Toward Valid Generative Clinical Trial Data with Survival Endpoints

Perrine Chassat[*] PERRINE.CHASSAT@INRIA.FR
Inria, Université Paris Cité, Inserm, HeKA, F-75015 Paris, France

Van Tuan Nguyen[*] VAN-TUAN.NGUYEN@INRIA.FR
Inria, Université Paris Cité, Inserm, HeKA, F-75015 Paris, France

Lucas Ducrot LUCAS.DUCROT@UNIV-ROUEN.FR
Inria, Université Paris Cité, Inserm, HeKA, F-75015 Paris, France

Emilie Lanoy EMILIE.LANOY@APHP.FR
AP-HP, Hôpital Européen Georges Pompidou, Unité de Recherche Clinique, APHP Centre, Paris, France.
Inserm, Centre d'Investigation Clinique 1418 (CIC1418) Epidémiologie Clinique, Paris, France.

Agathe Guilloux AGATHE.GUILLOUX@INRIA.FR
Inria, Université Paris Cité, Inserm, HeKA, F-75015 Paris, France

Abstract

Clinical trials face mounting challenges: fragmented patient populations, slow enrollment, and unsustainable costs, particularly for late phase trials in oncology and rare diseases. While external control arms built from real-world data have been explored, a promising alternative is the generation of synthetic control arms using generative AI. A central challenge is the generation of time-to-event outcomes, which constitute primary endpoints in oncology and rare disease trials, but are difficult to model under censoring and small sample sizes. Existing generative approaches, largely GAN-based, are data-hungry, unstable, and rely on strong assumptions such as independent censoring. We introduce a variational autoencoder (VAE) that jointly generates mixed-type covariates and survival outcomes within a unified latent variable framework, without assuming independent censoring. Across synthetic and real trial datasets, we evaluate our model in two realistic scenarios: (i) data sharing under privacy constraints, where synthetic controls substitute for original data, and (ii) control-arm augmentation, where synthetic patients mitigate imbalances between treated and control groups. Our method outperforms GAN baselines on fidelity, utility, and privacy metrics, while revealing systematic miscalibration of type I error and power. We propose a post-generation selection procedure that improves calibration, highlighting both progress and open challenges for generative survival modeling.

Keywords: Generative models, Clinical trial simulation, Survival endpoints, Variational autoencoder (VAE), Type I error and power, Control arm augmentation, Privacy-preserving data sharing

Data and Code Availability We used phase III clinical trial datasets in our experiments. We accessed the publicly available AIDS Clinical Trials Group (ACTG) 320 study via Stanford University HIV drug resistance database. The other 3 datasets - trials NCT00119613, NCT00113763, NCT00339183 - are accessible after registration at the Project Data Sphere. The code is publicly available at https://github.com/aguilloux/survgen-clinical-trials.

Institutional Review Board (IRB) This research does not require IRB approval.

1. Introduction

Synthetic clinical data generation is an emerging frontier in machine learning for health, with applications ranging from privacy-preserving data sharing to benchmarking and clinical trial simulation. The clinical trial context is particularly compelling: as precision medicine fragments patient populations into increasingly fine strata (e.g., molecular subgroups), late Phase II-III randomized trials face slow enrollment,

[*] These authors contributed equally

early discontinuation before reaching planned sample size, and in consequence escalating costs. These issues are most acute in oncology and rare diseases, where limited populations make traditional randomized designs difficult to execute. Synthetic data has therefore been proposed as a way to alleviate these barriers—by enabling external validation, supporting meta-analyses, or augmenting underpowered control arms.

Several approaches exist for generating patient-level data. Those based on mechanistic models provide interpretable scenarios grounded in biological knowledge (Carlier et al., 2018; Delobel et al., 2023; Wang et al., 2024). More recently, data-driven generative models have been applied to tabular clinical datasets, with methods such as TGAN, TVAE (Xu et al., 2019), and CTAB-GAN (Zhao et al., 2021) producing synthetic samples that preserve statistical properties while protecting privacy (D'Amico et al., 2023; Petrakos et al., 2025).

A central open challenge is the generation of time-to-event outcomes, frequently defined as the primary endpoints in most late-phase trials. Existing extensions of GANs to this setting, including SurvivalGAN (Norcliffe et al., 2023), have shown initial promise (D'Amico et al., 2023; Eckardt et al., 2024; Akiya et al., 2024; El Kababji et al., 2025; Elvatun et al., 2025), but face important limitations: they require large training datasets (Wang and Pai, 2023), suffer from instability (Thanh-Tung and Tran, 2020), and assume independent random censoring, which is rarely realistic.

Contributions and Paper Outline Our contributions are threefold: (i) We introduce a novel variational autoencoder (VAE) that jointly generates mixed-type covariates and survival outcomes within a unified latent variable framework, without assuming independent censoring. (ii) We propose a calibration-oriented evaluation framework, going beyond fidelity, utility, and privacy to assess type I error and power in downstream survival analyses. (iii) We investigate two realistic trial scenarios—data sharing under privacy constraints and control-arm augmentation—and show that while our HI-VAE variants outperform GAN and VAE baselines on classical metrics, all models fail to reproduce nominal type I error and power without additional safeguards. We propose a post-generation selection procedure that improves calibration, partially restoring statistical validity. Beyond its methodological contribution, this work also aims to inform current regulatory initiatives at the European Medicines Agency (EMA[1]) and Food and Drug Administration (FDA[2]) on the evaluation and potential integration of AI-based and model-based evidence in clinical development.

The remainder of the paper is organized as follows. Section 2 details our model. Section 3 presents the experimental setup, including datasets, baselines, and evaluation metrics. Section 4 reports our findings, and discussion.

2. Method

Formally, we consider that the observed tabular data from a clinical trial can be represented as

$$\mathcal{D} = \{(x_i, t_i, \delta_i)\}_{i=1}^{N},$$

where $x_i \in \mathbb{R}^d$ denotes a vector of covariates for subject i; unless otherwise specified, this vector includes the treatment assignment variable $e_i \in \{0, 1\}$, which indicates whether the individual received the treatment (1) or was assigned to the control group (0). The variable $t_i > 0$ represents the observed time, corresponding either to an event time (primary endpoint) or a censoring time, and $\delta_i \in \{0, 1\}$ is the event indicator (1 if the event occurred, 0 if censored). Finally, $N = N^T + N^C$ denotes the total number of trial participants, with N^T (respectively N^C) individuals assigned to the treatment (respectively control) arm.

To generate synthetic clinical trial data, we extend the HI-VAE (Heterogeneous and Incomplete Variational Autoencoder) of Nazábal et al. (2020) to explicitly model censored survival times. The HI-VAE is a VAE-based model designed to handle missing data, in which the Gaussian prior is replaced by a mixture of Gaussians, allowing it to capture greater heterogeneity in complex clinical data. This choice is further motivated by the advantages of VAE-based formulations for tabular data generation over GAN-based models in terms of training stability, data efficiency, and interpretable latent representations (Xu et al., 2019).

More specifically, we consider a hierarchical model with three latent variables for each subject i. The first two variables jointly define a Gaussian mixture in the latent space: a one-hot vector $\mathbf{s}_i \in \{0, 1\}^L$

[1] https://www.ema.europa.eu/en/human-regulatory-overview/research-development/innovation-task-force-briefing-meetings

[2] https://www.fda.gov/about-fda/cder-center-clinical-trial-innovation-c3ti/c3ti-areas-interest

indicates the mixture component and is drawn from a categorical prior with uniform mixing probabilities, and a continuous latent variable $\mathbf{z}_i \in \mathbb{R}^K$ is drawn conditionally on $\mathbf{s}_i$. Their priors are given by

$$p(\mathbf{s}_i) = \mathrm{Cat}(\mathbf{s}_i \mid \boldsymbol{\pi}), \quad \boldsymbol{\pi}_\ell = 1/L,$$
$$p(\mathbf{z}_i \mid \mathbf{s}_i) = \mathcal{N}\big(\mathbf{z}_i \mid \boldsymbol{\mu}_p(\mathbf{s}_i), \mathrm{Id}\big),$$

where $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^L$ denote the component means. Finally, an intermediate representation $\mathbf{y}_i = g(\mathbf{z}_i) \in \mathbb{R}^H$ is obtained through a neural network g, which maps $\mathbf{z}_i$ to a homogeneous latent space that captures dependencies among heterogeneous attributes. The generative process is parameterized by the joint likelihood of the observed data (x_i, t_i, δ_i), factorized as:

$$p(x_i, t_i, \delta_i \mid \mathbf{z}_i, \mathbf{s}_i) = \prod_{j=1}^{d} p\big(x_i^j \mid h_j^i\big) p(t_i, \delta_i \mid \mathbf{z}_i, \mathbf{s}_i)$$

where $h_j^i = h_j(\mathbf{y}_i, \mathbf{s}_i)$, h_j is neural network, which outputs the parameters of the feature-specific conditional density $p(x_i^j \mid h_j(\mathbf{y}_i, \mathbf{s}_i))$ determined by the type of the j-th variable. Common choices include, but are not limited to, the Gaussian distribution for continuous variables, with

$$h_j(\mathbf{y}_i, \mathbf{s}_i) = \big(\mu_j(\mathbf{y}_i, \mathbf{s}_i), \sigma_j^2(\mathbf{y}_i, \mathbf{s}_i)\big),$$

the log-normal or Poisson distributions for strictly positive or count data, and the multinomial-logit distribution for categorical variables (for more details, we refer the reader to Appendix A.1).

For the part of the distribution associated with the time-to-event outcomes $p(t_i, \delta_i \mid \mathbf{z}_i, \mathbf{s}_i)$, we chose to parameterize the complete density (rather than only the distribution of the event times) as

$$\Big[p\big(t_i \mid \eta_T(\mathbf{y}_i, \mathbf{s}_i)\big)\, \bar{P}\big(t_i \mid \eta_C(\mathbf{y}_i, \mathbf{s}_i)\big)\Big]^{\delta_i}$$
$$\Big[\bar{P}\big(t_i \mid \eta_T(\mathbf{y}_i, \mathbf{s}_i)\big)\, p\big(t_i \mid \eta_C(\mathbf{y}_i, \mathbf{s}_i)\big)\Big]^{1-\delta_i},$$

where p denotes a density and $\bar{P}$ the associated survival function. These are parameterized by neural networks: $\eta_T(\mathbf{y}_i, \mathbf{s}_i)$ for the event time and $\eta_C(\mathbf{y}_i, \mathbf{s}_i)$ for the censoring time. In practice, several choices are possible for p. In our implementation, it is modeled using either a Weibull distribution or a piecewise-constant density model. We refer to these two variants as `HI-VAE_Weibull` and `HI-VAE_piecewise`, respectively. The detailed formulations of these two choices are provided in Appendix A.1.

For recognition models, we define the variational approximate posteriors as $q(\mathbf{s}_i \mid x_i, t_i, \delta_i) = \mathrm{Cat}(\mathbf{s}_i \mid \boldsymbol{\pi}(x_i, t_i, \delta_i))$, $q(\mathbf{z}_i \mid x_i, t_i, \delta_i, \mathbf{s}_i) = \mathcal{N}(\mathbf{z}_i \mid \boldsymbol{\mu}_q(x_i, t_i, \delta_i, \mathbf{s}_i), \boldsymbol{\Sigma}_q(x_i, t_i, \delta_i, \mathbf{s}_i))$ where $\boldsymbol{\pi}$, $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$ are parametrized by independent neural networks.

The Evidence Lower Bound (ELBO) to be optimized from the data writes

$$\log p(\mathcal{D}) = \log p\big(x_i, t_i, \delta_i, i = 1, \dots, N\big)$$
$$\geq \mathcal{L}_{\text{HI-VAE}} = \sum_{i=1}^{N} \mathcal{L}^i_{\text{HI-VAE}} \qquad (1)$$

and

$$\mathcal{L}^i_{\text{HI-VAE}} = \mathbb{E}_{q(\mathbf{z}_i, \mathbf{s}_i | x_i, t_i, \delta_i)}\big[p(x_i, t_i, \delta_i \mid \mathbf{z}_i, \mathbf{s}_i)\big]$$
$$- \mathbb{E}_{\{q(\mathbf{s}_i | x_i, t_i, \delta_i)\}}\big[\mathrm{KL}(q(\mathbf{z}_i | x_i, t_i, \delta_i, \mathbf{s}_i) | p(\mathbf{z}_i | \mathbf{s}_i))\big]$$
$$- \mathrm{KL}(q(\mathbf{s}_i | x_i, t_i, \delta_i) | p(\mathbf{s}_i)),$$

here KL stands for the Kullback–Leibler divergence.

Synthetic samples are then generated through posterior sampling. Specifically, each original data point (x_i, t_i, δ_i) is first encoded into latent representations using the variational encoders $q(\mathbf{s}_i \mid x_i, t_i, \delta_i)$ and $q(\mathbf{z}_i \mid x_i, t_i, \delta_i, \mathbf{s}_i)$. Differentiable samples are then drawn from these learned posteriors using the Gumbel–Softmax trick (Li et al., 2019) for $\mathbf{s}_i$, and the Gaussian reparameterization trick (Kingma and Welling, 2013) for $\mathbf{z}_i \mid \mathbf{s}_i$. Finally, the sampled latent variables are passed through the decoder $p(x_i, t_i, \delta_i \mid \mathbf{z}_i, \mathbf{s}_i)$ to generate synthetic data. This sampling procedure ensures that the generated samples reflect the statistical structure learned from the original dataset (Nazábal et al., 2020).

3. Experiments

We frame our experiments in the standard late phase randomized clinical trial setting. A typical analysis proceeds in two steps. First, the performance of randomization in balancing patients characteristics across experimental and control treatment arms is assessed as recommended in established reporting guidelines (Moher et al., 2010). Second, the experimental treatment effect on survival outcomes is evaluated, most commonly using a log-rank test to compare survival distributions and/or by estimating the hazard ratio with a Cox proportional hazards model (Fleming and Harrington, 2013). Our experimental question is whether these conclusions—obtained when using the original data—can be faithfully reproduced when the control arm is replaced or augmented by synthetic data generated with state-of-the-art models.

3.1. Control arm generation strategies

More specifically, we consider two distinct scenarios, both of which correspond to pressing issues in clinical research:

Data sharing under privacy constraints An investigating team has access to the full clinical trial dataset, including N^T treated and N^C control patients. To enable external validation or meta-analysis, the control arm must be shared with another group. However, direct release of patient-level data is often prohibited by privacy regulations and data-use restrictions. In this setting, the task is to generate a synthetic control arm that closely reproduces the original distribution, preserving utility while ensuring confidentiality.

Control-arm augmentation In many trials, particularly for rare diseases, highly targeted therapies, or early-phase studies, the control group is much smaller than the treated group ($N^C \ll N^T$). This imbalance reduces statistical power and compromises treatment effect estimation. Here, synthetic data are used to augment the control arm, partially correcting the imbalance and improving the robustness of downstream analyses.

In our experiments, the generator is trained on a fraction $\upsilon \in \{1/3, 2/3, 1\}$ of the available control patients ($N_{\text{train}} = \upsilon N^C$). We then generate N_{gen} synthetic control arms, each of size $N_{\text{sim}} = N^T$ (matching the treated group). When $\upsilon = 1$, this corresponds to the data-sharing scenario; when $\upsilon < 1$, it corresponds to augmentation, with effective control-arm expansion factors of ×3 and ×1.5. An alternative training strategy, where treated-arm data are also used, is described in Section 4.4.

3.2. Competing algorithms

To benchmark our HI-VAE-based generative models, we compared them to two state-of-the-art baselines proposed in Norcliffe et al. (2023) and available in the `synthcity` library (Qian et al., 2023a): SurvivalGAN and SurvivalVAE. Both methods are based on a modular architecture where the generation of synthetic covariates is separated from the modeling of survival outcomes, allowing flexibility in the choice of generative model—either GAN- or VAE-based—and independent strategies for assigning event times and censoring. This separated two-stage modeling pipeline is a key difference with our proposed HI-VAE model, which jointly learns covariates and time-to-event outcomes within a unified latent framework. Other tabular data generators such as CTGAN, TVAE, or normalizing flows were compared to SurvivalGAN in Norcliffe et al. (2023) and found less effective for survival modeling, and were therefore excluded from our comparison. We refer to these two competing methods as `Surv-GAN` and `Surv-VAE` (see Appendix A.1).

3.3. Training and hyperparameter tuning

We train the models on a dataset $\mathcal{D}_{\text{train}}$ with the same structure as described in Section 2. In our model, we optimize the parameters by maximizing the evidence lower bound (ELBO, Eq. 1) in a batch learning setting with the Adam optimizer (Kingma and Ba, 2014). Moreover, we monitor training convergence using early stopping based on this ELBO to avoid overfitting. For competing methods with modular architectures, the parameters of each module are optimized in a sequential manner, see Norcliffe et al. (2023) for more details.

Hyperparameters - such as the learning rate, batch size, and latent state dimensions - are selected by minimizing the distance between the Kaplan–Meier survival curves estimated from the observed data and those derived from synthetic data. We perform this optimization using the Optuna framework (Akiba et al., 2019), conducting up to 150 trials to ensure reliable selection of hyperparameters. We refer the reader to Appendix A.3 and B.4 for full details of the hyperparameter selection process.

3.4. Metrics

We evaluate the quality of the synthetic data along three dimensions—resemblance, utility, and privacy—using the metrics implemented in the `synthcity` package (Qian et al., 2023b). Our selection of metrics is guided by the scoping review of Kaabachi et al. (2025).

Data resemblance We quantify the similarity between synthetic and original control arm data distributions using the Jensen-Shannon distance (*J-S distance*). The J-S distance measures the dissimilarity between two probability distributions, with smaller values indicating closer resemblance (range from 0 to 1).

Utility We evaluate whether the synthetic data can replicate clinically relevant statistical patterns using the *Survival curves distance*. The survival curves distance integrates the absolute difference between Kaplan–Meier survival curves estimated from the original and synthetic control arms, with smaller distances indicating better preservation of survival patterns.

Privacy Following Steier et al. (2025), we evaluate privacy preservation using the *K-map score*. The K-map score measures the minimum group size of quasi-identifier combinations in the synthetic control arm, with higher values indicating lower re-identification risk and thus better privacy preservation.

Additional metrics, including the Kolmogorov–Smirnov test, an XGBoost classifier performance score, and the Nearest Neighbor Distance Ratio, are described in Appendix A.4.

Computation of expected type I errors and powers For the simulation experiments, we assess type I error and power by computing the proportion of rejected null hypotheses from the log-rank test (the standard nonparametric test for comparing survival curves between groups, Fleming and Harrington (2013)) and comparing them with the theoretical (asymptotic) power given by the formula of Schoenfeld (1983). Further details are provided in Appendix A.5.

Multiple test adjustment For each Monte Carlo replication, we compute the p-values of the log-rank tests comparing survival curves between the original control and each generated control set, yielding N_{sim} p-values in total. To account for multiple comparisons, we then apply the Benjamini–Hochberg (BH) procedure Benjamini and Hochberg (1995) to control the false discovery rate (FDR) at a nominal level $\alpha = 0.05$.

3.5. Simulations

To evaluate our method, we simulate $M = 100$ independent Monte Carlo replications of a dataset obtained as follows. For each subject $i \in \{1, \ldots, N\}$, covariates $x_i \in \mathbb{R}^d$ are sampled from a multivariate normal distribution with Toeplitz covariance. Continuous features remain unchanged, while binary ones are obtained by thresholding $x_i^j \leftarrow \mathbf{1}\{x_i^j > 0\}$. Treatments e_i are drawn from $\mathcal{B}(1, 0.5)$. Event times (τ_i) and censoring times (c_i) follow Weibull distributions parameterized by covariates and a treatment coefficient $\beta \in \{0, 0.2, \ldots, 1.0\}$. Distributions are either fully independent or conditionally independent given covariates. By construction, $\beta = 0$ indicates no risk difference between control and treated groups, with the effect size increasing in β. Further details are given in Appendix A.6.

3.6. Clinical trial data

We further validate our approach on four phase III clinical trial datasets.

NCT00119613 Placebo-controlled trial of darbepoetin alfa with platinum-based chemotherapy in extensive-stage small-cell lung cancer (Pirker et al., 2008).

NCT00113763 Trial of panitumumab plus chemotherapy versus chemotherapy alone in metastatic colorectal cancer (Van Cutsem et al., 2007).

NCT00339183 Trial of chemotherapy with or without panitumumab in metastatic or recurrent head and neck squamous cell carcinoma (Vermorken et al., 2013).

ACTG 320 Trial showing triple-drug antiretroviral therapy reduced AIDS progression or death compared with dual therapy in HIV-infected patients (Hammer et al., 1997).

A summary of the main characteristics of the simulated and clinical datasets is provided in Table 1 and details on the distribution of the variables are provided in Appendix A.7.

Although our empirical evaluation focuses on oncology and HIV datasets, the proposed framework is readily applicable to any clinical domain where the primary endpoint is of time-to-event type (e.g., overall survival, disease-free survival, progression-free survival), which represents a substantial share of phase III trials across therapeutic areas.

4. Results

4.1. Performance comparisons on classical metrics

We first evaluated the performance of our algorithm in its two variants, `HI-VAE_Weibull` and

Table 1: Dataset characteristics: number of control samples (N^C), treated samples (N^T), and number of static features by type: categorical (d^{cat}), positive-valued (d^{pos}), and continuous real-valued (d^{real}).

Dataset	N^C	N^T	d^{cat}	d^{pos}	d^{real}
Simulation	300	300	6	0	6
ACTG 320	577	574	5	2	1
NCT00119613	236	235	7	0	2
NCT00113763	475	470	10	0	2
NCT00339183	260	260	5	0	1

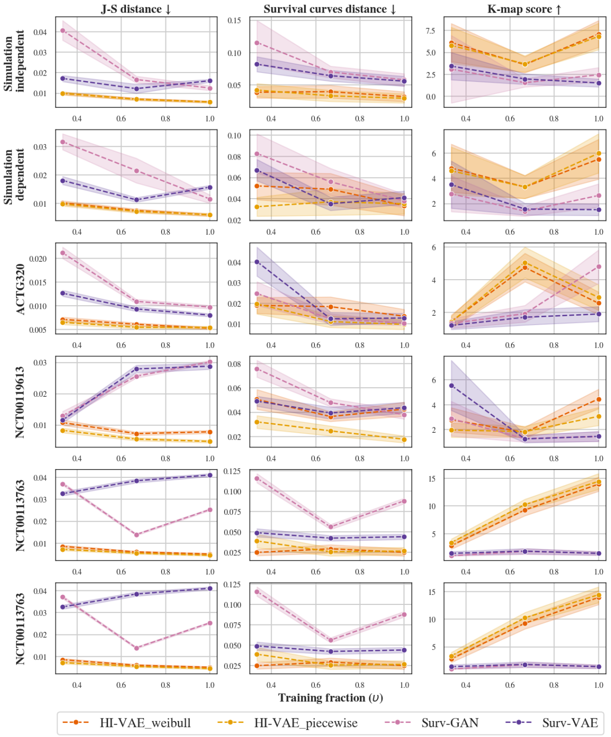

Figure 1: Performance comparison on simulated and real datasets, using J-S distance, survival curve distance, and K-map score. Arrows indicate directions of better performance.

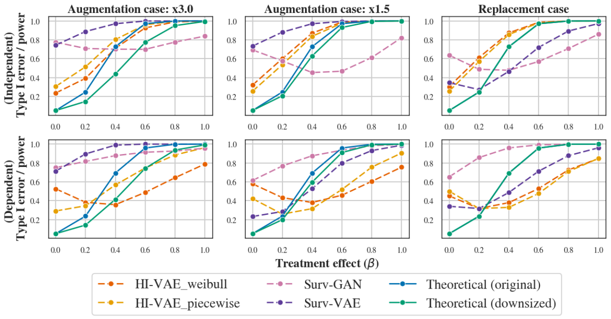

Figure 2: Type I error and power estimation for **independent** case (**top**) and **dependent** case (**bottom**). Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

HI-VAE_piecewise, against competing methods Surv-GAN and Surv-VAE, using the fidelity, utility, and privacy metrics described in Section 3. Figure 1 reports results on both simulated and real datasets. Our HI-VAE consistently achieved lower divergence from the original distributions and higher fidelity, particularly on survival outcomes, while maintaining competitive privacy scores. Supplementary results with additional metrics are provided in Appendix B.1.

4.2. Type I error and powers estimation in generated data

We then assessed whether generative models reproduce the expected type I error ($\beta = 0$) and statistical power ($\beta \in \{0.2, 0.4, \ldots, 1.0\}$) predicted by Schoenfeld's formula (Schoenfeld, 1983). Figures 2 summarize results for both independent and dependent cases over $M = 100$ Monte Carlo replications.

Across all methods, type I error rates were consistently inflated, and power curves often deviated sharply from theoretical expectations, highlighting that good fidelity scores or usual utility metrics do not guarantee valid downstream inference.

4.3. Post-generation selection

We next investigated the reasons for the poor statistical performance of the generative algorithms. Figure 3 shows the proportion of Monte Carlo replications in which at least one of the $N_{\text{gen}} = 200$ generated datasets for which the multiple-testing–adjusted log-rank test of equal survival between original and generated controls was not rejected. Our method achieved this in at least 80% of replications across all experimental settings—and often considerably more—making it substantially more reliable and robust than competing approaches.

We then retained, for each case, the best generated control dataset from among the $N_{\text{gen}} = 200$ candidates. "Best" was defined as the dataset yielding the highest p-value in a log-rank test comparing survival

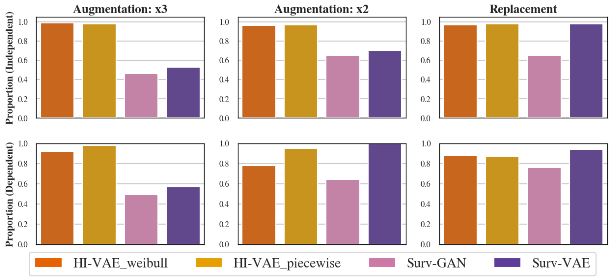

Figure 3: Proportion of Monte Carlo replications with at least one generated dataset not rejected by the adjusted log-rank test (at the 5% level) against the original controls.

distributions between the original training and generated controls. In the simulation study, this procedure resulted in one selected dataset per Monte Carlo replication. Further details on the selection procedure and on the computation of type I error rates and power are provided in Appendix A.5.

Figure 4 reports the type I error rates and powers for both independent and dependent settings. In the independent case, the datasets generated by our algorithm closely reproduce the target theoretical power for a control size of $N^C = 300$ (blue line), even when trained on as few as $N_{\text{train}} = 100$ or 200 original controls. This indicates a genuine augmentation effect, particularly in the ×3 augmentation scenario. By contrast, this effect is attenuated or inexistent in the dependent setting. Type I error rates are, however, somewhat inflated in augmentation settings, with inflation increasing with the augmentation factor.

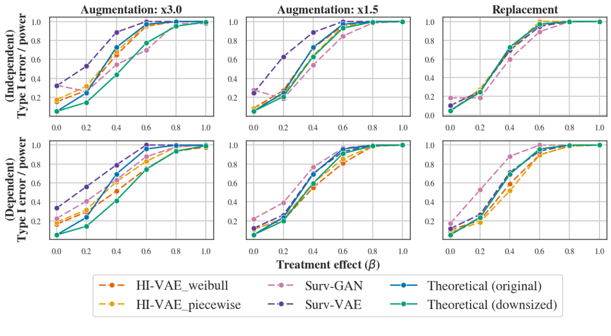

Figure 4: Type I error and power estimation after post-generation selection for **independent** case (**top**) and **dependent** case (**bottom**). Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

Conversely, the `Surv-GAN` algorithm generally fails to reproduce the expected type I error rates or powers, possibly due to the limited proportion of generated control survival data resembling the originals. The `Surv-VAE` algorithm attains the target theoretical power in three of the six experiments, most notably in the replacement scenario, but falls short in the others and does not adequately control type I error in most cases. Its heterogeneous performance across experiments may partly reflect variability in how closely the generated control survival data approximate the originals. These observations are consistent with Figure 3.

Importantly, other performance metrics are not degraded by the selection procedure (see Figure 13 in Appendix B.2).

Finally, reducing the candidate pool (e.g., to 50 or 100 datasets instead of 200) does not alter these conclusions (see Figure 15 in Appendix B.2). However, selecting the top 20% of generated datasets leads to poor performance for all methods (Figure 14 in Appendix B.2).

4.4. Extended experiments

Risk-model discrimination and calibration To compare the discrimination and calibration of the risk model trained on synthetic versus real control data, we performed an additional experiment in which the Cox proportional hazards model (Cox, 1972) was trained separately on the two datasets. The predictive performance was then evaluated on an independent real test set using the C-index (Harrell et al., 1996) and the integrated Brier score (Graf et al., 1999). As shown in Appendix B.3 (Figures 16 - 17), the results indicate that our method produces synthetic-trained models whose performance closely matches that of real controls in both discrimination and calibration, whereas Surv-GAN and Surv-VAE achieve good discrimination but exhibit poorer calibration.

Impact of hyperparameter optimization We conducted additional experiments to assess the impact of both the hyperparameter search strategy and the random seed within the Optuna framework. As detailed in Appendix B.4 (Figures 18–21), we compared three search methods (1–3) for two models (`HI-VAE_piecewise` and `Surv-VAE`) in the independent setting, training only on the control arm.

Overall, strategies that evaluated performance on generated synthetic datasets of full control size (methods 1 and 3) proved more robust than relying solely on validation-set evaluation, which can be limited in size (method 2). With these approaches, the relative performance of our model versus `Surv-VAE` remained stable across random seeds. However, seed choice still influenced the absolute performance within each algorithm, highlighting the sensitivity to hyperparameter selection. In our experiments, we fixed the number of Optuna trials to 150, but in practice, a more extensive search would likely be needed to ensure optimal results.

Training setup: control vs. control + treated In our initial experiments, the generator was trained exclusively on the available controls—or on a subset thereof in the augmentation setting. We then explored an alternative strategy in which treated-arm data were also incorporated during training, as illustrated in Figure 7 of Appendix A.2. The rationale was that in clinical trial contexts, where datasets are often small (e.g., $\upsilon = 1/3$ with 100 controls and 300 treated), leveraging treated data might yield more robust models.

However, results on both simulated and real datasets did not clearly support this intuition (see Figure 5 and Appendix B.5). On simulated data, resemblance and utility metrics were generally superior when training on controls only, whereas privacy metrics improved when both controls and treated were used—consistent with the expected utility–privacy trade-off. Among the models, `Surv-VAE` was the only one to show a modest benefit from including treated data; the others did not. On real datasets, all models performed better on average when trained solely on controls, particularly with respect to data resemblance metrics.

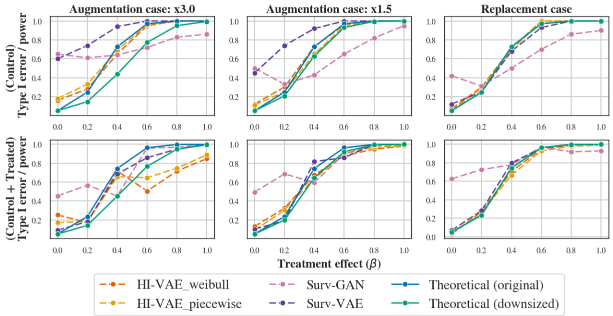

Figure 5: Type I error and power estimation after post-generation selection for independent case under two training strategies: using only available control samples (**top**) and using both control and treated arms (**bottom**). Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

Privacy considerations: prior sampling and differential privacy Our privacy analysis showed that posterior-based sampling falls short of the thresholds required for effective data sharing under current standards such as EMA Policy 0070 (EMA, 2025), which recommends K-map values above 11 for public release. In our setting, posterior sampling yielded K-map values of about 4–6 and NNDR values around 0.2–0.4. These levels may still be acceptable under controlled-access regimes (e.g., data use agreements) or for augmentation scenarios, but they remain insufficient for regulatory-grade data sharing. To address this gap, we explored two complementary strategies.

The first strategy was to replace posterior sampling with prior-based sampling. Unlike posterior sampling, which draws latent variables conditional on the observed data, prior sampling generates them directly from the model priors, producing fully synthetic samples that are in principle less dependent on the original dataset and therefore potentially more privacy-preserving. In practice, however, we found only minor differences between the two approaches. On independent simulated datasets, posterior sampling performed slightly better overall (Figure 6, Table 2). On real datasets, results were more mixed: prior sampling improved some K-map scores but not NNDR values. Overall, we did not observe consistent privacy benefits from switching to prior-based sampling.

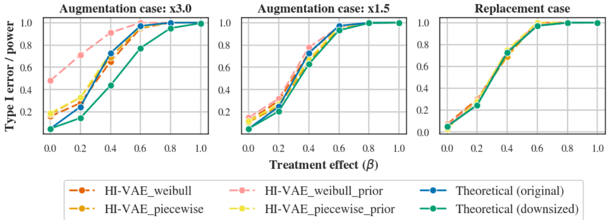

Figure 6: Type I error and power estimation after post-generation selection for independent case. Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

The second strategy was a preliminary application of differential privacy using the `Opacus` framework (Yousefpour et al., 2021). We trained the HI-VAE models with per-step noise injection (`noise_multiplier = 2.0`). Under this configuration, privacy metrics did not improve substantially compared to the non-private models (Appendix B.7). Stronger protection would likely require more aggres-

Dataset	HI-VAE	Metric	Post.	Prior
Simulation (independent)	_piecewise	K-map	$\mathbf{5.44 \pm 3.41}$	4.97 ± 3.17
		NNDR	$\mathbf{0.42 \pm 0.07}$	0.41 ± 0.07
	_weibull	K-map	$\mathbf{5.61 \pm 3.57}$	5.03 ± 3.62
		NNDR	$\mathbf{0.42 \pm 0.07}$	0.41 ± 0.06
NCT00119613	_piecewise	K-map	3.07 ± 1.45	$\mathbf{3.49 \pm 1.70}$
		NNDR	$\mathbf{0.18 \pm 0.04}$	0.17 ± 0.04
	_weibull	K-map	4.47 ± 1.57	$\mathbf{4.86 \pm 1.53}$
		NNDR	0.21 ± 0.05	$\mathbf{0.22 \pm 0.04}$

Table 2: Comparison of K-map (↑) and NNDR (↑) scores for posterior vs. prior sampling in the replacement setting ($\upsilon = 1$), evaluated on the independent simulated dataset and the NCT00119613 dataset.

sive parameter settings, which in turn would degrade utility.

In summary, while synthetic patients generated with posterior or prior sampling are not direct replicas of real ones, neither approach provided sufficient privacy guarantees for open data release. Differential privacy offered no improvement under moderate noise levels, underscoring the difficulty of balancing utility and privacy in this setting.

5. Conclusion

We introduced a VAE-based framework for generating synthetic control arms with time-to-event outcomes. Across synthetic and real clinical trial datasets, and under both data-sharing and data-augmentation scenarios, our method outperformed baselines in terms of fidelity, utility, and privacy. However, a key limitation emerged: despite strong performance on classical metrics, all models—including ours—yielded miscalibrated survival analyses, with inflated type I error and biased power. Post-generation selection improved calibration and restored power in most settings, though type I error remained partially elevated.

These findings highlight the importance of evaluating generative models for health not only on fidelity and privacy, but also on their downstream statistical calibration. A model that looks strong by standard ML metrics may still fail when applied to clinical inference.

Our privacy analysis also underscored important limitations. Posterior sampling yielded K-map values consistently below the EMA Policy 0070 (EMA, 2025) benchmark (≥ 11), confirming that current configurations remain insufficient for public data release. Nevertheless, such values may still be acceptable under controlled-access regimes (e.g., data use agreements) or in augmentation scenarios. Complementary metrics (NNDR, detection tests) indicated only partial protection against re-identification. A preliminary attempt to integrate differential privacy (Opacus) did not yield substantial gains, underscoring the sensitivity of privacy–utility trade-offs to parameter choices.

Future work should explore stronger privacy-preserving techniques—such as calibrated ε–δ differential privacy, diffusion- or transformer-based generators with privacy-aware objectives—and the development of calibration-aware training strategies. However, adapting these models to survival settings requires further methodological development. More broadly, bridging generative modeling with domain-specific notions of validity, such as error control in survival analysis, will be essential to ensure reliability in real-world applications.

In summary, we provide the first systematic evaluation of type I error and power in generative survival models. Our results demonstrate both the promise of VAEs for survival data generation and the need for methodological advances before such models can be safely applied in clinical research.

Acknowledgments & Fundings

For this work, AG and VTN have benefited from the support of the National Agency for Research under the France 2030 program with the reference ANR-22-PESN-0016. PC, LD and AG received funding from the MEDITWIN Bpifrance i-Demo project as part of the France 2030 program. AG has also benefited from the Institut National du Cancer. All authors thank Armelle Arnoux and Sandrine Katsahian, who helped design the experimental settings.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Ippei Akiya, Takuma Ishihara, Keiichi Yamamoto, et al. Comparison of synthetic data generation techniques for control group survival data in oncology clinical trials: simulation study. *JMIR medical informatics*, 12(1):e55118, 2024.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57 (1):289–300, 1995.

Aurélie Carlier, A Vasilevich, M Marechal, Jan de Boer, and L Geris. In silico clinical trials for pediatric orphan diseases. *Scientific reports*, 8(1): 2465, 2018.

Aziliz Cottin, Nicolas Pecuchet, Marine Zulian, Agathe Guilloux, and Sandrine Katsahian. Idnetwork: A deep illness-death network based on multistate event history process for disease prognostication. *Statistics in Medicine*, 41(9):1573–1598, 2022.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

Saverio D'Amico, Daniele Dall'Olio, Claudia Sala, Lorenzo Dall'Olio, Elisabetta Sauta, Matteo Zampini, Gianluca Asti, Luca Lanino, Giulia Maggioni, Alessia Campagna, et al. Synthetic data generation by artificial intelligence to accelerate research and precision medicine in hematology. *JCO clinical cancer informatics*, 7:e2300021, 2023.

Thibault Delobel, Luis E Ayala-Hernández, Jesús J Bosque, Julián Pérez-Beteta, Salvador Chulián, Manuel García-Ferrer, Pilar Piñero, Philippe Schucht, Michael Murek, and Víctor M Pérez-García. Overcoming chemotherapy resistance in low-grade gliomas: A computational approach. *PLoS computational biology*, 19(11): e1011208, 2023.

Jan-Niklas Eckardt, Waldemar Hahn, Christoph Röllig, Sebastian Stasik, Uwe Platzbecker, Carsten Müller-Tidow, Hubert Serve, Claudia D Baldus, Christoph Schliemann, Kerstin Schäfer-Eckart, et al. Mimicking clinical trials with synthetic acute myeloid leukemia patients using generative artificial intelligence. *NPJ digital medicine*, 7(1):76, 2024.

Samer El Kababji, Nicholas Mitsakakis, Elizabeth Jonker, Ana-Alicia Beltran-Bless, Gregory Pond, Lisa Vandermeer, Dhenuka Radhakrishnan, Lucy Mosquera, Alexander Paterson, Lois Shepherd, et al. Augmenting insufficiently accruing oncology clinical trials using generative models: validation study. *Journal of medical Internet research*, 27: e66821, 2025.

Severin Elvatun, Daan Knoors, Simon Brant, Christian Jonasson, and Jan F Nygård. Synthetic data as external control arms in scarce single-arm clinical trials. *PLOS Digital Health*, 4(1):e0000581, 2025.

EMA. External guidance on the implementation of the European Medicines Agency policy 0070 on the publication of clinical data for medicinal products for human use. Technical Report EMA/90915/2016, Rev. 1, version 1.5, May 2025. URL https://www.ema.europa.eu/en/documents/regulatory-procedural-guideline/external-guidance-implementation-european-medicines-agency-policy-publication-clinical-data-medicinal-products-human-use-version-15_en.pdf. Accessed: 2025-10-09.

Thomas R Fleming and David P Harrington. *Counting processes and survival analysis*. John Wiley & Sons, 2013.

Michael Friedman et al. Piecewise exponential models for survival data with covariates. *The Annals of Statistics*, 10(1):101–113, 1982.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Scott M Hammer, Kathleen E Squires, Michael D Hughes, Janet M Grimes, Lisa M Demeter, Judith S Currier, Joseph J Eron Jr, Judith E Feinberg, Henry H Balfour Jr, Lawrence R Deyton, et al. A controlled trial of two nucleoside analogues plus indinavir in persons with human immunodeficiency virus infection and cd4 cell counts of 200 per cubic millimeter or less. *New England Journal of Medicine*, 337(11):725–733, 1997.

Frank E Harrell, Kerry L Lee, and Daniel B Mark. Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15:361–387, 1996.

Frank E Harrell Jr. Package 'hmisc'. *CRAN2018*, 2019:235–236, 2019.

Bayrem Kaabachi, Jérémie Despraz, Thierry Meurers, Karen Otte, Mehmed Halilovic, Bogdan Kulynych, Fabian Prasser, and Jean Louis Raisaro. A scoping review of privacy and utility metrics in medical synthetic data. *NPJ digital medicine*, 8(1): 60, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL https://api.semanticscholar.org/CorpusID:216078090.

Håvard Kvamme and Ørnulf Borgan. Continuous and discrete-time survival prediction with neural networks. *arXiv preprint arXiv:1910.06724*, 2019.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela van der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11842. URL https://ojs.aaai.org/index.php/AAAI/article/view/11842.

Xiaopeng Li, Zhourong Chen, Leonard K. M. Poon, and Nevin Lianwen Zhang. Learning latent superstructures in variational autoencoders for deep multidimensional clustering. In *International Conference on Learning Representations*, 2019. URL https://api.semanticscholar.org/CorpusID:56657907.

David Moher, Sally Hopewell, Kenneth F Schulz, Victor Montori, Peter C Gøtzsche, Philip J Devereaux, Diana Elbourne, Matthias Egger, and Douglas G Altman. Consort 2010 explanation and elaboration: updated guidelines for reporting parallel group randomised trials. *Bmj*, 340, 2010.

Alfredo Nazábal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes, 2020. ISSN 0031-3203. URL https://www.sciencedirect.com/science/article/pii/S0031320320303046.

Alexander Norcliffe, Bogdan Cebere, Fergus Imrie, Pietro Lio, and Mihaela van der Schaar. Survivalgan: Generating time-to-event data for survival analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 10279–10304. PMLR, 2023.

Niki Z Petrakos, Erica EM Moodie, and Nicolas Savy. A framework for generating realistic synthetic tabular data in a randomized controlled trial setting. *Statistics in Medicine*, 44(18-19):e70227, 2025.

Robert Pirker, Rodryg A Ramlau, Wolfgang Schuette, Petr Zatloukal, Irene Ferreira, Tom Lillie, and Johan F Vansteenkiste. Safety and efficacy of darbepoetin alfa in previously untreated extensive-stage small-cell lung cancer treated with platinum plus etoposide. *Journal of Clinical Oncology*, 26 (14):2342–2349, 2008.

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

Zhaozhi Qian, Rob Davis, and Mihaela Van Der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. *Advances in neural information processing systems*, 36:3173–3188, 2023b.

David A Schoenfeld. Sample-size formula for the proportional-hazards regression model. *Biometrics*, pages 499–503, 1983.

Amy Steier, Lipika Ramaswamy, Andre Manoel, and Alexa Haushalter. Synthetic data privacy metrics. *arXiv preprint arXiv:2501.03941*, 2025.

Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020.

Eric Van Cutsem, Marc Peeters, Salvatore Siena, Yves Humblet, Alain Hendlisz, Bart Neyns, Jean-Luc Canon, Jean-Luc Van Laethem, Joan Maurel, Gary Richardson, et al. Open-label phase iii trial of panitumumab plus best supportive care compared with best supportive care alone in patients with chemotherapy-refractory metastatic colorectal cancer. *Journal of clinical oncology*, 25(13):1658–1664, 2007.

Jan B Vermorken, Jan Stöhlmacher-Williams, Irina Davidenko, Lisa Licitra, Eric Winquist, Cristian Villanueva, Paolo Foa, Sylvie Rottey, Krzysztof Skladowski, Makoto Tahara, et al. Cisplatin and fluorouracil with or without panitumumab in patients with recurrent or metastatic squamous-cell carcinoma of the head and neck (spectrum): an open-label phase 3 randomised trial. *The lancet oncology*, 14(8):697–710, 2013.

Hanwen Wang, Theinmozhi Arulraj, Alberto Ippolito, and Aleksander S Popel. From virtual patients to digital twins in immuno-oncology: lessons learned from mechanistic quantitative systems pharmacology modeling. *NPJ digital medicine*, 7 (1):189, 2024.

Winston Wang and Tun-Wen Pai. Enhancing small tabular clinical trial dataset through hybrid data augmentation: combining smote and wcgan-gp. *Data*, 8(9):135, 2023.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Jinsung Yoon, Lydia N. Drumright, and Mihaela van der Schaar. Anonymization through data synthesis using generative adversarial networks (adsgan). *IEEE Journal of Biomedical and Health Informatics*, 24(8):2378–2388, 2020. doi: 10.1109/JBHI.2020.2980262.

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.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian conference on machine learning*, pages 97–112. PMLR, 2021.

Appendix A. Supplementary materials for Section 2

A.1. Algorithmic and Implementation Details

Details on the feature-specific conditional densities

- Continuous real-valued variables (Normal): $h_j(\mathbf{y}_i, \mathbf{s}_i) = \big(\mu_j(\mathbf{y}_i, \mathbf{s}_i), \sigma_j^2(\mathbf{y}_i, \mathbf{s}_i)\big)$,
- Positive real-valued variables (Log-Normal): $h_j(\mathbf{y}_i, \mathbf{s}_i) = \big(\mu_j(\mathbf{y}_i, \mathbf{s}_i), \sigma_j^2(\mathbf{y}_i, \mathbf{s}_i)\big)$,
- Count variables (Poisson): $h_j(\mathbf{y}_i, \mathbf{s}_i) = \lambda_j(\mathbf{y}_i, \mathbf{s}_i)$,
- Categorical variables (Multinomial-logit): $h_j(\mathbf{y}_i, \mathbf{s}_i) = \big(h_{j0}(\mathbf{y}_i, \mathbf{s}_i), \dots, h_{j(R-1)}(\mathbf{y}_i, \mathbf{s}_i)\big)$.

Details on the parameterization of the time-to-event density We consider two parameterizations of the density p in our implementation. In the `HI-VAE_Weibull` variant, the density p is modeled using a Weibull distribution. For each subject, the neural network $\eta(\mathbf{y}_i, \mathbf{s}_i)$ outputs the scale and shape parameters, denoted as $\mathrm{sc}(\mathbf{y}_i, \mathbf{s}_i)$ and $\mathrm{sh}(\mathbf{y}_i, \mathbf{s}_i)$, respectively. The survival function $\bar{P}$ then writes

$$\bar{P}(t) = \exp\left(-\left(\tfrac{t}{\mathrm{sc}}\right)^{\mathrm{sh}}\right).$$

In the `HI-VAE_piecewise` variant, we follow Friedman et al. (1982); Lee et al. (2018); Kvamme and Borgan (2019); Cottin et al. (2022) and discretize the time axis into K disjoint intervals $i_1, \dots, i_K$, with the right endpoint of i_K chosen larger than the maximal observed time. A neural network $\eta(\mathbf{y}_i, \mathbf{s}_i)$ with a softmax output layer provides interval probabilities $(p_1, \dots, p_K)$ (with $\sum_{k=1}^K p_k = 1$). The survival function $\bar{P}$ is then given by

$$1 - \bar{P}(t) = \sum_{k=1}^{i(t)-1} p_k + \frac{(t - i_{i(t)-1})}{|i_{i(t)}|} p_{i(t)},$$

where $|i_k|$ denotes the length of the interval i_k and $i(t)$ the index of the interval containing t.

More details on survivalGAN and survivalVAE SurvivalGAN first encodes covariates via Gaussian mixture and one-hot encodings, then constructs a conditional GAN (based on ADS-GAN (Yoon et al., 2020)) to generate synthetic covariates guided by a compact condition vector. A survival function (parameterized by DeepHit (Lee et al., 2018)) estimates survival probabilities, followed by a time regressor that outputs event or censoring times. As a result, synthetic samples are generated in sequential stages: sampling latent noise, generating covariates, estimating survival curves, and regressing event times. SurvivalVAE adopts the same pipeline framework, replacing the GAN module with a Tabular Variational Autoencoder (Xu et al., 2019) to generate covariates.

A.2. Details on the control arm generation process

We provide in Figure 7 additional details on how synthetic control arms are generated, illustrating both training on the control arm only and training on the combined control and treated arms (corresponding to additional experiments in Section 4.4).

A.3. Hyperparameter optimization

We list the hyperparameter search space for each algorithm.

- `HI-VAE`. Learning rate: $\{2\mathrm{e}^{-2}, \mathrm{e}^{-3}, \mathrm{e}^{-4}\}$; batch size: $\{0.25, 0.4, 0.6, 0.75\} \times N_{\text{train}} \cup \{100\}$; latent dimensions: $z \in [10, 200]$ (step 10), $y \in [10, 200]$ (step 5), $s \in [10, 200]$ (step 10); number of survival layers (for `HI-VAE_piecewise` only): $\{1, 2\}$; number of piecewise intervals (for `HI-VAE_piecewise` only): $\{5, 10, 15, 20\}$.

- `Surv-VAE`. (from `synthcity` Qian et al. (2023b)) Number of max epochs $\{100, 200, 300, 400, 500\}$; learning rate: $\{\mathrm{e}^{-3}, 2\mathrm{e}^{-4}, \mathrm{e}^{-4}\}$; weight decay: $\{\mathrm{e}^{-3}, \mathrm{e}^{-4}\}$; batch size: $\{64, 128, 256, 512\}$; embedding units: $[50, 500]$ (step 50); encoder/decoder: hidden layers $\in [1, 5]$, hidden units $\in [50, 500]$ (step 50); nonlinearities: $\{\texttt{relu}, \texttt{leaky\_relu}, \texttt{tanh}, \texttt{elu}\}$; dropout $\in [0, 0.2]$.

- `Surv-GAN`. (from `synthcity` Qian et al. (2023b)) Learning rate: $\{\mathrm{e}^{-3}, 2\mathrm{e}^{-4}, \mathrm{e}^{-4}\}$; weight decay: $\{\mathrm{e}^{-3}, \mathrm{e}^{-4}\}$; encoder clusters $\in [2, 20]$; Generator: hidden layers $\in [1, 4]$, hidden units $\{50, 100, 150\}$, nonlinearities $\{\texttt{relu}, \texttt{leaky\_relu}, \texttt{tanh}, \texttt{elu}\}$, dropout $\in [0, 0.2]$; Discriminator: hidden layers $\in [1, 4]$, hidden units $\{50, 100, 150\}$, same nonlinearities, dropout $\in [0, 0.2]$.

A.4. Additional evaluation metrics

Data resemblance The Kolmogorov-Smirnov test (*KS test*) compares the empirical cumulative distribution functions of synthetic and original features, producing a score between 0 and 1, where 1 indicates identical distributions.

Utility We also compute a performance score by training an XGBoost classifier (*Detection XGB*) to distinguish between original and synthetic samples. A result of 0 means the two datasets are indistinguishable (best), and 1 means they are completely distinguishable (worst for privacy).

Privacy We additionally compute the Nearest Neighbor Distance Ratio (*NNDR*), which compares, for each real sample, the distance to its nearest synthetic neighbor relative to the distance to its nearest real neighbor. A score of 0 indicates exact reproduction of real samples in the synthetic dataset (privacy leakage), while a score of 1 indicates that all synthetic samples are far from any real data (stronger privacy).

A.5. Type I error and power computations

Monte Carlo experiments and estimations of type I error and powers In our Monte Carlo experiments, for each value (size effect of the treatment e_i) β in $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, we generated $M = 100$ control and treated arms of resp. sizes N^C and N^T. We obtained M initial p-values for the log-rank tests performed on each replication: $\mathrm{pv}\big(\mathcal{D}_{\text{control},1}, \mathcal{D}_{\text{treated},1}, \beta\big)$ to $\mathrm{pv}\big(\mathcal{D}_{\text{control,M}}, \mathcal{D}_{\text{treated},M}, \beta\big)$. We then computed an approximate power

$$\text{power}_{\text{inital}}(\beta) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\{\mathrm{pv}\big(\mathcal{D}_{\text{control,m}}, \mathcal{D}_{\text{treated},m}, \beta\big) < 0.05\}, \tag{2}$$

notice that, for $\beta = 0$, it corresponds to the approximate type I error.

In a second time, for each replication m, we generated N_{gen} synthetic control arms, $\mathcal{D}^1_{\text{gen,m}}$ to $\mathcal{D}^{N_{\text{gen}}}_{\text{gen,m}}$, from $\mathcal{D}_{\text{control},m}$ the initial control arm (and the $\mathcal{D}_{\text{treated},m}$ treated arm) as depicted in Figure 7. For each of them, we computed the log-rank test p-value $\mathrm{pv}\big(\mathcal{D}^n_{\text{gen,m}}, \mathcal{D}_{\text{treated},m}, \beta\big)$. This is summarized for one Monte Carlo experiment in Figure 7. We then defined the approximate power (resp. type I level) reached by the synthetic control arms as

$$\text{power}_{\text{gen}}(\beta) = \frac{1}{M N_{\text{gen}}} \sum_{m=1}^{M} \sum_{n=1}^{N_{\text{gen}}} \mathbf{1}\{\mathrm{pv}\big(\mathcal{D}^n_{\text{gen,m}}, \mathcal{D}_{\text{treated},m}, \beta\big) < 0.05\},$$

these values are reported in Figure 2.

Finally, we selected only the best generated control dataset in each case (among $N_{\text{gen}} = 200$ candidates) by comparing the p-values in a log-rank test comparing survival distributions between the original training and generated controls.

$$n_{\text{best},m} = \text{argmin}_{n=1,\dots,N_{\text{gen}}}\ \mathrm{pv}\big(\mathcal{D}^n_{\text{gen,m}}, \mathcal{D}_{\text{control},m}\big).$$

We then compute the approximate type I error and power reached by these selected datasets by appliying the following formula

$$\text{power}_{\text{gen,best}}(\beta) = \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\{\text{pv}\big(\mathcal{D}_{\text{gen,m}}^{n_{\text{best},m}}, \mathcal{D}_{\text{treated},m}, \beta\big) < 0.05\}.$$

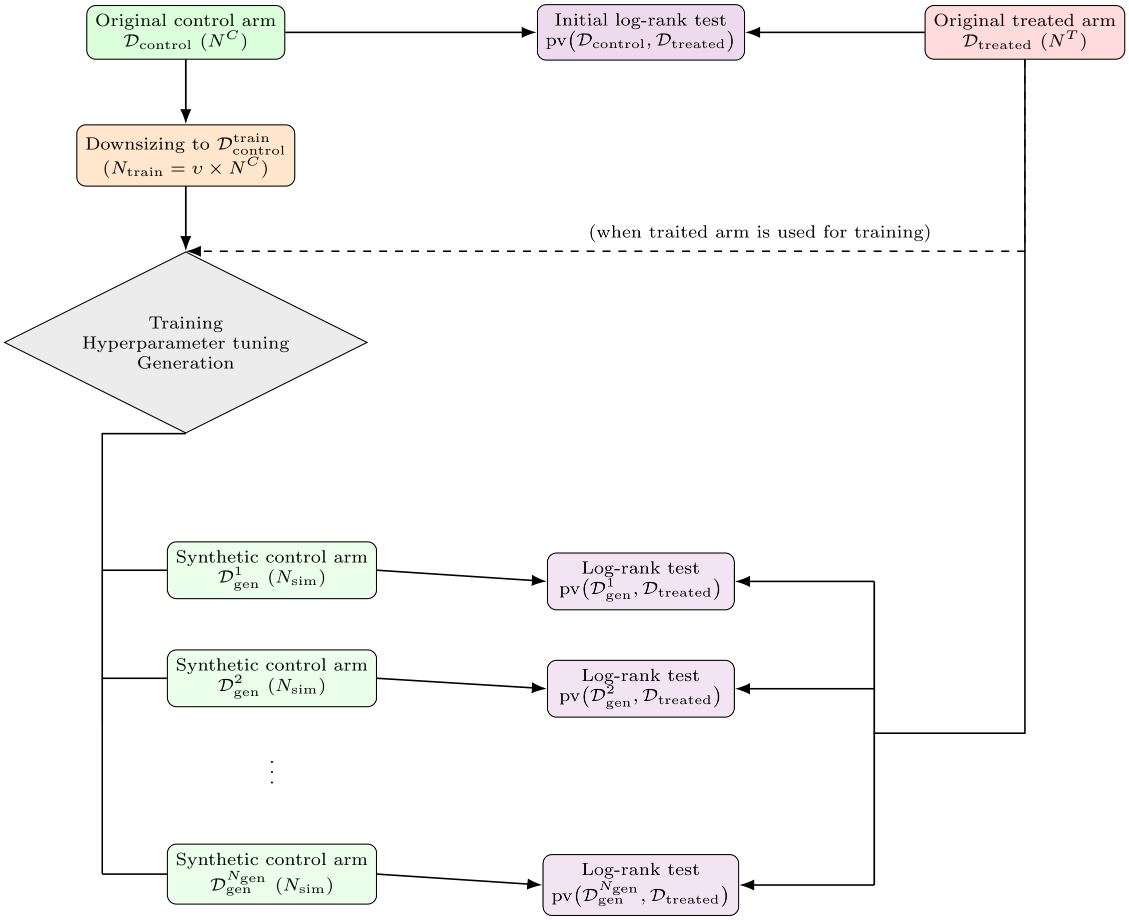
Original control arm
$\mathcal{D}_{\text{control}}$ (N^C)
Initial log-rank test
$\text{pv}(\mathcal{D}_{\text{control}}, \mathcal{D}_{\text{treated}})$
Original treated arm
$\mathcal{D}_{\text{treated}}$ (N^T)
Downsizing to $\mathcal{D}_{\text{control}}^{\text{train}}$
$(N_{\text{train}} = \upsilon \times N^C)$
(when traited arm is used for training)
Training
Hyperparameter tuning
Generation
Synthetic control arm
$\mathcal{D}_{\text{gen}}^{1}$ (N_{sim})
Log-rank test
$\text{pv}(\mathcal{D}_{\text{gen}}^{1}, \mathcal{D}_{\text{treated}})$
Synthetic control arm
$\mathcal{D}_{\text{gen}}^{2}$ (N_{sim})
Log-rank test
$\text{pv}(\mathcal{D}_{\text{gen}}^{2}, \mathcal{D}_{\text{treated}})$
⋮
Synthetic control arm
$\mathcal{D}_{\text{gen}}^{N_{\text{gen}}}$ (N_{sim})
Log-rank test
$\text{pv}(\mathcal{D}_{\text{gen}}^{N_{\text{gen}}}, \mathcal{D}_{\text{treated}})$

Figure 7: Workflow for generating synthetic control arms and evaluating treatment effects. Original control data are downsized and used to train the generator (optionally including treated data). Multiple synthetic control replicates are then produced and compared with the treated arm using log-rank tests.

Theoretical power formula from Schoenfeld (1983) Wherever reported, the theoretical power has been computed the following way, following the implementation of the function `cpower` of the `Hmisc` R package (Harrell Jr, 2019).

$$\text{power}(\tilde{\beta}) = 1 - \Big(\Phi\big(\Phi^{-1}(\alpha/2) - \frac{|\tilde{\beta}|}{\sigma}\big) - \Phi\big(-\Phi^{-1}(\alpha/2) - \frac{|\tilde{\beta}|}{\sigma}\big)\Big)$$

where

- Φ is the cumulative distribution function of the standard Normal,

- $\sigma = \sqrt{\frac{1}{\text{ns}_T} + \frac{1}{\text{ns}_C}}$ with ns_T, ns_C are the (average) number of survivors in the treated and control groups
- $\tilde{\beta}$ is the univariate equivalent of the treatment coefficient β (computed via Monte Carlo experiments in each simulation setting).

A.6. Details on the simulation settings

The static covariates $x_i \in \mathbb{R}^d$ from a multivariate normal distribution with Toeplitz covariance $\Sigma \in \mathbb{R}^{d \times d}$:

$$x_i \sim \mathcal{N}(0, \Sigma), \quad \Sigma_{jk} = \rho^{|j-k|}, \quad j, k = 1, \ldots, d.$$

Event (τ_i) and censoring (c_i) times are simulated as

$$\begin{aligned} \tau_i &= (-\log(1 - u_i) / \exp(\alpha^\top x_i + \beta e_i))^{1/\kappa_T} \\ c_i &= \lambda_C * (-\log(1 - v_i))^{1/\kappa_C} \quad \text{(independent case)} \\ c_i &= \lambda_C * (-\log(1 - v_i) / \exp(\alpha^\top x_i + \beta e_i))^{1/\kappa_C} \quad \text{(dependent case)} \end{aligned}$$

where

- u_i and v_i are drawn from two independent uniform distributions
- $\alpha = (1, -\exp(-1/10), \exp(-2/10), 0, 0, 0)$ and β varies in $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$
- $\lambda_C, \kappa_T, \kappa_C$ have been chosen to reach a censoring level of about 15%.

The observed outcomes are then defined as $t_i = \min(\tau_i, c_i), \quad \delta_i = \mathbf{1}\{\tau_i \leq c_i\}$.

A.7. Variable distribution in real datasets

Across all datasets, each figure consists of three panels: Kaplan–Meier survival curves of observed times (panel **A**), distribution plots of positive and continuous variables (panel **B**), and count plots of categorical and ordinal variables (panel **C**). All plots are stratified by treatment arm (control vs. treated).

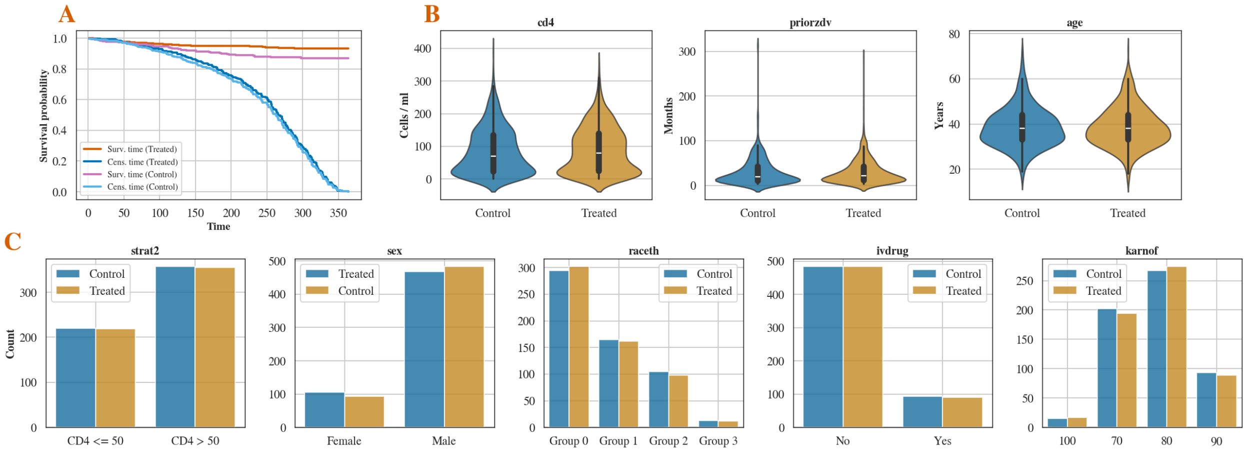

Figure 8: Variable distributions in the ACTG320 dataset.

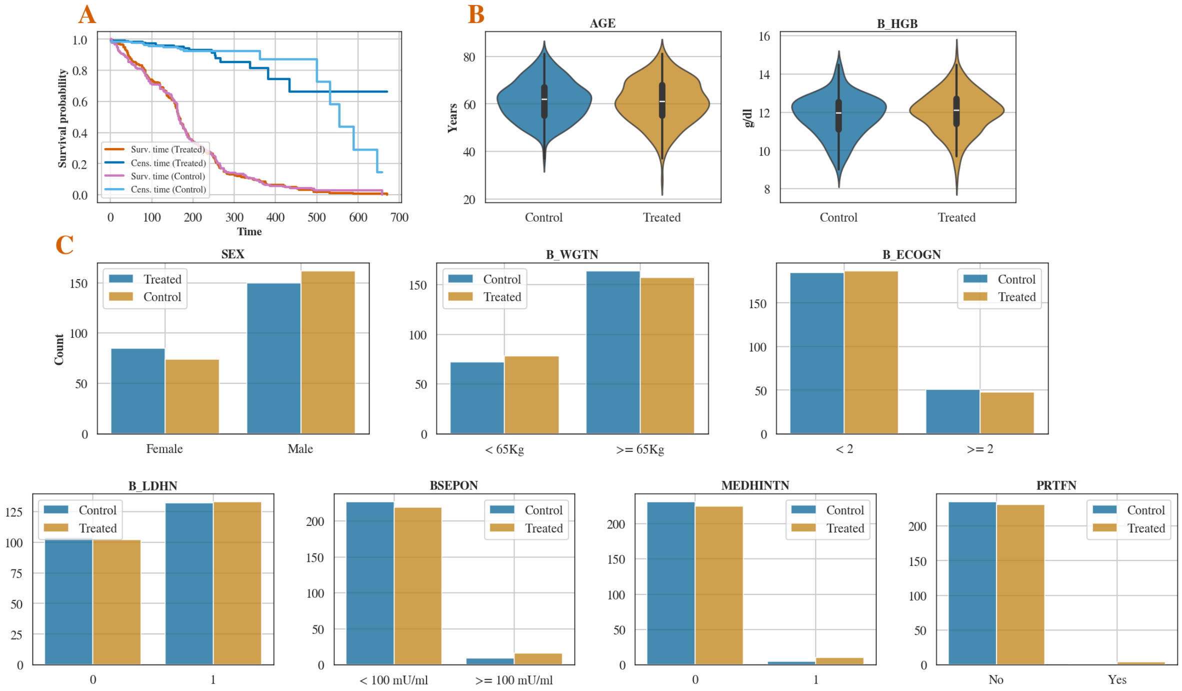

Figure 9: Variable distributions in the NCT00119613 dataset.

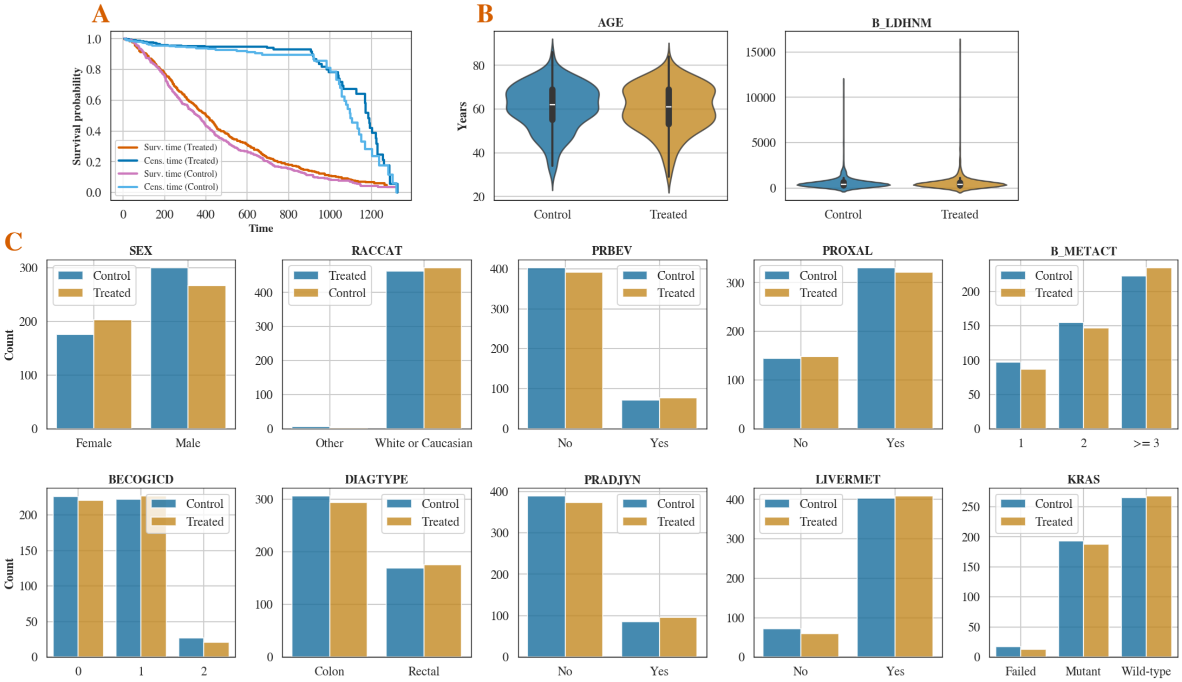

Figure 10: Variable distributions in the NCT00113763 dataset.

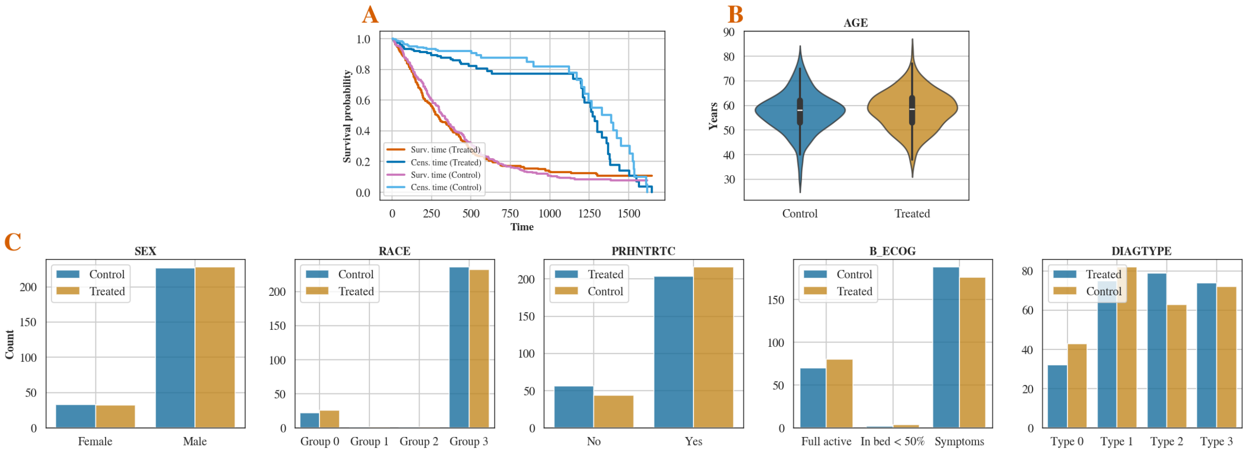

Figure 11: Variable distributions in the NCT00339183 dataset.

Appendix B. Supplementary materials for Section 4

B.1. Additional metrics

We also evaluate the performance of our HI-VAE models against competing methods using the additional metrics described in Appendix A.4, on both simulated and real datasets.

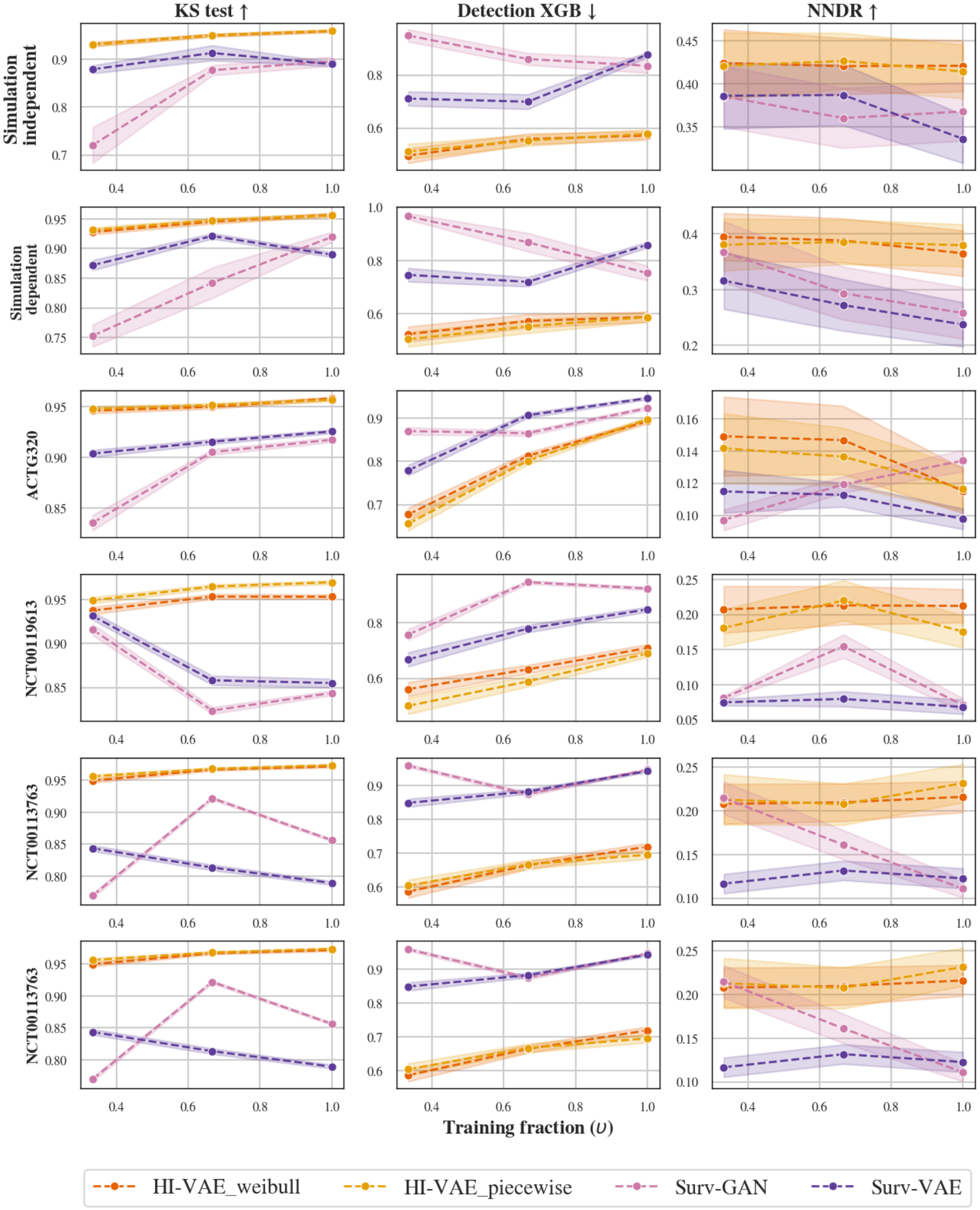

Figure 12: Generative performance comparison on simulated and real datasets, using KS test, detection XGB, NNDR. Arrows indicate the direction corresponding to better performance for each metric.

B.2. Post-generation selection

Here we present additional results on post-generation selection, as discussed in Section 4.3 of the main paper.

	Algorithm	1/3	2/3	1
(Independent) Simulation	HI-VAE_weibull	0.970	0.768	0.745
	HI-VAE_piecewise	0.922	0.842	0.790
	Surv-GAN	0.300	0.318	0.390
	Surv-VAE	0.540	0.684	0.280
(Dependent) Simulation	HI-VAE_weibull	0.599	0.628	0.707
	HI-VAE_piecewise	0.947	0.491	0.154
	Surv-GAN	0.440	0.395	0.285
	Surv-VAE	0.252	0.603	0.454
ACTG 320	HI-VAE_weibull	1.000	0.870	1.000
	HI-VAE_piecewise	1.000	1.000	1.000
	Surv-GAN	0.970	1.000	1.000
	Surv-VAE	0.970	0.990	0.995
NCT00119613	HI-VAE_weibull	1.000	0.970	0.990
	HI-VAE_piecewise	1.000	1.000	0.990
	Surv-GAN	0.990	0.810	0.000
	Surv-VAE	0.925	0.960	0.990
NCT00113763	HI-VAE_weibull	0.990	0.965	0.880
	HI-VAE_piecewise	0.870	0.970	0.940
	Surv-GAN	1.000	0.000	0.000
	Surv-VAE	0.975	0.970	0.975
NCT00339183	HI-VAE_weibull	1.000	1.000	0.995
	HI-VAE_piecewise	1.000	0.995	1.000
	Surv-GAN	1.000	1.000	0.360
	Surv-VAE	0.915	0.820	0.990

Table 3: Proportion of accepted H_0 ($\alpha = 0.05$) in log-rank tests comparing original and generated controls.

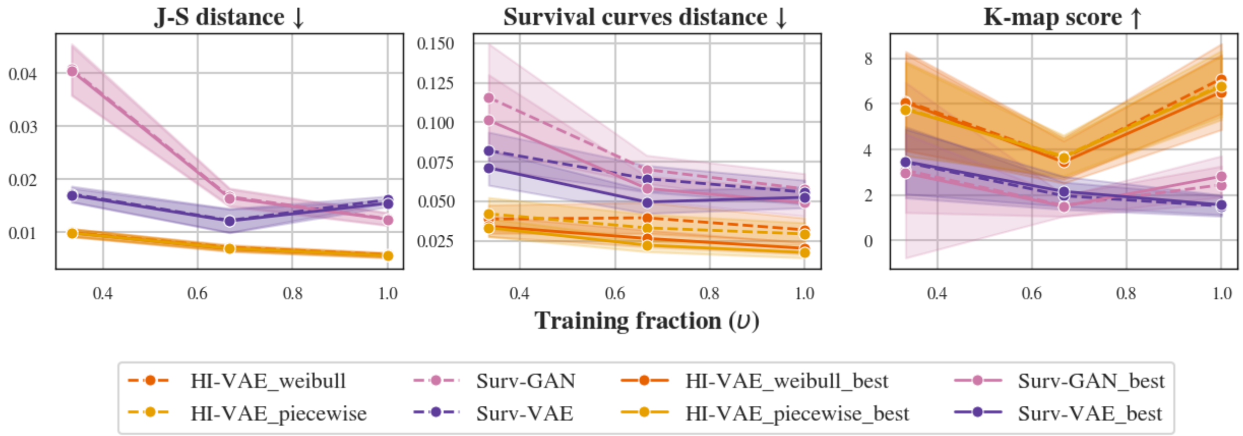

Figure 13: Comparison of generative performance between the best generated dataset and the full set of generated datasets across Monte Carlo experiments, under the independent simulation setting with training restricted to control-arm samples.

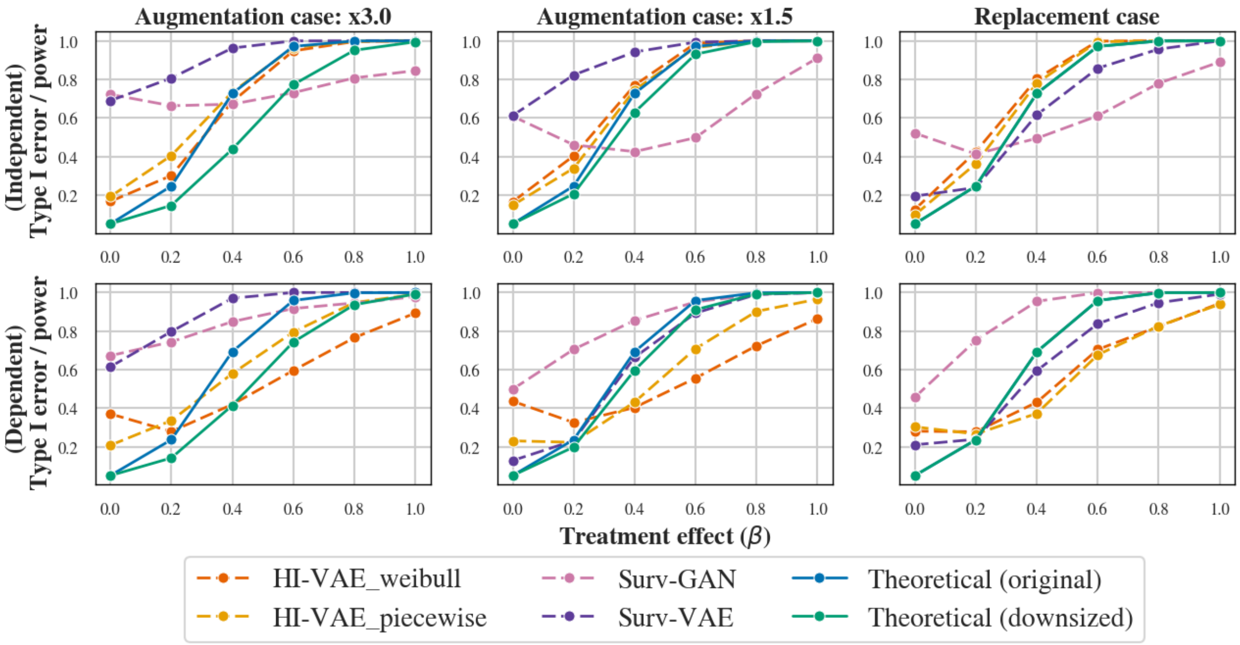

Figure 14: Type I error and power estimation after post-generation selection, based on subsets of the top 20% best generated control arm, for **independent** case (**top**) and **dependent** case (**bottom**). Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

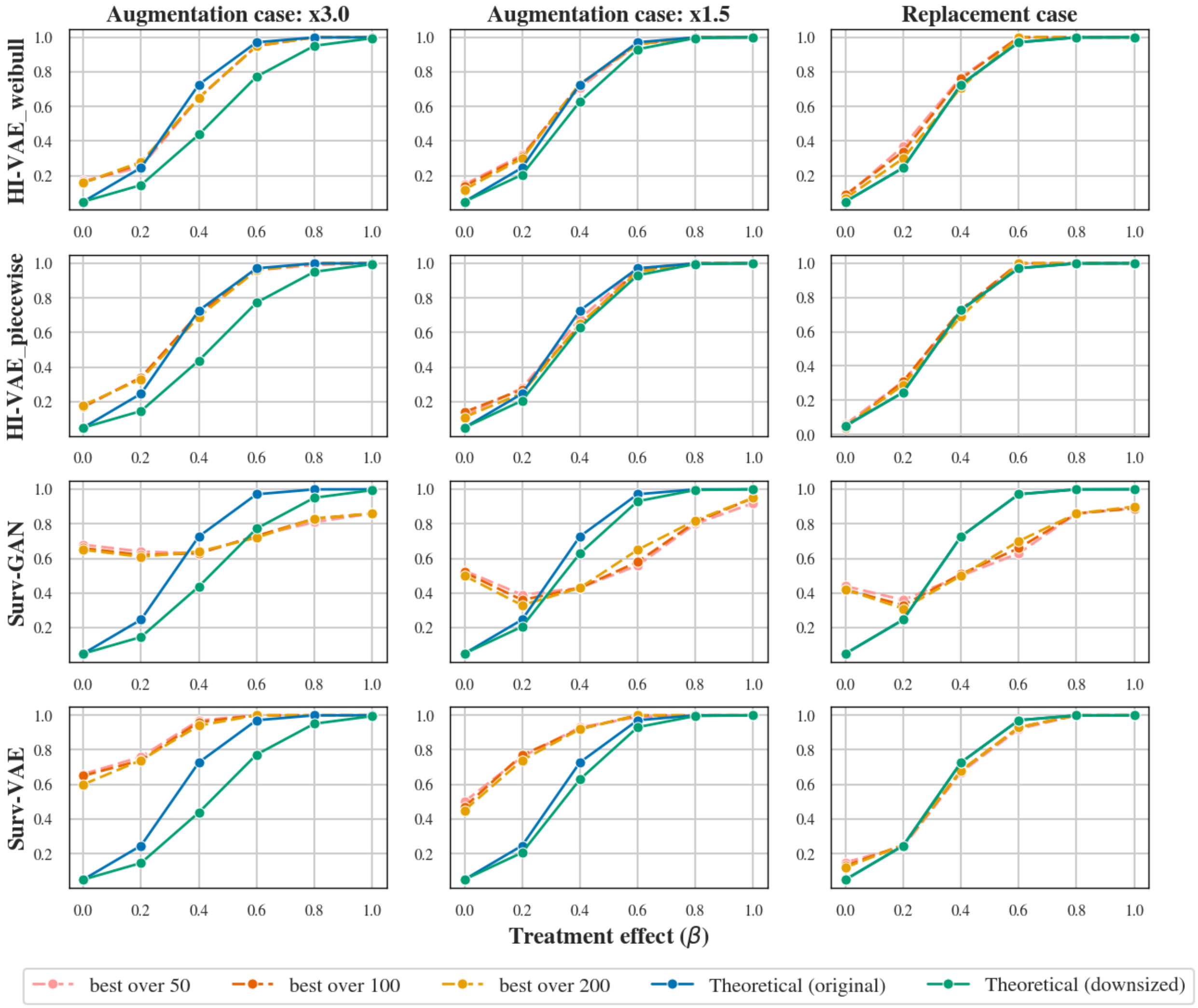

Figure 15: Type I error and power estimation after post-generation selection, based on subsets of varying sizes from the well-performing generated datasets, for an independent simulation setting. Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

B.3. Risk-model discrimination/calibration

We report here the discrimination (C-index) and calibration (integrated Brier score) of Cox models trained separately on either real control data (using a fraction $\upsilon = 2/3$ of the available controls in the augmentation setting) or on the corresponding synthetic data. The predictive performance was then evaluated on an independent real test set consisting of the remaining control samples. Our methods yield synthetic-trained models whose performance closely aligns with that of real controls in both discrimination and calibration, whereas Surv-GAN and Surv-VAE exhibit good discrimination but poorer calibration.

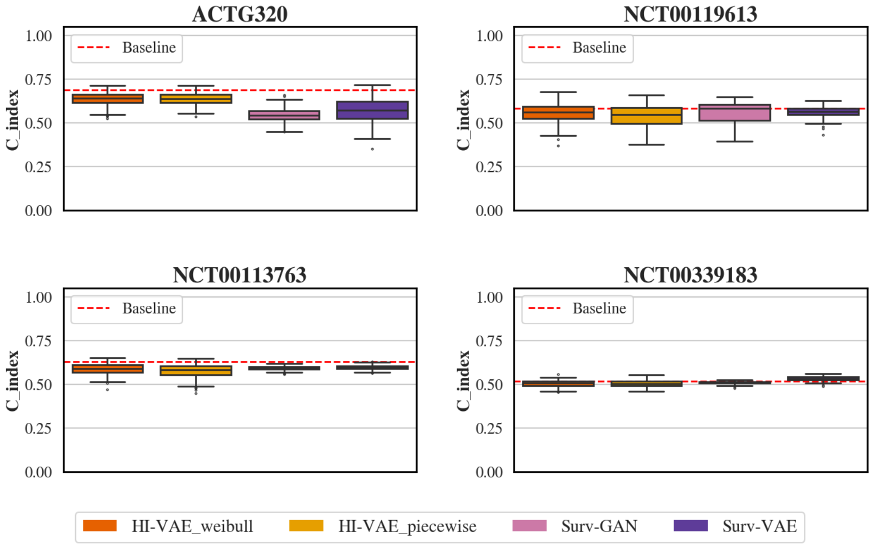

Figure 16: Discrimination performance comparison on real datasets, using C-index metric (*higher* is better). The dashed line represents the performance of the model trained on real control data.

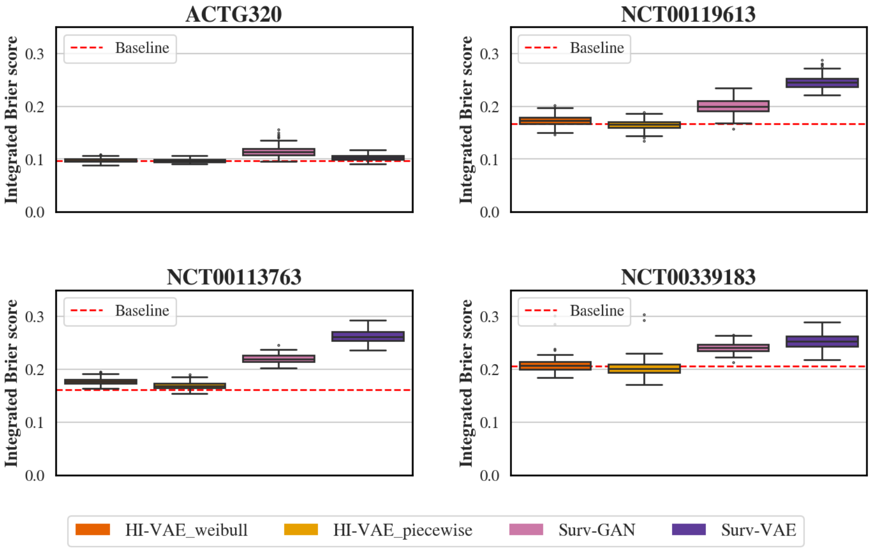

Figure 17: Calibration performance comparison on real datasets, using integrated Brier score metric (*lower* is better). The dashed line represents the performance of the model trained on real control data.

B.4. Impact of hyperparameter optimization

We report here the results of the additional experiments assessing the impact of both the hyperparameter search strategy and the random seed within the Optuna framework (Section 4.4 of the main paper).

Influence of the chosen method for the hyperparameters search We compare the following hyperparameter search methods:

1. Train on the full dataset of shape N_C, then generate N_{gen} synthetic datasets of size $N_{sim} = N_C$.
2. Split the dataset into training and validation sets. Train on the training set, then generate N_{gen} synthetic datasets from the validation set of size $N_{sim} = N_{val}$.
3. Split the dataset into training and validation sets. Train on the training set, then generate N_{gen} synthetic datasets from the full dataset (train + validation) of size $N_{sim} = N_C$.

In all cases, the score is the survival curve distance between the generated and original control arms. For this comparison, we focus on the replacement case ($\upsilon = 1$) in the independent setting trained on controls only.

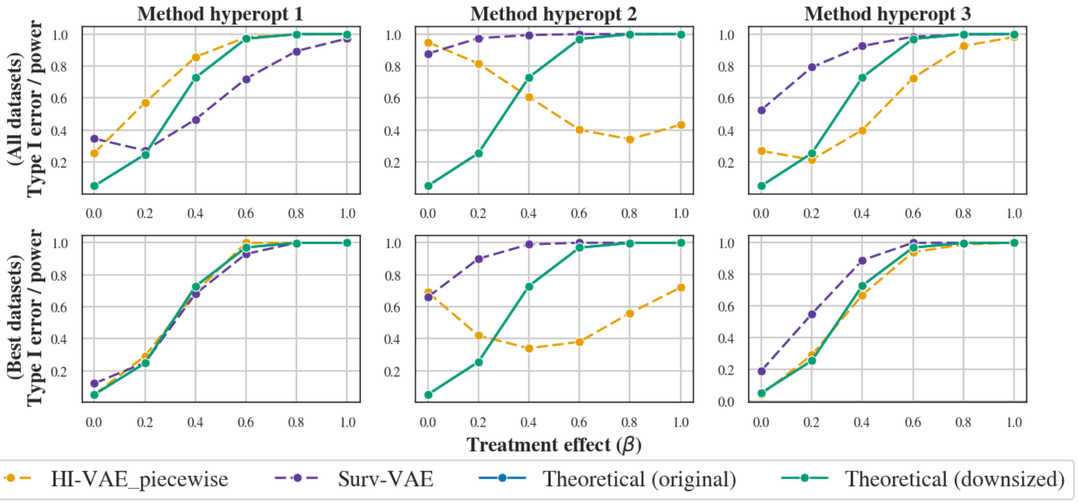

Figure 18: Type I error and power estimation (independent simulation setting, trained on controls only, replacement case, with **different hyperparameter search methods** and post-generation selection). Top: all generated datasets; bottom: best generated dataset. Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

Influence of the seed in the Optuna hyperparameters search framework We investigate the influence of the random seed in the Optuna hyperparameter search framework (Akiba et al., 2019), focusing on the replacement case $\upsilon = 1$ in the independent setting trained on the control arm only, and using method 1 for the hyperparameter search.

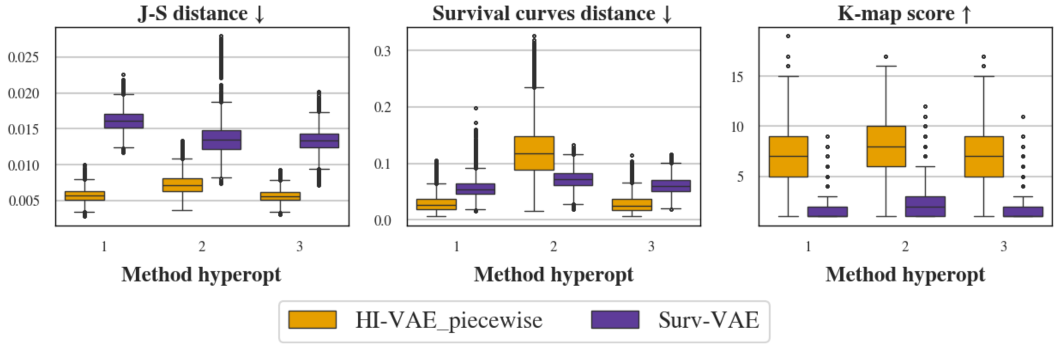

Figure 19: Comparison of generative performance metrics (J–S distance, survival distance, and K-map) across hyperparameter search methods, under the independent simulation setting (trained on controls only, replacement case).

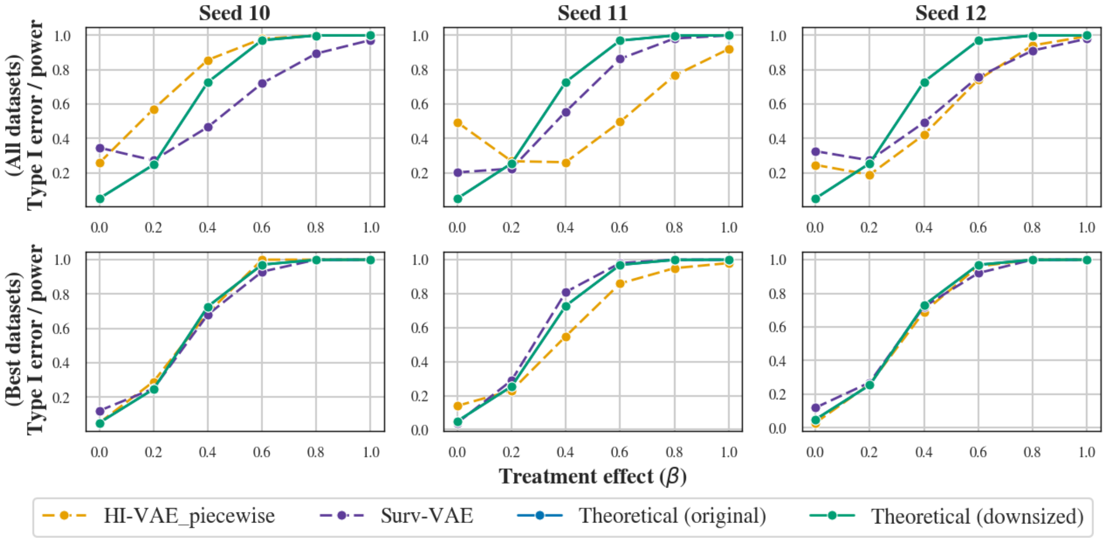

Figure 20: Type I error and power estimation (independent simulation setting, trained on controls only, replacement case, with **different random seed values in the hyperparameter search framework** and post-generation selection). Top: all generated datasets; bottom: best generated dataset. Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

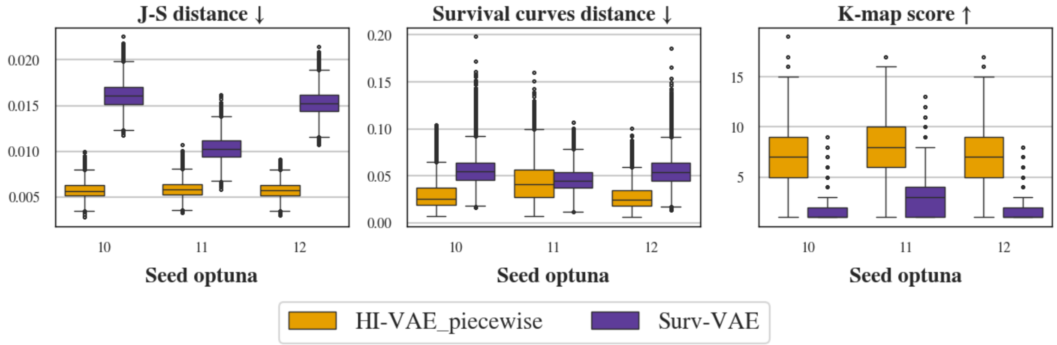

Figure 21: Comparison of generative performance metrics (J–S distance, survival distance, and K-map) across random seed values in hyperparameter search framework, in the independent simulation setting (trained on controls only, replacement case).

B.5. Influence of the training setup: control vs. control + treated

Here we provide additional results comparing models trained only on controls versus models trained on both control and treated data, as described in Section 4.4 of the main paper.

Table 4: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across training strategies (controls only vs. controls + treated) in the independent simulation setting.

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	**0.010±0.001**	0.011±0.002	**0.039±0.018**	0.072±0.032	6.127±4.345	**6.254±4.479**
	HI-VAE_piecewise	**0.010±0.001**	0.011±0.002	**0.042±0.021**	0.067±0.032	5.797±4.14	**6.208±4.466**
	Surv-GAN	0.041±0.01	**0.021±0.006**	0.116±0.068	**0.071±0.025**	3.091±7.692	**3.541±3.338**
	Surv-VAE	0.017±0.003	**0.016±0.002**	0.082±0.023	**0.065±0.025**	3.440±3.116	**4.864±3.809**
2/3	HI-VAE_weibull	**0.007±0.001**	0.008±0.001	**0.040±0.02**	0.040±0.019	3.606±1.845	**4.038±1.982**
	HI-VAE_piecewise	**0.007±0.001**	0.008±0.001	**0.033±0.017**	0.042±0.021	3.702±1.859	**3.960±1.97**
	Surv-GAN	**0.017±0.003**	0.022±0.008	**0.070±0.018**	0.072±0.029	**1.544±0.947**	1.373±0.762
	Surv-VAE	**0.012±0.004**	0.015±0.001	0.064±0.017	**0.059±0.018**	**1.969±1.20**	1.503±0.819
3/3	HI-VAE_weibull	**0.006±0.001**	0.006±0.001	**0.032±0.015**	0.033±0.015	7.105±3.069	**7.280±3.153**
	HI-VAE_piecewise	**0.006±0.001**	0.006±0.001	**0.029±0.014**	0.035±0.017	6.826±3.025	**7.529±3.16**
	Surv-GAN	**0.013±0.002**	0.032±0.006	**0.058±0.019**	0.091±0.044	**2.449±1.646**	1.237±0.615
	Surv-VAE	0.016±0.001	**0.014±0.001**	0.056±0.015	**0.049±0.013**	1.517±0.888	**2.103±1.383**

Table 5: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across training strategies (controls only vs. controls + treated) in the independent simulation setting.

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	**0.930±0.008**	0.923±0.01	**0.496±0.056**	0.547±0.057	0.424±0.078	**0.430±0.073**
	HI-VAE_piecewise	**0.930±0.008**	0.924±0.01	**0.512±0.057**	0.541±0.056	0.421±0.077	**0.426±0.075**
	Surv-GAN	0.720±0.074	**0.843±0.04**	0.950±0.046	**0.840±0.078**	0.385±0.071	**0.402±0.078**
	Surv-VAE	0.878±0.016	**0.890±0.012**	**0.711±0.054**	0.753±0.045	**0.386±0.076**	0.380±0.084
2/3	HI-VAE_weibull	**0.949±0.006**	0.946±0.007	0.559±0.04	**0.554±0.041**	0.421±0.065	**0.432±0.068**
	HI-VAE_piecewise	**0.949±0.006**	0.946±0.007	**0.553±0.041**	0.557±0.04	0.427±0.065	**0.429±0.068**
	Surv-GAN	**0.876±0.021**	0.831±0.052	**0.861±0.044**	0.921±0.047	0.360±0.07	**0.387±0.068**
	Surv-VAE	**0.912±0.031**	0.901±0.008	**0.700±0.056**	0.771±0.032	**0.387±0.07**	0.358±0.064
3/3	HI-VAE_weibull	**0.958±0.005**	0.955±0.006	**0.572±0.033**	0.572±0.033	**0.421±0.059**	0.419±0.063
	HI-VAE_piecewise	**0.958±0.005**	0.955±0.006	0.578±0.033	**0.573±0.032**	0.414±0.062	**0.422±0.062**
	Surv-GAN	**0.895±0.019**	0.725±0.043	**0.835±0.053**	0.993±0.009	0.368±0.068	**0.414±0.067**
	Surv-VAE	0.889±0.009	**0.905±0.008**	0.877±0.02	**0.846±0.023**	0.336±0.056	**0.341±0.058**

Table 6: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across training strategies (controls only vs. controls + treated) for the ACTG320 dataset.

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	**0.007±0.001**	0.007±0.001	**0.019±0.008**	0.023±0.01	1.435±0.799	**1.570±0.905**
	HI-VAE_piecewise	**0.006±0.001**	0.007±0.001	**0.020±0.01**	0.032±0.013	**1.415±0.689**	1.375±0.661
	Surv-GAN	0.021±0.002	**0.013±0.001**	**0.025±0.012**	0.059±0.012	**1.325±0.584**	1.175±0.535
	Surv-VAE	0.013±0.001	**0.012±0.001**	**0.040±0.014**	0.050±0.014	1.200±0.491	**1.715±0.779**
2/3	HI-VAE_weibull	0.006±0.001	**0.006±0.001**	0.018±0.01	**0.015±0.006**	**4.740±1.737**	4.620±1.726
	HI-VAE_piecewise	**0.005±0.001**	0.006±0.001	**0.011±0.006**	0.012±0.006	**5.050±1.875**	3.385±1.438
	Surv-GAN	0.011±0.001	**0.009±0.001**	0.012±0.005	**0.009±0.001**	1.915±1.016	**2.235±1.432**
	Surv-VAE	**0.009±0.001**	0.010±0.001	**0.012±0.006**	0.015±0.008	1.700±0.946	**3.315±1.472**
3/3	HI-VAE_weibull	**0.005±0.001**	0.006±0.001	**0.014±0.007**	0.014±0.004	2.560±1.37	**2.870±1.642**
	HI-VAE_piecewise	**0.005±0.001**	0.005±0.001	**0.010±0.005**	0.014±0.007	**2.915±1.552**	2.475±1.507
	Surv-GAN	**0.010±0.001**	0.015±0.001	**0.010±0.002**	0.012±0.002	**4.790±2.046**	2.030±1.219
	Surv-VAE	**0.008±0.001**	0.009±0.001	**0.013±0.005**	0.027±0.007	1.895±0.91	**1.910±1.261**

Table 7: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across training strategies (controls only vs. controls + treated) for the ACTG320 dataset.

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	**0.946±0.005**	0.944±0.005	**0.677±0.03**	0.685±0.032	**0.149±0.049**	0.136±0.046
	HI-VAE_piecewise	**0.947±0.005**	0.940±0.006	**0.656±0.031**	0.669±0.032	**0.142±0.043**	0.139±0.047
	Surv-GAN	0.836±0.015	**0.902±0.008**	0.870±0.017	**0.813±0.021**	0.097±0.013	**0.110±0.021**
	Surv-VAE	0.904±0.007	**0.908±0.008**	0.780±0.024	**0.756±0.023**	**0.115±0.026**	0.094±0.007
2/3	HI-VAE_weibull	0.950±0.004	**0.952±0.004**	**0.813±0.015**	0.814±0.016	**0.147±0.043**	0.145±0.039
	HI-VAE_piecewise	0.951±0.004	**0.954±0.004**	**0.801±0.016**	0.813±0.018	**0.137±0.036**	0.124±0.037
	Surv-GAN	0.905±0.006	**0.916±0.005**	**0.865±0.012**	0.880±0.011	**0.119±0.01**	0.098±0.02
	Surv-VAE	**0.915±0.005**	0.913±0.006	0.907±0.011	**0.890±0.011**	**0.113±0.015**	0.105±0.009
3/3	HI-VAE_weibull	**0.958±0.003**	0.956±0.004	**0.892±0.01**	0.894±0.009	0.115±0.028	**0.118±0.026**
	HI-VAE_piecewise	0.956±0.003	**0.957±0.003**	**0.897±0.011**	0.898±0.011	**0.117±0.027**	0.110±0.027
	Surv-GAN	**0.917±0.005**	0.896±0.004	**0.923±0.008**	0.946±0.005	**0.134±0.013**	0.095±0.017
	Surv-VAE	**0.925±0.004**	0.920±0.004	0.946±0.006	**0.938±0.007**	**0.098±0.013**	0.076±0.004

Table 8: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across training strategies (controls only vs. controls + treated) for the NCT00119613 dataset.

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	0.011±0.002	**0.010±0.002**	0.051±0.016	**0.043±0.013**	**2.795±1.963**	2.735±2.209
	HI-VAE_piecewise	**0.008±0.001**	0.011±0.002	**0.032±0.01**	0.041±0.015	1.970±1.147	**2.780±2.028**
	Surv-GAN	**0.013±0.003**	0.024±0.002	0.076±0.014	**0.073±0.008**	2.890±2.789	**3.405±3.662**
	Surv-VAE	**0.012±0.002**	0.050±0.004	**0.049±0.008**	0.056±0.021	**5.565±4.023**	2.630±2.289
2/3	HI-VAE_weibull	**0.007±0.001**	0.010±0.002	**0.036±0.008**	0.047±0.013	1.790±0.854	**1.965±0.974**
	HI-VAE_piecewise	**0.006±0.001**	0.006±0.001	0.025±0.008	**0.023±0.007**	**1.845±0.88**	1.625±0.865
	Surv-GAN	0.026±0.001	**0.016±0.002**	**0.048±0.006**	0.070±0.008	**1.285±0.553**	1.255±0.549
	Surv-VAE	**0.028±0.002**	0.028±0.003	**0.039±0.01**	0.042±0.01	1.260±0.612	**1.265±0.571**
3/3	HI-VAE_weibull	0.008±0.001	**0.006±0.001**	0.043±0.011	**0.029±0.006**	**4.470±1.569**	3.315±1.542
	HI-VAE_piecewise	**0.005±0.001**	0.006±0.001	**0.018±0.006**	0.020±0.006	**3.070±1.455**	2.945±1.579
	Surv-GAN	**0.030±0.001**	0.038±0.001	**0.038±0.007**	0.093±0.005	**1.455±0.794**	1.060±0.356
	Surv-VAE	**0.029±0.002**	0.052±0.002	**0.044±0.009**	0.047±0.009	**1.465±0.795**	1.355±0.641

Table 9: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across training strategies (controls only vs. controls + treated) for the NCT00119613 dataset.

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	0.937±0.008	**0.938±0.009**	0.561±0.051	**0.556±0.054**	**0.207±0.066**	0.193±0.066
	HI-VAE_piecewise	**0.949±0.007**	0.937±0.009	**0.502±0.059**	0.542±0.062	0.181±0.053	**0.189±0.067**
	Surv-GAN	**0.916±0.013**	0.864±0.009	**0.758±0.039**	0.838±0.036	0.082±0.006	**0.088±0.007**
	Surv-VAE	**0.931±0.009**	0.757±0.018	**0.669±0.051**	0.884±0.03	0.075±0.009	**0.079±0.007**
2/3	HI-VAE_weibull	**0.953±0.006**	0.944±0.007	**0.633±0.031**	0.635±0.036	0.213±0.055	**0.229±0.045**
	HI-VAE_piecewise	**0.965±0.005**	0.962±0.005	**0.589±0.038**	0.599±0.036	**0.220±0.057**	0.190±0.057
	Surv-GAN	0.824±0.008	**0.897±0.01**	0.946±0.012	**0.778±0.026**	**0.155±0.033**	0.077±0.021
	Surv-VAE	0.858±0.011	**0.861±0.012**	**0.779±0.027**	0.836±0.023	0.080±0.022	**0.080±0.027**
3/3	HI-VAE_weibull	0.953±0.005	**0.962±0.005**	0.710±0.026	**0.694±0.025**	**0.213±0.047**	0.177±0.043
	HI-VAE_piecewise	**0.970±0.004**	0.964±0.005	0.689±0.026	**0.688±0.028**	**0.176±0.045**	0.172±0.043
	Surv-GAN	**0.844±0.005**	0.765±0.005	**0.924±0.011**	0.967±0.005	0.071±0.019	**0.082±0.01**
	Surv-VAE	**0.855±0.01**	0.746±0.011	**0.848±0.018**	0.937±0.011	**0.069±0.02**	0.062±0.009

Table 10: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across training strategies (controls only vs. controls + treated) for the NCT00113763 dataset.

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	0.009±0.001	**0.008±0.001**	**0.025±0.008**	0.026±0.008	**2.835±1.251**	2.530±1.173
	HI-VAE_piecewise	**0.007±0.001**	0.008±0.001	**0.039±0.017**	0.043±0.017	3.385±1.279	**4.370±1.346**
	Surv-GAN	0.037±0.001	**0.016±0.002**	0.116±0.011	**0.059±0.009**	1.010±0.10	**1.275±0.584**
	Surv-VAE	0.033±0.002	**0.011±0.001**	**0.049±0.01**	0.062±0.01	**1.435±0.691**	1.150±0.422
2/3	HI-VAE_weibull	0.006±0.001	**0.006±0.001**	0.029±0.01	**0.025±0.008**	**9.275±2.045**	8.545±2.136
	HI-VAE_piecewise	**0.005±0.001**	0.007±0.001	**0.025±0.01**	0.028±0.011	**10.245±1.973**	10.175±1.855
	Surv-GAN	**0.014±0.001**	0.041±0.001	**0.056±0.006**	0.107±0.005	**1.780±1.033**	1.120±0.355
	Surv-VAE	0.038±0.002	**0.030±0.002**	**0.042±0.007**	0.046±0.007	**1.840±0.969**	1.630±0.999
3/3	HI-VAE_weibull	**0.005±0.001**	0.005±0.001	**0.025±0.009**	0.029±0.01	**13.935±2.55**	13.875±2.773
	HI-VAE_piecewise	**0.004±0.001**	0.005±0.001	0.026±0.009	**0.022±0.008**	**14.425±2.847**	14.27±2.674
	Surv-GAN	**0.025±0.001**	0.037±0.001	0.088±0.007	**0.055±0.005**	1.435±0.699	**2.770±1.70**
	Surv-VAE	0.041±0.001	**0.035±0.001**	0.044±0.007	**0.041±0.007**	1.435±0.615	**2.105±1.188**

Table 11: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across training strategies (controls only vs. controls + treated) for the NCT00113763 dataset.

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	0.949±0.006	**0.957±0.005**	**0.586±0.036**	0.590±0.037	**0.208±0.046**	0.179±0.048
	HI-VAE_piecewise	**0.956±0.006**	0.951±0.006	**0.604±0.035**	0.607±0.035	0.213±0.057	**0.251±0.054**
	Surv-GAN	0.770±0.007	**0.916±0.008**	0.959±0.009	**0.716±0.032**	**0.215±0.037**	0.127±0.022
	Surv-VAE	0.844±0.009	**0.938±0.005**	0.849±0.024	**0.842±0.022**	0.117±0.022	**0.145±0.022**
2/3	HI-VAE_weibull	0.966±0.004	**0.967±0.004**	0.664±0.024	**0.649±0.025**	**0.210±0.043**	0.198±0.042
	HI-VAE_piecewise	**0.967±0.004**	0.962±0.004	0.667±0.026	**0.660±0.023**	0.208±0.047	**0.216±0.052**
	Surv-GAN	**0.922±0.005**	0.792±0.004	**0.875±0.012**	0.971±0.005	**0.161±0.034**	0.140±0.022
	Surv-VAE	0.814±0.009	**0.855±0.008**	0.882±0.014	**0.876±0.013**	**0.132±0.022**	0.124±0.02
3/3	HI-VAE_weibull	**0.971±0.003**	0.971±0.003	0.719±0.022	**0.703±0.023**	**0.216±0.035**	0.211±0.045
	HI-VAE_piecewise	**0.973±0.003**	0.973±0.003	**0.695±0.025**	0.700±0.02	**0.232±0.044**	0.211±0.043
	Surv-GAN	**0.856±0.004**	0.828±0.004	0.947±0.006	**0.944±0.006**	0.111±0.021	**0.118±0.016**
	Surv-VAE	0.790±0.006	**0.833±0.007**	0.942±0.006	**0.890±0.01**	**0.123±0.023**	0.116±0.014

Table 12: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across training strategies (controls only vs. controls + treated) for the NCT00339183 dataset.

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	**0.007±0.001**	0.009±0.002	0.056±0.023	**0.047±0.018**	**7.155±2.122**	5.455±1.989
	HI-VAE_piecewise	**0.007±0.001**	0.008±0.002	0.049±0.023	**0.044±0.019**	**7.475±2.062**	5.155±1.838
	Surv-GAN	**0.010±0.001**	0.015±0.002	**0.065±0.021**	0.138±0.017	2.515±1.91	**3.025±2.939**
	Surv-VAE	**0.010±0.002**	0.014±0.002	0.065±0.022	**0.054±0.018**	2.380±1.655	**2.710±2.128**
2/3	HI-VAE_weibull	**0.005±0.001**	0.007±0.001	**0.038±0.013**	0.049±0.018	2.165±1.016	**2.185±1.042**
	HI-VAE_piecewise	**0.004±0.001**	0.006±0.001	**0.025±0.01**	0.027±0.012	**2.045±0.846**	1.825±0.859
	Surv-GAN	0.022±0.001	**0.013±0.001**	0.121±0.01	**0.082±0.012**	**1.570±0.938**	1.075±0.282
	Surv-VAE	**0.009±0.001**	0.021±0.002	0.069±0.014	**0.049±0.015**	**1.390±0.686**	1.120±0.383
3/3	HI-VAE_weibull	**0.005±0.001**	0.005±0.001	**0.034±0.01**	0.039±0.011	**2.490±1.173**	2.400±1.032
	HI-VAE_piecewise	**0.004±0.001**	0.004±0.001	**0.022±0.009**	0.023±0.01	**2.720±1.13**	2.645±1.089
	Surv-GAN	**0.012±0.001**	0.025±0.001	**0.062±0.01**	0.067±0.008	1.155±0.415	**1.310±0.817**
	Surv-VAE	0.011±0.001	**0.010±0.001**	0.041±0.012	**0.040±0.013**	**1.185±0.502**	1.125±0.425

Table 13: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across training strategies (controls only vs. controls + treated) for the NCT00339183 dataset.

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Control	Control + Treated	Control	Control + Treated	Control	Control + Treated
1/3	HI-VAE_weibull	**0.957±0.007**	0.948±0.008	**0.526±0.046**	0.541±0.053	**0.171±0.057**	0.132±0.032
	HI-VAE_piecewise	**0.955±0.007**	0.950±0.008	**0.519±0.05**	0.534±0.052	**0.189±0.055**	0.156±0.039
	Surv-GAN	**0.935±0.008**	0.887±0.01	**0.679±0.048**	0.766±0.037	**0.113±0.021**	0.109±0.009
	Surv-VAE	**0.937±0.009**	0.918±0.009	**0.620±0.049**	0.723±0.043	**0.110±0.02**	0.108±0.023
2/3	HI-VAE_weibull	**0.967±0.005**	0.958±0.006	**0.676±0.037**	0.676±0.033	**0.124±0.047**	0.116±0.037
	HI-VAE_piecewise	**0.969±0.005**	0.963±0.005	0.658±0.034	**0.654±0.039**	0.127±0.037	**0.130±0.037**
	Surv-GAN	0.855±0.008	**0.927±0.006**	0.855±0.022	**0.811±0.024**	**0.069±0.012**	0.068±0.011
	Surv-VAE	**0.945±0.006**	0.895±0.008	**0.809±0.028**	0.810±0.026	0.065±0.014	**0.081±0.023**
3/3	HI-VAE_weibull	**0.969±0.004**	0.967±0.005	**0.803±0.026**	0.805±0.023	**0.125±0.037**	0.123±0.036
	HI-VAE_piecewise	**0.973±0.004**	0.972±0.005	0.805±0.022	**0.797±0.022**	0.135±0.039	**0.137±0.033**
	Surv-GAN	**0.910±0.005**	0.846±0.006	0.903±0.016	**0.892±0.013**	**0.069±0.022**	0.057±0.01
	Surv-VAE	0.936±0.007	**0.940±0.006**	**0.852±0.018**	0.872±0.018	0.060±0.018	**0.063±0.019**

B.6. Differences with prior vs. posterior samplings

The following tables illustrate the comparison between prior- and posterior-based sampling for generating synthetic data, introduced in Section 4.4 of the main paper.

Table 14: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across sampling strategies (posterior vs. prior) in the independent simulation setting (trained on controls only).

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.008±0.002**	0.008±0.002	**0.035±0.018**	0.036±0.021	**5.442±3.407**	4.974±3.174
	HI-VAE_weibull	**0.008±0.002**	0.008±0.003	**0.037±0.018**	0.058±0.042	**5.612±3.569**	5.027±3.619
2/3	HI-VAE_piecewise	**0.008±0.002**	0.008±0.002	**0.035±0.018**	0.036±0.021	**5.442±3.407**	4.974±3.174
	HI-VAE_weibull	**0.008±0.002**	0.008±0.003	**0.037±0.018**	0.058±0.042	**5.612±3.569**	5.027±3.619
3/3	HI-VAE_piecewise	**0.008±0.002**	0.008±0.002	**0.035±0.018**	0.036±0.021	**5.442±3.407**	4.974±3.174
	HI-VAE_weibull	**0.008±0.002**	0.008±0.003	**0.037±0.018**	0.058±0.042	**5.612±3.569**	5.027±3.619

Table 15: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across sampling strategies (posterior vs. prior) in the independent simulation setting (trained on controls only)

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.946±0.014**	0.944±0.014	**0.548±0.052**	0.560±0.051	**0.421±0.068**	0.413±0.067
	HI-VAE_weibull	**0.946±0.013**	0.941±0.017	**0.542±0.055**	0.584±0.048	**0.422±0.068**	0.410±0.065
2/3	HI-VAE_piecewise	**0.946±0.014**	0.944±0.014	**0.548±0.052**	0.560±0.051	**0.421±0.068**	0.413±0.067
	HI-VAE_weibull	**0.946±0.013**	0.941±0.017	**0.542±0.055**	0.584±0.048	**0.422±0.068**	0.410±0.065
3/3	HI-VAE_piecewise	**0.946±0.014**	0.944±0.014	**0.548±0.052**	0.560±0.051	**0.421±0.068**	0.413±0.067
	HI-VAE_weibull	**0.946±0.013**	0.941±0.017	**0.542±0.055**	0.584±0.048	**0.422±0.068**	0.410±0.065

Table 16: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across sampling strategies (posterior vs. prior) for the ACTG320 dataset (trained on controls only).

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	0.006±0.001	**0.006±0.001**	**0.020±0.01**	0.020±0.008	1.415±0.689	**1.445±0.755**
	HI-VAE_weibull	**0.007±0.001**	0.007±0.001	0.019±0.008	**0.018±0.008**	**1.435±0.799**	1.365±0.731
2/3	HI-VAE_piecewise	**0.005±0.001**	0.007±0.001	**0.011±0.006**	0.014±0.009	5.050±1.875	**5.115±1.666**
	HI-VAE_weibull	**0.006±0.001**	0.007±0.001	0.018±0.01	**0.014±0.006**	**4.740±1.737**	4.045±1.714
3/3	HI-VAE_piecewise	**0.005±0.001**	0.005±0.001	0.010±0.005	**0.009±0.006**	**2.915±1.552**	2.675±1.49
	HI-VAE_weibull	**0.005±0.001**	0.006±0.00	0.014±0.007	**0.012±0.005**	**2.560±1.37**	2.220±1.364

Table 17: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across sampling strategies (posterior vs. prior) for the ACTG320 dataset (trained on controls only).

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.947±0.005**	0.947±0.005	**0.656±0.031**	0.677±0.032	**0.142±0.043**	0.128±0.036
	HI-VAE_weibull	**0.946±0.005**	0.940±0.006	**0.677±0.03**	0.707±0.03	**0.149±0.049**	0.144±0.038
2/3	HI-VAE_piecewise	**0.951±0.004**	0.944±0.004	**0.801±0.016**	0.812±0.016	0.137±0.036	**0.144±0.042**
	HI-VAE_weibull	**0.950±0.004**	0.948±0.004	**0.813±0.015**	0.839±0.015	**0.147±0.043**	0.135±0.043
3/3	HI-VAE_piecewise	**0.956±0.003**	0.954±0.004	0.897±0.011	**0.893±0.011**	**0.117±0.027**	0.109±0.023
	HI-VAE_weibull	**0.958±0.003**	0.956±0.003	**0.892±0.01**	0.905±0.009	**0.115±0.028**	0.101±0.025

Table 18: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across sampling strategies (posterior vs. prior) for the NCT00119613 dataset (trained on controls only).

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.008±0.001**	0.011±0.002	**0.032±0.01**	0.056±0.024	1.970±1.147	**3.185±2.686**
	HI-VAE_weibull	0.011±0.002	**0.009±0.001**	0.051±0.016	**0.040±0.011**	**2.795±1.963**	2.790±2.133
2/3	HI-VAE_piecewise	**0.006±0.001**	0.006±0.001	**0.025±0.008**	0.030±0.011	1.845±0.88	**1.975±0.932**
	HI-VAE_weibull	**0.007±0.001**	0.007±0.001	**0.036±0.008**	0.038±0.008	1.790±0.854	**2.090±0.936**
3/3	HI-VAE_piecewise	**0.005±0.001**	0.005±0.001	**0.018±0.006**	0.022±0.008	3.070±1.455	**3.490±1.701**
	HI-VAE_weibull	0.008±0.001	**0.007±0.001**	**0.043±0.011**	0.044±0.01	4.470±1.569	**4.865±1.526**

Table 19: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across sampling strategies (posterior vs. prior) for the NCT00119613 dataset (trained on controls only).

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.949±0.007**	0.936±0.008	**0.502±0.059**	0.559±0.052	**0.181±0.053**	0.158±0.063
	HI-VAE_weibull	0.937±0.008	**0.941±0.008**	0.561±0.051	**0.542±0.063**	**0.207±0.066**	0.152±0.062
2/3	HI-VAE_piecewise	**0.965±0.005**	0.961±0.006	**0.589±0.038**	0.612±0.038	**0.220±0.057**	0.218±0.056
	HI-VAE_weibull	**0.953±0.006**	0.953±0.006	**0.633±0.031**	0.637±0.035	0.213±0.055	**0.232±0.048**
3/3	HI-VAE_piecewise	**0.970±0.004**	0.966±0.005	**0.689±0.026**	0.691±0.027	**0.176±0.045**	0.175±0.044
	HI-VAE_weibull	0.953±0.005	**0.958±0.004**	**0.710±0.026**	0.714±0.025	0.213±0.047	**0.214±0.043**

Table 20: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across sampling strategies (posterior vs. prior) for the NCT00113763 dataset (trained on controls only).

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.007±0.001**	0.008±0.001	0.039±0.017	**0.027±0.013**	**3.385±1.279**	2.910±1.157
	HI-VAE_weibull	0.009±0.001	**0.008±0.001**	**0.025±0.008**	0.026±0.009	2.835±1.251	**3.125±1.125**
2/3	HI-VAE_piecewise	**0.005±0.001**	0.007±0.001	**0.025±0.01**	0.048±0.014	**10.245±1.973**	9.455±1.93
	HI-VAE_weibull	**0.006±0.001**	0.006±0.001	0.029±0.01	**0.023±0.008**	**9.275±2.045**	8.770±1.959
3/3	HI-VAE_piecewise	**0.004±0.001**	0.005±0.001	0.026±0.009	**0.017±0.007**	**14.425±2.847**	13.44±2.465
	HI-VAE_weibull	0.005±0.001	**0.005±0.001**	0.025±0.009	**0.020±0.007**	**13.935±2.55**	13.46±2.536

Table 21: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across sampling strategies (posterior vs. prior) for the NCT00113763 dataset (trained on controls only).

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	**0.956±0.006**	0.953±0.006	0.604±0.035	**0.597±0.038**	**0.213±0.057**	0.210±0.063
	HI-VAE_weibull	0.949±0.006	**0.953±0.005**	**0.586±0.036**	0.599±0.036	0.208±0.046	**0.211±0.047**
2/3	HI-VAE_piecewise	**0.967±0.004**	0.958±0.004	**0.667±0.026**	0.677±0.023	**0.208±0.047**	0.194±0.048
	HI-VAE_weibull	**0.966±0.004**	0.964±0.004	**0.664±0.024**	0.673±0.025	**0.210±0.043**	0.204±0.037
3/3	HI-VAE_piecewise	**0.973±0.003**	0.972±0.003	**0.695±0.025**	0.702±0.022	**0.232±0.044**	0.185±0.046
	HI-VAE_weibull	**0.971±0.003**	0.971±0.003	0.719±0.022	**0.713±0.021**	**0.216±0.035**	0.207±0.034

Table 22: Comparison of generative performance metrics (J-S distance, survival distance, K-map) across sampling strategies (posterior vs. prior) for the NCT00339183 dataset (trained on controls only).

v	**Algorithm**	J-S dist ↓		Surv dist ↓		K-map score ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	0.007±0.001	**0.007±0.001**	0.049±0.023	**0.039±0.016**	**7.475±2.062**	6.595±1.926
	HI-VAE_weibull	0.007±0.001	**0.007±0.001**	0.056±0.023	**0.042±0.014**	**7.155±2.122**	5.750±1.875
2/3	HI-VAE_piecewise	0.004±0.001	**0.004±0.001**	0.025±0.01	**0.023±0.009**	2.045±0.846	**2.135±0.97**
	HI-VAE_weibull	**0.005±0.001**	0.005±0.001	**0.038±0.013**	0.044±0.013	2.165±1.016	**2.220±0.993**
3/3	HI-VAE_piecewise	0.004±0.001	**0.004±0.001**	0.022±0.009	**0.022±0.009**	2.720±1.13	**3.000±1.211**
	HI-VAE_weibull	0.005±0.001	**0.004±0.001**	**0.034±0.01**	0.038±0.011	2.490±1.173	**2.600±1.199**

Table 23: Comparison of generative performance metrics (KS test, Detect XGB, NNDR) across sampling strategies (posterior vs. prior) for the NCT00339183 dataset (trained on controls only).

v	**Algorithm**	KS test ↑		Detection XGB ↓		NNDR ↑	
		Posterior	Prior	Posterior	Prior	Posterior	Prior
1/3	HI-VAE_piecewise	0.955±0.007	**0.958±0.007**	**0.519±0.05**	0.522±0.05	**0.189±0.055**	0.173±0.053
	HI-VAE_weibull	0.957±0.007	**0.958±0.007**	0.526±0.046	**0.517±0.05**	**0.171±0.057**	0.161±0.052
2/3	HI-VAE_piecewise	0.969±0.005	**0.971±0.004**	**0.658±0.034**	0.667±0.034	**0.127±0.037**	0.122±0.035
	HI-VAE_weibull	**0.967±0.005**	0.966±0.005	**0.676±0.037**	0.677±0.037	0.124±0.047	**0.131±0.039**
3/3	HI-VAE_piecewise	**0.973±0.004**	0.973±0.004	0.805±0.022	**0.803±0.023**	0.135±0.039	**0.147±0.037**
	HI-VAE_weibull	0.969±0.004	**0.973±0.004**	**0.803±0.026**	0.809±0.023	0.125±0.037	**0.130±0.04**

B.7. Preliminary exploration of differential privacy

We present here our preliminary results exploring differential privacy in our models, using the `Opacus` framework (Yousefpour et al., 2021) (Section 4.4 of the main paper). Specifically, we applied the privacy engine to our HI-VAE model with the following parameters:

```
hivae_model, optimizer, data_loader = privacy_engine.make_private(
        module=hivae_model,
        optimizer=optimizer,
        data_loader=data_loader,
        noise_multiplier=2.0,
        max_grad_norm=1.0)
```

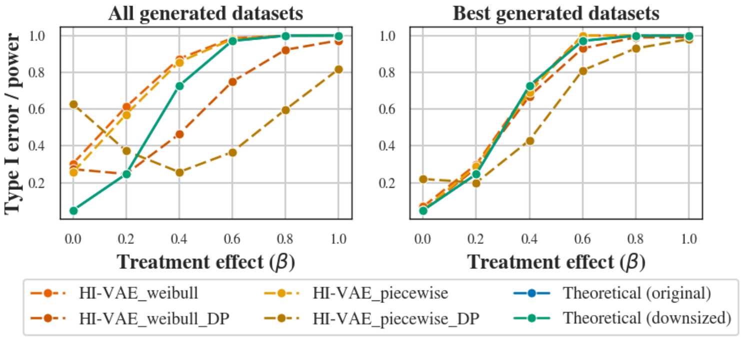

Figure 22: Type I error and power estimation in the independent simulation setting, (trained on controls only, replacement case), with or without **the differential privacy method**. Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.

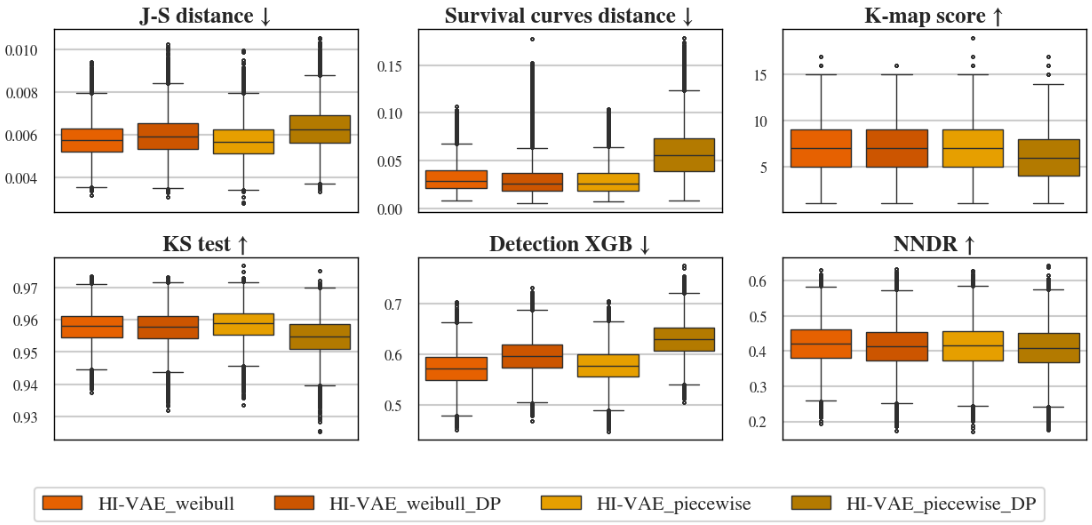

Figure 23: Comparison of generative performance metrics (J-S distance, survival distance, K-map) in the independent simulation setting, (trained on controls only, replacement case), with or without **the differential privacy method**. Dashed lines: empirical power. Green: theoretical power with reduced control size. Blue: theoretical power with generated control size.