GENERATION AND EVALUATION OF SYNTHETIC DATA CONTAINING TREATMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal inference on medical data containing treatments, such as estimation of treatment effects, is crucial to ensure the efficacy and safety of interventions. However, privacy concerns can often limit access to the patient data necessary for such analyses. Generative models can produce synthetic data that preserve privacy and closely approximate the real data distribution, yet existing methods do not consider downstream tasks for data containing treatments, nor the unique challenges these pose. With our work we establish a set of desiderata that synthetic data containing treatments should satisfy to maximise downstream utility: preservation of (i) the covariate distribution, (ii) the treatment assignment mechanism, and (iii) the outcome generation mechanism. Based on these desiderata, we propose a set of evaluation metrics to assess such synthetic data. Finally, we present STEAM: a novel method for generating ***S**ynthetic data for **T**reatment **E**ffect **A**nalysis in **M**edicine*. STEAM mimics the data-generating process of data containing treatments and optimises for our desiderata, while allowing differentially private generation. We empirically demonstrate that STEAM achieves state-of-the-art performance across our metrics as compared to existing generative models, particularly as the complexity of the generative task increases.

1 INTRODUCTION

Medical data sharing is crucial for advancing research in healthcare, enabling the replication of results to establish validity, and the discovery of new insights through alternative analyses (Bauchner et al., 2016; Wirth et al., 2021). However, such sharing is hindered by stringent regulations which restrict access to patient data for research purposes (Annas, 2003; Voigt & Von dem Bussche, 2017).

A potential solution is offered by synthetic data, which has gained increasing recognition in medical literature (Jadon & Kumar, 2023). Generative models can produce synthetic copies of sensitive data, which can be shared more freely (Jordon et al., 2018). If necessary, generation can satisfy formal definitions of privacy, such as differential privacy (DP) (Dwork et al., 2006), ensuring provable guarantees (Pan et al., 2024). Importantly, the promise of synthetic data hinges on its ability to preserve information critical to relevant downstream tasks. Among the existing synthetic data literature (Bauer et al., 2024), most works focus on downstream *predictive* (supervised) tasks, shaping standard evaluation and generation practices to this setting.

However, medical data typically contain *treatment assignment variables*, which invite unique downstream analysis. Data containing treatments are typically analysed via *causal inference* methods (e.g. treatment effect estimation methods) which examine the causal relationships between covariates, treatments, and outcomes, in a manner distinct from associative prediction (Feuerriegel et al., 2024). Despite this, synthetic data papers which use medical data containing treatments for motivation and validation (Choi et al., 2018; Kotelnikov et al., 2022; Yan et al., 2022; Borisov et al., 2023) generally employ standard, prediction-oriented, generation and evaluation techniques (see examples in Appendix A). Failure to acknowledge the likely downstream use of synthetic data containing treatments leads to low-quality generation, which is masked by misaligned evaluation metrics.

Evaluation. Standard synthetic data evaluation involves statistical comparison of synthetic and real data (Table 1), and assessing the accuracy of synthetically-trained models in predicting a target variable. In this evaluation paradigm, causal inference tasks are not considered, as treatments are handled like any other feature, limiting the relevance of such assessment for synthetic data containing

treatments. To illustrate this, consider the following key questions that an analyst working with a synthetic dataset containing treatments, $\mathcal{D}_{\text{synth}}$, may ask: ***Q1*** *How representative are the patient covariates in $\mathcal{D}_{synth}$?*; ***Q2*** *How accurate are the treatment assignment decisions in $\mathcal{D}_{synth}$?*; and ***Q3*** *How much error might be introduced in treatment effect estimates derived from $\mathcal{D}_{synth}$?* These questions require differentiation between covariates, treatments, and outcomes, and they cannot be accurately answered with current evaluation protocols.

Generation. *Generic synthetic data generation* (Table 2) seeks to minimise the difference in synthetic and real joint distributions, and all variables are generated simultaneously from this distribution. This overlooks the data-generating process (DGP) from which data containing treatments arise, where covariates drive treatment assignments, and both covariates and treatments influence outcomes (Pearl, 2009). In doing so, such generation fails to capitalise on a valuable inductive bias, producing synthetic data which poorly preserve these important relationships for causal inference. Existing *causal generative models*, on the other hand, generally assume access to the full causal graph $\mathcal{G}$, which is overly restrictive in complex settings, such as medicine, where $\mathcal{G}$ is unlikely to be known.

In this work we address these limitations by conducting an analysis of synthetic data containing treatments, proposing novel approaches to evaluation and generation which operate under reasonable assumptions and explicitly consider the likely downstream use of such data. In doing so, we make the following contributions:

1. **Desiderata:** By examining the typical analysis conducted on data containing treatments, we establish a set of desiderata that synthetic data should satisfy in this context (Section 4).
2. **Evaluation:** We show that existing evaluation metrics for synthetic data are inadequate in this setting, as they do not measure how well these desiderata are respected. As a remedy, we propose a principled set of metrics derived from our desiderata, allowing meaningful evaluation of synthetic data containing treatments (Section 5).
3. **Generation:** We propose STEAM, a novel method for synthetic data generation that contains inductive biases to optimise for our desiderata and mimic the real DGP of data containing treatments. Furthermore, STEAM can satisfy DP if desired (Section 6).
4. **Empirical Analysis:** Using our newly established metrics, we demonstrate that STEAM exhibits state-of-the-art performance in generating synthetic data containing treatments, particularly as the real DGP grows in complexity, and in high-dimensional scenarios (Sections 7). Our code is available via `https://anonymous.4open.science/r/STEAM-35EC`.

2 Problem Formulation

Setup. We consider a *data owner* with access to observational or experimental real data $\mathcal{D}_{\text{real}} = \{(\mathbf{X}^{(i)}_{\text{real}}, W^{(i)}_{\text{real}}, Y^{(i)}_{\text{real}})\}_{i=1}^{n}$ sampled from a population $P_{\mathbf{X},W,Y}$, where $\mathbf{X}^{(i)}_{\text{real}} = \{X_j\}_{j=1}^{d} \in \mathcal{X}^{(d)}$ is a vector of d binary or continuous covariates, $W^{(i)}_{\text{real}} \in \{0,1\}$ is a binary treatment assignment, and $Y^{(i)}_{\text{real}} \in \mathcal{Y}$ is a binary or continuous outcome. We refer to the set of all variables in $\mathcal{D}_{\text{real}}$ as $\mathcal{V} = \{X_1, ..., X_d, W, Y\}$. We denote the propensity score with $\pi(\mathbf{x}) = P_{W|\mathbf{X}}(W = 1|\mathbf{X} = \mathbf{x})$.

Objective. We wish to enable the release of synthetic data to downstream users with various analysis goals, such as estimation of propensity scores, average treatment effects (ATEs), and conditional average treatment effects (CATEs).[1] To do so, we aim to generate synthetic data $\mathcal{D}_{\text{synth}} = \{(\mathbf{X}^{(i)}_{\text{synth}}, W^{(i)}_{\text{synth}}, Y^{(i)}_{\text{synth}})\}_{i=1}^{n}$ from a distribution Q and evaluate how well $\mathcal{D}_{\text{synth}}$ captures information relevant to likely downstream tasks with a set of metrics $\mathcal{M}(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}})$.

Terminology. To avoid confusion, we clarify that 'real data' refers to the specific data of the data owner (rather than simply any real-world observational data). Further, 'synthetic data' refers strictly to data that serve as synthetic copies of real data. This should not be confused with *simulated* or *semi-simulated* data, which are often used to benchmark CATE learners (Curth et al., 2021).

3 Related Work

Evaluation. Evaluation of synthetic tabular data, the modality we focus on in this paper, is diverse, although there are two common themes: *resemblance* and *predictive utility* (Murtaza et al., 2023).

[1] Denoting the potential outcomes as Y(0) and Y(1), ATE is defined as ATE $= \mathbb{E}_P[Y(1) - Y(0)]$ and CATE is $\tau(\mathbf{x}) = \mathbb{E}_P[Y(1) - Y(0)|\mathbf{X} = \mathbf{x}]$.

Table 1: Tabular synthetic data evaluation methods applied to data containing treatments. $d(\cdot)$ is an abstract distance function. For 'Prec., Rec.', we use $\mathcal{S}_P$ to denote the support of distribution P. For 'Discriminator', $\mathcal{D}_{\text{comb}} = \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth}}$, and $c^{(i)}$ is the dataset label for instance i. '**Q. addressed?**': which, if any, of the key questions from Section 1 does the method answer?

	Method	Formula	Differentiates between $\mathbf{X}, W, Y$?	Q. addressed?
Existing	Marginal	$\frac{1}{\lvert\mathcal{V}\rvert} \sum_{i \in \mathcal{V}} d(P_i, Q_i)$	✗	-
	Correlation	$\frac{1}{\lvert\mathcal{V}\rvert(\lvert\mathcal{V}\rvert-1)} \sum_{\substack{i,j \in \mathcal{V} \\ i \neq j}} d(\text{corr}(P_i, P_j), \text{corr}(Q_i, Q_j))$	✗	-
	Joint	$d(P_{\mathcal{V}}, Q_{\mathcal{V}})$	✗	-
	Prec., Rec.	$\lvert\mathcal{S}_P \cap \mathcal{S}_Q\rvert / \lvert\mathcal{S}_Q\rvert$, $\lvert\mathcal{S}_P \cap \mathcal{S}_Q\rvert / \lvert\mathcal{S}_P\rvert$	✗	-
	Discriminator	$\frac{1}{2n} \sum_{i=1}^{2n} \mathbb{1}\{C(\mathcal{D}_{\text{comb}}^{(i)}) = c^{(i)}\}$	✗	-
Ours	$P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$	Equations 3 and 4	✓	***Q1***
	JSD_π	Equation 5	✓	***Q2***
	U_{PEHE}	Equation 6	✓	***Q3***

Assessing *predictive utility* involves training a predictive model on $\mathcal{D}_{\text{synth}}$ and measuring its accuracy. Since key causal inference tasks, such as treatment effect estimation, do not have observable ground-truths (Holland, 1986), model validation becomes a non-trivial task (Curth & Van Der Schaar, 2023) and such utility assessment does not apply to our setting. On the other hand, methods for assessing *resemblance*, which can involve comparison of synthetic and real marginals, 2-way correlation matrices, joint distributions, supports via precision and recall, and assessment of a discriminator model C in separating real and synthetic data, are summarised in Table 1. Importantly, these methods handle all features within a dataset similarly, not differentiating between the variable classes $\mathbf{X}$, W, and Y, and they therefore cannot directly answer any of the key questions ***Q1-3*** posed in Section 1. Our metrics, proposed in Section 5, remedy this.

Generation. We consider two related approaches to generative modeling, with a high-level comparison in Table 2. Firstly, *generic generative models* are those which make minimal assumptions on $\mathcal{D}_{\text{real}}$, and seek to minimise the difference between real and synthetic joint distributions. *Causal generative models*, on the other hand, assume access to the full causal graph $\mathcal{G}$, and then seek to minimize the difference between each real and synthetic conditional distribution, as dictated by the causal relationships in $\mathcal{G}$. The assumptions for our proposed method, STEAM, fall in between these two. While we assume that the underlying DGP of $\mathcal{D}_{\text{real}}$ is of the form $X \sim P_{\mathbf{X}}$, $W \sim P_{W|\mathbf{X}}$, $Y \sim P_{Y|W,\mathbf{X}}$, this will hold for a wide array of datasets containing treatments, and it is generally less restrictive than assuming complete knowledge of $\mathcal{G}$, as we do not need the individual causal relationships between variables. Therefore, STEAM is more applicable in complex settings. Furthermore, neither of these existing approaches are tailored to downstream treatment effect estimation, and they do not target our desiderata for synthetic data containing treatments (Section 4). For empirical comparisons against generic generative models, see Section 7, and see Appendix O for comparison with causal generative models. For further elaboration on specific evaluation and generation methods see Appendix B.

4 Desiderata for synthetic data containing treatments

We now consider three distributions that are critical in downstream analysis of data containing treatments: (i) the covariate distribution $P_{\mathbf{X}}$, (ii) the treatment assignment mechanism $P_{W|\mathbf{X}}$, and (iii) the outcome generation mechanism $P_{Y|W,\mathbf{X}}$. Their importance, which we describe below, is clear in the causal inference community, giving rise to the key questions ***Q1-3*** which downstream analysts may ask when using $\mathcal{D}_{\text{synth}}$. However, this has not been addressed by the synthetic data community, and methods designed to specifically preserve them are missing. To bridge this gap, we establish our desiderata for synthetic data containing treatments based on these distributions.

(i) The covariate distribution $P_{\mathbf{X}}$

$P_{\mathbf{X}}$ describes the population of interest and, in medical practice, it is standard to report its characteristics (Wolff et al., 2019), as it determines to whom analysis will be relevant.

Why is its preservation important? Failure to cover covariate levels in $\mathcal{D}_{\text{synth}}$ can result in exclusion from downstream analysis of members of the population whose covariates are not well

Table 2: Tabular synthetic data generation methods. $d(\cdot)$ is an abstract distance function. $\text{PA}_{\mathcal{G}}(\mathcal{V}_i)$ refers to the set of parents of node $\mathcal{V}_i$ in the causal graph $\mathcal{G}$. [1]: Rezende & Mohamed (2016), [2]: Xu et al. (2019), [3]: Kotelnikov et al. (2022), [4]: Watson et al. (2023), [5]: ANM (Hoyer et al., 2008), [6]: Sánchez-Martin et al. (2022), [7]: Chao et al. (2024)

	Methods	Distributional target	Assumptions on $\mathcal{D}_{\text{real}}$	$\mathcal{D}_{\text{synth}}$ application
Generic gen. models	NFlow [1] CTGAN [2] TVAE [2] TabDDPM [3] ARF [4]	$\min d(Q_{\mathcal{V}}, P_{\mathcal{V}})$	None	Prediction
Causal gen. models	ANM [5] VACA [6] CGM [7]	$\min d\big(Q_{\mathcal{V}_1\|\text{PA}_{\mathcal{G}}(\mathcal{V}_1)}, P_{\mathcal{V}_1\|\text{PA}_{\mathcal{G}}(\mathcal{V}_1)}\big)$ $\vdots$ $\min d\big(Q_{\mathcal{V}_{\|\mathcal{V}\|}\|\text{PA}_{\mathcal{G}}(\mathcal{V}_{\|\mathcal{V}\|})}, P_{\mathcal{V}_{\|\mathcal{V}\|}\|\text{PA}_{\mathcal{G}}(\mathcal{V}_{\|\mathcal{V}\|})}\big)$	Known $\mathcal{G}$	Interventional and counterfactual queries on $\mathcal{G}$
Ours	STEAM	$\min d(Q_{\mathbf{X}}, P_{\mathbf{X}})$ $\min d(Q_{W\|\mathbf{X}}, P_{W\|\mathbf{X}})$ $\min d(Q_{Y\|W,\mathbf{X}}, P_{Y\|W,\mathbf{X}})$	Valid DGP	Treatment effect estimation

explored, as making reliable inferences can become infeasible (Petersen et al., 2010; Rudolph et al., 2022). On the other hand, generating out-of-distribution covariates in $\mathcal{D}_{\text{synth}}$ can cause groundless extrapolation by synthetically-trained models, leading to potential misuse.

(ii) The treatment assignment mechanism $P_{W|\mathbf{X}}$

$P_{W|\mathbf{X}}$ is often used as a nuisance parameter in treatment effect models (Austin, 2011; Curth & van der Schaar, 2021), and it can be a target for analysis itself when examining treatment protocols.

Why is its preservation important? Given the use of $P_{W|\mathbf{X}}$ as a nuisance parameter, errors in its modelling will propagate to errors in treatment effect estimates derived from $\mathcal{D}_{\text{synth}}$. Furthermore, $P_{W|\mathbf{X}}$ can guide the difficult task of CATE model selection (Hüyük et al., 2024), so poor preservation may lead to inconsistency in this area between $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$, which is unideal (Hansen et al., 2023). Finally, misrepresenting $P_{W|\mathbf{X}}$ can lead to misreporting of treatment protocols. Given that extreme propensities of $\pi(\mathbf{x}) \approx 0$ (or $\pi(\mathbf{x}) \approx 1$) are common in high-dimensional data, such as electronic health records (Li et al., 2018), an inaccurate $Q_{W|\mathbf{X}}$ could lead to subsequent exploration of treatments in patient subgroups for which they are unsafe.

(iii) The outcome generation mechanism $P_{Y|W,\mathbf{X}}$

$P_{Y|W,\mathbf{X}}$ is the distribution through which treatment effects can be estimated by comparing the statistical functionals of $P_{Y|W=1,\mathbf{X}}$ and $P_{Y|W=0,\mathbf{X}}$.

Why is its preservation important? $P_{Y|W,\mathbf{X}}$ must be preserved, so that $\mathcal{D}_{\text{synth}}$ can permit accurate estimation of treatment effects. If $Q_{Y|W,\mathbf{X}}$ is inaccurate, then even a perfect model could not estimate correct treatment effects from $\mathcal{D}_{\text{synth}}$, and the worse this relationship is preserved, the less useful it becomes.

Preserving (i)-(iii) is *necessary* and *sufficient* for Q to be a high quality approximation of P. Modelling each distribution well is evidently *necessary* given the above reasons, and it is also *sufficient*, which is clear from the following decomposition of $P_{\mathbf{X},W,Y}$:

$$P_{\mathbf{X},W,Y}(\mathbf{X}, W, Y) = \underbrace{P_{\mathbf{X}}(\mathbf{X})}_{\text{(i)}}\ \underbrace{P_{W|\mathbf{X}}(W|\mathbf{X})}_{\text{(ii)}}\ \underbrace{P_{Y|W,\mathbf{X}}(Y|W,\mathbf{X})}_{\text{(iii)}} \tag{1}$$

The components (i)-(iii) offer a complete factorisation of the joint distribution, and therefore Q matching P in each component is sufficient for Q to match P entirely. As such, accurate modelling of (i)-(iii) forms our desiderata for synthetic data containing treatments. Generation methods should seek to *maximise adherence to these desiderata*, and evaluation metrics should *assess how successful $\mathcal{D}_{\text{synth}}$ is in this regard* (and therefore answer ***Q1-3***). In the following sections, we show that existing metrics (Section 5.1), and generation methods (Section 7) perform poorly in this regard.

On causal assumptions. Even if these desiderata are satisfied, $\mathcal{D}_{\text{synth}}$ may not permit correct causal inference. Required assumptions, such as typical identifiability assumptions,[2] must still be critically examined, since any violations in $\mathcal{D}_{\text{real}}$ will almost surely be violated in a faithful $\mathcal{D}_{\text{synth}}$ as well. Identifying and accounting for such violated assumptions is a task orthogonal to synthetic data generation, with existing literature (Kallus et al., 2019; Frauen & Feuerriegel, 2022), and we do not consider it necessary for Q to improve upon such factors. Instead, any biases in P should be maintained in Q, allowing post-generation methods to rectify them if necessary.

5 How to evaluate synthetic data containing treatments

With our desiderata established, we now investigate how to evaluate the adherence of $\mathcal{D}_{\text{synth}}$.

5.1 Inadequacy of existing metrics

Existing evaluation metrics, discussed in Section 3, do not offer a clear sense of how well $\mathcal{D}_{\text{synth}}$ satisfies our desiderata. These metrics do not differentiate between $\mathbf{X}$, W, and Y, and they therefore cannot directly assess any of $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, or $Q_{Y|W,\mathbf{X}}$. Within these existing metrics, joint-distribution-level metrics, such as Kullback–Leibler divergence (KL) (Kullback & Leibler, 1951), are most popular, since they offer a complete, holistic assessment of how well Q models P. However these are, at best, loosely related to our desiderata, and they do not allow a user to disentangle how each of (i)-(iii) is preserved, limiting the depth of information offered on $\mathcal{D}_{\text{synth}}$. Furthermore, we argue that these metrics will tend to be dominated by the covariate distribution as $\mathbf{X}$ grows in dimensionality, and they will lose *sensitivity* to the treatment assignment and outcome generation mechanisms. In this sense, *sensitivity* refers to the effect that differences in the modelling of $P_{W|\mathbf{X}}$ or $P_{Y|W,\mathbf{X}}$ by a proposal distribution Q have on a metric $\mathcal{M}$.

To demonstrate this more formally, consider a simple $P_{\mathbf{X},W,Y}$ which can be factorized as $P_{\mathbf{X},W,Y} = \prod_{i=1}^{d} P_{\mathbf{X}_i}\ P_{W|\mathbf{X}}\ P_{Y|W,\mathbf{X}}$. Let there be two learnable distributions $Q^{\theta_1}_{\mathbf{X},W,Y}$ and $Q^{\theta_2}_{\mathbf{X},W,Y}$, which estimate $P_{\mathbf{X},W,Y}$, with the same form $Q^{\theta_k}_{\mathbf{X},W,Y} = \prod_{i=1}^{d} Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i}\ Q^{\theta_{W,k}}_{W|\mathbf{X}}\ Q^{\theta_{Y,k}}_{Y|W,\mathbf{X}}$, and which only differ in either $\theta_{W,k}$ or $\theta_{Y,k}$ (i.e. they either model $P_{W|\mathbf{X}}$ or $P_{Y|W,\mathbf{X}}$ differently). In this setting, the following holds:

Theorem 1. *Let P, Q^{θ_1}, Q^{θ_2} be of the above form, and $\mathcal{M}$ be KL divergence. If we assume that Q^{θ_1} and Q^{θ_2} have sufficient capacity to have bounded error on each component, i.e. $\forall i$, $0 < \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i}) < \varepsilon_{\mathbf{X}}$, and $0 < \mathcal{M}(P_{W|\mathbf{X}}, Q^{\theta_{W,k}}_{W|\mathbf{X}}) < \varepsilon_{W,k}$, and $0 < \mathcal{M}(P_{Y|W,\mathbf{X}}, Q^{\theta_{Y,k}}_{Y|W,\mathbf{X}}) < \varepsilon_{Y,k}$, then:*

$$\frac{\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_1}_{\mathbf{X},W,Y})}{\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_2}_{\mathbf{X},W,Y})} \to 1,\ \textit{as}\ d \to \infty \tag{2}$$

Proof. See Appendix B. □

Theorem 1 shows that KL divergence loses sensitivity to $W|\mathbf{X}$ and $Y|W,\mathbf{X}$ as d grows, suggesting that this metric will struggle in selecting between $Q^{\theta_1}_{\mathbf{X},W,Y}$ and $Q^{\theta_2}_{\mathbf{X},W,Y}$, since their scores will converge to the same value despite any difference in their modelling of $P_{W|\mathbf{X}}$ or $P_{Y|W,\mathbf{X}}$. For an empirical example of this phenomena, with an extended array of joint-distribution-level metrics, see Appendix D.

5.2 Metrics tailored to synthetic data containing treatments

These formal and empirical findings motivate us to design our own metrics for synthetic data containing treatments. We now propose an appropriate set of metrics $\mathcal{M} = (P_{\alpha,\mathbf{X}}, R_{\beta,\mathbf{X}}, JSD_{\pi}, U_{\text{PEHE}})$ which directly measure performance in line with desiderata (i)-(iii), and can offer answers to ***Q1-3***.

5.2.1 The covariate distribution $P_{\mathbf{X}}$

Evaluation of the preservation of $P_{\mathbf{X}}$ requires direct comparison of the generally high-dimensional covariate distributions of $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$, which is non-trivial. Nevertheless this is a standard synthetic data evaluation task, as $\mathbf{X}_{\text{real}}$ and $\mathbf{X}_{\text{synth}}$ do not contain treatments. We see precision/recall analysis as the most useful evaluation practice in this context. There is typically a trade-off between these two qualities, which generative models approach differently (Sajjadi et al., 2018; Bayat, 2023),

[2] Consistency: $Y^{(i)} = Y(W^{(i)})$, overlap: $0 < \pi(\mathbf{x}) < 1$, and unconfoundedness: $Y(0), Y(1) \perp\!\!\!\perp W|\mathbf{X}$

and by measuring them both a data holder can guide generation towards their preferences of covariate realism and diversity. If the data holder has no strong preference, balancing the two is recommended to achieve the best downstream results (Jordon et al., 2022).

We propose the use of the integrated P_α and R_β scores, introduced by Alaa et al. (2022). Intuitively, P_α captures how much of the synthetic data falls within the support of the real data, and R_β reflects how much of the real data is covered by the support of synthetic data. We denote the covariate precision and recall with $P_{\alpha,\mathbf{X}}$ and $R_{\beta,\mathbf{X}}$ respectively, which are calculated by applying integrated P_α and R_β to the covariate distribution only, as in (3) and (4).

To assess the preservation of $P_{\mathbf{X}}$

$$P_{\alpha,\mathbf{X}}(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}) = 1 - 2\int_0^1 |\mathbb{P}(\tilde{\mathbf{X}}_{\text{synth}} \in \mathcal{S}^{\alpha}_{\text{real}}) - \alpha|\, d\alpha \tag{3}$$

$$R_{\beta,\mathbf{X}}(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}) = 1 - 2\int_0^1 |\mathbb{P}(\tilde{\mathbf{X}}_{\text{real}} \in \mathcal{S}^{\beta}_{\text{synth}}) - \beta|\, d\beta \tag{4}$$

where $\tilde{\mathbf{X}}_\diamond$ and $\mathcal{S}^{\square}_\diamond$ are the embedding $\tilde{\mathbf{X}}_\diamond = \Phi(\mathbf{X}_\diamond)$ and $\square$-support as defined by Alaa et al. (2022), respectively.

We have $0 < P_{\alpha,\mathbf{X}}, R_{\beta,\mathbf{X}} < 1$, and scores near 1 indicate a realistic and diverse $Q_{\mathbf{X}}$. Together, these metrics can be used to answer ***Q1***.

5.2.2 The treatment assignment mechanism $P_{W|\mathbf{X}}$

While in general we do not have access to $P_{W|\mathbf{X}}$ and $Q_{W|\mathbf{X}}$, we know that, for each $\mathbf{X} = \mathbf{x}$, they are Bernoulli distributions, since W is a binary variable. The success probabilities can be estimated from $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$ with a probabilistic classifier, which can be used to form approximations of $P_{W|\mathbf{X}}$ and $Q_{W|\mathbf{X}}$. There is then an array of valid options to compare these approximations. We propose the use of Jensen-Shannon distance[3] given its desirable properties of symmetry, smoothness, and boundedness (we discuss alternatives in Appendix E). For a given probabilistic classifier $\hat{\pi}$, we define $\hat{P}_{W|\mathbf{X}=\mathbf{x}} = \text{Bern}(\hat{\pi}_{\text{real}}(\mathbf{x}))$ and $\hat{Q}_{W|\mathbf{X}=\mathbf{x}} = \text{Bern}(\hat{\pi}_{\text{synth}}(\mathbf{x}))$ where $\hat{\pi}_{\text{real}}$ and $\hat{\pi}_{\text{synth}}$ are trained on $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$ respectively, and we measure the preservation of $P_{W|\mathbf{X}}$ as in (5).

To assess the preservation of $P_{W|\mathbf{X}}$

$$\text{JSD}_\pi(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}) = 1 - \mathbb{E}_{P_{\mathbf{X}}}\left[\sqrt{\frac{1}{2}D_{KL}(\hat{P}_{W|\mathbf{X}=\mathbf{x}}||M) + \frac{1}{2}D_{KL}(\hat{Q}_{W|\mathbf{X}=\mathbf{x}}||M)}\right] \tag{5}$$

where $M = \frac{1}{2}(\hat{P}_{W|\mathbf{X}=\mathbf{x}} + \hat{Q}_{W|\mathbf{X}=\mathbf{x}})$ and D_{KL} is KL divergence using $\log_2$.

JSD_π can be used to answer ***Q2***. We have $0 < \text{JSD}_\pi < 1$, with scores near 1 indicating that $Q_{W|\mathbf{X}}$ matches $P_{W|\mathbf{X}}$ well. The validity of JSD_π will depend on the accuracy of $\hat{\pi}$, so conducting $\hat{\pi}$ model selection is an important pre-evaluation step, although amongst reasonable model choices which exhibit similar performance, the information offered by JSD_π will not significantly differ.

5.2.3 The outcome generation mechanism $P_{Y|W,\mathbf{X}}$

To evaluate the preservation of $P_{Y|W,\mathbf{X}}$, we consider a treatment effect analogue of predictive utility. In this, we address the unavailability of ground-truths by aiming for agreement in performance on $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$, rather than attempting to quantify error from an oracle value. Such evaluation is inherently task dependent, yet the specific quantity $\mathcal{D}_{\text{synth}}$ may be used to estimate is unclear. Assessment should therefore centre on a complex task, in which comparable performance will likely imply the same for simpler tasks. In this case, we consider the most difficult treatment effect task likely to arise in the medical field—CATE estimation—as similarity in this between $\mathcal{D}_{\text{synth}}$ and $\mathcal{D}_{\text{real}}$ will tend to imply similarity in simpler tasks, such as ATE estimation. Therefore, we evaluate how well $Q_{Y|W,\mathbf{X}}$ preserves $P_{Y|W,\mathbf{X}}$ by calculating the PEHE between synthetic- and real-trained CATE

[3] $\text{JSD}(P \,\|\, Q) = \sqrt{\frac{1}{2}D_{\text{KL}}(P \,\|\, M) + \frac{1}{2}D_{\text{KL}}(Q \,\|\, M)}$, where $M = \frac{1}{2}(P + Q)$ and D_{KL} is KL divergence.

learners (see Appendix E for alternatives). Given a family $\mathcal{F}$ of CATE learners $\hat{\tau}$, where $\hat{\tau}_{\text{real}}$ and $\hat{\tau}_{\text{synth}}$ are trained on $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$ respectively, we assess the preservation of $P_{Y|W,\mathbf{X}}$ as in (6).

To assess the preservation of $P_{Y|W,\mathbf{X}}$

$$U_{\text{PEHE}}(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}) = \frac{1}{|\mathcal{F}|} \sum_{\hat{\tau} \in \mathcal{F}} \sqrt{\mathbb{E}_{P_{\mathbf{X}}}[(\hat{\tau}_{\text{synth}}(\mathbf{X}) - \hat{\tau}_{\text{real}}(\mathbf{X}))^2]} \tag{6}$$

U_{PEHE} can answer ***Q3***. We average over $\mathcal{F}$ since CATE model validation is difficult (Curth & Van Der Schaar, 2023), so $\hat{\tau}$ cannot be set as the best performing model in a similar fashion as is done for JSD_{π} (we discuss choices for $\mathcal{F}$ in Appendix E.3.2). As such, U_{PEHE} rewards generators which permit proximity in CATE estimations across a wide array of potential learners, where a lower U_{PEHE} indicates better preservation of $P_{Y|W,\mathbf{X}}$.

6 Generating Synthetic Data Containing Treatments

To illustrate the standard DGP of data contraining treatments, shown in the middle of Figure 9, consider a simple hospital dataset. Patient covariates $\mathbf{X}$, such as height, weight etc., are drawn from an underlying covariate distribution $P_{\mathbf{X}}$, which is dictated by the local population. Treatments are then assigned by a domain expert, such as a doctor, conditioned on $\mathbf{X}$, i.e. $W \sim P_{W|\mathbf{X}}$. Finally, patients' outcomes are dictated by the dynamics of their ailments, conditional upon W and $\mathbf{X}$, i.e. $Y \sim P_{Y|\mathbf{X},W}$. We now propose STEAM, a novel method for generating ***S**ynthetic data for **T**reatment **E**ffect **A**nalysis in **M**edicine* which mimics the real DGP.

6.1 STEAM

Mimicry of the real DGP acts as an inductive bias, pushing Q closer towards the P in structure, and directly targeting each distributions from our desiderata. STEAM, shown on the right of Figure 9, conducts a three-step generation process, involving the following:

1. $\boldsymbol{Q_{\mathbf{X}}}$. $\mathbf{X}$ is generated from a generative model trained to match the covariate distribution $P_{\mathbf{X}}$.
2. $\boldsymbol{Q_{W|\mathbf{X}}}$. Treatments are assigned according to a propensity function trained on $\mathcal{D}_{\text{real}}$. If $\mathcal{D}_{\text{real}}$ is experimental data with known $P_{W|\mathbf{X}}$, then $Q_{W|\mathbf{X}}$ can be directly set as the true distribution, negating the need for any optimisation at this step.
3. $\boldsymbol{Q_{Y|W,\mathbf{X}}}$. PO estimators are trained to match $P_{Y|W=0,\mathbf{X}}$ and $P_{Y|W=1,\mathbf{X}}$, and the relevant outcome is generated for each instance based on their assigned treatment.

Each component can be defined with any relevant model. $Q_{\mathbf{X}}$ can be any generative model, $Q_{W|\mathbf{X}}$ can be any classifier, and $Q_{Y|W,\mathbf{X}}$ can use any regressors.

6.2 Differential Privacy with STEAM

Theoretical guarantees of the privacy of synthetic data are often required in high-stakes scenarios, such as medicine. STEAM can permit this, satisfying DP when its three component models do, as an application of the post-processing and composition theorems of DP (Dwork & Roth, 2014).

Proposition 1. *If $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, and $Q_{Y|W,\mathbf{X}}$ satisfy $(\epsilon_{\mathbf{X}}, \delta_{\mathbf{X}})$-, (ϵ_W, δ_W)-, and (ϵ_Y, δ_Y)-differential privacy respectively, STEAM satisfies $(\epsilon_{total}, \delta_{total})$-differential privacy, where $\epsilon_{total} = \epsilon_{\mathbf{X}} + \epsilon_W + \epsilon_Y$, $\delta_{total} = \delta_{\mathbf{X}} + \delta_W + \delta_Y$.*

Proof. See Appendix G. □

There are a number of existing DP generative models, classifiers, and regressors which can be set as $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, and $Q_{Y|W,\mathbf{X}}$ respectively to enable this.

7 Empirical Analysis

We now demonstrate the superior performance of STEAM. In Section 7.1, we compare STEAM with generic generation methods in the non-DP setting. In Section 7.2, we examine performance in targeted settings to better understand where STEAM is particularly successful. In Section 7.3 we demonstrate STEAM's capability in satisfying DP generation. To avoid infeasible model selection and unwieldy notation, in STEAM we consistently model $Q_{W|\mathbf{X}}$ using logistic regression, and $Q_{Y|W,\mathbf{X}}$ using T-learner (Künzel et al., 2019) PO estimators. We use the open source `synthcity` (Qian et al., 2023) for all generative models, and we indicate which we set for $Q_{\mathbf{X}}$ in STEAM with subscript, i.e. $\text{STEAM}_{\diamond}$ uses generative model $\diamond$ for $Q_{\mathbf{X}}$. We detail experimental set-ups in Appendix H.

Table 3: $P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$, JSD_π, and U_{PEHE} for the best performing standard and STEAM models on medical data. Full results in Table 11. Averaged over 20 runs, with 95% CIs. Bold indicates significant differences.

Dataset	Model	$P_{\alpha,\mathbf{X}}$ (↑)	$R_{\beta,\mathbf{X}}$ (↑)	JSD_π (↑)	U_{PEHE} (↓)
ACTG	TVAE	0.926 ± 0.013	0.483 ± 0.010	0.946 ± 0.004	0.564 ± 0.017
	STEAM $_{\text{TVAE}}$	0.929 ± 0.008	0.486 ± 0.009	$\mathbf{0.958 \pm 0.004}$	$\mathbf{0.492 \pm 0.011}$
IHDP	CTGAN	0.663 ± 0.018	0.419 ± 0.013	0.888 ± 0.010	2.521 ± 0.161
	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014	0.424 ± 0.011	$\mathbf{0.928 \pm 0.009}$	$\mathbf{1.709 \pm 0.052}$
ACIC	TVAE	0.763 ± 0.011	0.515 ± 0.006	0.926 ± 0.007	4.202 ± 0.134
	STEAM $_{\text{TVAE}}$	0.767 ± 0.009	0.514 ± 0.004	$\mathbf{0.972 \pm 0.002}$	$\mathbf{2.013 \pm 0.112}$

7.1 Generation of medical data containing treatments

Setup

We compare STEAM with state-of-the-art generic tabular data generators in the non-DP setting, across three medical datasets:

1. **AIDS Clinical Trial Group (ACTG) study 175.** A clinical trial on subjects with HIV-1 (Hammer et al., 1996).
2. **Infant Health and Development Program (IHDP).** A semi-synthetic medical dataset, with real covariates and simulated outcomes, using data from a randomised experiment designed to evaluate the effect of specialist childcare on the cognitive test scores of premature infants (Brooks-Gunn et al., 1992).
3. **Atlantic Causal Inference Competition 2016 (ACIC).** A semi-synthetic medical dataset, with real covariates and simulated outcomes, containing data from the Collaborative Perinatal Project (Niswander, 1972).

ACTG allows us to assess performance on real-world medical data, and the IHDP and ACIC will be familiar to the causal inference community. We use baselines across the major families of tabular data generators (CTGAN, TVAE (Xu et al., 2019), TabDDPM (Kotelnikov et al., 2022), ARF (Watson et al., 2023), and normalising flow (Rezende & Mohamed, 2016)).

Takeaway

We display the performance across our metrics of the best performing standard model, and its STEAM analogue, on each dataset in Table 3 (extended results in Appendix I). STEAM and standard generation demonstrate similar performance in terms of desiderata (i), as $P_{\alpha,\mathbf{X}}$ and $R_{\beta,\mathbf{X}}$ are similar across datasets. This is expected, since both methods approach the modelling of $P_{\mathbf{X}}$ similarly. Larger differences occur in terms of desiderata (ii) and (iii). JSD_π and U_{PEHE} are improved by STEAM at a statistically significant level across all datasets, indicating that targeted modelling of $P_{W|\mathbf{X}}$ and $P_{Y|W,\mathbf{X}}$ improves their preservation. The most notable improvement is in the U_{PEHE} metric, which is up to twice as good in STEAM models.

7.2 Comparisons on simulated data

To investigate the performance delta between STEAM and standard generation, we design experiments on simulated data using a DGP with tunable *experimental knobs*, similar to that proposed in Crabbé et al. (2022). Our tunable knobs include covariate dimensionality d, propensity function $\pi : \mathcal{X}^{(d)} \to [0, 1]$, and prognostic and predictive functions $\mu_{\text{prog.}}, \mu_{\text{pred.}} : \mathcal{X}^{(d)} \to \mathbb{R}$.[4] Sample i is generated by drawing $\mathbf{X}^{(i)} \sim \mathcal{N}(0, I_d)$, $W^{(i)} \sim \text{Bern}[\pi(\mathbf{X}^{(i)})]$, and $Y^{(i)} \sim \mathcal{N}(\mu_{\text{prog.}}(\mathbf{X}^{(i)}) + W^{(i)} \cdot \mu_{\text{pred.}}(\mathbf{X}^{(i)}), 1)$. With this DGP, we can assess performance on datasets tailored to specific situations. Across experiments, we consistently compare between TabDDPM and $\text{STEAM}_{\text{TabDDPM}}$, and the default settings for each experimental knob are:

$d = 10$, $\pi(\mathbf{X}) = (1 + e^{-1/2({X_1}^2 + {X_2}^2)})^{-1}$, $\mu_{\text{prog.}}(\mathbf{X}) = {X_1}^2 + {X_2}^2$, $\mu_{\text{pred.}}(\mathbf{X}) = {X_3}^2 + {X_4}^2$

[4] Prognostic variables affect an outcome regardless of treatment, while predictive variables only affect treated outcomes. Prognostic and predictive functions dictate the effect of each covariate on the outcome.

7.2.1 COVARIATE DIMENSIONALITY

Setup

To investigate performance as $\mathcal{D}_{\text{real}}$ increases in dimensionality, we vary $d \in \{5, 10, 20, 50\}$, with all other settings at default.

Takeaway

The performance delta between STEAM and standard generation grows with the dimensionality of $\mathbf{X}$. This follows the intuition that, as d grows, $P_{\mathbf{X}}$ will dominate the joint distribution, and the comparatively small $P_{W|\mathbf{X}}$ and $P_{Y|W,\mathbf{X}}$ will be overlooked by standard models. The top of Figure 1 shows that, as d increases, both $\text{STEAM}_{\text{TabDDPM}}$ and TabDDPM preserve $P_{Y|W,\mathbf{X}}$ worse, however $\text{STEAM}_{\text{TabDDPM}}$ is less affected by d. The bottom of Figure 1 is similar, showing that TabDDPM degrades in performance more than $\text{STEAM}_{\text{TabDDPM}}$ in preserving $P_{W|\mathbf{X}}$ as d grows. Direct modelling with $Q_{W|\mathbf{X}}$ and $Q_{Y|W,\mathbf{X}}$ allows these small, but important, components to be better preserved in high dimensions.

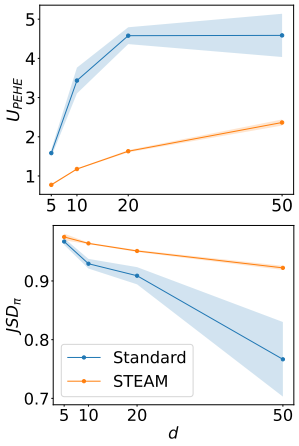

Figure 1: U_{PEHE} (↓) and JSD_π (↑) as d increases. Averaged over 10 runs, shaded area represents 95% CIs.

7.2.2 TREATMENT ASSIGNMENT COMPLEXITY

Setup

To investigate performance as $P_{W|\mathbf{X}}$ increases in complexity, we vary the number of covariates upon which it depends. We set $\pi(\mathbf{X}) = (1 + e^{-1/K \sum_{k=1}^{K} X_k^2})^{-1}$ for $K \in \{1, 2, 3, 4, 5\}$, with all other settings at default.

Takeaway

STEAM increasingly outperforms standard generation in preserving more complex $P_{W|\mathbf{X}}$. Figure 2 shows that, as K increases, $\text{STEAM}_{\text{TabDDPM}}$ maintains a good estimate of $P_{W|\mathbf{X}}$, with JSD_π consistently near 1. On the other hand, the estimate by standard TabDDPM degrades with K, widening the performance gap. Direct modelling allows more complex $P_{W|\mathbf{X}}$ to be preserved.

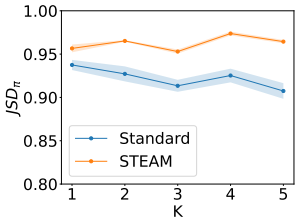

Figure 2: JSD_π (↑) as K increases. Averaged over 10 runs, shaded area represents 95% CIs.

7.2.3 OUTCOME HETEROGENEITY

Setup

To investigate performance as outcomes become increasingly heterogeneous, we vary the number of covariates upon which $P_{Y|W,\mathbf{X}}$ depends. We set $\mu_{\text{pred.}}(\mathbf{X}) = \sum_{k=3}^{K} X_k^2$, $K \in \{3, 4, 5, 6, 7\}$, with all other settings at default.

Takeaway

As $P_{Y|W,\mathbf{X}}$ becomes increasingly heterogeneous, its preservation by $\text{STEAM}_{\text{TabDDPM}}$ degrades slightly, and much more dramatically for TabDDPM, as shown in Figure 3. Again, direct modelling with $Q_{Y|W,\mathbf{X}}$ better preserves complex distributions.

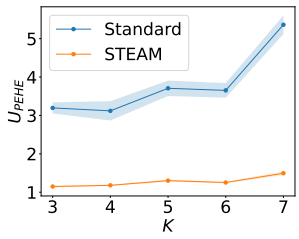

Figure 3: U_{PEHE} (↓) as K increases. Averaged over 10 runs, shaded area represents 95% CIs.

These experiments demonstrate that the performance delta between STEAM and standard generation grows in complex settings. Whether difficulty arises from high-dimensionality, or through complex dependencies in $P_{W|\mathbf{X}}$ or $P_{Y|W,\mathbf{X}}$, STEAM increasingly outperforms in the more difficult scenarios. These situations are likely to emerge in real-world data, which is often highly complex, heightening the relevance of STEAM to the medical setting.

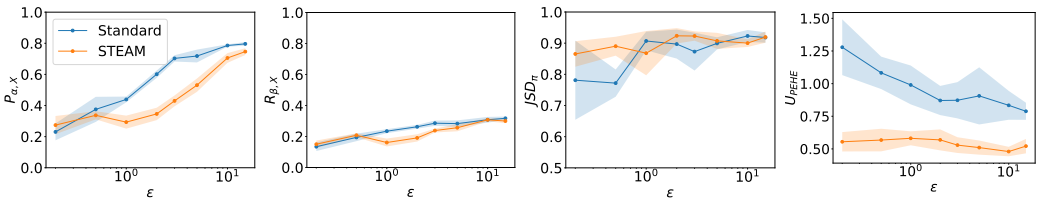

Figure 4: $P_{\alpha,\mathbf{X}}$ (↑), $R_{\beta,\mathbf{X}}$ (↑), JSD_π (↑), and U_{PEHE} (↓) evaluating $\text{STEAM}_{\text{AIM}}$ and standard AIM across privacy budgets. Averaged over 5 runs, shaded area represents 95% CIs.

7.3 Differentially private generation with STEAM

Setup

We examine STEAM's performance in (ϵ, δ)-DP generation. For comparison we use (ϵ, δ)-AIM (McKenna et al., 2024), and for STEAM, we set $Q_{\mathbf{X}}$ as $(\epsilon/3, \delta/3)$-AIM, $Q_{W|\mathbf{X}}$ as an $(\epsilon/3, \delta/3)$-DP random forest, and $Q_{Y|W,\mathbf{X}}$ as an $(\epsilon/3, \delta/3)$-T-Learner, such that STEAM is also (ϵ, δ)-DP. We compare performance on the ACTG dataset across $\epsilon \in \{0.25, 0.5, 1, 2, 3, 5, 10, 15\}$ with $\delta = 10^{-6}$.

Takeaway

Figure 4 shows the results, and results with more baselines are in Appendix I.2. $\text{STEAM}_{\text{AIM}}$ models $P_{Y|W,\mathbf{X}}$ better on all tested values of ϵ, as U_{PEHE} is significantly lower than for standard AIM. $P_{W|\mathbf{X}}$ is better modelled by $\text{STEAM}_{\text{AIM}}$ at small ϵ, with equivalent performance between the methods at less conservative budgets. $P_{\mathbf{X}}$, on the other hand, is better preserved by standard AIM, scoring higher on $P_{\alpha,\mathbf{X}}$ and $R_{\beta,\mathbf{X}}$ at most ϵ. This is likely because assigning $Q_{\mathbf{X}}$ one third of the budget of the standard AIM model and having it model largely the same distribution, save for the removed W and Y, is prohibitively restrictive given the high-dimensionality of $\mathbf{X}$. As such, with uniform distribution of (ϵ, δ) across each component, there is a trade-off between $\text{STEAM}_{\text{AIM}}$ and standard AIM, where $\text{STEAM}_{\text{AIM}}$ better preserves $P_{W|\mathbf{X}}$ and $P_{Y|W,\mathbf{X}}$, while standard AIM preserves $P_{\mathbf{X}}$ better. Distributing (ϵ, δ) differently amongst $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, and $Q_{Y|W,\mathbf{X}}$ could address this trade-off, as we discuss in Section 8 and Appendix M.

8 Discussion

Impact. In this paper, we tackle a problem impeding progress in the causal inference for medicine community—unavailability of data. Existing synthetic data solutions are inadequate, producing poor quality data containing treatments, which are evaluated with misaligned metrics. Our evaluation and generation proposals, grounded in our desiderata which stem from the needs of analysts, remedy this. We enable generation of synthetic data of substantially higher quality, which we demonstrate across a range of experiments in Section 7, as well as in an additional ablative study (Appendix J) and hyperparameter stability study (Appendix K). Furthermore, we allow meaningful evaluation with our metrics, proposed in Section 5, that can answer the key questions ***Q1-3*** of downstream analysts from Section 1. Our paper's impact is heightened by the fact that STEAM increasingly outperforms standard generation in complex situations likely to arise in real-world settings. While we focus on medical data, our methods are also applicable to other fields where data contain treatments, such as education, marketing, and public policy, broadening our impact.

Limitations. STEAM has room for refinement. Uniform distribution of the privacy budget across $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, and $Q_{Y|W,\mathbf{X}}$ during DP generation, as we do in Section 7.3, is sub-optimal, as particular component models may benefit from a larger share of ϵ depending on their importance and complexity (Appendix M). Also, generative models, used to model $Q_{\mathbf{X}}$, can struggle when covariate shift is high (Appendix N). This does not uniquely affect STEAM, as it occurs under standard generation as well, however it is important to acknowledge that poor performance may occur in this setting.

Future work. There are many future research directions in this setting. These include improving the limitations discussed above, and examining further applications of STEAM. For example, in this work we focus on static medical data, and longitudinal data, with continuous measurement of covariates, treatments, and outcomes, may require further novel thought (Appendix P).

REFERENCES

Haleh Akrami, Sergul Aydore, Richard M. Leahy, and Anand A. Joshi. Robust variational autoencoder for tabular data with beta divergence, 2020.

Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 290–306. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/alaa22a.html.

George J Annas. Hipaa regulations: a new era of medical-record privacy? *New England Journal of Medicine*, 348:1486, 2003.

Peter C Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behav Res*, 46(3):399–424, June 2011.

Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*, pp. 457–467. PMLR, 2021.

Howard Bauchner, Robert M. Golub, and Phil B. Fontanarosa. Data Sharing: An Ethical and Scientific Imperative. *JAMA*, 315(12):1238–1240, 03 2016. ISSN 0098-7484. doi: 10.1001/jama.2016.2420. URL https://doi.org/10.1001/jama.2016.2420.

André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. Comprehensive exploration of synthetic data generation: A survey, 2024. URL https://arxiv.org/abs/2401.02524.

Reza Bayat. A study on sample diversity in generative models: Gans vs. diffusion models. In *URL https://openreview. net/forum*, 2023.

Patrick Blöbaum, Peter Götz, Kailash Budhathoki, Atalanti A Mastakouri, and Dominik Janzing. Dowhy-gcm: An extension of dowhy for causal inference in graphical causal models. *Journal of Machine Learning Research*, 25(147):1–7, 2024.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators, 2023. URL https://arxiv.org/abs/2210.06280.

Jeanne Brooks-Gunn, Fong-ruey Liaw, and Pamela Kato Klebanov. Effects of early intervention on cognitive function of low birth weight preterm infants. *The Journal of pediatrics*, 120(3):350–359, 1992.

Patrick Chao, Patrick Blöbaum, Sapan Patel, and Shiva Prasad Kasiviswanathan. Modeling causal mechanisms with diffusion models for interventional and counterfactual queries, 2024. URL https://arxiv.org/abs/2302.00860.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks, 2018. URL https://arxiv.org/abs/1703.06490.

Jonathan Crabbé, Alicia Curth, Ioana Bica, and Mihaela van der Schaar. Benchmarking heterogeneous treatment effect models through the lens of interpretability. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 12295–12309. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4fd7b4ed13f78b9ba7afcd9d01615896-Paper-Datasets_and_Benchmarks.pdf.

Alicia Curth and Mihaela van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1810–1818. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/curth21a.html.

Alicia Curth and Mihaela Van Der Schaar. In search of insights, not magic bullets: Towards demystification of the model selection dilemma in heterogeneous treatment effect estimation. In *International Conference on Machine Learning*, pp. 6623–6642. PMLR, 2023.

Alicia Curth, David Svensson, Jim Weatherall, and Mihaela van der Schaar. Really doing great at estimating cate? a critical look at ml benchmarking practices in treatment effect estimation. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*, 2021.

Rajeev H Dehejia and Sadek Wahba. Causal effects in non-experimental studies: Re-evaluating the evaluation of training programs. Working Paper 6586, National Bureau of Economic Research, June 1998. URL http://www.nber.org/papers/w6586.

Rajeev H. Dehejia and Sadek Wahba. Propensity Score-Matching Methods for Nonexperimental Causal Studies. *The Review of Economics and Statistics*, 84(1):151–161, 02 2002. ISSN 0034-6535. doi: 10.1162/003465302317331982. URL https://doi.org/10.1162/003465302317331982.

Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition, 2018.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Khaled El Emam, Lucy Mosquera, and Chaoyi Zheng. Optimizing the synthesis of clinical trial data using sequential trees. *Journal of the American Medical Informatics Association*, 28(1):3–13, 2021.

Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans, 2017.

Fiona Katharina Ewald, Ludwig Bothmann, Marvin N. Wright, Bernd Bischl, Giuseppe Casalicchio, and Gunnar König. *A Guide to Feature Importance Methods for Scientific Inference*, pp. 440–464. Springer Nature Switzerland, 2024. ISBN 9783031637971. doi: 10.1007/978-3-031-63797-1_22. URL http://dx.doi.org/10.1007/978-3-031-63797-1_22.

Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 30(4):958–968, Apr 2024. ISSN 1546-170X. doi: 10.1038/s41591-024-02902-1. URL https://doi.org/10.1038/s41591-024-02902-1.

Dennis Frauen and Stefan Feuerriegel. Estimating individual treatment effects under unobserved confounding using binary instruments. *arXiv preprint arXiv:2208.08544*, 2022.

Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023.

Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. Generation and evaluation of synthetic patient data. *BMC medical research methodology*, 20:1–40, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Scott M. Hammer, David A. Katzenstein, Michael D. Hughes, Holly Gundacker, Robert T. Schooley, Richard H. Haubrich, W. Keith Henry, Michael M. Lederman, John P. Phair, Manette Niu, Martin S. Hirsch, and Thomas C. Merigan. A trial comparing nucleoside monotherapy with combination therapy in hiv-infected adults with cd4 cell counts from 200 to 500 per cubic millimeter. *New England Journal of Medicine*, 335(15):1081–1090, 1996. doi: 10.1056/NEJM199610103351501. URL https://www.nejm.org/doi/full/10.1056/NEJM199610103351501.

Lasse Hansen, Nabeel Seedat, Mihaela van der Schaar, and Andrija Petrovic. Reimagining synthetic tabular data generation through data-centric ai: A comprehensive benchmark, 2023. URL https://arxiv.org/abs/2310.16981.

Tobias Hatt, Jeroen Berrevoets, Alicia Curth, Stefan Feuerriegel, and Mihaela van der Schaar. Combining observational and randomized data for estimating heterogeneous treatment effects, 2022. URL https://arxiv.org/abs/2202.12891.

Erik Hermansson and David Svensson. *On Discovering Treatment-Effect Modifiers Using Virtual Twins and Causal Forest ML in the Presence of Prognostic Biomarkers*, pp. 624–640. Springer-Verlag, 09 2021. ISBN 978-3-030-86972-4. doi: 10.1007/978-3-030-86973-1_44.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. doi: 10.1109/MSP.2012.2205597.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81 (396):945–960, 1986.

Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: the IBM differential privacy library. *ArXiv e-prints*, 1907.02444 [cs.CR], July 2019.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Alihan Hüyük, Qiyao Wei, Alicia Curth, and Mihaela van der Schaar. Defining expertise: Applications to treatment effect estimation, 2024.

Aryan Jadon and Shashank Kumar. Leveraging generative ai models for synthetic data generation in healthcare: Balancing research and privacy. In *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, July 2023. doi: 10.1109/smartnets58706.2023.10215825. URL http://dx.doi.org/10.1109/SmartNets58706.2023.10215825.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2018. URL https://api.semanticscholar.org/CorpusID:53342261.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic data–what, why and how? *arXiv preprint arXiv:2205.03257*, 2022.

Nathan Kallus, Xiaojie Mao, and Angela Zhou. Interval estimation of individual-level causal effects under unobserved confounding. In *The 22nd international conference on artificial intelligence and statistics*, pp. 2281–2290. PMLR, 2019.

L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960. doi: 10.1287/mnsc.6.4.366. URL https://doi.org/10.1287/mnsc.6.4.366.

Dhamanpreet Kaur, Matthew Sobiesk, Shubham Patil, Jin Liu, Puran Bhagat, Amar Gupta, and Natasha Markuzon. Application of bayesian networks to generate synthetic health data. *Journal of the American Medical Informatics Association*, 28, 12 2020. doi: 10.1093/jamia/ocaa303.

Edward H Kennedy et al. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 5, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

Aneesh Komanduri, Xintao Wu, Yongkai Wu, and Feng Chen. From identifiable causal representations to controllable counterfactual generation: A survey on causal generative modeling, 2024. URL https://arxiv.org/abs/2310.11011.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models, 2022.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pp. 604–620, 1986.

Dongha Lee, Hwanjo Yu, Xiaoqian Jiang, Deevakar Rogith, Meghana Gudala, Mubeen Tejani, Qiuchen Zhang, and Li Xiong. Generating sequential electronic health records using dual adversarial autoencoder. *Journal of the American Medical Informatics Association*, 27(9):1411–1419, 2020.

Fan Li, Laine E Thomas, and Fan Li. Addressing Extreme Propensity Scores via the Overlap Weights. *American Journal of Epidemiology*, 188(1):250–257, 09 2018. ISSN 0002-9262. doi: 10.1093/aje/kwy201. URL https://doi.org/10.1093/aje/kwy201.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Terrance Liu, Giuseppe Vietri, and Steven Z Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34: 690–702, 2021.

Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models, 2017.

Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *arXiv preprint arXiv:2201.12677*, 2022.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: An adaptive and iterative mechanism for differentially private synthetic data, 2024. URL https://arxiv.org/abs/2201.12677.

Hajra Murtaza, Musharif Ahmed, Naurin Farooq Khan, Ghulam Murtaza, Saad Zafar, and Ambreen Bano. Synthetic data generation: State of the art in health care domain. *Computer Science Review*, 48:100546, 2023.

Kenneth R Niswander. The collaborative perinatal study of the national institute of neurological diseases and stroke. *The Woman and Their Pregnancies*, 1972.

Ke Pan, Yew-Soon Ong, Maoguo Gong, Hui Li, A.K. Qin, and Yuan Gao. Differential privacy in deep learning: A literature survey. *Neurocomputing*, 589:127663, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2024.127663. URL https://www.sciencedirect.com/science/article/pii/S092523122400434X.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, June 2018. ISSN 2150-8097. doi: 10.14778/3231751.3231757. URL http://dx.doi.org/10.14778/3231751.3231757.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Maya L Petersen, Kristin E Porter, Susan Gruber, Yue Wang, and Mark J van der Laan. Diagnosing and responding to violations in the positivity assumption. *Stat Methods Med Res*, 21(1):31–54, October 2010.

Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023. URL https://arxiv.org/abs/2301.07573.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016. URL https://arxiv.org/abs/1505.05770.

Kara E Rudolph, Catherine Gimbrone, Ellicott C Matthay, Iván Díaz, Corey S Davis, Katherine Keyes, and Magdalena Cerdá. When effects cannot be estimated: Redefining estimands to understand the effects of naloxone access laws. *Epidemiology*, 33(5):689–698, June 2022.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/f7696a9b362ac5a51c3dc8f098b73923-Paper.pdf.

Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Designing variational graph autoencoders for causal queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8159–8168, 2022.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pp. 3076–3085. PMLR, 2017. URL https://proceedings.mlr.press/v70/shalit17a.html.

Claudia Shi, David Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32, 2019.

Tao Shi and Steve Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138, 2006.

Yishai Shimoni, Ehud Karavani, Sivan Ravid, Peter Bak, Tan Hung Ng, Sharon Hensley Alford, Denise Meade, and Yaara Goldschmidt. An evaluation toolkit to guide model selection and cohort definition in causal inference, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

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

Allan Tucker, Zhenchen Wang, Ylenia Rotalinti, and Puja Myles. Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. *NPJ digital medicine*, 3(1):1–13, 2020.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

David S. Watson, Kristin Blesch, Jan Kapar, and Marvin N. Wright. Adversarial random forests for density estimation and generative modeling, 2023.

Felix Nikolaus Wirth, Thierry Meurers, Marco Johns, and Fabian Prasser. Privacy-preserving data sharing infrastructures for medical research: systematization and comparison. *BMC Medical Informatics and Decision Making*, 21:1–13, 2021.

Robert F Wolff, Karel G M Moons, Richard D Riley, Penny F Whiting, Marie Westwood, Gary S Collins, Johannes B Reitsma, Jos Kleijnen, Sue Mallett, and PROBAST Group†. PROBAST: A tool to assess the risk of bias and applicability of prediction model studies. *Ann Intern Med*, 170 (1):51–58, January 2019.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network, 2018.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan, 2019.

Chao Yan, Yao Yan, Zhiyu Wan, Ziqi Zhang, Larsson Omberg, Justin Guinney, Sean D Mooney, and Bradley A Malin. A multifaceted benchmarking of synthetic electronic health record generation models. *Nature communications*, 13(1):7609, 2022.

Jinsung Yoon, Lydia N Drumright, and Mihaela van der Schaar. Anonymization through data synthesis using generative adversarial networks (ADS-GAN). *IEEE J Biomed Health Inform*, 24 (8):2378–2388, March 2020.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

A EXAMPLES OF MISUSE OF MEDICAL DATA CONTAINING TREATMENTS

Here, we detail a select few synthetic data papers which use medical data, containing treatments, to demonstrate on, yet they do not consider the downstream task of causal inference and how their methods may need to be altered for this. Note that we do not claim that this list is exhaustive, as this is a pervasive problem in the synthetic data literature, and we mean only to provide a few examples here to demonstrate this problem, and provide motivation for our paper.

A.1 MEDGAN

In the paper 'Generating Multi-label Discrete Patient Records using Generative Adversarial Networks' (Choi et al., 2018), the authors propose a GAN-based approach to generate 'realistic synthetic patient records'. In doing so, they experiment on multiple datasets containing treatments, including one from Sutter Palo Alto Medical Foundation (PAMF) which consists of longitudinal medical records of 258,000 patients, as well as the MIMIC-III dataset (Johnson et al., 2016), which includes 46,000 intensive care unit patient records. Given the nature of these datasets, they both include treatments administered to patients, and they therefore invite downstream analysts to conduct causal inference tasks, such as treatment effect estimation.

Nevertheless, in this paper, standard generation and evaluation practices are followed, not differentiating between covariates, treatments, and outcomes. In particular, the evaluation protocol relies on marginal comparisons, and predictive utility assessment, which offers limited information as to how well the generation method, `medGAN`, produces useful data for causal inference. Furthermore, `medGAN` itself draws samples directly from the joint distribution, not mimicking the DGP of treatment data, or optimising for the distributions most important for causal inference.

A.2 TABDDPM

The paper 'TabDDPM: Modelling Tabular Data with Diffusion Models' (Kotelnikov et al., 2022) proposes a diffusion-based tabular data generation method. While not explicitly geared towards medical data, this paper does use medical data with variables that could be seen as treatments in its experiments. It therefore, at least implicitly, positions itself to work on data containing treatments, and invites users to conduct generation with `TabDDPM` on such data. Specifically, the cardiovascular disease dataset from `https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset` is used, and this data could be analysed via causal inference by setting 'physical activity' as a treatment, to estimate its effect on cardiovascular disease.

However, in this paper, only standard evaluation and generation methods are used, and the needs of downstream analysts pursuing causal inference tasks are not acknowledged. Evaluation involves only predictive utility measures, and the `TabDDPM` method generates all variables in a sample simultaneously, not optimising for the distributions most important for causal inference.

A.3 GREAT

The paper 'Language Models are Realistic Tabular Data Generators' (Borisov et al., 2023) proposes an LLM-based generator. Similar to the `TabDDPM` paper, this paper is not explicitly geared towards medical data, but it demonstrates on medical data containing treatments, thereby implicitly condoning its use on this type of data. Specifically, the dataset `sick` from `https://www.openml.org/search?type=data&sort=runs&id=38&status=active` is demonstrated on, which could be analysed via causal inference to assess the effect of 'thyroxine' or 'antithyroid' treatments. Nevertheless, once again this paper does not consider downstream analysis involving causal inference, only evaluating its `GReaT` method with predictive utility metrics and a discriminator score.

A.4 BENCHMARKING PROCESS FOR SYNTHETIC ELECTRONIC HEALTH RECORDS

Finally, the paper 'A Multifaceted benchmarking of synthetic electronic health record generation models' proposes a benchmarking framework for use on synthetic electronic health record (EHR) data (Yan et al., 2022). Naturally, EHRs will include treatments administered to patients, and they will likely be analysed with treatment effect estimation in mind. In the proposed benchmarking

framework, the evaluation procedures—including marginal comparison, correlation comparison, and predictive utility—do not differentiate between covariates, treatments, and outcomes, or acknowledge the needs of downstream analysts conducting causal inference.

B EXTENDED LITERATURE REVIEW

To provide useful context for readers, we extend on our literature review here.

Evaluation. We extend on the synthetic data evaluation practices summarised in Table 1 here.

Marginal comparison. Assessing the distributional distance between synthetic and real marginals is often used to offer a quantitative assessment of how well individual variables are modelled (Yan et al., 2022; Tucker et al., 2020; Goncalves et al., 2020). The distance function d to conduct this can be set from a variety of choices, including including KL divergence (Kullback & Leibler, 1951), Jensen-Shannon distance (Lin, 1991), Wasserstein distance (Kantorovich, 1960), Kolmogorov-Smirnov score (Massey Jr, 1951), MMD (Gretton et al., 2012), and many more.

Correlation matrix comparison. Correlation-based assessment can offer a sense of how well inter-dependencies between variables are modelled in synthetic data (Murtaza et al., 2023). This commonly involves calculating synthetic and real 2-way correlation matrices, and assessing their difference, by setting d as a distance such as Frobenius norm (Goncalves et al., 2020) and absolute error (Kaur et al., 2020).

Joint distribution comparison. Metrics based on notions of statistical divergence can offer a means of quantifying how different the entire joint distributions of real and synthetic data are (Yoon et al., 2020; Tucker et al., 2020; Torfi et al., 2022). The distance function d can be set to largely to the same family of functions as in the marginal comparison case.

Precision and recall analysis. Precision and recall, originally proposed for generative model assessment in Sajjadi et al. (2018), measure if generated samples are covered by real samples, and vice versa. Alpha precision and beta recall (Alaa et al., 2022) are refined versions of the original metrics which account for the densities of the real and generative distributions, rather than just comparing supports.

Discriminator performance. Discriminator performance is a slightly unique evaluation practice, involving a 'discriminator', which predicts whether instances are synthetic or real, where poor performance of the discriminator indicates realism in the synthetic data (Kaur et al., 2020; Lee et al., 2020; Emam et al., 2021; Borisov et al., 2023).

Predictive utility. Predictive utility metrics offer a practical evaluation of synthetic data by quantifying the performance of a synthetically-trained predictive model. The "train on synthetic, test on real" (TSTR) paradigm (Esteban et al., 2017) is the common approach to such assessment, measuring the accuracy of a synthetically-trained model in predicting a target label on a real test set.

Generic generative models. We expand on the generic generative modelling paradigm outlined in Table 2 by describing specific existing generative models which adhere to it. These models approximate the real joint distribution using a diverse range of techniques.

GAN-based models. GANs (Goodfellow et al., 2020) consist of a generator and discriminator network, which are trained adversarially to generate and identify synthetic data, respectively, until the samples are realistic. Originally proposed for image generation, GANs have been adapted to tabular generation, and there are many methods which adopt this popular architecture (e.g. CTGAN (Xu et al., 2019), TableGAN (Park et al., 2018)), including those specifically designed for medical data (e.g. MedGAN (Choi et al., 2018)).

VAE-based models. VAEs (Kingma & Welling, 2022) are another common architecture, which learn to encode data into a lower-dimensional latent space and then decode it back to reconstruct the original data. They generate new data samples by sampling from the latent distribution and decoding these samples, and their application to tabular data involve techniques to handle mixed data types (e.g.TVAE (Xu et al., 2019)), and regularisation for improved robustness (e.g. RTVAE Akrami et al. (2020)).

Diffusion-based models. Diffusion models (Ho et al., 2020; Song et al., 2020) learn the gradient of the data distribution, and generate data via progressive denoising, beginning with a noisy sample and using a neural network to predict and remove noise over a number of timesteps, including for tabular data (e.g. TabDDPM (Kotelnikov et al., 2022)).

Forest-based models. Random forests can estimate the density of a probability distribution, as leaf nodes partition the data space into distinct hyper-rectangles with estimated densities of the proportion of samples which fall into them. Samples can then be drawn from this estimated density. Random forests can easily handle heteorgeneous data types, so their application to tabular data synthesis is natural. They are particularly fast to train and generate from (Shi & Horvath, 2006; Watson et al., 2023)).

Normalizing flow-based models. Normalizing flows estimate the target density by transforming a tractable density (e.g. a Gaussian) into the target through a series of invertible transformations, called 'flows'. Probabilities from the target distribution can then be found using the change of variables formula (e.g. Rezende & Mohamed (2016)). Recent work has shown their theoretical similarity to diffusion models (Lipman et al., 2023).

Causal generative models. Causal generative models (Komanduri et al., 2024; Blöbaum et al., 2024; Chao et al., 2024) are a class of generative model, distinct from generic tabular data generators, that approximate the underlying structural causal model (SCM) (Pearl, 2009) of a dataset. While such models are related to our work with STEAM, and are likely to better preserve causal relationships than generic generators in settings where they can be used, their assumptions can restrict their practical use cases. In comparison to STEAM, they generally differ in terms of their (1) assumptions, (2) motivation, and (3) flexibility, which we detail here.

(1): Importantly, causal generative models typically assume that the data holder has knowledge of the entire causal graph of the real data, which is a more restrictive assumption than we make in this work. We assume that our specification of the underlying DGP $\mathbf{X} \sim P_{\mathbf{X}},\ W \sim P_{W|\mathbf{X}},\ Y \sim P_{Y|W,\mathbf{X}}$ is correct for datasets containing treatments, but we do not require knowledge of the causal graph, as we do not need to know the causal links between individual variables. We do not assume knowledge of the causal relationships amongst the covariates, nor knowledge of which covariates cause treatment assignment or patient outcomes. As such, we make less restrictive assumptions than works which require knowledge of a causal graph, and we suggest that our approach is more realistic in complex, real-world scenarios, such as those that arise in medicine, where the true causal graph is unlikely to be available.

(2): The motivation for causal generative models is typically to allow generation of data to answer graph-specific interventional and counterfactual queries, that require knowledge of the full causal graph. With STEAM, however, we seek to generate useful synthetic data only from the observational distribution, for use by analysts with goals such as treatment effect estimation (e.g., CATE).

(3): STEAM's design can, in principle, incorporate any generative model for $Q_{\mathbf{X}}$, essentially acting as a wrapper around $Q_{\mathbf{X}}$ to improve its generation quality for causal inference tasks. This allows STEAM to very easily empower many existing generative modelling frameworks, without having to incorporate bespoke generators. Also, it allows STEAM models to continuously improve along with the base generative model. Existing causal generative models do not generally allow such flexibility, and therefore cannot as easily benefit from future improvements in underlying generic generative models.

Despite these differences, we conduct empirical comparisons between STEAM and some baseline causal generative models in Appendix O.

Privacy. Despite the popularity of some of the above generators, memorisation of training samples is a phenomenon observed in generative models (Ghalebikesabi et al., 2023). Therefore, provably private generation is often desired to limit the amount of information leaked. Differential privacy Dwork et al. (2006) is the most common standard adopted, and there are multiple generators which guarantee this, including GAN-based methods (e.g. PATE-GAN (Jordon et al., 2018), DP-GAN (Xie

et al., 2018)) and query-based methods (e.g. GEM (Liu et al., 2021), MST (McKenna et al., 2021), RAP (Aydore et al., 2021), AIM (McKenna et al., 2022)).

C THEOREM 1 PROOF

Theorem 1. *Let P, Q^{θ_1}, Q^{θ_2} be of the form described in Section 5.1, and $\mathcal{M}$ be KL divergence. If we assume that Q^{θ_1} and Q^{θ_2} have sufficient capacity to have bounded error on each component, i.e. $\forall i$, $0 < \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i}) < \varepsilon_{\mathbf{X}}$, and $0 < \mathcal{M}(P_{W|\mathbf{X}}, Q^{\theta_{W,k}}_{W|\mathbf{X}}) < \varepsilon_{W,k}$, and $0 < \mathcal{M}(P_{Y|W,\mathbf{X}}, Q^{\theta_{Y,k}}_{Y|W,\mathbf{X}}) < \varepsilon_{Y,k}$, then:*

$$\frac{\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_1}_{\mathbf{X},W,Y})}{\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_2}_{\mathbf{X},W,Y})} \to 1, \text{ as } d \to \infty \tag{7}$$

Proof. From the factorizations of P and Q, KL divergence decomposes:

$$D_{\mathrm{KL}}(P\|Q^{\theta_k}) = \sum_{i=1}^{d} D_{\mathrm{KL}}(P_{\mathbf{X}_i}\|Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i}) + \mathbb{E}_{P_{\mathbf{X}}}\left[D_{\mathrm{KL}}(P_{W|\mathbf{X}}\|Q^{\theta_{W,k}}_{W|\mathbf{X}})\right] + \mathbb{E}_{P_{\mathbf{X},W}}\left[D_{\mathrm{KL}}(P_{Y|W,\mathbf{X}}\|Q^{\theta_{Y,k}}_{Y|W,\mathbf{X}})\right] \tag{8}$$

As such, the following holds for when KL divergence is set as $\mathcal{M}$.

Define the ratio:

$$R(d) = \frac{\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_1}_{\mathbf{X},W,Y})}{\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_2}_{\mathbf{X},W,Y})}.$$

Substituting the decompositions, we have:

$$R(d) = \frac{\sum_{i=1}^{d} \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i}) + \mathbb{E}_{P_X}\left[\mathcal{M}(P_{W|\mathbf{X}}, Q^{\theta_{W,1}}_{W|\mathbf{X}})\right] + \mathbb{E}_{P_{X,W}}\left[\mathcal{M}(P_{Y|W,\mathbf{X}}, Q^{\theta_{Y,1}}_{Y|W,\mathbf{X}})\right]}{\sum_{i=1}^{d} \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i}) + \mathbb{E}_{P_X}\left[\mathcal{M}(P_{W|\mathbf{X}}, Q^{\theta_{W,2}}_{W|\mathbf{X}})\right] + \mathbb{E}_{P_{X,W}}\left[\mathcal{M}(P_{Y|W,\mathbf{X}}, Q^{\theta_{Y,2}}_{Y|W,\mathbf{X}})\right]}.$$

As the dimensionality d increases, the marginal summations $\sum_{i=1}^{d} \mathcal{M}(P_{\mathbf{X}_i}\|Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i})$ grow linearly with d, since each $\mathcal{M}(P_{\mathbf{X}_i}\|Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i})$ is, by assumption, non-negative, and they therefore dominate the bounded conditional contributions:

$$\mathbb{E}_{P_X}\left[\mathcal{M}(P_{W|\mathbf{X}}, Q^{\theta_{W,k}}_{W|\mathbf{X}})\right] < \varepsilon_{W,k},$$

$$\mathbb{E}_{P_{X,W}}\left[\mathcal{M}(P_{Y|W,\mathbf{X}}, Q^{\theta_{Y,k}}_{Y|W,\mathbf{X}})\right] < \varepsilon_{Y,k}.$$

Thus, $\mathcal{M}(P_{\mathbf{X},W,Y}, Q^{\theta_k}_{\mathbf{X},W,Y}) \sim \sum_{i=1}^{d} \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i})$ and $R(d) \sim \frac{\sum_{i=1}^{d} \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i})}{\sum_{i=1}^{d} \mathcal{M}(P_{\mathbf{X}_i}, Q^{\theta_{\mathbf{X}}}_{\mathbf{X}_i})} = 1.$

Therefore:

$$R(d) \to 1, \text{ as } d \to \infty$$

□

D EMPIRICAL DEMONSTRATIONS OF CURRENT METRIC FAILURE

Table 4: Joint-distribution-level metrics on $\mathcal{D}^i_{\text{synth}}$ which differ in $Q^i_{Y|W,\mathbf{X}}$ architecture only. Averaged over 10 runs, with 95% CIs.

$Q^i_{Y\|W,\mathbf{X}}$	P_α (↑)	R_β (↑)	Inv. KL (↑)	KS (↑)	WD (↓)	JSD (↓)	Oracle (↓)
T-Learner	0.927 ± 0.001	0.584 ± 0.006	0.947 ± 0.000	0.979 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.525 ± 0.012
TARNet	0.919 ± 0.002	0.573 ± 0.005	0.950 ± 0.006	0.985 ± 0.001	0.002 ± 0.000	0.002 ± 0.000	0.616 ± 0.015
DragonNet	0.921 ± 0.001	0.574 ± 0.004	0.947 ± 0.000	0.984 ± 0.001	0.002 ± 0.000	0.002 ± 0.000	0.618 ± 0.007
S-Learner	0.926 ± 0.002	0.579 ± 0.007	0.957 ± 0.009	0.990 ± 0.000	0.002 ± 0.000	0.001 ± 0.000	1.279 ± 0.015

D.1 Failure to identify changes to the outcome generation mechanism

We demonstrate this with a simple experiment investigating how four $\mathcal{D}_{\text{synth}}$ of size $n = 1000$, which only differ in their outcome generation mechanisms, are assessed by an array of current metrics. We simulate $\mathcal{D}_{\text{real}}$ from a simple DGP with 10 covariates with $P_{\mathbf{X}} = \mathcal{N}(0, I)$, $P_{W|\mathbf{X}} = \text{Bern}(0.5)$, $P_{Y|W,\mathbf{X}} = \mathcal{N}(W \cdot {X_1}^2, 1)$. We generate four $\mathcal{D}^i_{\text{synth}}$ with the same $Q^i_{\mathbf{X}} \overset{d}{=} P_{\mathbf{X}}$, $Q^i_{W|\mathbf{X}} \overset{d}{=} P_{W|\mathbf{X}}$, $\forall i \in \{1, 2, 3, 4\}$. We vary each $Q^i_{Y|W,\mathbf{X}} \sim \mathcal{N}(W \cdot \Phi_i(\mathbf{X}, 1) + (1 - W) \cdot \Phi_i(\mathbf{X}, 0), 1)$ where Φ_i represents a potential outcome (PO) estimator with the architecture from either an S-Learner, T-Learner (Künzel et al., 2019), DragonNet (Shi et al., 2019), or TARNet (Shalit et al., 2017). These four architectures will model $Q^i_{Y|W,X}$ differently, inducing the only point of variation amongst the $\mathcal{D}^i_{\text{synth}}$.

Since we simulate $\mathcal{D}_{\text{real}}$, we know the ground-truth treatment effects, and an oracle metric can be established to determine the true quality of each $\mathcal{D}^i_{\text{synth}}$. We define this as the precision of estimating heterogeneous effects (PEHE) (Hill, 2011) of estimates from a CATE learner trained on $\mathcal{D}^i_{\text{synth}}$ and the ground-truth CATEs. In Table 4 we report the scores of P_α, R_β (Alaa et al., 2022), inverse KL divergence (Kullback & Leibler, 1951), Kolmogorov-Smirnov (KS) score (Massey Jr, 1951), Wasserstein distance (WD) (Kantorovich, 1960), and Jensen-Shannon distance (JSD) (Lin, 1991) on each $\mathcal{D}^i_{\text{synth}}$. All report very similar scores across the $\mathcal{D}^i_{\text{synth}}$, with most offering no statistically significant best option, suggesting that their quality is the same. The oracle metric, however, determines that $\mathcal{D}^i_{\text{synth}}$ using a T-Learner for Φ_i is a clear best, and $\mathcal{D}^i_{\text{synth}}$ with an S-Learner as Φ_i is more than twice as bad at preserving the true treatment effects. Clearly, even in a moderately sized dataset, these metrics cannot reliably identify changes in $Q^i_{Y|W,\mathbf{X}}$, despite the large effect that this distribution has on downstream performance.

Comparison to U_{PEHE} Since we only alter $Q^i_{Y|W,\mathbf{X}}$ between each $\mathcal{D}^i_{\text{synth}}$, all have the same $P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$, and JSD_π. In Table 5 we report U_{PEHE} on each dataset, and we see that it fully reproduces the oracle ranking, and correctly identifies the best dataset to a statistically significant level, which no existing metric could do.

Table 5: U_{PEHE} on $\mathcal{D}^i_{\text{synth}}$ with varied $Q^i_{Y|W,\mathbf{X}}$. Averaged over 10 runs, with 95% CIs.

$Q^i_{Y\|W,\mathbf{X}}$	U_{PEHE} (↓)	Oracle (↓)
T-Learner	0.693 ± 0.013	0.525 ± 0.012
TARNet	0.731 ± 0.016	0.616 ± 0.015
DragonNet	0.754 ± 0.019	0.618 ± 0.007
S-Learner	0.906 ± 0.019	1.279 ± 0.015

D.2 Failure to identify changes to the treatment assignment mechanism

We conduct a similar experiment varying $Q^i_{W|\mathbf{X}}$ across three $\mathcal{D}^i_{\text{synth}}$. We simulate $\mathcal{D}_{\text{real}} \sim P_{\mathbf{X},W,Y}$ from a DGP with 5 covariates, all of which contribute to the propensity score. We set $P_{\mathbf{X}} = \mathcal{N}(0, I)$, $P_{W|\mathbf{X}} = \text{Bern}(\pi(\mathbf{X}))$, $\pi(\mathbf{X}) = (1 + e^{-1/5 \sum_{i=1}^5 X_i})^{-1}$, $P_{Y|W,\mathbf{X}} = \mathcal{N}(0, 1)$. We generate three $\mathcal{D}^i_{\text{synth}} \sim Q^i_{\mathbf{X},W,Y}$ which vary only in the degree to which they correctly model $\pi(\mathbf{X})$ by setting $Q^i_{\mathbf{X}} \overset{d}{=} P_{\mathbf{X}}$, $Q^i_{Y|W,\mathbf{X}} \overset{d}{=} P_{Y|W,\mathbf{X}}$, $\forall i \in \{1, 2, 3\}$ and $Q^i_{W|\mathbf{X}} = \text{Bern}(\pi_i(\mathbf{X}))$ where $\pi_1(\mathbf{X}) = (1 + e^{-X_1})^{-1}$, $\pi_2(\mathbf{X}) = (1 + e^{-1/3 \sum_{i=1}^3 X_i})^{-1}$, and $\pi_3(\mathbf{X}) = (1 + e^{-1/5 \sum_{i=1}^5 X_i})^{-1}$. In this way, we know that, in truth, $Q^3_{\mathbf{X},W,Y}$ is a better model than $Q^2_{\mathbf{X},W,Y}$, which in turn is better than $Q^1_{\mathbf{X},W,Y}$, and we can now assess how well existing metrics, and our JSD_π metric, recover this ranking.

We display the scores of P_α, R_β, inverse KL, Kolmogorov-Smirnov score, Wasserstein distance, Jensen-Shannon distance, and our metric JSD_π on each $\mathcal{D}^i_{\text{synth}}$ in Table 6. We see that the existing metrics report very similar scores across the three datasets, and none offer a statistically significant

Table 6: # correct var.: The number of correctly identified variables in the propensity score. P_α: α precision. R_β: β recall. Inv. KL: Inverse KL divergence. KS: Kolmogorov-Smirnov score. WD: Wasserstein distance. JSD: Jensen-Shannon distance. JSD_π: Ours. Averaged over 10 runs, with 95% CIs.

# correct var.	P_α (↑)	R_β (↑)	Inv. KL (↑)	KS (↑)	WD (↓)	JSD (↓)	JSD_π (↑)
5	0.863 ± 0.024	0.456 ± 0.017	0.989 ± 0.002	0.965 ± 0.003	0.017 ± 0.001	0.004 ± 0.000	0.963 ± 0.007
3	0.868 ± 0.022	0.457 ± 0.012	0.981 ± 0.006	0.966 ± 0.003	0.018 ± 0.001	0.004 ± 0.000	0.942 ± 0.006
1	0.866 ± 0.021	0.453 ± 0.013	0.985 ± 0.003	0.965 ± 0.003	0.018 ± 0.001	0.004 ± 0.000	0.908 ± 0.013

best option. This contrasts with our JSD_π metric, which correctly orders the three models, and selects $Q^3_{\mathbf{X},W,Y}$ as the best option to a statistically significant level.

To further elucidate the differences in rankings between existing and our metrics, both in this experiment and the outcome generation comparison in Section 5, we list each ranking and their Spearman's rank correlation coefficient with the oracle ranking in Tables 7 and 8. Assessment via our metrics is the only protocol that reproduces the oracle ranking across both experiments.

Table 7: Treatment assignment experiment: rankings by different metrics, sorted by Spearman's rank correlation coefficient (r_s) with oracle ranking. Numbering indicates the oracle order of $\pi_i(\mathbf{X})$.

Metric	Ranking	r_s (↑)
P_α	2,3,1	-0.5
KS	2,1,3	0.5
Inv. KL	1,3,2	0.5
R_β	2,1,3	0.5
WD	1,2,3	1
JSD_π	1,2,3	1

Table 8: Outcome generation experiment: Rankings by different metrics, sorted by Spearman's rank correlation coefficient (r_s) with oracle ranking. $Q^i_{Y|W,\mathbf{X}}$ are number by oracle ranking, 1: T-Learner, 2: TARNet, 3: DragonNet, 4: S-Learner.

Metric	Ranking	r_s (↑)
KS	4,2,3,1	-0.80
Inv. KL	4,2,1,3	-0.40
P_α	1,4,3,2	0.20
R_β	1,4,3,2	0.20
WD	2,3,1,4	0.40
U_{PEHE}	1,2,3,4	1

D.3 Existing metric failure: extreme example

As a 'proof by contradiction' that current metrics can offer a good level on information on the preservation of (i)-(iii), we present some extreme examples. We show that joint-distribution-level metrics do not have enough resolution to identify how well (i)-(iii) are preserved, even if Q comprehensively fails in modelling any one of the component distributions $P_{\mathbf{X}}$, $P_{W|\mathbf{X}}$, or $P_{Y|W,\mathbf{X}}$.

We perform a series of experiments where we evaluate adversarial synthetic versions of a simulated dataset, with each synthetic version failing in one of the above components, and we show that standard metrics do not identify these failure modes. We simulate real data using the DGP in `CATENets` from Curth & van der Schaar (2021) and we create three $\mathcal{D}_{\text{synth}}$ that perfectly model two component distributions of $\mathcal{D}_{\text{real}}$ but poorly approximate the remaining one. For poorly modelled $P_{\mathbf{X}}$, we set $\mathbf{X} = \mathbf{0}$; for poorly modelled $P_{W|\mathbf{X}}$, we assign all instances with $W = 0$; and for poorly modelled $P_{Y|W,\mathbf{X}}$, we draw Y from a normal distribution with mean 0 regardless of treatment. All such $\mathcal{D}_{\text{synth}}$ are useless for treatment effect estimation.

Table 9: Scores on adversarially created $\mathcal{D}_{\text{synth}}$ which poorly perform on desiderata (i), (ii), or (iii).

	Inv. KL (↑)	$\boldsymbol{P_\alpha}$ (↑)	$\boldsymbol{R_\beta}$ (↑)	**MMD** (↓)
Poor (i)	0.681	0.902	0.368	0.085
Poor (ii)	0.685	0.501	0.333	0.074
Poor (iii)	0.844	0.905	0.430	0.008

We report the inverse of KL divergence, P_α, R_β and MMD, which all have range $[0, 1]$, for these synthetic datasets in Table 9. We see that these conventional evaluation metrics do not accurately reflect the invalidity of each $\mathcal{D}_{\text{synth}}$ for treatment effect estimation. None report significantly low scores, despite the failure of each $\mathcal{D}_{\text{synth}}$. R_β reflects these failures best, although its scores still do not adequately reflect how these datasets render correct treatment effect analysis impossible, and it does not allow a granular enough analysis to disentangle which component distribution is poorly modelled.

In further detail, for the experiments that examine poor modelling of $P_{\mathbf{X}}$ and $P_{W|\mathbf{X}}$, we simulate $\mathcal{D}_{\text{real}}$ of size $n = 1000$ with $d = 1$ covariate as follows:

$$X \sim \mathcal{N}(0, 1) \tag{9}$$
$$W \sim \text{Bernoulli}(0.5) \tag{10}$$
$$Y(0), Y(1) = 0 \tag{11}$$
$$Y = (1 - W)Y(0) + WY(1) + \epsilon,\ \epsilon \sim \mathcal{N}(0, 1) \tag{12}$$

We manufacture $\mathcal{D}_{\text{synth}}$ that exhibits poor modelling of $P_{\mathbf{X}}$ by generating W and Y from the true distributions as above, but set $\mathbf{X} = \mathbf{0}$. For $\mathcal{D}_{\text{synth}}$ that exhibits poor modelling of $P_{W|\mathbf{X}}$, we generate $\mathbf{X}$ and Y from their true distributions, but set all $W = 0$.

To demonstrate assessment under poor modelling of $P_{Y|W,\mathbf{X}}$, we set the covariate in $\mathcal{D}_{\text{real}}$ to be predictive, such that it affects the value of the potential outcome $Y(1)$, but not $Y(0)$. The distributions remain the same as the above, although now the potential outcomes are:

$$Y(0) = 0 \tag{13}$$
$$Y(1) = X^2 \tag{14}$$

We manufacture $\mathcal{D}_{\text{synth}}$ that poorly models of $P_{Y|W,\mathbf{X}}$ by generating $\mathbf{X}$ and W from their true distributions, but we set $Y(0), Y(1) = 0$.

E Discussion on alternative metrics

While we propose a set of metrics $\mathcal{M}$ for evaluation of $\mathcal{D}_{\text{synth}}$, there are many possible alternatives to each choice we make. Our choices enable evaluation of how well $\mathcal{D}_{\text{synth}}$ adheres to our desiderata, but, like any metrics, they may be sub-optimal for certain data holders with specific preferences. Here, we list some alternative definitions, and we detail when they may be preferable. We would like to emphasise that conducting *any* reasonable assessment of the preservation of (i)-(iii) is beneficial compared to standard evaluation practices.

E.1 Alternative covariate distribution assessment

As we state in the main paper, comparison of P_X and Q_X is essentially a standard synthetic data evaluation problem, and therefore any standard protocol can be applied.

For example, if the dimensionality of $\mathbf{X}$ is small, manual evaluation via visualisation may be preferable to the precision/recall analysis we suggest, as this can provide a more granular and

interpretable assessment. On the other hand, if a single all-encompassing score is desired, rather than the two-dimensional metric ($P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$), then statistical divergence metrics can offer this. These one-dimensional metrics can lead to more straightforward model selection than ($P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$), as ordering based on a two-dimensional metric can be ambiguous.

E.2 Alternative treatment assignment mechanism assessment

Similarly, there is a vast array of metrics which could be substituted into (5) over Jensen-Shannon distance which could measure the difference between $P_{W|\mathbf{X}}$ and $Q_{W|\mathbf{X}}$. These include metrics such as KL divergence and Wasserstein distance, which are also very common in machine learning literature. For example, a data holder may prefer KL divergence if they want to more harshly punish $Q_{W|\mathbf{X}}$ for failing to place density where $P_{W|\mathbf{X}}$ is probable, encouraging *mode-covering* behaviour. On the other hand, if a data holder wants to more harshly punish $Q_{W|\mathbf{X}}$ for spreading mass away from the modes, Wasserstein distance may be preferable, leading to *mode-seeking* behaviour. JSD achieves a balance between these two focuses, but if a data holder has a strong preference for one or the other, these alternate choices would be preferable. Nevertheless, we suggest that, apart from extreme scenarios, most reasonable methods to assess the preservation of $P_{W|\mathbf{X}}$ will lead to similar analysis.

E.3 Alternative outcome generation mechanism metrics

Raw similarity of PEHE in CATE estimation between $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$ may not be the most important quantity of interest for certain data holders. This can be particularly true in medical practice, as raw performance is not the only important aspect of a downstream model. We propose some alternatives which may be more applicable in the following situations:

1. Correct estimation of the *sign* of the CATE may be of heightened importance if the CATE learner is assisting with policy decisions. The wrong CATE sign will lead to incorrect policy administration, whereas the magnitude of the effect may not be as important for decision making.
2. Discovering the correct drivers of effect heterogeneity may be important, as how a learner arrives at its final estimation is particularly important to consider in applications such as in drug discovery or clinical practice (Hermansson & Svensson, 2021; Crabbé et al., 2022).

E.3.1 Policy assignment

If policy guidance is of interest, then quantification of how well the sign of CATE estimates is preserved between $\mathcal{D}_{\text{synth}}$ and $\mathcal{D}_{\text{real}}$ may be desired, which can be done as follows:

$$U_{\text{policy}}(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}) = \frac{1}{|\mathcal{F}|} \sum_{\hat{\tau} \in \mathcal{F}} \mathbb{E}_{P_{\mathbf{X}}}[I(\hat{\tau}_{\text{synth}}(\mathbf{X}) \times \hat{\tau}_{\text{real}}(\mathbf{X}) > 0)] \tag{15}$$

where I is the indicator function.

E.3.2 Feature importance

If assessing how well $\mathcal{D}_{\text{synth}}$ permits the discovery of the correct drivers of effect heterogeneity is important, this can be quantified through the use of feature importance methods. Given a CATE learner $\hat{\tau}$, feature importance methods offer a means to measure the sensitivity of the model to each covariate by assigning an importance score $a_i(\hat{\tau}, \mathbf{x})$ to each feature x_i that reflects its importance in the prediction of the CATE $\hat{\tau}(\mathbf{x})$. There are many different instantiations of feature importance methods with different strengths (Ewald et al., 2024), and the metric we propose here is method-agnostic. We quantify how well $P_{Y|W,\mathbf{X}}$ is modelled according to feature importance similarity between $\mathcal{D}_{\text{synth}}$ and $\mathcal{D}_{\text{real}}$ as follows:

$$U_{\text{int}}(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}) = \frac{1}{|\mathcal{F}|} \sum_{\hat{\tau} \in \mathcal{F}} S_C(A_{\text{real},\hat{\tau}}, A_{\text{synth},\hat{\tau}}) \tag{16}$$

where S_C is cosine similarity, and $A_{\text{real},\hat{\tau}}$ and $A_{\text{synth},\hat{\tau}}$ are d-dimensional vectors with i^{th} entries

$$A^i_{\diamond,\hat{\tau}} = \mathbb{E}_{P_{\mathbf{X}}}[a_i(\hat{\tau}_\diamond, \mathbf{X})],\ \diamond \in \{\text{real}, \text{synth}\} \tag{17}$$

$U_{\text{int}} \in [-1, 1]$, where $U_{\text{int}} = 1$ indicates total agreement in the feature importances of $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{synth}}$, while $U_{\text{int}} = 0$ indicates that the feature importances are uncorrelated, suggesting that $Q_{Y|W,\mathbf{X}}$ does not allow discovery of the correct drivers of heterogeneity. Finally, $U_{\text{int}} = -1$ indicates antithetical feature importances, suggesting a drastic failure of $Q_{Y|W,\mathbf{X}}$ in estimating $P_{Y|W,\mathbf{X}}$.

F Defining $\mathcal{F}$ for U_{PEHE}

In CATE estimation, model validation is a difficult task (Curth & Van Der Schaar, 2023). As such, it is reasonable to expect that a set of downstream analysts conducting CATE estimation on $\mathcal{D}_{\text{synth}}$ will use different learners. Therefore, we want U_{PEHE} to reflect the expected difference in downstream performance between $\mathcal{D}_{\text{synth}}$ and $\mathcal{D}_{\text{real}}$ across a diverse array of potential learners, such that it is representative for the entire population of analysts, and has limited bias towards any particular learner class. To achieve this, we propose averaging U_{PEHE} across a family of CATE learners $\mathcal{F}$, and we suggest that larger $|\mathcal{F}|$, and diverse selection of the learners within $\mathcal{F}$, is preferable.

Of course, there is a trade off between the size of $\mathcal{F}$, and therefore the stability of U_{PEHE}, and the computational cost of repeated CATE estimation. With this in mind, to limit the computation involved in calculating U_{PEHE}, we suggest that users should be selective of the learners included in $\mathcal{F}$ to maximize learner diversity, and minimise $|\mathcal{F}|$. For example, in our experiments we set $|\mathcal{F}| = 4$, and we chose learners from both of the high-level CATE learning strategies described in Curth & van der Schaar (2021) (i.e. one-step plug-in learners, and two-step learners). Specifically, for the one-step learners we use S- and T- learners (Künzel et al., 2019), and for the two-step learners we use RA and DR learners (Kennedy et al., 2020). All four of these learners conduct CATE estimation differently, and encode different inductive biases in their approaches, and thus they form a good diverse base for $\mathcal{F}$.

For our experiments, on each of the real datasets from Section 7.1, the runtime for calculating U_{PEHE} for one run are shown in Table 10. Note that these are much less than the typical generation times for each dataset, so this step is unlikely to be a large time burden for the data holder. Also note that these calculations can be parallelized across the learner classes, which we did not do, and this can improve the computational feasibility of using a larger $|\mathcal{F}|$.

Table 10: Runtime to calculate U_{PEHE}

Dataset	U_{PEHE} runtime (s)
ACTG	26
IHDP	60
ACIC	191

G STEAM differential privacy proof

The theoretical guarantee of STEAM's differential privacy (DP) when using individual DP components is grounded in the post-processing and composition theorems of DP (Dwork & Roth, 2014), as we state in the main body of the paper. We make this derivation clear here, by first outlining the post-processing and composition theorems in full.

Theorem (Post-Processing Theorem)**.** *Let $M : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}$ be a randomized algorithm that is (ϵ, δ)-differentially private. Let $f : \mathcal{R} \to \mathcal{R}'$ be an arbitrary randomized mapping. Then the composition $f \circ M : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}'$ is (ϵ, δ)-differentially private.*

Theorem (Composition Theorem)**.** *Let $M_i : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_i$ be an (ϵ_i, δ_i)-differentially private algorithm for $i \in [k]$. Define $M_{[k]} : \mathbb{N}^{|\mathcal{X}|} \to \prod_{i=1}^k \mathcal{R}_i$ as:*

$$M_{[k]}(x) = (M_1(x), M_2(x), \ldots, M_k(x)),$$

then $M_{[k]}$ *is* $\left(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i\right)$*-differentially private.*

Given these theorems, we have our guarantee of DP generation with STEAM. Specifically:

Proposition 1. *If* $Q_{\mathbf{X}}$ *satisfies* $(\epsilon_{\mathbf{X}}, \delta_{\mathbf{X}})$*-differential privacy,* $Q_{W|\mathbf{X}}$ *satisfies* (ϵ_W, δ_W)*-differential privacy, and* $Q_{Y|W,\mathbf{X}}$ *satisfies* (ϵ_Y, δ_Y)*-differential privacy, STEAM satisfies* $(\epsilon_{total}, \delta_{total})$*-differential privacy, where* $\epsilon_{total} = \epsilon_{\mathbf{X}} + \epsilon_W + \epsilon_Y,\ \delta_{total} = \delta_{\mathbf{X}} + \delta_W + \delta_Y$.

Proof. $Q_{\mathbf{X}}$ generates $\mathbf{X}$, and satisfies $(\epsilon_{\mathbf{X}}, \delta_{\mathbf{X}})$-differential privacy by assumption.

By the post-processing theorem, inputting $\mathbf{X}$ as the condition to $Q_{W|\mathbf{X}}$ does not affect its privacy. $Q_{W|\mathbf{X}}$ generates W, and satisfies (ϵ_W, δ_W)-differential privacy by assumption.

By the post-processing theorem, inputting W and $\mathbf{X}$ as the conditions to $Q_{Y|W,\mathbf{X}}$ does not affect their privacy. $Q_{Y|W,\mathbf{X}}$ generates Y, and satisfies (ϵ_Y, δ_Y)-differential privacy by assumption.

STEAM generates $(\mathbf{X}, W, Y)$, and is the composition of $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, and $Q_{Y|W,\mathbf{X}}$, i.e. STEAM = $(Q_{\mathbf{X}}, Q_{W|\mathbf{X}}, Q_{Y|W,\mathbf{X}})$

Therefore, by the composition theorem STEAM satisfies $(\epsilon_{\text{total}}, \delta_{\text{total}})$-differential privacy, where $\epsilon_{\text{total}} = \epsilon_{\mathbf{X}} + \epsilon_W + \epsilon_Y,\ \delta_{\text{total}} = \delta_{\mathbf{X}} + \delta_W + \delta_Y$. □

H MAIN EXPERIMENTAL DETAILS

Here we add any additional details to the experiment set-ups from Section 7. All experiments were run on an Azure VM with a 48-Core AMD EPYC Milan CPU, an A100 GPU with 80GB of VRAM, and 880GB of RAM. We report typical runtimes where relevant. An estimated total compute time for all experimental runs is ~72 hours. This does not include the compute required for preliminary experimentation.

For all generative models, we use the open source library `synthcity` Qian et al. (2023) (Apache-2.0 License), and we do not change the default hyperparameters. We set the treatment and outcome generators of STEAM as a logistic regression function from `scikit-learn` Pedregosa et al. (2011) and T-Learner from `CATENets` Curth & van der Schaar (2021), respectively.

H.1 GENERATION OF MEDICAL DATA CONTAINING TREATMENTS

To assess sequential generation in a number of real-world scenarios, we evaluate performance on ACTG (Hammer et al., 1996) and on the popular treatment effect estimation datasets IHDP (Hill, 2011) and ACIC (Dorie et al., 2018). We also report further results in Table 11 on a non-medical dataset, Jobs (LaLonde, 1986), which is also popular amongst the treatment effect estimation community, to show that STEAM can be applied beyond the medical context, to any dataset containing treatments. More in depth descriptions of the datasets used are here:

1. **AIDS Clinical Trial Group (ACTG) study 175.** A clinical trial on subjects with HIV-1 (Hammer et al., 1996). Preprocessed as in Hatt et al. (2022) to compare CD4 counts at the beginning of the study and after 20 ± 5 weeks across treatment arms using zidovudine (ZDV) and zalcitabine (ZAL) vs. ZDV only. The ACTG dataset contains $n = 1056$ instances with $d = 12$ covariates and a continuous outcome, and we use the publicly available version from `https://github.com/tobhatt/CorNet`.
2. **Infant Health and Development Program (IHDP).** A semi-synthetic medical dataset, with real covariates and simulated outcomes, using data from a randomised experiment designed to evaluate the effect of specialist childcare on the cognitive test scores of premature infants (Brooks-Gunn et al., 1992). Confounding and treatment imbalance were introduced in Hill (2011) to mimic an observational dataset. The IHDP dataset consists of $n = 747$ instances with $d = 25$ covariates and a continuous outcome. We use the publicly available version from `https://github.com/AMLab-Amsterdam/CEVAE` (Louizos et al., 2017), with the first batch of simulated outcomes.
3. **Atlantic Causal Inference Competition 2016 (ACIC).** A semi-synthetic medical dataset, with real covariates and simulated outcomes, containing data from the Collaborative Perinatal

Project (Niswander, 1972). The data was modified in Dorie et al. (2018) to simulate an observational study examining the impact of birth weight in twins on IQ. The ACIC dataset consists of $n = 4802$ instances with $d = 58$ covariates and a continuous outcome. We use the publicly available version from the `causallib` package (Shimoni et al., 2019) (Apache-2.0 License) available here `https://github.com/BiomedSciAI/causallib`, using the first simulated set of treatments and potential outcomes.

4. **Jobs.** Jobs contains experimental data from a male sub-sample from the National Supported Work Demonstration from LaLonde (1986) to evaluate the effect of job training on income. The Jobs dataset consists of $n = 722$ instances with $d = 7$ covariates and a continuous outcome. We use the publicly available version used in Dehejia & Wahba (1998; 2002), from `https://users.nber.org/~rdehejia/data/.nswdata2.html`.

We report extended results for all models tested, and the further results on the Jobs dataset, in Table 11. For each model on each dataset we conduct 20 runs. A typical run for a given real dataset and generative model took 15 minutes.

H.2 Simulated experiments

For our simulated insight experiments, we compare performance of a standard TabDDPM with $\text{STEAM}_{\text{TabDDPM}}$, and we report average results over 10 runs. A typical run took 5 minutes. For simulation of $\mathcal{D}_{\text{real}}$, we use the DGP from `CATENets` (Curth & van der Schaar, 2021).

H.3 Differentially private generation

For our experiment which showcases the performance of STEAM when satisfying DP, we compare the generative performance of baseline methods AIM (McKenna et al., 2022), GEM (Liu et al., 2021), MST (McKenna et al., 2021), RAP (Aydore et al., 2021) with their STEAM counterparts. We use the code provided by McKenna et al. (2022) in their GitHub `https://github.com/ryan112358/private-pgm` for the AIM and MST implementations, and we use the code provided in the GitHub `https://github.com/terranceliu/dp-query-release` for the GEM and RAP implementations. We use the default hyperparameter settings of these implementations, with the workload set as 3-way marginals. For the STEAM models, we use the relevant base model for $Q_{\mathbf{X}}$, DP random forest from the `diffprivlib` library (Holohan et al., 2019) (MIT License) for $Q_{W|\mathbf{X}}$, and a custom implementation of a T-Learner Künzel et al. (2019) based on Curth & van der Schaar (2021) which guarantees DP by training with DP stochastic gradient descent, implemented with the `Opacus` library (Yousefpour et al., 2021) (Apache-2.0 License). We report comparative results on varying ϵ, averaged over 5 runs.

I Extended results

I.1 Section 7.1 extended results

We report the full set of results for each model and dataset from Section 7.1 in Table 11. We pair each standard model with its STEAM analogue, and report the relative difference between them for each metric, where (green) indicates better performance by STEAM, and (red) indicates better performance by standard modelling. We see that STEAM clearly outperforms. Almost all STEAM models perform better in each metric than all standard models.

I.2 Section 7.3 extended results

We report the full (ϵ, δ)-DP generation results on the ACTG across a set of baseline models with the same set-ups as in Section 7.3. In Figure 5 we compare GEM (Liu et al., 2021) with $\text{STEAM}_{\text{GEM}}$, in Figure 6 we compare MST (McKenna et al., 2021) with $\text{STEAM}_{\text{MST}}$, and in Figure 7 we compare RAP (Aydore et al., 2021) with $\text{STEAM}_{\text{RAP}}$. While there are some nuances to each baseline comparison,

Table 11: $P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$, JSD_{π}, and U_{PEHE} values for STEAM and standard models. Averaged over 20 runs, with 95% confidence intervals. Each STEAM model is placed after its corresponding standard model. Coloured numbers in brackets indicate relative difference between standard and STEAM model, where (green) indicates better performance by STEAM, and (red) indicates better performance by standard modelling.

Dataset	Model	$P_{\alpha,\mathbf{X}}$ (↑)	$R_{\beta,\mathbf{X}}$ (↑)	JSD_{π} (↑)	U_{PEHE} (↓)
ACTG	TVAE	0.926 ± 0.013	0.483 ± 0.010	0.946 ± 0.004	0.564 ± 0.017
	STEAM $_{\text{TVAE}}$	0.929 ± 0.008 (+0.003)	0.486 ± 0.009 (+0.003)	0.958 ± 0.004 (+0.012)	0.492 ± 0.011 (-0.072)
	ARF	0.818 ± 0.012	0.453 ± 0.007	0.960 ± 0.004	0.577 ± 0.015
	STEAM $_{\text{ARF}}$	0.836 ± 0.008 (+0.018)	0.464 ± 0.007 (+0.011)	0.962 ± 0.004 (+0.002)	0.423 ± 0.016 (-0.154)
	CTGAN	0.889 ± 0.020	0.446 ± 0.014	0.934 ± 0.008	0.586 ± 0.017
	STEAM $_{\text{CTGAN}}$	0.892 ± 0.017 (+0.003)	0.435 ± 0.012 (-0.011)	0.959 ± 0.005 (+0.025)	0.436 ± 0.012 (-0.150)
	NFlow	0.817 ± 0.032	0.418 ± 0.008	0.913 ± 0.016	0.643 ± 0.026
	STEAM $_{\text{NFlow}}$	0.837 ± 0.040 (+0.020)	0.417 ± 0.015 (-0.001)	0.962 ± 0.005 (+0.049)	0.445 ± 0.020 (-0.198)
	TabDDPM	0.067 ± 0.060	0.036 ± 0.035	0.812 ± 0.029	1.761 ± 0.230
	STEAM $_{\text{TabDDPM}}$	0.612 ± 0.106 (+0.545)	0.310 ± 0.055 (+0.274)	0.952 ± 0.009 (+0.140)	0.468 ± 0.013 (-1.293)
IHDP	CTGAN	0.663 ± 0.018	0.419 ± 0.013	0.888 ± 0.010	2.521 ± 0.161
	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014 (+0.011)	0.424 ± 0.011 (+0.005)	0.928 ± 0.009 (+0.040)	1.709 ± 0.052 (-0.812)
	TabDDPM	0.477 ± 0.036	0.340 ± 0.022	0.862 ± 0.011	2.706 ± 0.138
	STEAM $_{\text{TabDDPM}}$	0.553 ± 0.029 (+0.076)	0.396 ± 0.015 (+0.056)	0.918 ± 0.011 (+0.056)	2.346 ± 0.088 (-0.360)
	ARF	0.528 ± 0.009	0.381 ± 0.010	0.921 ± 0.009	3.019 ± 0.117
	STEAM $_{\text{ARF}}$	0.565 ± 0.014 (+0.037)	0.394 ± 0.010 (+0.013)	0.921 ± 0.009 (+0.000)	1.629 ± 0.056 (-1.390)
	TVAE	0.622 ± 0.014	0.410 ± 0.010	0.880 ± 0.014	3.198 ± 0.172
	STEAM $_{\text{TVAE}}$	0.629 ± 0.015 (+0.007)	0.412 ± 0.011 (+0.002)	0.927 ± 0.007 (+0.047)	2.100 ± 0.075 (-1.098)
	NFlow	0.406 ± 0.028	0.309 ± 0.012	0.882 ± 0.012	3.835 ± 0.345
	STEAM $_{\text{NFlow}}$	0.435 ± 0.034 (+0.029)	0.333 ± 0.020 (+0.024)	0.921 ± 0.007 (+0.039)	2.177 ± 0.118 (-1.658)
ACIC	TVAE	0.763 ± 0.011	0.515 ± 0.006	0.926 ± 0.007	4.202 ± 0.134
	STEAM $_{\text{TVAE}}$	0.767 ± 0.009 (+0.004)	0.514 ± 0.004 (-0.001)	0.972 ± 0.002 (+0.046)	2.013 ± 0.112 (-2.189)
	ARF	0.936 ± 0.003	0.396 ± 0.003	0.948 ± 0.002	4.742 ± 0.165
	STEAM $_{\text{ARF}}$	0.939 ± 0.004 (+0.003)	0.393 ± 0.004 (-0.003)	0.977 ± 0.002 (+0.029)	2.176 ± 0.141 (-2.566)
	CTGAN	0.880 ± 0.016	0.421 ± 0.013	0.942 ± 0.005	4.518 ± 0.186
	STEAM $_{\text{CTGAN}}$	0.873 ± 0.014 (-0.007)	0.424 ± 0.014 (+0.003)	0.972 ± 0.002 (+0.030)	2.268 ± 0.154 (-2.250)
	NFlow	0.691 ± 0.052	0.298 ± 0.014	0.872 ± 0.024	5.222 ± 0.332
	STEAM $_{\text{NFlow}}$	0.673 ± 0.044 (-0.018)	0.285 ± 0.019 (-0.013)	0.973 ± 0.002 (+0.101)	2.790 ± 0.337 (-2.432)
	TabDDPM	0.260 ± 0.043	0.001 ± 0.000	0.787 ± 0.032	10.104 ± 1.205
	STEAM $_{\text{TabDDPM}}$	0.273 ± 0.035 (+0.013)	0.001 ± 0.000 (+0.000)	0.941 ± 0.020 (+0.154)	6.178 ± 0.619 (-3.926)
Jobs	TabDDPM	0.890 ± 0.014	0.477 ± 0.011	0.949 ± 0.004	3.335 ± 0.516
	STEAM $_{\text{TabDDPM}}$	0.929 ± 0.009 (+0.039)	0.493 ± 0.008 (+0.016)	0.954 ± 0.003 (+0.005)	1.446 ± 0.052(-1.889)
	ARF	0.832 ± 0.010	0.431 ± 0.019	0.964 ± 0.004	3.173 ± 0.691
	STEAM $_{\text{ARF}}$	0.863 ± 0.011 (+0.031)	0.481 ± 0.016 (+0.050)	0.953 ± 0.004 (-0.011)	2.280 ± 0.381 (-0.893)
	TVAE	0.886 ± 0.017	0.288 ± 0.009	0.944 ± 0.006	4.471 ± 0.336
	STEAM $_{\text{TVAE}}$	0.887 ± 0.014 (+0.001)	0.300 ± 0.012 (+0.012)	0.949 ± 0.004 (+0.005)	1.540 ± 0.167 (-2.931)
	CTGAN	0.830 ± 0.049	0.339 ± 0.023	0.925 ± 0.033	4.608 ± 0.792
	STEAM $_{\text{CTGAN}}$	0.778 ± 0.076 (-0.052)	0.298 ± 0.030 (-0.041)	0.939 ± 0.007 (+0.014)	1.846 ± 0.270 (-2.762)
	NFlow	0.716 ± 0.058	0.374 ± 0.017	0.920 ± 0.018	5.445 ± 0.883
	STEAM $_{\text{NFlow}}$	0.800 ± 0.041 (+0.084)	0.375 ± 0.017 (+0.001)	0.952 ± 0.006 (+0.032)	2.666 ± 0.200 (-2.779)

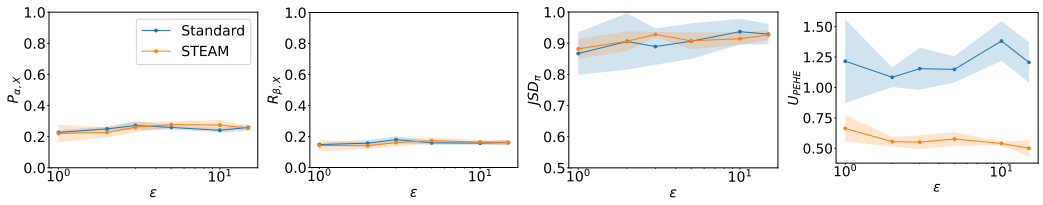

Figure 5: $P_{\alpha,\mathbf{X}}$ (↑), $R_{\beta,\mathbf{X}}$ (↑), JSD_π (↑), and U_{PEHE} (↓) evaluating $\text{STEAM}_{\text{GEM}}$ and standard GEM across privacy budgets. Averaged over 5 runs, shaded area represents 95% CIs.

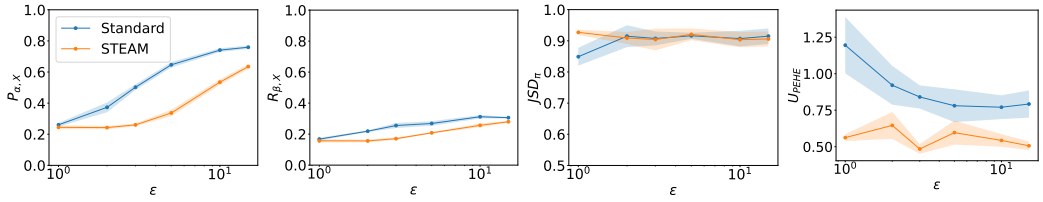

Figure 6: $P_{\alpha,\mathbf{X}}$ (↑), $R_{\beta,\mathbf{X}}$ (↑), JSD_π (↑), and U_{PEHE} (↓) evaluating $\text{STEAM}_{\text{MST}}$ and standard MST across privacy budgets. Averaged over 5 runs, shaded area represents 95% CIs.

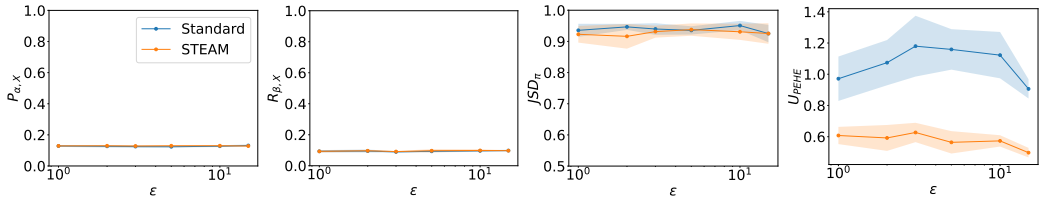

Figure 7: $P_{\alpha,\mathbf{X}}$ (↑), $R_{\beta,\mathbf{X}}$ (↑), JSD_π (↑), and U_{PEHE} (↓) evaluating $\text{STEAM}_{\text{RAP}}$ and standard RAP across privacy budgets. Averaged over 5 runs, shaded area represents 95% CIs.

the general takeaway remains similar to those reported in Section 7.3 - STEAM models preserve $P_{W|\mathbf{X}}$ and $P_{Y|W,\mathbf{X}}$ better, while standard models preserve $P_{\mathbf{X}}$ better.

It is worth noting, however, that when baseline models perform poorly in modelling $P_{\mathbf{X}}$, as is the case for GEM and RAP, then the relevant STEAM model exhibits similar performance in this regard.

J ABLATIVE STUDY

To add to the evidence of STEAM's efficacy, we conduct an ablative study by assessing how jointly modelling $P_{\mathbf{X},W}$ affects performance. On the medical datasets used in the main body of the paper, we compare performance of the best standard models with their relevant ablation STEAM $_{\diamond, \text{ joint X,W}}$, which models $P_{\mathbf{X},W}$ with the generative model and $P_{Y|W,\mathbf{X}}$ with a PO estimator, and regular STEAM. We report the results in Table 12.

We see that the ablative model, while often improving upon standard generation, is not as effective as STEAM. Directly modelling $P_{W|\mathbf{X}}$, as STEAM does, better preserves the treatment assignment and outcome generation mechanisms, and both JSD_π and U_{PEHE} are significantly improved by STEAM in most cases. Using the full inductive bias of directly modelling each distribution of our desiderata, and following the true DGP of data containing treatments is the best approach to generation.

Table 12: $P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$, JSD_π, and U_{PEHE} values on standard, ablation, and STEAM models. Averaged over 10 runs, with 95% CIs.

Dataset	Model	$\boldsymbol{P_{\alpha,\mathbf{X}}}$ (↑)	$\boldsymbol{R_{\beta,\mathbf{X}}}$ (↑)	$\textbf{JSD}_{\boldsymbol{\pi}}$ (↑)	$\boldsymbol{U}_{\textbf{PEHE}}$ (↓)
ACTG	TVAE	0.926 ± 0.013	0.483 ± 0.010	0.946 ± 0.004	0.564 ± 0.017
	STEAM $_{\text{TVAE, joint X,W}}$ (*ablation*)	0.918 ± 0.021	0.473 ± 0.012	0.939 ± 0.010	0.475 ± 0.012
	STEAM $_{\text{TVAE}}$	0.929 ± 0.008	0.486 ± 0.009	0.958 ± 0.004	0.492 ± 0.011
IHDP	CTGAN	0.663 ± 0.018	0.419 ± 0.013	0.888 ± 0.010	2.521 ± 0.161
	STEAM $_{\text{CTGAN, joint X,W}}$ (*ablation*)	0.639 ± 0.021	0.428 ± 0.009	0.908 ± 0.019	2.140 ± 0.134
	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014	0.424 ± 0.011	0.928 ± 0.009	1.709 ± 0.052
ACIC	TVAE	0.763 ± 0.011	0.515 ± 0.006	0.926 ± 0.007	4.202 ± 0.134
	STEAM $_{\text{TVAE, joint X,W}}$ (*ablation*)	0.747 ± 0.023	0.506 ± 0.005	0.920 ± 0.009s	2.530 ± 0.187
	STEAM $_{\text{TVAE}}$	0.767 ± 0.009	0.514 ± 0.004	0.972 ± 0.002	2.013 ± 0.112

K HYPERPARAMETER STABILITY

Generative modelling performnce is typically sensitive to hyperparameters. To assess the stability of STEAM's performance across hyperparameters, on IHDP we compare the performance of CTGAN with $\text{STEAM}_{\text{CTGAN}}$ with multiple hyperparameter configurations. We report results by changing three hyperparameters: number of hidden units within the generator layers (`generator_n_hidden_units`) (Table 13), number of hidden layers within the generator (`generator_n_hidden_layers`) (Table 14), and activation functions used in the generator (`generator_nonlin`) (Table 15), keeping all other hyperparameters default.

The performance gap between $\text{STEAM}_{\text{CTGAN}}$ and CTGAN is relatively stable across these configurations. $\text{STEAM}_{\text{CTGAN}}$ outperforms CTGAN in each metric at almost all hyperparameter levels. The most statistically significant differences are consistently noted in the JSD_π and U_{PEHE} metrics, which is compatible with the results displayed in the main paper.

Table 13: Comparison of STEAM with standard generation on IHDP at different `generator_n_hidden_units` levels. Averaged over 5 runs, with 95% CIs.

`generator_n_hidden_units`	Model	$\boldsymbol{P_{\alpha,\mathbf{X}}}$ (↑)	$\boldsymbol{R_{\beta,\mathbf{X}}}$ (↑)	$\textbf{JSD}_{\boldsymbol{\pi}}$ (↑)	$\boldsymbol{U}_{\textbf{PEHE}}$ (↓)
5	CTGAN	0.517 ± 0.026	0.396 ± 0.015	0.863 ± 0.033	2.914 ± 0.047
	STEAM $_{\text{CTGAN}}$	0.565 ± 0.011	0.405 ± 0.011	0.941 ± 0.000	2.194 ± 0.265
50	CTGAN	0.622 ± 0.028	0.411 ± 0.043	0.916 ± 0.15	2.282 ± 0.141
	STEAM $_{\text{CTGAN}}$	0.664 ± 0.020	0.444 ± 0.017	0.905 ± 0.041	1.960 ± 0.174
100	CTGAN	0.607 ± 0.038	0.418 ± 0.032	0.894 ± 0.010	2.560 ± 0.289
	STEAM $_{\text{CTGAN}}$	0.682 ± 0.016	0.439 ± 0.018	0.912 ± 0.004	2.097 ± 0.095
300	CTGAN	0.619 ± 0.030	0.434 ± 0.030	0.908 ± 0.023	2.426 ± 0.289
	STEAM $_{\text{CTGAN}}$	0.699 ± 0.018	0.458 ± 0.015	0.928 ± 0.016	2.028 ± 0.163
500	CTGAN	0.663 ± 0.018	0.419 ± 0.013	0.888 ± 0.010	2.521 ± 0.161
	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014	0.424 ± 0.011	0.928 ± 0.009	1.709 ± 0.052

Table 14: Comparison of STEAM with standard generation on IHDP at different `generator_n_hidden_layers` levels. Averaged over 5 runs, with 95% CIs.

`generator_n_hidden_layers`	Model	$P_{\alpha,X}$ (↑)	$R_{\beta,X}$ (↑)	JSD_{π} (↑)	U_{PEHE} (↓)	
2	CTGAN	0.663 ± 0.018	0.419 ± 0.013	0.888 ± 0.010	2.521 ± 0.161	
	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014	0.424 ± 0.011	0.928 ± 0.009	1.709 ± 0.052	
3	CTGAN	0.595 ± 0.067	0.395 ± 0.066	0.868 ± 0.064	2.982 ± 0.647	
	STEAM $_{\text{CTGAN}}$	0.693 ± 0.075	0.441 ± 0.043	0.924 ± 0.018	2.028 ± 0.143	
4	CTGAN	0.583 ± 0.049	0.259 ± 0.074	0.807 ± 0.054	3.278 ± 0.191	
	STEAM $_{\text{CTGAN}}$	0.596 ± 0.220	0.301 ± 0.084	0.886 ± 0.014	2.690 ± 0.836	
5	CTGAN	0.490 ± 0.092	0.313 ± 0.069	0.770 ± 0.127	2.871 ± 0.599	
	STEAM $_{\text{CTGAN}}$	0.691 ± 0.071	0.386 ± 0.041	0.915 ± 0.010	2.498 ± 0.536	

Table 15: Comparison of STEAM with standard generation on IHDP at different `generator_nonlin` settings. Averaged over 5 runs, with 95% CIs.

`generator_nonlin`	Model	$P_{\alpha,X}$ (↑)	$R_{\beta,X}$ (↑)	JSD_{π} (↑)	U_{PEHE} (↓)
ReLU	CTGAN	0.663 ± 0.018	0.419 ± 0.013	0.888 ± 0.010	2.521 ± 0.161
	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014	0.424 ± 0.011	0.928 ± 0.009	1.709 ± 0.052
SELU	CTGAN	0.604 ± 0.020	0.419 ± 0.015	0.855 ± 0.023	2.509 ± 0.160
	STEAM $_{\text{CTGAN}}$	0.699 ± 0.017	0.445 ± 0.025	0.929 ± 0.014	2.043 ± 0.130
Leaky ReLU	CTGAN	0.648 ± 0.045	0.415 ± 0.015	0.889 ± 0.016	2.482 ± 0.210
	STEAM $_{\text{CTGAN}}$	0.699 ± 0.028	0.457 ± 0.019	0.916 ± 0.011	2.036 ± 0.135

L Congeniality bias

Congeniality bias (Curth & Van Der Schaar, 2023) is a phenomenon which may arise from generation with STEAM. In this scenario it refers to the fact that downstream models which are structurally similar to the outcome generator, $Q_{Y|W,\mathbf{X}}$, may be advantaged in their performance on $\mathcal{D}_{\text{synth}}$. For example, if the POs from an S-learner are used for $Q_{Y|W,\mathbf{X}}$, the outcome generation mechanism in $\mathcal{D}_{\text{synth}}$ may be modelled in such a way that it allows downstream S-learners to better estimate CATEs than other learners. While we acknowledge this phenomenon may disadvantage certain downstream models, we note that our outcome error metric, U_{PEHE}, averages across a number of downstream learner types, such that conducting generative model selection with U_{PEHE} should lead to good performance across a wide variety of downstream learners, not just those similar to $Q_{Y|W,\mathbf{X}}$, helping to reduce this congeniality bias.

M Allocation of the privacy budget in STEAM

In STEAM, uniform distribution of the privacy budget ϵ amongst the three component models ensures (ϵ, δ)-DP. However, such allocation is uninformed on the difficulty of modelling of $P_{\mathbf{X}}$, $P_{W|\mathbf{X}}$, and $P_{Y|W,\mathbf{X}}$, and their relative importance to downstream analysts.

In relation to the importance of each distribution, one immediate improvement can be to distribute ϵ according to some preference function $f : (0, \infty) \times \triangle^2 \to \epsilon \cdot \triangle^2$ (where $\triangle^2$ is the 2-simplex) which takes input of the budget ϵ and weights $\mathbf{w}$ for the relative importance of good modelling in $Q_{\mathbf{X}}$, $Q_{W|\mathbf{X}}$, and $Q_{Y|W,\mathbf{X}}$, and outputs a corresponding ϵ distribution. For example, a simple preference function definition would be $f(\epsilon, \mathbf{w}) = \epsilon \cdot \mathbf{w}$ where $\mathbf{w}$ could be defined by a data holder with some prior knowledge of the importance level of each component distribution to downstream analysts. Another approach, if it is not necessary to specify the desired ϵ distribution *a priori*, is to treat it as a hyperparameter, to be tuned over a series of runs to optimize some metric, such as a combination of $P_{\alpha,X}$, $R_{\beta,X}$, JSD_{π}, and U_{PEHE}.

Incorporating knowledge of the complexity of modelling $P_{\mathbf{X}}$, $P_{W|\mathbf{X}}$, and $P_{Y|W,\mathbf{X}}$ is more difficult. While some proxy measures could be established, such as the number of covariates in $\mathbf{X}$ indicating the

complexity of $P_{W|\mathbf{X}}$, establishing a robust understanding of how the complexity of these distributions relate and compare, is highly non-trivial, and as such we leave this for future work.

N GENERATION UNDER COVARIATE SHIFT

Generative model performance degrades under *covariate shift*. Covariate shift is a phenomenon which arises in data containing treatments, as the covariates of treated and untreated patients tend to differ, i.e. treated and untreated covariates tend to be drawn from different distributions. As the difference between these distributions grows, so too does covariate shift. We demonstrate degradation under high covariate shift through a simple experiment, where we create $\mathcal{D}_{\text{real}}$ with $d = 50$ covariates that are drawn from the following mixture model:

$$\mathbf{X} \sim \frac{9}{10}\mathcal{N}(\mu, I) + \frac{1}{10}\mathcal{N}(-\mu, I) \tag{18}$$

We set the mixture to have uneven weights between the two distributions to more closely simulate a real-world scenario, as treated instances are typically outnumbered in datasets, such as in IHDP and Jobs, which have under 20% treated. We model $\mathcal{D}_{\text{synth}}$ on $\mathcal{D}_{\text{real}}$ with varied μ using a TabDDPM model, and we see in Figure 8 that the metrics P_α and R_β decrease as μ, and therefore the covariate shift, increases.

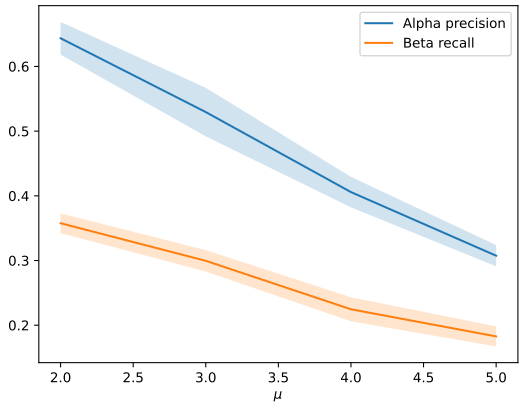

Figure 8: P_α and R_β for $\mathcal{D}_{\text{synth}}$ as covariate shift within $\mathcal{D}_{\text{real}}$ increases.

Since this effect is observed in standard generative models, it will also be observed in STEAM, which uses a generative model for $Q_{\mathbf{X}}$. Performance degredation under this setting is unideal, since some degree of covariate shift is likely to arise in observational data, and therefore future works to remedy this are important.

A simple idea we explored was to use separate models to generate the covariates of each treatment group. We found, however, that the reduction in sample size that each generator received, as a result of the data splitting, negated any gains made by removing covariate shift, except in extreme shift scenarios. This was especially true for treated instances, due to their typical under-representation in data. There are many potential avenues that stem from this approach. Generation with fine-tuning (Hinton et al., 2012) on each treatment group is one possible direction. Another possibility is designing custom generative models that approach generation similarly to plug-in CATE learners, by scaling the degree to which a representation space is shared between treated and untreated instances. Given the non-trivial nature of this issue, we leave such explorations to future work.

O CAUSAL GENERATIVE MODEL COMPARISON

Causal generative models, which estimate the underlying structural causal model of a dataset, are a related family of generative models. We discuss the differences in positioning and assumptions

of our work compared to causal generative models in our extended related works, and here we produce empirical results for these model on the ACTG, IHDP, and ACIC datasets from Section 7.1. For baseline models, we consider two causal generative models: the additive noise model (ANM) (Hoyer et al., 2008) implementation in the `DoWhy-GCM` python package (Blöbaum et al., 2024), and a diffusion-based causal model (DCM) from Chao et al. (2024). We use the code provided by Chao et al. (2024) in their GitHub `https://github.com/patrickrchao/DiffusionBasedCausalModels` for the baseline implementations, and we use the same hyperparameter settings for both ANM and DCM as in that work.

However, in order to fairly compare these models with STEAM, we must first reconcile the differences in assumptions made, as discussed in Appendix B. For ACTG, IHDP, and ACIC, we do not have knowledge of the true causal graph, we simply know which features are the treatment and outcome. As such, we must first construct some reasonable causal graph using this knowledge to supply to these methods. We do so with three methods:

1. Construction of a naive graph $\mathcal{G}_{\text{naive}}$, in which each covariate causes W and Y, W causes Y, and every pair of covariates has a causal relationship between them;
2. Using the constraint-based PC causal discovery algorithm (Spirtes et al., 2001) to discover an estimated graph $\mathcal{G}_{\text{discovered}}$ from the Markov Equivalence Class for the true causal graph and;
3. Pruning the discovered causal graph $\mathcal{G}_{\text{discovered}}$ by removing any edges which contradict the DGP we assume. As such, any edges from Y to W or $\mathbf{X}$, or from W to $\mathbf{X}$ are removed to form $\mathcal{G}_{\text{pruned}}$.

In Table 16 we report the results for ANM and DCM with each of these graph discovery methods, and we compare to the best performing STEAM models from Appendix I. For each dataset, we see that the relevant STEAM model outperforms all instantiations of the causal generative models in almost every metric, only being outperformed to a statistically significant level in $R_{\beta,X}$ on the ACIC dataset. These results validate that, when the true causal graph is not known, our less restrictive assumptions enable more useful generation of synthetic data containing treatments. We also see that the differences between the graph discovery methods are relatively small.

P Future Work: Extensions to other DGPs

STEAM places minimal assumptions upon the causal graph of $\mathcal{D}_{\text{real}}$. We see that our assumed DGP involving $P_{\mathbf{X}}$, $P_{W|\mathbf{X}}$, and $P_{Y|W,\mathbf{X}}$ is applicable across a very wide range of scenarios for datasets containing treatments, particularly in medical settings, as we describe in Section 6. If this DGP is known to not hold for $\mathcal{D}_{\text{real}}$, then it would be unadvisable to apply STEAM in its current form, as its inductive biases may not be helpful. However, if a similar setting arises where the overarching DGP is known, but the specific causal graph is not, then altering the generation order between $\mathbf{X}$, W, and Y to mimic this alternative DGP, with a similar underlying motivation as in STEAM - that generation mimicking the real DGP is preferable to joint-level generation - can be done.

An example of a more complicated setting, where STEAM is not immediately applicable is when dealing with longitudinal data, which has time-varying covariates, treatments, and outcomes. In this setting, our assumed DGP would not hold in general, as features at a specific time point will likely have temporal causal relationships with earlier features, e.g. at time t, $\mathbf{X}_t$ could be affected by W_{t-1}, which is not modelled in our assumed DGP. Potential alterations to STEAM could adjust for this time-varying DGP, where one could consider a 'base' STEAM model, as described in this work, operating at each time step, with additional inputs from the immediately preceding time step to model temporal causal relationships.

Table 16: $P_{\alpha,\mathbf{X}}$, $R_{\beta,\mathbf{X}}$, JSD_{π}, and U_{PEHE} values for the best performing STEAM models from Table I in comparison to causal generative models. Averaged over 20 runs, with 95% confidence intervals. Bold indicates the statistically significant best performing model.

Dataset	Model	$P_{\alpha,\mathbf{X}}$ (↑)	$R_{\beta,\mathbf{X}}$ (↑)	JSD_{π} (↑)	U_{PEHE} (↓)
ACTG	STEAM $_{\text{TVAE}}$	**0.929 ± 0.008**	**0.486 ± 0.009**	0.958 ± 0.004	**0.492 ± 0.011**
	DCM $\mathcal{G}_{\text{naive}}$	0.773 ± 0.013	0.369 ± 0.006	0.937 ± 0.006	0.665 ± 0.034
	DCM $\mathcal{G}_{\text{discovered}}$	0.756 ± 0.011	0.350 ± 0.007	0.956 ± 0.005	0.605 ± 0.023
	DCM $\mathcal{G}_{\text{pruned}}$	0.758 ± 0.013	0.358 ± 0.007	0.957 ± 0.003	0.596 ± 0.017
	ANM $\mathcal{G}_{\text{naive}}$	0.787 ± 0.007	0.389 ± 0.008	0.954 ± 0.005	0.580 ± 0.017
	ANM $\mathcal{G}_{\text{discovered}}$	0.836 ± 0.007	0.419 ± 0.007	0.952 ± 0.004	0.578 ± 0.019
	ANM $\mathcal{G}_{\text{pruned}}$	0.839 ± 0.008	0.412 ± 0.005	0.952 ± 0.005	0.582 ± 0.014
IHDP	STEAM $_{\text{CTGAN}}$	0.674 ± 0.014	**0.424 ± 0.011**	**0.928 ± 0.009**	**1.709 ± 0.052**
	DCM $\mathcal{G}_{\text{naive}}$	0.557 ± 0.010	0.340 ± 0.009	0.883 ± 0.016	4.878 ± 0.395
	DCM $\mathcal{G}_{\text{discovered}}$	0.658 ± 0.011	0.360 ± 0.007	0.893 ± 0.008	2.059 ± 0.140
	DCM $\mathcal{G}_{\text{pruned}}$ *	0.658 ± 0.011	0.360 ± 0.007	0.893 ± 0.008	2.059 ± 0.140
	ANM $\mathcal{G}_{\text{naive}}$	0.597 ± 0.029	0.379 ± 0.011	0.900 ± 0.005	1.868 ± 0.147
	ANM $\mathcal{G}_{\text{discovered}}$	0.589 ± 0.012	0.359 ± 0.009	0.892 ± 0.008	1.865 ± 0.059
	ANM $\mathcal{G}_{\text{pruned}}$ *	0.589 ± 0.012	0.359 ± 0.009	0.892 ± 0.008	1.865 ± 0.059
ACIC†	STEAM $_{\text{ARF}}$	0.939 ± 0.004	0.393 ± 0.004	**0.977 ± 0.002**	**2.176 ± 0.141**
	DCM $\mathcal{G}_{\text{discovered}}$	0.942 ± 0.004	0.422 ± 0.003	0.957 ± 0.003	4.249 ± 0.132
	DCM $\mathcal{G}_{\text{pruned}}$	0.939 ± 0.004	0.420 ± 0.004	0.959 ± 0.002	4.340 ± 0.159
	ANM $\mathcal{G}_{\text{discovered}}$	0.929 ± 0.003	0.404 ± 0.003	0.872 ± 0.002	4.193 ± 0.127
	ANM $\mathcal{G}_{\text{pruned}}$	0.930 ± 0.004	0.404 ± 0.003	0.880 ± 0.002	4.481 ± 0.174

* $\mathcal{G}_{\text{pruned}}$ is the same as $\mathcal{G}_{\text{discovered}}$ for IHDP

† Excessive runtime caused the exclusion of $\mathcal{G}_{\text{naive}}$ ACIC results

Q STEAM DIAGRAM

See the below for a pictoral representation of the DGPs generic synthetic data generation methods, real datasets containing treatments, and STEAM. STEAM is designed to closely mimic the real DGP.

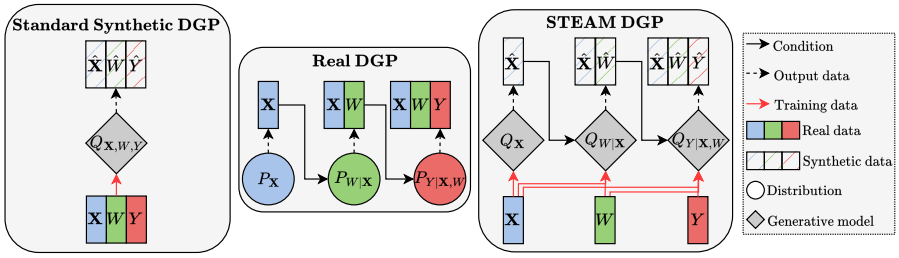

Figure 9: DGPs for generic generative models (left), real datasets (middle), and STEAM (right).