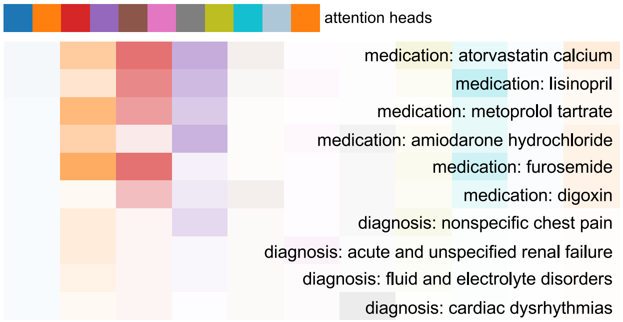


Figure 2. The visualization of the top 10 attention weights associated with the special token [CLS] of a patient from the apixaban treatment group

Colors identify the corresponding attention heads, and color intensity reflects the attention score.

given only 5%–10% labeled data, CURE achieves comparable performance to the Base Model, which is trained on the fully labeled data. Specifically, the performance gains are large when given a small fraction of labeled data (1%–5%), and the curve tends to gently increase after the fraction is larger than 10%. With increased data size, the performance gradually achieves the upper bound of fine-tuning on fully labeled data. The results demonstrate that unsupervised pre-training benefits low-resource downstream tasks even when only a limited number of labeled data are available for fine-tuning. Additional results of AUPR scores are shown in Figure S5.

Effect of pre-training data size

We further explore the effect of pre-training data volume on the performance of downstream tasks. In Figure 5, we show the AUC given different fractions of pre-training data. Here, the 0% training set size denotes the Base Model, which is trained on the downstream data from scratch. Generally, the performance improves with the increase of pre-train data. The results indicate that pre-training is beneficial for downstream tasks by learning contextualized patient representations from large-scale unlabeled patient data. Additional results of AUPR scores are shown in Figure S6.

DISCUSSION

In this paper, we study the problem of TEE from observational data. We propose a new Transformer-based TEE framework called CURE, which adopts the pre-training and fine-tuning paradigm. CURE is pre-trained on large-scale unlabeled patient data and then fine-tuned on labeled patient data for TEE. We convert the structured patient data into sequence and design a new sequence encoding method to encode the structure and time into a comprehensive patient embedding. Thorough experiments show that pre-training significantly boosts the TEE performance on 4 downstream tasks compared to state-of-the-art methods. We further demonstrate the data scalability of CURE and verify the results with corresponding published RCTs. One promising application of our model is to help generate useful hypotheses of treatment effects and serve as a complementary tool to standard RCTs, e.g., exploring new uses of existing drugs. Future work could be done to improve the model performance by engaging more patient data from diverse disease cohorts as pre-training data.

Deep learning for TEE

Generally, existing NN-based methods formulate the TEE as several regression tasks (i.e., regression on potential outcomes and treatment) with different levels of information shared among the nuisance estimation tasks using representation learning.[37] TARNet,[8] for example, learns shared representations for two potential outcomes, while SNet[8] learns five different representations on the combinations of treatment and potential outcomes. Recently, Transformer has been introduced as an encoder block for TEE[9,11] and yields better performance than the state-of-the-art methods. Despite the promising results, the main limitation is that the model performance can be diminished if the labeled dataset is limited. The model trained for one particular problem or data may fail to generalize to other scenarios.

Pre-training and fine-tuning of Transformer

Since Transformer is based on a flexible architecture with few assumptions on the input data structure, it is difficult to directly train the model on small-scale data. Therefore, various PTMs are first pre-trained on the large-scale unlabeled data and then fine-tuned for labeled tasks at hand. PTMs learn universal and contextualized representations, which can boost various downstream tasks and avoid developing and training a new model from scratch. Among the existing PTMs in NLP, BERT[15] is one of the most popular models. BERT[15] is pre-trained on large-scale unlabeled corpus via self-supervised pre-training tasks (i.e., masked language modeling [MLM] and next sentence

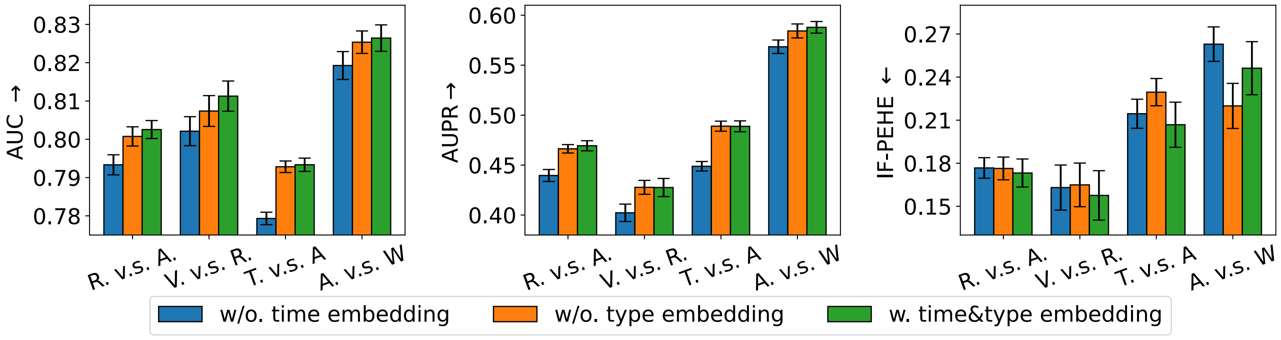


Figure 3. The effect of different embedding layer designs on four downstream tasks. The error bars represent the standard error of the mean (SEM).

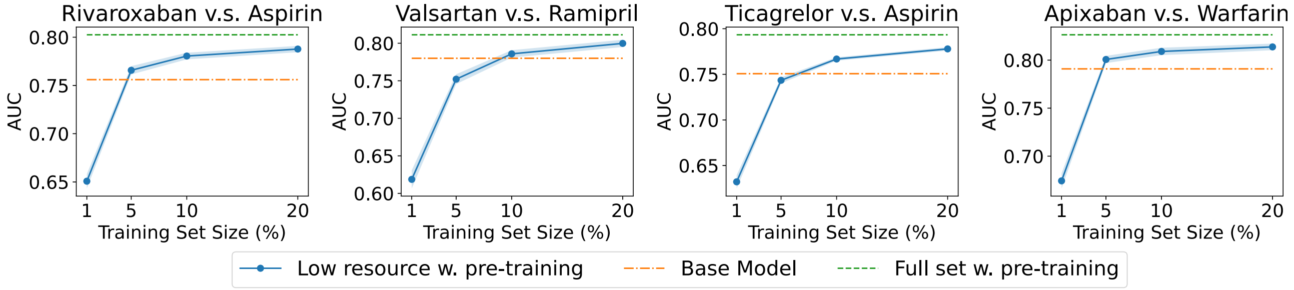


Figure 4. The effect of limited resources in fine-tuning datasets on performance across four downstream tasks with varied proportions of the labeled training set (x axes)

prediction) and fine-tuned on downstream tasks with an additional linear head. The patient data are similar to natural language text, as they both contain sequential information. However, patient data, with their hierarchical structure and irregular time information, pose distinct challenges compared to standard textual data. To address these specific challenges effectively, we proposed a new patient data encoding method. This method is designed to fully accommodate the unique characteristics of patient data, enabling a more comprehensive understanding and interpretation of such information.

Comparison with existing pre-trained models using patient data

Recently, several pre-trained models, such as BEHRT[38] and Med-BERT,[39] have been developed for clinical risk prediction using patient data. BEHRT, for instance, is a Transformer model directly adapted from the BERT architecture to encode patient visits for disease prediction. Similarly, Med-BERT, another Transformer model, incorporates customized pre-training objectives tailored for disease prediction problems. Despite these advancements, most existing models tend to employ the standard BERT encoding method or only introduce minimal modifications when applied to patient data. This approach can be inadequate, as it may not fully capture the complex hierarchical relationships and temporal irregularities inherent in patient data. This limitation is evident in Table 5, where our proposed patient encoding method in CURE demonstrates superior performance by effectively addressing these unique challenges of patient data representation. More detailed comparisons of our method and existing work are provided in Tables S11 and S12.

Real-world data selection and model generalization

The dataset employed in our study, obtained from MarketScan,[23] was selected for its comprehensive scope, encompassing approximately 3 M patient medical sequences, which is crucial for capturing a broad spectrum of heart disease manifestations. Additionally, we developed a semi-synthetic dataset, which contains ground truths of treatment effects, to facilitate the evaluation of our model (refer to Table S10). In terms of generalization, our approach can be applied to datasets comparable in scale and content to MarketScan, such as Cosmos,[40] Optum,[41] and IQVIA.[42] These datasets share similar characteristics to MarketScan, such as information on medications, diagnoses, and demographics. To support the adaptation of our methodologies by other researchers, we have detailed the MarketScan data structure on GitHub (https://github.com/ruoqi-liu/CURE).

Heterogeneity of large real-world data

The potential heterogeneity in care protocols across different healthcare institutions might impact the model's performance. Such variability introduces confounding factors, given that patients may receive care that adheres to differing standards. Unfortunately, MarketScan data do not contain institute-level information, which precludes a direct analysis of how specific care protocols might influence treatment outcomes. To account for this, our approach includes a comprehensive analysis of a wide array of covariates—totaling 9,435—including historical medication use, diagnoses, and demographic factors. These covariates are selected to broadly account for potential confounders that might influence treatment outcomes, serving as proxies for the unavailable institute-level data. Furthermore, if available, the institute-level information (e.g., hospital locations, care protocols, etc.) can be easily integrated into our model to refine our findings.

Class imbalance in downstream data

The downstream datasets do not exhibit a high degree of class imbalance. The percentage of positive labels across all datasets

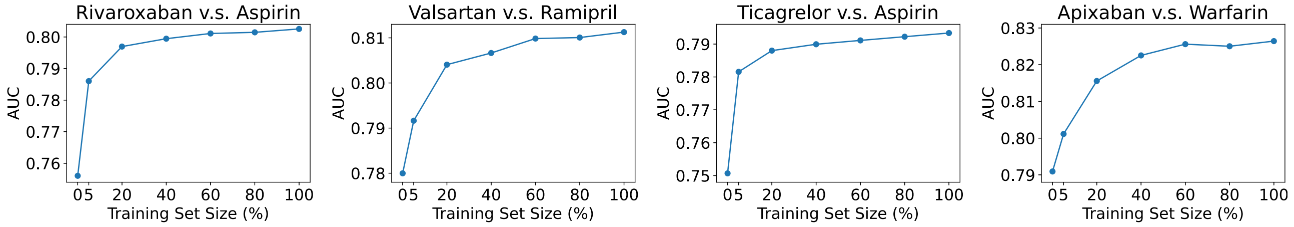


Figure 5. The effect of pre-training data volume on four downstream tasks (average of 20 runs)

Table 5. Comparison with existing pre-trained models on four downstream datasets

Method	Rivaroxaban vs. aspirin			Valsartan vs. ramipril		
	AUC ↑	AUPR ↑	IF-PEHE ↓	AUC ↑	AUPR ↑	IF-PEHE ↓
BEHRT	0.765 ± 0.013	0.424 ± 0.018	0.179 ± 0.031	0.784 ± 0.017	0.394 ± 0.021	0.181 ± 0.024
Med-BERT	0.771 ± 0.012	0.432 ± 0.022	0.176 ± 0.027	0.789 ± 0.015	0.401 ± 0.029	0.177 ± 0.034
CURE	0.803 ± 0.011	0.469 ± 0.023	0.173 ± 0.038	0.811 ± 0.018	0.428 ± 0.041	0.158 ± 0.062
Method	Ticagrelor vs. aspirin			Apixaban vs. warfarin		
	AUC ↑	AUPR ↑	IF-PEHE ↓	AUC ↑	AUPR ↑	IF-PEHE ↓
BEHRT	0.755 ± 0.017	0.433 ± 0.012	0.232 ± 0.018	0.796 ± 0.014	0.554 ± 0.019	0.246 ± 0.021
Med-BERT	0.761 ± 0.015	0.447 ± 0.014	0.230 ± 0.027	0.802 ± 0.011	0.561 ± 0.022	0.242 ± 0.031
CURE	0.793 ± 0.008	0.489 ± 0.024	0.198 ± 0.068	0.826 ± 0.014	0.588 ± 0.024	0.224 ± 0.066

The results are the average and standard deviation over 20 runs. CURE outperforms all other patient data encoding methods.

ranges from 12% to 19%. Notably, even in the dataset with the lowest proportion of positive instances (11.9%), a substantial count of 1,529 positive instances was observed. Recognizing the importance of selecting an appropriate evaluation metric for imbalanced data, we opted for the AUPR. AUPR is especially advantageous in scenarios of class imbalance, as it emphasizes the trade-off between precision and recall, a critical consideration when the positive class is underrepresented. This metric's suitability is underscored by our findings, where our model demonstrated consistent superiority over state-of-the-art models across various datasets. Specifically, our model exhibited an average improvement of 7% in AUPR, including a notable 4% enhancement in performance on the dataset with the lowest percentage of positive labels, compared to the best baseline.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Dr. Ping Zhang (zhang.10631@osu.edu).

Materials availability

This study did not generate any new materials.

Data and code availability

The data we use are from MarketScan CCAE (more than 100 M patients, from 2012 to 2017). Access to the MarketScan data analyzed in this manuscript is provided by The Ohio State University. The dataset is available at https://www.merative.com/real-world-evidence. Our source code is available at GitHub (https://github.com/ruoqi-liu/CURE) and has been archived at Zenodo.[43]

Methods

TEE from observational data

In this article, we are interested in observational patient data. Each patient sample consists of pre-treatment covariates $\boldsymbol{x}$ (i.e., historical co-medication, co-morbidities, and demographics) and treatment a of interest. Following the potential outcome framework,[44] the potential outcome y_a is defined as the response to treatment a out of all available treatment options. Typically, we consider the comparative treatment effects of two treatments and denote two potential outcomes as y_1 and y_0 for simplicity.

We aim to estimate the individual-level treatment effect (ITE) as the difference between the potential outcomes under two treatment arms as $y_1(\boldsymbol{x}) - y_0(\boldsymbol{x})$. We are also interested in the average treatment effect (ATE), which is the average effect among the entire population, denoted as $\mathbb{E}[y_1(\boldsymbol{x}) - y_0(\boldsymbol{x})]$. In observational data, only one of the potential outcomes is available and the remaining counterfactual outcomes are missing in nature, which makes this task more difficult than classical supervised learning. We follow the standard assumptions[45] (i.e., consistency, positivity, and strong ignorability). The potential outcome can be defined as $y_a(\boldsymbol{x}) = \mathbb{E}[y|a,\boldsymbol{x}]$ and can be estimated from observational data. More details of assumptions and analysis of propensity are illustrated in supplemental experimental procedures 1 and Figure S1.

Encoding structured patient data

In this work, we focus on longitudinal observational patient data. We first introduce the data for pre-training and fine-tuning, respectively. Then, we illustrate how to convert structured patient data into sequential input for the Transformer encoder.

Pre-train data structure

The pre-training is based on large-scale unlabeled patient data. Here, to distinguish from downstream data, we denote the pre-train data as unlabeled data ($\boldsymbol{x} \sim \mathcal{X}$), while downstream data with treatment a and outcome y are denoted as labeled data ($\{\boldsymbol{x}, a, y\} \sim \mathcal{Z}$). The unlabeled patient data consist of (1) co-medication $m_1, m_2, \ldots, m_{|\mathcal{M}|} \in \mathcal{M}$, where $|\mathcal{M}|$ is the number of unique medication names; (2) co-morbidities $d_1, d_2, \ldots, d_{|\mathcal{D}|} \in \mathcal{D}$, where $|\mathcal{D}|$ is the number of

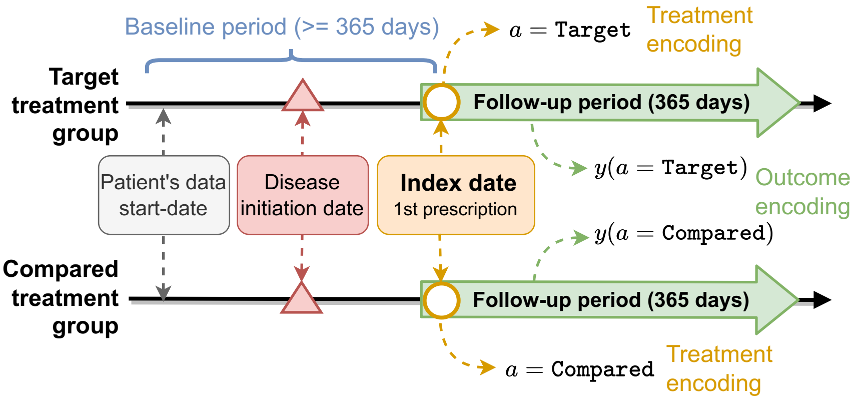


Figure 6. Illustration of the downstream data construction with retrospective study design

Index date refers to the first prescription of the target treatment or the compared treatment, which should not be prior to the disease initiation date. The baseline period is no less than 1 year, and the follow-up period as an outcome observation is also 1 year. The treatments of interests and outcomes are obtained at the index date and follow-up period, respectively.

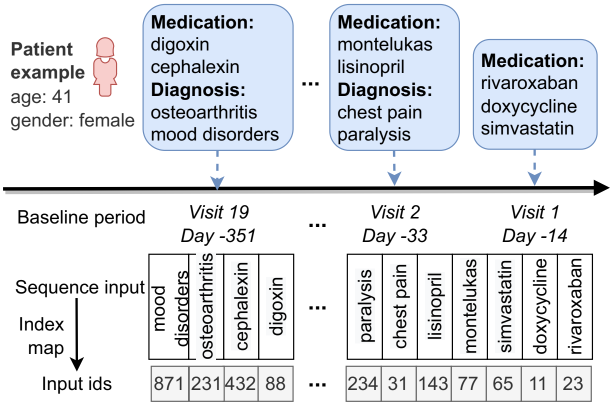


Figure 7. Illustration of encoding structured patient data into sequential input

The patient data are recorded in a hierarchical structure such that a patient contains multiple visits and each visit contains multiple medications and diagnoses. The structured data are converted into a sequence by flattening all covariates in each visit and aligning them chronologically.

unique diagnosis codes, and (3) demographics c, with age encoded as a categorical value and gender encoded as a binary value. A patient can have multiple visits $\{v_1, \ldots, v_T\}$, where each of visit v_t contains a subset of medication and diagnosis codes ($v_t \in \mathcal{M} \cup \mathcal{D}$). We denote the unlabeled patient data as $\boldsymbol{x} = \{c, \{v_t\}_{t=1}^T\}$, and all the covariates are obtained from the baseline period as shown in Figure 6. We build a medical vocabulary from all patient covariates as $\mathcal{V} = \{\mathcal{M}, \mathcal{D}, c\}$.

Fine-tuning data structure

The fine-tuning is based on small-scale labeled patient data, which are not used for pre-training. Besides the co-medication, co-morbidities, and demographics, the labeled patient data contain treatment $a \in \mathcal{M}_{\text{task}}$ (i.e., can be either the target treatment or compared treatment from task-specific medication group $\mathcal{M}_{\text{task}}$) and outcome $y_a \in \{0,1\}$ under the observed treatment a. In Figure 5, we show the retrospective study design of how to construct downstream data and obtain labels for treatments and outcomes. In particular, we collect patient data from two different treatment groups for comparison. For each group of patients, the covariates (i.e., co-medication, co-morbidities, and demographics) are obtained from the baseline period (also known as pre-treatment covariates) as potential confounders, and the outcomes are obtained from the follow-up period. More illustrations of the study design can be found in supplemental experimental procedures 4.

Structured patient data to sequential input

As introduced above, the original patient data are recorded naturally in a hierarchical structure. Unlike natural language text, which is inherently encoded as a sequence of words, the patient data need to be pre-processed into a "sequence-like" format before being sent to the Transformer encoder. As shown in Figure 7, we flatten the structured patient data by chronologically going through each medication and diagnosis in each visit and aligning them in one sequence. Each medication or diagnosis is encoded as an individual token, which is comparable to text tokenization. The token IDs are obtained from the medical vocabulary $\mathcal{V}$.

Pre-training CURE

As shown in Figure 8, the pre-training consists of three modules: (1) an embedding layer to convert input patient data into embedding representations, (2) Transformer encoders to generate contextualized hidden representations, and (3) a final project layer for the pre-training objective. More formally, given the encoded patient sequence $\boldsymbol{x} = [x_1, \ldots, x_m, \ldots, x_T]$, the pre-training procedure can be decomposed into the following steps:

$$\boldsymbol{x} \xrightarrow{\text{Mask}} [x_1, \ldots, [MASK]_m, \ldots, x_T] \xrightarrow{\text{Embedding}} \{\boldsymbol{e}_i\}_{i=1}^{T} \xrightarrow{f_\theta} \{\boldsymbol{h}_i\}_{i=1}^{T} \xrightarrow{\text{MLM}} \mathcal{L}_{\text{MLM}}(\theta) \qquad \text{(Equation 1)}$$

We randomly replace 15% of input tokens with special [MASK] tokens, e.g., token x_m in the sequence. $\boldsymbol{e}_i \in \mathbb{R}^B$ denotes the embedding representation with embedding dimension B generated by the comprehensive embedding layer. $\boldsymbol{h}_i \in \mathbb{R}^H$ denotes the contextualized representation with hidden dimension H generated by Transformer encoder f_θ. The MLM[15] aims to predict the masked tokens x_m from the established vocabulary $\mathcal{V}$ using hidden representation $\boldsymbol{h}_m$. The pre-training loss function of MLM is denoted as $\mathcal{L}_{\text{MLM}}(\theta)$ with optimization parameters θ.

Comprehensive embedding layer

Pre-trained language models like BERT[15] have achieved great success in natural language text, demonstrating strong power in modeling sequential data. Although longitudinal patient data can be considered sequential data when organized chronologically, significant and unignorable differences exist between text and patient data. It is hard to directly apply the existing pre-trained language model to our unique patient data. Our ablation study shows that the standard embedding design adopted in NLP (i.e., token embedding and position embedding) will be sub-optimal in our scenario (see ablation studies for more details).

Compared to the natural language text, (1) longitudinal patient data contain a more complex hierarchical structure than the text data: a patient record contains a number of visits, and each visit also contains a number of different types of medical codes (i.e., medication or diagnosis). (2) The patient data are irregularly sampled (i.e., the time interval among visits is not regular), while the text data are regularly organized. As shown in Figure 7, the visit dates are not regularly distributed along the time: the first visit happened on day 0, the second visit happened on day 14, the third visit on day 33, etc. On visit 2 (day 33), the patient received two types of codes: montelukast and lisinopril as medications and chest pain and paralysis as diagnoses.

To accommodate the above issues of complex hierarchical structure and irregularity of the observational patient data, we propose a more comprehensive embedding layer than the original BERT[15] embedding layer by including associated code-type information and time information. For each input token, the patient embedding $\boldsymbol{e}_i$ is obtained as

$$\boldsymbol{e}_i = \boldsymbol{w}_{token} + \boldsymbol{t}_{type} + \boldsymbol{v}_{visit} + \boldsymbol{p}_{physical} \qquad \text{(Equation 2)}$$

where $\boldsymbol{w}_{token}$ is the original input token embedding and $\boldsymbol{t}_{type}$ is the type embedding of the input token. According to our data, there are three types in total: {demographics, medication, diagnosis}. The visit time embedding $\boldsymbol{v}_{visit}$ is the visit time corresponding to a visit. The physical time embedding $\boldsymbol{p}_{physical}$ is the physical time associated with the visit. Here, the physical time is measured by month (i.e., 30 day fixed window). Both visit and physical times are organized relative to the treatment index date (i.e., the absolute distance between the visit/physical time and the index date).

For example, Figure 7 illustrates an input sequence of patient data, including type and timing information: rivaroxaban falls under the medication type, prescribed during visit 1 (day 14), and chest pain is categorized under the diagnosis type, recorded during visit 2 (day 33). The input token embedding, time embedding, and type embedding are integrated and used as the input to the Transformer encoder.

Transformer encoder and pre-training objective

We use an N-stacked Transformer as our encoding backbone as it has been a widely adopted architecture. Each Transformer encoder block comprises a multihead self-attention layer succeeded by a fully connected feedforward layer.[14] Further information on the Transformer architecture can be found in supplemental experimental procedures 3.

The Transformer encoder f_θ takes the comprehensive embedding representations as input and generates contextualized hidden representations as $f_\theta(\boldsymbol{e})$. Given unlabeled patient data $\mathcal{X}$, the pre-training is to minimize the MLM loss of predicting the masked token with position $j \in \mathcal{J}$ using the input token embedding and hidden representation:

$$\mathcal{L}_{\text{MLM}}(\theta) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{X}}\left[-\sum_{j \in \mathcal{J}} \log(\mathrm{P}(\boldsymbol{w}_j | f_\theta(\boldsymbol{e}))) \right] \qquad \text{(Equation 3)}$$

where $\mathrm{P}(\boldsymbol{w}_j | f_\theta(\boldsymbol{e}))$ is the softmax probability of the masked token over all tokens in the vocabulary.

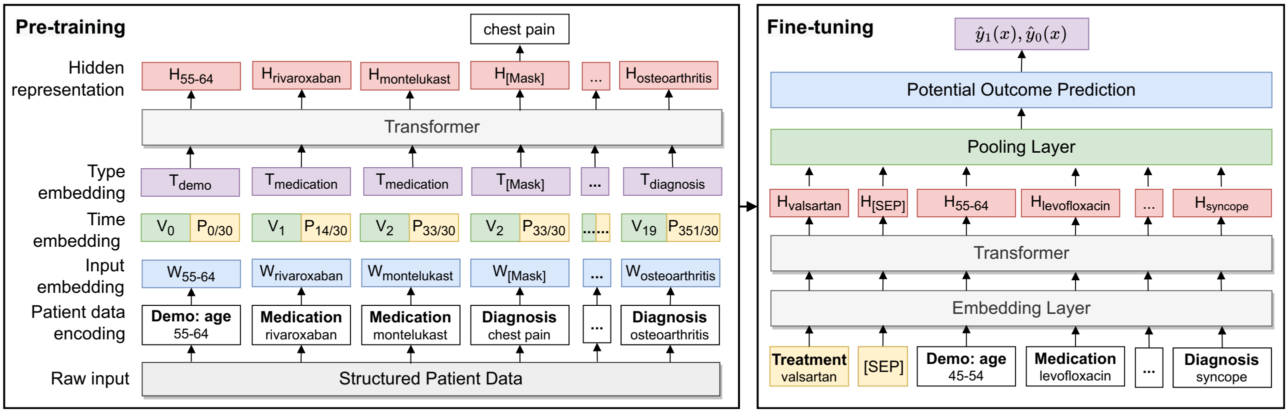


Figure 8. Illustration of pre-training and fine-tuning of CURE

The unlabeled structured patient data are first converted into a sequential input and processed for the embedding layer and encoder. During the fine-tuning, the treatments of interest are appended for potential outcome prediction.

Fine-tuning CURE for TEE

Given downstream labeled data $\{\boldsymbol{x}, a, y\} \sim \mathcal{Z}$, we fine-tune the model on different downstream TEE tasks. Here, we are interested in the comparative causal treatment effect of the target treatment over another compared treatment according to the downstream tasks. For each task, we plug in the task-specific input and outputs into CURE. We add a linear head f_φ to the hidden representations learned from the pre-training stage. We fully fine-tune all model parameters end to end by jointly updating θ^* obtained from optimizing Equation 3 and a randomized φ.

Specifically, we append the original input sequence with the index treatment (i.e., target treatment or compared treatment), which is separated by the special [SEP] token. As shown in Figure 8, the treatment valsartan is appended to the original inputs to indicate that the patient is from the treatment group of valsartan. The model processes the new inputs through the embedding layer and the Transformer encoder with parameters initialized with θ^*. We use the final hidden vector corresponding to the first input token ([CLS]) as the pooled representation $\boldsymbol{h}_{[CLS]}$ from the pooling layer. We predict the potential outcomes under the treatment a via the linear head as $f_\varphi \circ f_{\theta^*}(\boldsymbol{h}_{[CLS]}(a))$. The fine-tuning objective is the BCE of the potential outcome prediction:

$$\mathcal{L}_{\text{TEE}}(\theta^*, \varphi) = \mathbb{E}_{\{\boldsymbol{x},a,y\}\sim\mathcal{Z}}\left[\text{BCE}\left(f_\varphi \circ f_{\theta^*}\left(\boldsymbol{h}_{[CLS]}(a)\right), y\right)\right] \qquad \text{(Equation 4)}$$

Here, only the factual outcomes are used for training loss computation, as the counterfactual outcomes are unavailable in the observational data. After model fine-tuning, we infer the ITE δ and ATE Δ as the difference between two predicted potential outcomes under the target and compared treatment:

$$\begin{aligned} \hat{\delta} &= \hat{y}_{a\,=\,\text{Target}} - \hat{y}_{a\,=\,\text{Compared}} \\ \hat{\Delta} &= \mathbb{E}\left[\hat{y}_{a\,=\,\text{Target}} - \hat{y}_{a\,=\,\text{Compared}}\right] \end{aligned} \qquad \text{(Equation 5)}$$

Ethical consideration

The observational data used in the paper are from the MarketScan Research Database,[23] which is fully Health HIPAA compliant de-identified and has very minimal risk of the potential for loss of privacy. Our research protocol has been determined by The Office of Responsible Research Practices at The Ohio State University for IRB exemption under the study 2023E0357.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2024.100973.

ACKNOWLEDGMENTS

This work was funded in part by the National Institutes of Health (NIH) under award number R01GM141279. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

AUTHOR CONTRIBUTIONS

P.Z. conceived the project. R.L., P.-Y.C., and P.Z. developed the method. R.L. conducted the experiments. R.L., P.-Y.C., and P.Z. analyzed the results. R.L., P.-Y.C., and P.Z. wrote the manuscript. All authors read and approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: November 13, 2023
Revised: February 17, 2024
Accepted: April 3, 2024
Published: May 1, 2024

REFERENCES

1. Glass, T.A., Goodman, S.N., Hernán, M.A., and Samet, J.M. (2013). Causal inference in public health. Annu. Rev. Publ. Health *34*, 61–75.
2. Sibbald, B., and Roland, M. (1998). Understanding controlled trials. why are randomised controlled trials important? BMJ Br. Med. J. (Clin. Res. Ed.) *316*, 201.
3. Adebamowo, C., Bah-Sow, O., Binka, F., Bruzzone, R., Caplan, A., Delfraissy, J.-F., Heymann, D., Horby, P., Kaleebu, P., Tamfum, J.-J.M., et al. (2014). Randomised controlled trials for ebola: practical and ethical issues. Lancet *384*, 1423–1424.
4. Hernán, M.A., and Robins, J.M. (2016). Using big data to emulate a target trial when a randomized trial is not available. Am. J. Epidemiol. *183*, 758–764.
5. Shalit, U., Johansson, F.D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In International Conference on Machine Learning (PMLR), pp. 3076–3085.
6. Shi, C., Blei, D., and Veitch, V. (2019). Adapting neural networks for the estimation of treatment effects. Adv. Neural Inf. Process. Syst. *32*.
7. Hassanpour, N., and Greiner, R. (2019). Learning disentangled representations for counterfactual regression. In International Conference on Learning Representations.
8. Curth, A., and van der Schaar, M. (2021a). Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In International Conference on Artificial Intelligence and Statistics (PMLR), pp. 1810–1818.
9. Guo, Z., Zheng, S., Liu, Z., Yan, K., and Zhu, Z. (2021). Cetransformer: Casual effect estimation via transformer based representation learning.

In Chinese Conference on Pattern Recognition and Computer Vision (PRCV) (Springer), pp. 524–535.

10. Liu, R., Wei, L., and Zhang, P. (2021). A deep learning framework for drug repurposing via emulating clinical trials on real-world patient data. Nat. Mach. Intell. *3*, 68–75.

11. Zhang, Y.-F., Zhang, H., Lipton, Z.C., Li, L.E., and Xing, E.P. (2022). Can transformers be strong treatment effect estimators?. Preprint at arXiv. https://doi.org/10.48550/arXiv.2202.01336.

12. Chipman, H.A., George, E.I., and McCulloch, R.E. (2010). Bart: Bayesian additive regression trees. Ann. Appl. Stat. *4*, 266–298.

13. Wager, S., and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. J. Am. Stat. Assoc. *113*, 1228–1242.

14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. Adv. Neural Inf. Process. Syst. *30*.

15. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pp. 4171–4186.

16. Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-training (OpenAI blog).

17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog *1*, 9.

18. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. Adv. Neural Inf. Process. Syst. *33*, 1877–1901.

19. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. Preprint at arXiv. https://doi.org/10.48550/arXiv.1907.11692.

20. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In European conference on computer vision (Springer), pp. 213–229.

21. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations.

22. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image transformer. In International Conference on Machine Learning (PMLR), pp. 4055–4064.

23. Merative (2023a). Marketscan research databases. Available at. https://www.merative.com/real-world-evidence.

24. Healthcare Cost and Utilization Project (HCUP) (2017). Clinical Classifications Software (Ccs) (Agency for Healthcare Research and Quality). Avaiable at. www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp.

25. Merative (2023b). Micromedex red book. Available at. https://www.merative.com/micromedex-training-center/red-book.

26. Künzel, S.R., Sekhon, J.S., Bickel, P.J., and Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. Proc. Natl. Acad. Sci. USA *116*, 4156–4165.

27. Curth, A., and van der Schaar, M. (2021b). On inductive biases for heterogeneous treatment effect estimation. Adv. Neural Inf. Process. Syst. *34*.

28. Alaa, A., and Van Der Schaar, M. (2019). Validating causal inference models via influence functions. In International Conference on Machine Learning (PMLR), pp. 191–201.

29. Anand, S.S., Bosch, J., Eikelboom, J.W., Connolly, S.J., Diaz, R., Widimsky, P., Aboyans, V., Alings, M., Kakkar, A.K., Keltai, K., et al. (2018). Rivaroxaban with or without aspirin in patients with stable peripheral or carotid artery disease: an international, randomised, double-blind, placebo-controlled trial. Lancet *391*, 219–229.

30. Pfeffer, M.A., Claggett, B., Lewis, E.F., Granger, C.B., Køber, L., Maggioni, A.P., Mann, D.L., McMurray, J.J.V., Rouleau, J.-L., Solomon, S.D., et al. (2021). Angiotensin receptor–neprilysin inhibition in acute myocardial infarction. N. Engl. J. Med. *385*, 1845–1855.

31. Sandner, S.E., Schunkert, H., Kastrati, A., Wiedemann, D., Misfeld, M., Böning, A., Tebbe, U., Nowak, B., Stritzke, J., Laufer, G., et al. (2020). Ticagrelor monotherapy versus aspirin in patients undergoing multiple arterial or single arterial coronary artery bypass grafting: insights from the ticab trial. Eur. J. Cardio. Thorac. Surg. *57*, 732–739.

32. Granger, C.B., Alexander, J.H., McMurray, J.J.V., Lopes, R.D., Hylek, E.M., Hanna, M., Al-Khalidi, H.R., Ansell, J., Atar, D., Avezum, A., et al. (2011). Apixaban versus warfarin in patients with atrial fibrillation. N. Engl. J. Med. *365*, 981–992.

33. Hernán, M.A. (2004). A definition of causal effect for epidemiological research. J. Epidemiol. Community Health *58*, 265–271.

34. Vig, J. (2019). A multiscale visualization of attention in the transformer model. ACLPPinforma *2019*.

35. Roy, D., Talajic, M., Dorian, P., Connolly, S., Eisenberg, M.J., Green, M., Kus, T., Lambert, J., Dubuc, M., Gagné, P., et al. (2000). Amiodarone to prevent recurrence of atrial fibrillation. N. Engl. J. Med. *342*, 913–920.

36. Stanifer, J.W., Pokorney, S.D., Chertow, G.M., Hohnloser, S.H., Wojdyla, D.M., Garonzik, S., Byon, W., Hijazi, Z., Lopes, R.D., Alexander, J.H., et al. (2020). Apixaban versus warfarin in patients with atrial fibrillation and advanced chronic kidney disease. Circulation *141*, 1384–1392.

37. Yao, L., Chu, Z., Li, S., Li, Y., Gao, J., and Zhang, A. (2021). A survey on causal inference. ACM Trans. Knowl. Discov. Data *15*, 1–46.

38. Li, Y., Rao, S., Solares, J.R.A., Hassaine, A., Ramakrishnan, R., Canoy, D., Zhu, Y., Rahimi, K., and Salimi-Khorshidi, G. (2020). Behrt: transformer for electronic health records. Sci. Rep. *10*, 7155.

39. Rasmy, L., Xiang, Y., Xie, Z., Tao, C., and Zhi, D. (2021). Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. NPJ Digit. Med. *4*, 86.

40. Corporation, E.S. (2024). Cosmos. Available at. https://cosmos.epic.com/.

41. Optum, I. (2024). Optum claims data. Available at. https://www.optum.com/en/business/life-sciences/real-world-data/claims-data.html.

42. Inc., I (2023). Iqvia real world and health data sets. Available at. https://www.iqvia.com/solutions/real-world-evidence/real-world-data-and-insights.

43. Liu, R. (2024). Code for the Article “CURE: A Pre-training Deep Learning Framework on Large-Scale Patient Data for Treatment Effect Estimation. Zenodo. https://doi.org/10.5281/zenodo.10802189.

44. Rubin, D.B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. J. Am. Stat. Assoc. *100*, 322–331.

45. Imbens, G.W., and Rubin, D.B. (2015). Causal Inference in Statistics, Social, and Biomedical Sciences (Cambridge University Press).