JETO: JOINT ESTIMATION OF TREATMENT ASSIGNMENT AND OUTCOMES FOR HETEROGENEOUS TREATMENT EFFECT ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Heterogeneous treatment effect estimation seeks to characterize how the effect of a treatment varies across individuals with different covariates, despite only one potential outcome being observed for each unit. Classical methods primarily target the conditional average treatment effect, providing a point estimate of treatment benefit. Generative modeling of $p(Y \mid X, W)$ extends this view to treatment-specific outcome distributions, but leaves the treatment assignment mechanism outside the generative target and does not explicitly capture structure shared between propensity and outcomes. We propose JETO, which learns the joint conditional distribution $p(Y, W \mid X)$ with conditional Flow Matching, providing a treatment effect estimate, treatment-specific means, propensity scores, and conditional outcome densities from a single model. Across six synthetic and semi-synthetic benchmarks, JETO achieves the lowest mean relative PEHE on four datasets. Controlled experiments show that joint modeling exploits covariate-dependent structure shared by treatment assignment and outcomes, while density experiments demonstrate accurate multimodal outcome recovery and propensity estimation.

1 INTRODUCTION

Estimating how the effect of an intervention varies across individuals is a central problem in causal inference. Heterogeneous treatment effect (HTE) estimation typically focuses on the conditional average treatment effect (CATE), which describes how the expected treatment benefit changes with observed covariates. Most existing estimators target this quantity through treatment-specific response surfaces, direct treatment-effect regression, or nuisance-adjusted objectives. Although effective for point estimation, such approaches do not generally characterize the full treatment-specific outcome distributions.

Recent generative methods extend HTE estimation beyond conditional means. DiffPO uses conditional diffusion models to learn potential-outcome distributions (Ma et al., 2024), while PO-Flow adopts a flow-based generative formulation (Wu et al., 2026). Both model $p(Y \mid X, W)$, treating the observed treatment assignment as a conditioning variable. This provides access to treatment-specific outcome distributions, but leaves the treatment-assignment mechanism outside the generative target.

We propose JETO, a conditional generative model that instead learns the joint observational distribution

$$p(Y, W \mid X). \tag{1}$$

Treatment assignment is therefore generated jointly with the observed outcome rather than supplied only as an external condition. From the learned distribution, a single model provides treatment-specific conditional means, CATE estimates, treatment propensities, and conditional outcome densities. We instantiate JETO using conditional Flow Matching, jointly transporting a continuous treatment–outcome representation while conditioning on observed covariates.

Modeling $p(W \mid X)$ is not required for identification of CATE under the standard assumptions considered in this work. Instead, its role is representational: the treatment-assignment objective provides an additional learning signal about how treatment varies across the covariate space. This

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

signal may be useful when the assignment mechanism and outcome distribution contain related covariate-dependent structure, but it need not improve estimation in every data-generating setting. We examine this distinction through matched synthetic benchmarks and an architecture-controlled comparison between joint modeling of $p(Y, W \mid X)$ and conditional modeling of $p(Y \mid X, W)$.

Our main contributions are:

- We introduce JETO, a generative model for HTE analysis that jointly models treatment assignment and outcome through $p(Y, W \mid X)$, providing treatment effects, conditional outcome distributions, and propensity scores from a single fitted model.
- Across six synthetic and semi-synthetic benchmarks, JETO remains competitive with established HTE estimators and recent generative approaches, achieving the lowest mean relative PEHE on four datasets.
- Through an architecture-matched joint-versus-conditional comparison, we show that joint modeling improves CATE estimation when treatment assignment and outcome modality share covariate-dependent structure. Complementary gradient diagnostics indicate that the treatment and outcome objectives provide largely distinct, similarly scaled optimization signals without persistent strong interference.
- We demonstrate that JETO recovers information beyond point treatment effects. It captures multimodal treatment-specific outcome densities, achieves lower conditional-density KL divergence than PO-Flow in the considered density-recovery experiment, and provides consistent propensity estimates through both Monte Carlo sampling and direct density integration.

2 Related Work

CATE estimation and neural representations. A broad class of heterogeneous treatment effect estimators reduces CATE estimation to supervised-learning problems involving treatment-specific response surfaces, treatment effects, or transformed outcomes. The S- and T-learners estimate conditional mean outcomes, the X-learner combines imputed treatment effects, and the R- and doubly robust learners use residualized or pseudo-outcome objectives to reduce sensitivity to nuisance estimation (Künzel et al., 2019; Nie & Wager, 2021; Kennedy, 2023). Neural approaches additionally learn representations shared across treatment groups. TARNet uses a common representation followed by treatment-specific outcome heads, while CFR encourages balance between treated and control representations (Shalit et al., 2017). Dragonnet supplements the outcome heads with propensity prediction and targeted regularization (Shi et al., 2019). These methods demonstrate that treatment and outcome objectives can benefit from shared representations, but primarily target conditional means or CATE rather than the joint treatment–outcome distribution.

Generative models for treatment effects. Generative approaches provide access to richer outcome distributions. GANITE uses adversarial learning to impute potential outcomes (Yoon et al., 2018), whereas CEVAE introduces latent variables to model treatment assignment, outcomes, and unobserved confounding (Louizos et al., 2017). Outside HTE, SDPM jointly models event time and censoring status, $p(T, \delta \mid X)$, using diffusion with a continuous representation of the binary indicator, providing a related precedent for joint generative modeling of continuous and binary targets (Kirpichenko et al., 2026).

More recently, DiffPO applies conditional diffusion to treatment-specific potential-outcome distributions (Ma et al., 2024), and PO-Flow uses a flow-based generator for the same factorization (Wu et al., 2026). Both learn $p(Y \mid X, W)$, treating treatment assignment as a conditioning variable. JETO is most closely related to these methods but instead learns $p(Y, W \mid X)$, representing the assignment mechanism and treatment-specific outcome distributions within a single generative model.

Flow Matching. Flow Matching learns a time-dependent velocity field that transports a simple reference distribution toward the data distribution without simulating the generative process during training (Lipman et al., 2023). Conditional Flow Matching extends this construction to conditional probability paths, making it suitable for generative regression conditioned on observed covariates (Tong et al., 2024). JETO applies conditional Flow Matching to the joint treatment–outcome target.

In addition to sampling, the resulting continuous flow permits density evaluation through the instantaneous change-of-variables formula, which enables direct analysis of treatment-specific outcome densities and treatment propensity.

3 Heterogeneous Treatment Effect Estimation

Let $X \in \mathcal{X}$ denote observed pre-treatment covariates, $W \in \{0, 1\}$ a binary treatment assignment, and Y the observed outcome. Under the potential-outcomes framework (Rubin, 1974), each unit has potential outcomes $Y(0)$ and $Y(1)$, of which only $Y = Y(W)$ is observed. We define the treatment-specific conditional means as $\mu_w(x) = \mathbb{E}[Y(w) \mid X = x]$, $w \in \{0, 1\}$, and the conditional average treatment effect as $\tau(x) = \mu_1(x) - \mu_0(x)$.

Identification from observational data requires consistency, conditional exchangeability, and positivity (Rosenbaum & Rubin, 1983; Imbens & Rubin, 2015). Specifically, we assume $Y = Y(W)$, $\{Y(0), Y(1)\} \perp W \mid X$, and $0 < P(W = 1 \mid X = x) < 1$ over the population of interest. Under these assumptions, the response surfaces are identified by $\mu_w(x) = \mathbb{E}[Y \mid X = x, W = w]$.

The observed data also define the treatment propensity $e(x) = P(W = 1 \mid X = x)$. Both the treatment-specific outcome distributions and the assignment mechanism are contained in the conditional joint distribution

$$p(Y, W \mid X) = p(Y \mid X, W)\, p(W \mid X). \tag{2}$$

A model of this distribution therefore represents the quantities required for CATE estimation together with the treatment-assignment mechanism, motivating the joint generative formulation introduced next.

4 JETO

JETO is a conditional generative model that learns the joint observational distribution $p(Y, W \mid X)$. Unlike generative HTE methods that model $p(Y \mid X, W)$ with treatment supplied as a condition, JETO includes W in the generated target and represents treatment assignment and outcome within a single generative process.

Under the identification assumptions of Section 3, conditioning the learned distribution on $W = w$ provides the treatment-specific outcome distribution, while marginalization over Y provides the treatment propensity. Denoting the learned distribution by $\hat{p}$, JETO estimates

$$\hat{e}(x) = \int \hat{p}(y, 1 \mid x)\, dy, \tag{3}$$

$$\hat{\mu}_w(x) = \mathbb{E}_{\hat{p}}[Y \mid X = x, W = w], \qquad w \in \{0, 1\}, \tag{4}$$

$$\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x). \tag{5}$$

Thus, a single fitted model provides propensity scores, treatment-specific conditional means, CATE estimates, and conditional outcome densities. We implement this joint representation using conditional Flow Matching.

4.1 Conditional Flow Matching

JETO represents the transformed treatment–outcome target as $Z_1 = (\widetilde{Y}, \widetilde{W})^\top \in \mathbb{R}^2$ and uses the standard linear Flow Matching construction (Lipman et al., 2023; Tong et al., 2024). Given an independent source sample $Z_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $t \sim \mathcal{U}(0, 1)$, the interpolation path is $Z_t = (1 - t)Z_0 + tZ_1$, with target velocity $Z_1 - Z_0$. A conditional vector field $v_\theta(z, t, x)$ is trained by minimizing

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{X, Z_0, Z_1, t}\left[\left\| v_\theta(Z_t, t, X) - (Z_1 - Z_0) \right\|_2^2\right]. \tag{6}$$

At inference, source samples are transported from $t = 0$ to $t = 1$ by solving $dZ_t/dt = v_\theta(Z_t, t, x)$, producing joint samples from the learned treatment–outcome distribution conditional on $X = x$.

4.2 Implementation Details

Treatment representation and Monte Carlo estimation. In the main experiments, treatment assignment is encoded symmetrically as $\widetilde{W} = 2W - 1 \in \{-1, 1\}$. The generated treatment coordinate is decoded by its sign, such that $\hat{w}^{(k)} = \mathbb{1}[\widetilde{w}^{(k)} > 0]$. Given K joint samples $\{(y^{(k)}, \widetilde{w}^{(k)})\}_{k=1}^{K}$ conditional on $X = x$, JETO estimates

$$\hat{e}(x)_{\mathrm{MC}} = \frac{1}{K} \sum_{k=1}^{K} \hat{w}^{(k)}, \tag{7}$$

$$\hat{\mu}_w(x) = \frac{\sum_{k=1}^{K} \mathbb{1}[\hat{w}^{(k)} = w]\, y^{(k)}}{\sum_{k=1}^{K} \mathbb{1}[\hat{w}^{(k)} = w]}, \qquad w \in \{0, 1\}, \tag{8}$$

$$\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x). \tag{9}$$

Continuous relaxation and density evaluation. The deterministic treatment codes form a singular distribution along the treatment coordinate and therefore do not permit direct continuous-density evaluation. For density-based analyses, we instead use the Gaussian relaxation

$$\widetilde{W} \mid W = w \sim \mathcal{N}(2w - 1, \sigma_W^2), \qquad \sigma_W = 0.2. \tag{10}$$

The corresponding joint density $\hat{p}(y, \widetilde{w} \mid x)$ is evaluated through the continuous-time change-of-variables relation,

$$\log \hat{p}(z_1 \mid x) = \log p_0(z_0) - \int_0^1 \nabla_z \cdot v_\theta(z_t, t, x)\, dt, \tag{11}$$

where p_0 is the standard Gaussian source density.

Let $A_0 = (-\infty, 0]$ and $A_1 = (0, \infty)$ denote the regions associated with the two treatment classes. Marginalizing the relaxed coordinate gives

$$d_w(y \mid x) = \int_{A_w} \hat{p}(y, \widetilde{w} \mid x)\, d\widetilde{w}, \qquad M_w(x) = \int d_w(y \mid x)\, dy. \tag{12}$$

The treatment-specific conditional density and propensity estimate are then

$$\hat{p}(y \mid x, W = w) = \frac{d_w(y \mid x)}{M_w(x)}, \qquad \hat{e}_{\mathrm{int}}(x) = \frac{M_1(x)}{M_0(x) + M_1(x)}. \tag{13}$$

The normalization by $M_0(x) + M_1(x)$ also compensates for probability mass omitted when the integrals are evaluated over finite numerical bounds.

Architecture and numerical inference. The vector field is parameterized by a fully connected neural network using Fourier features for the covariates and an embedding of flow time. Outcomes are standardized using the training-set mean and standard deviation, $\widetilde{Y} = (Y - \overline{Y})/\mathrm{std}(Y)$, and generated values are transformed back to the original scale before estimation. The learned flow is integrated with the classical fourth-order Runge–Kutta method. Final inference uses ten integration steps and $K = 2048$ joint samples per covariate vector; validation and early stopping use three steps and $K = 512$ samples to reduce repeated computational cost. Additional architectural choices and hyperparameter search spaces are reported in Appendix A.

Figure 1 summarizes the JETO inference procedure, from numerical integration of the learned flow to the estimation of treatment-specific outcome quantities from generated joint samples.

5 Experimental Methodology

5.1 Evaluation Metrics

We evaluate CATE estimation primarily using relative Precision in Estimation of Heterogeneous Effect (rPEHE),

$$\mathrm{rPEHE} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\hat{\tau}(x_i) - \tau(x_i)\right)^2}}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} \tau(x_i)^2}}. \tag{14}$$

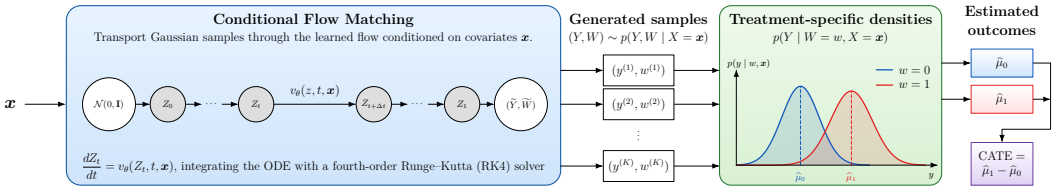


Figure 1: JETO inference. Given covariates x, the learned flow generates joint treatment–outcome samples. Grouping them by treatment yields treatment-specific outcome distributions, conditional means, and the CATE estimate.

Unlike absolute PEHE, rPEHE is invariant to a common multiplicative rescaling of the outcomes and therefore permits more meaningful aggregation across data-generating realizations with different treatment-effect scales.

As secondary diagnostics, we evaluate reconstruction of the treatment-specific conditional means using PO-RMSE and propensity estimation using $\mathrm{RMSE}_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[\hat{e}(x_i) - e(x_i)]^2}$. Complete definitions and PO-RMSE results are reported in Appendix C.

5.2 Baseline Methods

We compare JETO with five established meta-learners, a neural latent-variable model, and two generative potential-outcome methods. The meta-learning baselines comprise the S-, T-, and X-learners (Künzel et al., 2019), the R-learner (Nie & Wager, 2021), and the doubly robust learner (Kennedy, 2023). All were implemented using CausalML (Chen et al., 2020), with XGBoost used as the common regression model. We additionally include CEVAE (Louizos et al., 2017) as a neural latent-variable baseline.

The closest generative comparisons are DiffPO (Ma et al., 2024) and PO-Flow (Wu et al., 2026), which learn treatment-specific outcome distributions conditional on X and W. For both methods, we use the publicly released implementations provided by the authors.

Each method is tuned independently using the same Optuna trial budget. Comparable search ranges are used where parameters have analogous roles, while method-specific architectural and training choices from the released implementations are retained. Complete search spaces and implementation settings are reported in Appendix A.4.

5.3 Datasets

We evaluate all methods on six synthetic and semi-synthetic benchmarks. IHDP (Hill, 2011) and ACIC 2016 (Dorie et al., 2019) combine real covariates with simulated treatment assignments and outcomes; we use 100 realizations of each. We additionally consider Setups B and D of Nie and Wager (Nie & Wager, 2021), generated using CausalML (Chen et al., 2020) and denoted **NW-B** and **NW-D**. NW-B uses randomized treatment with $e(x) = 0.5$, whereas NW-D combines a nonconstant propensity score with distinct nonlinear response surfaces.

To evaluate distributional modeling, we introduce two matched Gaussian-mixture benchmarks, denoted **GM-I** and **GM-P**. Both use Gaussian covariates, nonlinear response surfaces constructed from random Fourier features, covariate-dependent treatment assignment, and bimodal potential-outcome distributions. They differ only in the probabilities of the outcome-mixture components:

$$\big(\pi_0(x),\ \pi_1(x)\big) = \begin{cases} \big(\frac{1}{2},\ \frac{1}{2}\big), & \text{GM-I}, \\ \big(\sigma(-\alpha),\ \sigma(\alpha + \rho\ \mathrm{logit}\, e(x))\big), & \text{GM-P}. \end{cases} \tag{15}$$

Thus, GM-I provides a control setting in which outcome modality is independent of treatment propensity, whereas GM-P links the treated-outcome mixture weight to the assignment mechanism. This matched design isolates whether joint modeling benefits from shared covariate-dependent structure between treatment assignment and outcome modality. The construction does not introduce unobserved confounding because treatment remains independent of potential outcomes conditional on

X. Both datasets use $n = 1000$ observations and $p = 10$ covariates. Complete data-generating processes and parameter settings are provided in Appendix B.

5.4 Evaluation Protocol

Each method was tuned independently for each dataset using 100 Optuna TPE trials on a dedicated realization excluded from final evaluation. The selected configuration was then fixed and evaluated on the remaining 99 IHDP and ACIC realizations and on 20 independently generated realizations of each synthetic dataset. Performance was measured exclusively on stratified held-out test sets. Flow- and diffusion-based models used the same total inference budget of $K = 2048$ samples per covariate vector. Complete data-splitting, cross-validation, early-stopping, and hyperparameter-search details are reported in Appendix A.

5.5 Additional Studies

Beyond the benchmark comparison, we examine when joint modeling is beneficial and whether JETO recovers distributional quantities beyond conditional means.

For conditional-density evaluation, we use GM-P with increased mixture separation $\Delta = 1.2$, producing clearly distinguishable modes while retaining the original response surfaces and propensity mechanism. JETO uses the continuous treatment relaxation and density-integration procedure described in Section 4.2. We compare it with PO-Flow, which uses the same generative process but models $p(Y \mid X, W)$ rather than $p(Y, W \mid X)$.

Qualitative recovery is evaluated at three representative covariate vectors with propensity scores close to 0.1, 0.5, and 0.9. Quantitative evaluation uses 100 independently sampled covariate vectors from each of three regions: $e(x) \in [0.05, 0.15]$, $e(x) \in [0.45, 0.55]$, and $e(x) \in [0.85, 0.95]$. Density accuracy is measured by $D_{\mathrm{KL}}(p_{\mathrm{true}} \| \hat{p})$, computed separately for each treatment arm. We additionally evaluate propensity recovery using both Monte Carlo sampling and direct integration of the learned joint density.

Further studies compare JETO with an architecture-matched conditional variant, analyze the cosine similarity and relative magnitudes of the outcome and treatment gradients, examine sensitivity to the Monte Carlo budget and ODE solver, and compare Flow Matching with a diffusion implementation of the same joint model. Complete protocols and results for these studies are provided in Appendix D.

6 Results

6.1 Benchmark Comparison

Table 1 compares CATE estimation accuracy across the six benchmarks. JETO achieves the lowest mean rPEHE on IHDP, GM-I, GM-P, and NW-D. The X-learner performs best on ACIC, while the R-learner obtains the lowest error on NW-B.

JETO's largest advantage is on IHDP: its mean rPEHE is 0.155, compared with 0.224 for the next-best method, the DR-learner. It also leads on both Gaussian-mixture benchmarks. On GM-I and GM-P, JETO obtains rPEHE values of 0.534 and 0.635, respectively, compared with 0.550 and 0.665 for the strongest competing methods. On these three datasets, all other methods differ significantly from JETO under the paired tests reported in Table 1.

The result on NW-D is closer: JETO obtains 0.387 and PO-Flow obtains 0.391, without a significant difference between them. On ACIC, the X-learner achieves the lowest rPEHE of 0.142, compared with 0.218 for JETO. NW-B is the least favorable setting for JETO: its mean rPEHE rises to 0.420 with substantial variability, while the R-learner achieves 0.331. DiffPO exhibits larger mean errors and variability than the other methods across the evaluated datasets.

Response-surface reconstruction provides a complementary perspective. The PO-RMSE results in Appendix C, Table 8, favor JETO on IHDP, GM-I, and GM-P, but favor PO-Flow on NW-B and NW-D. Thus, the benchmark results support competitive CATE estimation by JETO while also showing that its advantage depends on the data-generating setting.

Table 1: Comparison of rPEHE across datasets. Values are reported as means with standard deviations over experimental repetitions; the best mean for each dataset is shown in bold. An asterisk denotes a statistically significant difference from the best-mean method within the same dataset according to a two-sided paired Wilcoxon signed-rank test with Holm correction across methods ($p_{\text{adj}} < 0.05$).

Method	IHDP	ACIC	GM-I	GM-P	NW-B	NW-D
JETO	**0.155**$\pm$0.057	0.218$^*\pm$0.091	**0.534**$\pm$0.038	**0.635**$\pm$0.053	0.420$^*\pm$0.119	**0.387**$\pm$0.054
PO-Flow	0.255$^*\pm$0.135	0.231$^*\pm$0.079	0.576$^*\pm$0.073	0.682$^*\pm$0.061	0.374$\pm$0.058	0.391$\pm$0.028
DiffPO	0.753$^*\pm$0.300	0.557$^*\pm$0.272	2.513$^*\pm$3.100	3.361$^*\pm$2.563	1.491$^*\pm$1.275	1.744$^*\pm$1.139
CEVAE	0.330$^*\pm$0.159	0.572$^*\pm$0.179	0.602$^*\pm$0.056	0.744$^*\pm$0.058	0.357$\pm$0.064	0.592$^*\pm$0.073
T-Learner	0.232$^*\pm$0.135	0.184$^*\pm$0.098	0.572$^*\pm$0.034	0.668$^*\pm$0.036	0.388$^*\pm$0.040	0.492$^*\pm$0.044
S-Learner	0.311$^*\pm$0.190	0.242$^*\pm$0.082	0.588$^*\pm$0.029	0.787$^*\pm$0.024	0.465$^*\pm$0.043	0.719$^*\pm$0.065
X-Learner	0.300$^*\pm$0.157	**0.142**$\pm$0.076	0.550$^*\pm$0.046	0.665$^*\pm$0.039	0.335$\pm$0.045	0.549$^*\pm$0.043
R-Learner	0.334$^*\pm$0.182	0.189$^*\pm$0.087	0.603$^*\pm$0.038	0.679$^*\pm$0.039	**0.331**$\pm$0.051	0.575$^*\pm$0.054
DR-Learner	0.224$^*\pm$0.114	0.223$^*\pm$0.103	0.576$^*\pm$0.037	0.756$^*\pm$0.046	0.394$^*\pm$0.049	0.611$^*\pm$0.060

6.2 Joint versus Conditional Modeling

To isolate the effect of including treatment assignment in the generated target, we compare JETO with an architecture-matched variant that models $p(Y \mid X, W)$. Both variants use comparable capacity and the same total Monte Carlo budget. The experiment uses 20 paired realizations of each Gaussian-mixture benchmark, separate from those in the main comparison; further details are given in Appendix D.1.

Table 2: Joint versus conditional modeling on the Gaussian-mixture benchmarks. Paired differences are conditional minus joint rPEHE; positive values favor joint modeling.

Dataset	Joint JETO	Conditional JETO	Mean paired difference	Paired difference 95% CI
GM-I	0.5086	**0.4906**	-0.0179	[-0.0393, 0.0034]
GM-P	**0.5834**	0.6273	**0.0440**	[0.0180, 0.0699]

On GM-I, where outcome-mixture weights are independent of treatment propensity, the conditional variant obtains a slightly lower mean rPEHE, but the confidence interval for the paired difference includes zero. On GM-P, where the treated-outcome mixture weight depends on propensity, the conditional model obtains a mean rPEHE of 0.6273, approximately 7.5% higher than the 0.5834 achieved by joint JETO. The paired difference is positive, with a 95% confidence interval of $[0.0180, 0.0699]$.

The two formulations have similar errors when estimating $\mu_0(x)$ and $\mu_1(x)$ individually (Appendix D.1, Table 9). The GM-P improvement therefore does not reflect a uniform reduction in both response-surface errors; it arises when their estimates are combined to form CATE. Together, the GM-I and GM-P results indicate that the additional treatment-assignment objective can improve CATE estimation when assignment and outcome modality share covariate-dependent structure, without implying an advantage in every setting.

6.3 Conditional Density and Propensity Recovery

We next assess whether JETO recovers the shape of the treatment-specific outcome distributions and the treatment propensity. As described in Section 5.5, this experiment uses GM-P with increased mixture separation $\Delta = 1.2$. JETO is trained with the continuous treatment relaxation, and its conditional densities and propensity are computed from the learned joint density using the integration procedure in Section 4.2. For density recovery, we compare JETO with PO-Flow at low, intermediate, and high propensity.

Figure 2 shows the conditional outcome densities at three representative covariate vectors. Both methods recover multimodal structure, but JETO more closely tracks the locations and relative heights of the ground-truth modes in most panels. JETO also smooths some of the narrower components; neither method reproduces the target densities exactly.

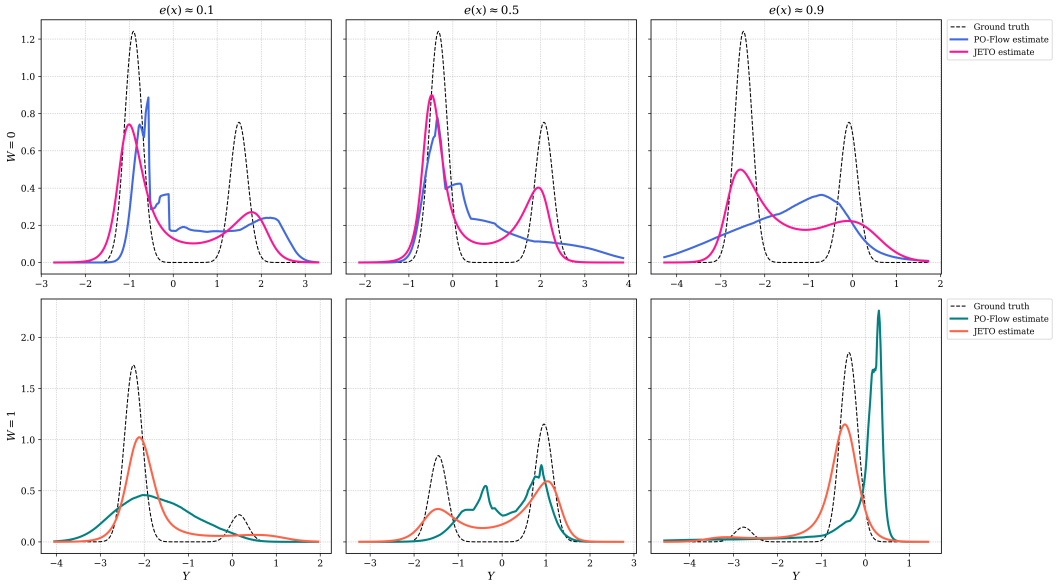


Figure 2: Treatment-specific conditional outcome densities on GM-P at representative covariate vectors with $e(x) \approx 0.1$, 0.5, and 0.9. The upper row shows $W = 0$ and the lower row $W = 1$. Dashed curves are the ground-truth Gaussian-mixture densities; solid curves are the densities estimated by JETO and PO-Flow.

The quantitative comparison uses 100 covariate vectors in each propensity region. Table 3 reports the KL divergence from the true conditional density to each learned density, separately for the two treatment arms. JETO has a lower mean KL divergence in all six region–arm comparisons. The difference is smallest for the treated arm in the high-propensity region, where the values are 0.607 and 0.622.

Table 3: Conditional-density recovery on GM-P measured by $D_{\mathrm{KL}}(p_{\mathrm{true}} \| \hat{p})$. Entries are mean $\pm$ standard deviation across covariate vectors; lower is better.

	$W = 0$			$W = 1$		
Method	Low	Middle	High	Low	Middle	High
JETO	**0.759** $\pm$ 0.048	**0.729** $\pm$ 0.055	**0.771** $\pm$ 0.057	**0.810** $\pm$ 0.056	**0.770** $\pm$ 0.043	**0.607** $\pm$ 0.071
PO-Flow	1.075 $\pm$ 0.085	1.059 $\pm$ 0.096	1.160 $\pm$ 0.090	1.219 $\pm$ 0.043	1.079 $\pm$ 0.065	0.622 $\pm$ 0.052

The low-overlap regions are particularly informative. For the minority treatment arms, JETO obtains a KL divergence of 0.810 versus 1.219 for treated outcomes when $e(x) \approx 0.1$, and 0.771 versus 1.160 for control outcomes when $e(x) \approx 0.9$. These results show a density recovery advantage in settings where observations from the corresponding arm are relatively scarce.

The relaxed JETO model also recovers treatment propensity: density integration and Monte Carlo sampling yield similar errors across all three regions (Appendix D.5).

7 Discussion

JETO learns $p(Y, W \mid X)$ and thus supports estimation of CATE, treatment-specific outcome distributions, and propensity from a joint generative model. It achieves the lowest mean rPEHE on four of six benchmarks, but the results on ACIC and NW-B show that joint modeling does not guarantee the best CATE estimate. In NW-B, treatment is randomized with $e(x) = 0.5$, so the assignment mechanism provides no covariate-dependent signal. This is consistent with the view that modeling treatment assignment is most useful when its structure is related to the outcome distribution, although the benchmark comparison alone cannot establish that mechanism.

The matched GM-I/GM-P comparison addresses this question more directly. In GM-I, where mixture weights are independent of propensity, joint and conditional models perform similarly. In GM-P, where the treated-outcome mixture weight depends on propensity, the conditional model has approximately 7.5% higher rPEHE than joint JETO. Errors in the individual response surfaces are similar, suggesting that the gain concerns how their estimates combine into CATE. Conditional models can learn the relevant dependence from X; the joint objective additionally supplies an explicit assignment signal. These controlled results support a benefit of that signal in the considered setting, without implying a universal advantage.

This distinction concerns estimation from finite samples, not causal identification. Under exchangeability and positivity, both models can in principle recover CATE from $p(Y \mid X, W)$; learning propensity is not a prerequisite. In GM-P, however, the same observed covariates determine assignment probabilities and the mixture weights of treated outcomes. The joint objective repeatedly exposes the network to that shared structure through both generated coordinates. Similar RMSE for the individual response surfaces does not preclude a CATE gain: the error of their difference also depends on how the two estimation errors vary together. The paired experiment demonstrates a gain in this constructed setting; whether the same inductive bias helps in other observational distributions remains an empirical question.

The density experiment examines a separate strength of the generative formulation. With a continuous treatment relaxation, JETO recovers the main modes of the GM-P outcome distributions and has lower mean conditional-density KL divergence than PO-Flow in all six evaluated region–arm combinations. The advantage persists for minority treatment arms: KL divergence is 0.810 versus 1.219 for treated outcomes at $e(x) \approx 0.1$, and 0.771 versus 1.160 for control outcomes at $e(x) \approx 0.9$. This is consistent with the assignment signal aiding density estimation where observations from the corresponding arm are scarce. Monte Carlo and density integration yield similar propensity errors (Appendix D.5). These distributional findings concern the relaxed treatment variant; the main CATE benchmarks use hard treatment codes.

Additional diagnostics clarify the implementation. Outcome and treatment gradients are largely near-orthogonal and similarly scaled, indicating a distinct assignment signal without persistent strong interference (Appendix D.2). Under the matched budgets studied, Flow Matching yields more stable hyperparameter search and lower final rPEHE than the diffusion variant (Appendix D.4). Inference ablations support the use of $K = 2048$ samples and ten RK4 steps (Appendix D.3).

Limitations. True CATE and conditional outcome densities are unavailable for direct validation in observational applications, so the evidence here relies on synthetic and semi-synthetic benchmarks. The controlled link between propensity and outcome modality in GM-P may not reflect the structure of a particular application. JETO requires ODE integration and Monte Carlo sampling at inference; rare treatment arms can require more samples for precise conditional means. Future work should assess distributional treatment effects in applied settings and reduce inference cost, especially under limited treatment overlap.

8 Conclusion

JETO uses conditional Flow Matching to model treatment assignment and outcome jointly. It achieves the lowest mean rPEHE on four of six benchmarks, and a matched comparison shows an advantage over conditional outcome generation when propensity and outcome modality share covariate-dependent structure. Joint generation also helps where data is scarce: in minority treatment arms under limited overlap, JETO lowers the treated-arm conditional-density KL on GM-P from 1.219 (PO-Flow) to 0.810 at $e(x) \approx 0.1$, with a matching gain in the control arm at $e(x) \approx 0.9$ (0.771 vs. 1.160). Gradient diagnostics show that outcome and assignment gradients remain near-orthogonal and similarly scaled during training (Appendix D.2), so the assignment objective acts as an independent, non-conflicting regularization signal. Together with the matched joint-versus-conditional comparison (Section 6.2), these findings establish joint generation as a viable approach to heterogeneous treatment effect analysis, with limited overlap and shared assignment–outcome structure as the regime where its benefit is realized.

AI USE STATEMENT

We used generative AI tools to identify potentially relevant literature, provide feedback on research methodology and experimental design, and review and refine the wording of parts of the manuscript. We also used these tools to generate isolated auxiliary code modules and to review the wording of the discussion of experimental results. We did not use generative AI to originate the central research idea or conduct the quantitative analysis. The authors independently assessed the results and determined the final scientific interpretation. Suggested references were checked against the original publications; AI-assisted text and code were reviewed, and code was tested before use. The authors take responsibility for the final content of this work, including all AI-assisted text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

The model and estimation procedures are described in the main paper. Dataset construction, including the new Gaussian-mixture benchmarks, is specified in Appendix B; experimental settings and hyperparameter selection are documented in Appendix A. We provide anonymized source code for training JETO, generating the synthetic data, and reproducing the experiments as supplementary material.

AUTHOR CONTRIBUTIONS

All authors contributed equally to this work.

REFERENCES

Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. Causalml: Python package for causal machine learning, 2020.

Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 34(1):43–68, 2019. doi: 10.1214/18-STS667.

Jennifer L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011. doi: 10.1198/jcgs.2010.08162.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015. ISBN 9781139025751. doi: 10.1017/CBO9781139025751.

Edward H. Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics*, 17(2):3008–3049, 2023. doi: 10.1214/23-EJS2157.

Stanislav R. Kirpichenko, Andrei V. Konstantinov, and Lev V. Utkin. Sdpm: Survival diffusion probabilistic model for continuous-time survival analysis, 2026. URL https://arxiv.org/abs/2605.22776.

Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, 2019. doi: 10.1073/pnas.1804597116.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Christos Louizos, Uri Shalit, Joris M. Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yuchen Ma, Valentyn Melnychuk, Jonas Schweisthal, and Stefan Feuerriegel. Diffpo: A causal diffusion model for learning distributions of potential outcomes. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-1384.

Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021. doi: 10.1093/biomet/asaa076.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983. doi: 10.1093/biomet/70.1.41.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974. doi: 10.1037/h0037350.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3076–3085. PMLR, 2017.

Claudia Shi, David M. Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 2507–2517, 2019.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. URL https://arxiv.org/abs/2302.00482.

Dongze Wu, David I. Inouye, and Yao Xie. Flow-based generative modeling of potential outcomes and counterfactuals. In *2026 IEEE International Symposium on Information Theory (ISIT)*, 2026.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations (ICLR)*, 2018.

A Experimental Protocol and Hyperparameter Search Spaces

A.1 Hyperparameter Selection

Hyperparameters were selected independently for every method and dataset using a single dedicated realization excluded from all subsequent evaluation. Optimization used the Optuna tree-structured Parzen estimator with 100 trials and rPEHE as the objective. Each trial was evaluated by three-fold cross-validation stratified by treatment assignment. Within the training portion of each fold, 20% of observations were reserved for validation and early stopping where supported.

After optimization, the selected configuration was fixed across all evaluation realizations. For IHDP and ACIC, one of the 100 available realizations was used for tuning and the remaining 99 for evaluation. For NW-B, NW-D, GM-I, and GM-P, tuning used a separate realization, followed by evaluation on 20 independently generated realizations.

A.2 Evaluation Splits and Training

For each evaluation realization, 30% of observations were held out as a test set using a split stratified by treatment assignment. Of the remaining observations, 20% were used for validation, resulting in approximately 56%, 14%, and 30% of the data being used for model fitting, validation, and testing, respectively. All reported metrics were computed exclusively on the held-out test set and aggregated across realizations.

JETO used validation-based early stopping. The XGBoost components of the meta-learners similarly used early stopping with a maximum of 1000 boosting iterations. The released CEVAE, DiffPO, and PO-Flow implementations use fixed training durations; their numbers of epochs were therefore included in the hyperparameter search.

A.3 Monte Carlo Inference

JETO, DiffPO, and PO-Flow used the same total final inference budget of $K = 2048$ generated samples per covariate vector. JETO drew all samples from its learned joint distribution. For DiffPO and PO-Flow, the budget was divided equally between treatment conditions, with 1024 samples generated for each of $W = 0$ and $W = 1$. Treatment-specific means were estimated by Monte Carlo averaging, and CATE was computed as their difference. JETO validation used $K = 512$ samples to reduce repeated inference cost.

A.4 Hyperparameter Search Spaces

The following tables report the complete search spaces and fixed implementation settings used for each method. Hyperparameters were optimized independently for each dataset using 100 Optuna trials with the TPE sampler and three-fold cross-validation. The optimization objective was rPEHE.

Table 4: Hyperparameter search space for JETO.

Hyperparameter	Search space	Sampling
Batch size	$\{32, 64, 128\}$	categorical
Learning rate	$[10^{-5}, 10^{-3}]$	log-uniform
Number of hidden layers	$\{2, 3, 4, 5\}$	integer
Hidden dimension	$\{64, 128, 256, 512\}$	categorical
Activation	$\{\mathrm{ReLU}, \mathrm{SiLU}\}$	categorical
Normalization	$\{\mathrm{none}, \mathrm{layer}, \mathrm{conditional}\}$	categorical
Dropout	0 or $[0.05, 0.25]$	categorical + uniform
Weight decay	$\{0, 10^{-6}, 10^{-5}, 10^{-4}\}$	categorical
Time-embedding dimension	$\{8, 16\}$	categorical
Feature-embedding noise	$\{\mathrm{False}, \mathrm{True}\}$	categorical
Number of Fourier frequencies	$\{8, 16, 32, 64\}$	categorical
Fourier frequency scale	$[10^{-2}, 1]$	log-uniform

The JETO architecture was fixed to a fully connected MLP with Fourier features and sinusoidal time encoding. The Flow Matching path used linear interpolation with $t \sim U(0, 1)$, and the outcome and treatment components were equally weighted in the training objective. Training used a maximum of 1000 epochs with validation performed every five epochs. Validation inference used three RK4 integration steps, whereas final inference used ten steps. The Monte Carlo budgets were 512 samples for validation and 2048 samples for final inference.

Table 5: Hyperparameter search space for the XGBoost-based meta-learners. Parameters marked with an asterisk are used only by the X-, R-, and DR-learners for propensity-score estimation.

Hyperparameter	Search space	Sampling
Maximum tree depth	$\{2, \ldots, 8\}$	integer
Learning rate	$[10^{-3}, 0.2]$	log-uniform
Subsample ratio	$[0.5, 1.0]$	uniform
Feature subsample ratio	$[0.5, 1.0]$	uniform
Minimum child weight	$[10^{-2}, 20]$	log-uniform
L_1 regularization	$[10^{-3}, 10]$	log-uniform
L_2 regularization	$[10^{-3}, 10]$	log-uniform
Minimum loss reduction (γ)	$[10^{-3}, 10]$	log-uniform
Propensity regularization C^*	$[10^{-4}, 10^{2}]$	log-uniform
Propensity L_1 ratio*	$[0, 1]$	uniform

For all XGBoost regression components, the maximum number of estimators was fixed to 1000 and early stopping was applied with a patience of 30 boosting rounds. The squared-error regression objective was used throughout. Propensity models required by the X-, R-, and DR-learners were implemented using elastic-net logistic regression.

CEVAE used 1000 latent Monte Carlo samples for prediction, with the learning rate decay fixed to one.

Table 6: Hyperparameter search space for CEVAE.

Hyperparameter	Search space	Sampling
Outcome distribution	{Normal, Laplace, Student-t}	categorical
Latent dimension	$\{8, 16, 32\}$	categorical
Hidden dimension	$\{64, 128, 256, 512\}$	categorical
Number of layers	$\{2, \ldots, 5\}$	integer
Batch size	$\{32, 64, 128\}$	categorical
Learning rate	$[10^{-5}, 10^{-3}]$	log-uniform
Weight decay	$[10^{-6}, 10^{-3}]$	log-uniform
Number of epochs	$\{50, 100, 200, 300\}$	categorical

Table 7: Hyperparameter search spaces for DiffPO and PO-Flow.

Model	Hyperparameter	Search space	Sampling
DiffPO / PO-Flow	Learning rate	$[10^{-5}, 10^{-3}]$	log-uniform
DiffPO / PO-Flow	Number of layers	$\{2, 3, 4\}$	integer
DiffPO / PO-Flow	Hidden dimension	$\{64, 128, 256\}$	categorical
DiffPO / PO-Flow	Batch size	$\{64, 128, 256\}$	categorical
DiffPO / PO-Flow	Number of epochs	$\{50, 100, 200, 500\}$	categorical
DiffPO	Final diffusion variance β_{end}	$\{0.1, 0.2, 0.5\}$	categorical
DiffPO	Propensity learning rate	$[10^{-4}, 10^{-3}]$	log-uniform
DiffPO	Propensity hidden dimension	$\{64, 100, 128\}$	categorical

For DiffPO, the number of diffusion steps was fixed to 100, with $\beta_{\text{start}} = 10^{-4}$, a quadratic noise schedule, and a diffusion embedding dimension of 128. Its auxiliary propensity network was trained for 1000 epochs with batch size 100 and weight decay 10^{-4}.

For PO-Flow, ODE integration used the adaptive Dormand–Prince solver (DOPRI5) with relative and absolute tolerances of 10^{-5}. For final Monte Carlo estimation, both conditional generative baselines used 1024 samples per treatment arm, corresponding to a total budget of 2048 generated outcomes per covariate vector.

B Dataset Details

B.1 Standard and Semi-Synthetic Benchmarks

The Infant Health and Development Program (IHDP) benchmark (Hill, 2011) combines covariates from a randomized study with simulated treatment assignments and response surfaces. Both treatment-specific conditional means are available for evaluation. We use the standard collection of 100 realizations.

The 2016 Atlantic Causal Inference Conference benchmark (Dorie et al., 2019) combines real covariates with synthetically generated treatment assignments and potential outcomes. Its realizations cover heterogeneous data-generating mechanisms of greater complexity than IHDP. We use a fixed subset of 100 realizations.

NW-B and NW-D correspond to Setups B and D of Nie and Wager (Nie & Wager, 2021), generated using the CausalML simulation suite (Chen et al., 2020). NW-B uses randomized treatment, $e(x) = 0.5$, with nonlinear response surfaces. NW-D uses a nonconstant propensity score and substantially different control and treatment response surfaces. Both provide the exact conditional means, CATE, and propensity score.

B.2 Gaussian-Mixture Benchmarks

The GM-I and GM-P benchmarks share the same covariate distribution, treatment-assignment mechanism, and central response surfaces. They differ only in how the potential-outcome mixture probabilities depend on the covariates and treatment propensity.

Covariates are sampled as $X \sim \mathcal{N}(0, I_p)$, and treatment is assigned according to

$$e(x) = P(W = 1 \mid X = x) = \sigma(\lambda \sin(x_0 - x_1)), \qquad W \mid X = x \sim \text{Bernoulli}(e(x)), \tag{16}$$

where $\sigma(a) = (1 + \exp(-a))^{-1}$, and λ controls the strength of treatment selection.

Nonlinear response surfaces are generated using random Fourier features. For a feature bank $\mathcal{B}$, we define

$$r_{\mathcal{B}}(x) = \sqrt{\frac{2}{M}} \sum_{j \in \mathcal{B}} a_j \cos\big(\omega_j^\top x + \phi_j\big), \tag{17}$$

where $a_j \sim \mathcal{N}(0, 1)$ and $\phi_j \sim \mathcal{U}(0, 2\pi)$. Each frequency vector ω_j is nonzero only on a randomly selected subset of covariates, producing interactions of different orders. Conditional on a support of size s_j, its nonzero entries are sampled with variance proportional to $1/s_j$.

Three independent feature banks define a common prognostic component $g(x)$ and treatment-specific components $u_0(x)$ and $u_1(x)$. The central response surfaces are

$$c_0(x) = g(x) + u_0(x), \qquad c_1(x) = g(x) + u_1(x). \tag{18}$$

The shared component induces structural similarity between treatment arms, while the arm-specific components produce nonlinear treatment-effect heterogeneity.

For each treatment w, a mixture indicator is sampled as $M_w \mid X = x \sim \text{Bernoulli}(\pi_w(x))$, and the corresponding potential outcome is

$$Y(w) = c_w(x) + \Delta(2M_w - 1) + \varepsilon_w, \qquad \varepsilon_w \sim \mathcal{N}(0, \sigma_Y^2). \tag{19}$$

Consequently, the conditional potential-outcome density is

$$\begin{aligned} p(y \mid x, W = w) = [1 - \pi_w(x)]\mathcal{N}\big(y;\, c_w(x) - \Delta, \sigma_Y^2\big) \\ + \pi_w(x)\mathcal{N}\big(y;\, c_w(x) + \Delta, \sigma_Y^2\big). \end{aligned} \tag{20}$$

Its conditional mean is

$$\mu_w(x) = c_w(x) + \Delta[2\pi_w(x) - 1], \tag{21}$$

and the true CATE is

$$\tau(x) = u_1(x) - u_0(x) + 2\Delta[\pi_1(x) - \pi_0(x)]. \tag{22}$$

In **GM-I**, the mixture probabilities are independent of the treatment propensity and identical between treatment arms:

$$\pi_0(x) = \pi_1(x) = \frac{1}{2}. \tag{23}$$

The conditional mixtures are therefore symmetric around their central surfaces, giving $\mu_w(x) = c_w(x)$ and $\tau(x) = u_1(x) - u_0(x)$. GM-I tests recovery of nonlinear CATE and bimodal outcome distributions without coupling outcome modality to treatment assignment.

In **GM-P**, the central surfaces and propensity score remain unchanged, but the mixture probabilities are

$$\pi_0(x) = \sigma(-\alpha), \qquad \pi_1(x) = \sigma(\alpha + \rho \operatorname{logit} e(x)). \tag{24}$$

Here, α controls baseline mixture asymmetry and ρ controls the strength of coupling between the treatment propensity and the treated-outcome mixture weight. GM-P is a mechanism-isolation benchmark: it tests whether joint modeling can exploit structure shared by treatment assignment and outcome modality.

This construction does not introduce unobserved confounding. Treatment assignment is sampled independently of the mixture indicators and outcome noise conditional on the observed covariates; therefore, $\{Y(0), Y(1)\} \perp W \mid X$ continues to hold.

Both Gaussian-mixture benchmarks use $n = 1000$ observations and $p = 10$ covariates. Each treatment-specific central surface contains 100 Fourier components: 75 shared between treatment arms and 25 treatment-specific components. The remaining parameters are

$$\Delta = 0.5, \qquad \sigma_Y = 0.2, \qquad \lambda = 2.5, \qquad \alpha = 0.5, \qquad \rho = 1. \tag{25}$$

The Fourier-feature realization is fixed across repetitions, whereas the covariates, treatment assignments, mixture indicators, and outcome noise are resampled independently. Repetitions therefore represent independent samples from a fixed data-generating process rather than different response surfaces.

C Additional Evaluation Metrics and PO-RMSE Results

In addition to the rPEHE reported in the main paper, we consider absolute Precision in Estimation of Heterogeneous Effect (PEHE), treatment-specific response-surface reconstruction, and propensity-score recovery. Absolute PEHE is defined as

$$\mathrm{PEHE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\tau}(x_i) - \tau(x_i)\right)^2}. \tag{26}$$

The rPEHE used as the primary metric in the main paper normalizes this quantity by the root mean square magnitude of the true treatment effect.

When both treatment-specific response surfaces are available, their reconstruction accuracy is measured as

$$\mathrm{RMSE}_w = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_w(x_i) - \mu_w(x_i)\right)^2}, \qquad w \in \{0,1\}, \tag{27}$$

and summarized by

$$\text{PO-RMSE} = \frac{1}{2}\left(\mathrm{RMSE}_0 + \mathrm{RMSE}_1\right). \tag{28}$$

Unlike rPEHE, PO-RMSE is not normalized by the outcome scale and is therefore used as a diagnostic of response-surface reconstruction rather than for comparisons across datasets with different outcome scales.

For datasets with a known treatment-assignment mechanism, propensity recovery is evaluated using

$$\mathrm{RMSE}_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{e}(x_i) - e(x_i)\right)^2}. \tag{29}$$

For $W \mid X = x \sim \mathrm{Bernoulli}(e(x))$, this metric corresponds to the reducible component of the expected Brier loss because

$$\mathbb{E}\left[\left(W - \hat{e}(x)\right)^2 \mid X = x\right] = e(x)[1 - e(x)] + \left[\hat{e}(x) - e(x)\right]^2. \tag{30}$$

The PO-RMSE comparison includes methods whose standard output provides both $\hat{\mu}_0(x)$ and $\hat{\mu}_1(x)$. The X- and R-learners, which primarily estimate treatment effects rather than reconstructing both response surfaces, are therefore omitted.

Table 8: Comparison of PO-RMSE across datasets. Values are reported as means with standard deviations over experimental repetitions; the best mean for each dataset is shown in bold. An asterisk denotes a statistically significant difference from the best-mean method within the same dataset according to a two-sided paired Wilcoxon signed-rank test with Holm correction across methods ($p_{\mathrm{adj}} < 0.05$).

Method	IHDP	ACIC	GM-I	GM-P	NW-B	NW-D
JETO	**0.838**±0.841	0.979*±0.375	**0.399**±0.022	**0.407**±0.021	0.663*±0.206	0.369±0.058
PO-Flow	1.902*±3.075	1.238*±0.361	0.420*±0.025	0.429*±0.030	**0.502**±0.041	**0.350**±0.029
DiffPO	9.428*±13.975	2.889*±0.915	1.414*±1.526	2.087*±0.835	1.475*±1.040	1.934*±0.836
CEVAE	2.125*±3.123	2.757*±0.708	0.474*±0.024	0.496*±0.030	0.710*±0.045	0.541*±0.056
T-Learner	1.611*±2.395	**0.824**±0.377	0.444*±0.025	0.468*±0.020	0.630*±0.068	0.441*±0.031
S-Learner	2.149*±3.372	0.986*±0.327	0.421*±0.024	0.486*±0.018	0.592*±0.029	0.638*±0.043
DR-Learner	1.488*±1.940	1.073*±0.375	0.511*±0.024	0.559*±0.024	0.711*±0.078	0.552*±0.050

JETO achieves the lowest mean PO-RMSE on IHDP, GM-P, and GM-I. PO-Flow performs best on NW-D and NW-B, while the T-learner obtains the lowest error on ACIC. These results broadly follow the rPEHE comparison on IHDP and the two Gaussian-mixture benchmarks. On NW-D, the two Flow Matching methods perform similarly, whereas PO-Flow has a clearer advantage on NW-B. Thus, joint modeling does not uniformly improve reconstruction of both response surfaces across all data-generating mechanisms.

D Additional Ablation and Diagnostic Studies

D.1 Joint versus Conditional Modeling

To isolate the effect of generating treatment assignment jointly with the outcome, we compare JETO with an architecture-matched conditional variant that models $p(Y \mid X, W)$. The two variants use essentially identical model capacity and the same total Monte Carlo budget of $K = 2048$ samples per covariate vector.

The comparison uses 20 paired realizations of GM-I and GM-P generated independently from those used in the main benchmark. The same data splits, training seeds, and sampling seeds are used for the two variants within each realization. Consequently, the absolute JETO results need not coincide with the main benchmark; the comparison of interest is the paired difference between the joint and conditional formulations.

As reported in Table 2, the conditional formulation is slightly better on GM-I, where the outcome-mixture probabilities are independent of treatment propensity, but its confidence interval for the paired difference includes zero. On GM-P, joint modeling reduces mean rPEHE from 0.6273 to 0.5834. The conditional variant has approximately 7.5% higher mean rPEHE than joint JETO.

To determine whether this improvement results from uniformly more accurate reconstruction of the individual response surfaces, Table 9 reports the RMSE of $\hat{\mu}_0(x)$ and $\hat{\mu}_1(x)$. The two formulations achieve similar accuracy for the individual conditional means. On GM-P, conditional JETO is slightly more accurate for μ_0, whereas joint JETO is slightly more accurate for μ_1; neither difference is large.

Table 9: RMSE of the treatment-specific conditional mean estimates for the joint and conditional JETO formulations.

	μ_0 RMSE		μ_1 RMSE	
Dataset	Joint	Conditional	Joint	Conditional
GM-I	**0.4005**	0.4005	0.3906	**0.3736**
GM-P	0.4000	**0.3922**	**0.4097**	0.4212

CATE accuracy depends not only on the marginal accuracy of the two response surfaces but also on the relationship between their errors. Defining $\varepsilon_w(x) = \hat{\mu}_w(x) - \mu_w(x)$, the CATE error is

$$\hat{\tau}(x) - \tau(x) = \varepsilon_1(x) - \varepsilon_0(x). \tag{31}$$

Thus, models with similar RMSE for μ_0 and μ_1 can differ in CATE accuracy when the errors of the two surfaces have different dependence structure. The GM-P results suggest that joint modeling improves the coherence of the treatment-specific estimates when their difference is used to form CATE, rather than uniformly reducing both response-surface errors.

The contrast between GM-I and GM-P is consistent with the intended role of joint modeling. In GM-I, outcome modality is independent of the assignment mechanism. In GM-P, the treated-outcome mixture weight depends on the propensity through $\pi_1(x) = \sigma(\alpha + \rho \operatorname{logit} e(x))$. A conditional model can represent this dependence through X, but its objective does not explicitly learn $p(W \mid X)$. The advantage observed only on GM-P therefore provides a controlled illustration of when the additional assignment signal can improve CATE estimation.

D.2 Gradient Interaction Analysis

Joint modeling requires the shared network parameters to support both the outcome and treatment components of the Flow Matching objective. We therefore measure whether their optimization signals are aligned, conflicting, or strongly imbalanced. Let $g_Y = \nabla_\theta \mathcal{L}_Y$ and $g_W = \nabla_\theta \mathcal{L}_W$ denote the gradients of the two coordinate losses with respect to the shared parameters. We monitor

$$\cos(g_Y, g_W) = \frac{g_Y^\top g_W}{\|g_Y\| \, \|g_W\|}, \qquad R_{\text{grad}} = \frac{\|g_Y\|}{\|g_W\|}. \tag{32}$$

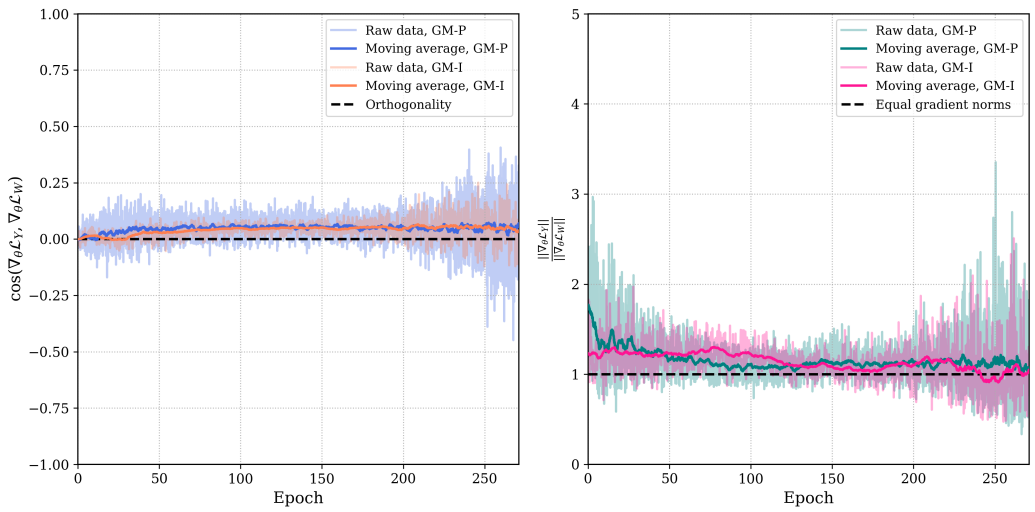


Figure 3: Interaction between the outcome and treatment-assignment gradients during JETO training on GM-I and GM-P, aggregated over 20 independent realizations. Left: gradient cosine similarity. Right: ratio of gradient norms. Solid curves show moving averages; lighter curves show unsmoothed trajectories.

A cosine near zero indicates approximately orthogonal update directions, while a norm ratio near one indicates that neither component systematically dominates by gradient magnitude.

For both datasets, the gradient cosine remains close to zero throughout training, with median values of 0.037 for GM-I and 0.047 for GM-P. The two components therefore provide largely distinct rather than redundant optimization signals. Their magnitudes are closely matched: the median gradient-norm ratio is 0.961 on GM-I and 0.997 on GM-P, and the smoothed trajectories remain near unity over most of training.

Local gradient conflict occurs in both settings and is more persistent on GM-P. During late training, approximately 38.2% of shared-parameter updates have negative cosine on GM-P, compared with 29.9% on GM-I. Strong negative alignment remains uncommon, occurring in 5.7% and 0.1% of late-phase updates, respectively. Thus, the propensity-linked structure does not make the objectives identical or uniformly aligned. Instead, treatment and outcome learning provide near-orthogonal signals of similar scale that occasionally compete without persistent strong interference.

D.3 Monte Carlo Budget and Numerical Integration

JETO estimates treatment-specific means and propensity scores from samples of the learned joint distribution. We evaluate sensitivity to the total Monte Carlo budget

$$K \in \{128, 256, 512, 1024, 2048, 4096, 8192\}. \tag{33}$$

The experiment uses 20 independent realizations of NW-B, NW-D, GM-I, and GM-P, with one fitted model per realization. For each test observation, $K_{\max} = 8192$ samples are generated once, and smaller budgets use nested prefixes of the same sample sequence. Comparisons therefore hold the fitted model, test observations, initial noise, and generated trajectories fixed.

Increasing K consistently improves CATE estimation, with most of the reduction in rPEHE occurring before $K = 2048$. The effect is strongest on NW-B, smaller on NW-D and GM-I, and weakest on GM-P. Propensity estimation is less sensitive once several hundred samples are available.

Because treatment assignment is generated jointly with outcome, only a fraction of the samples contributes to the conditional mean of the minority treatment arm. Increasing K therefore increases the effective sample size for the rarer potential outcome. At $K = 2048$, observations in poor-overlap regions receive hundreds of minority-arm samples on average, while further increases yield

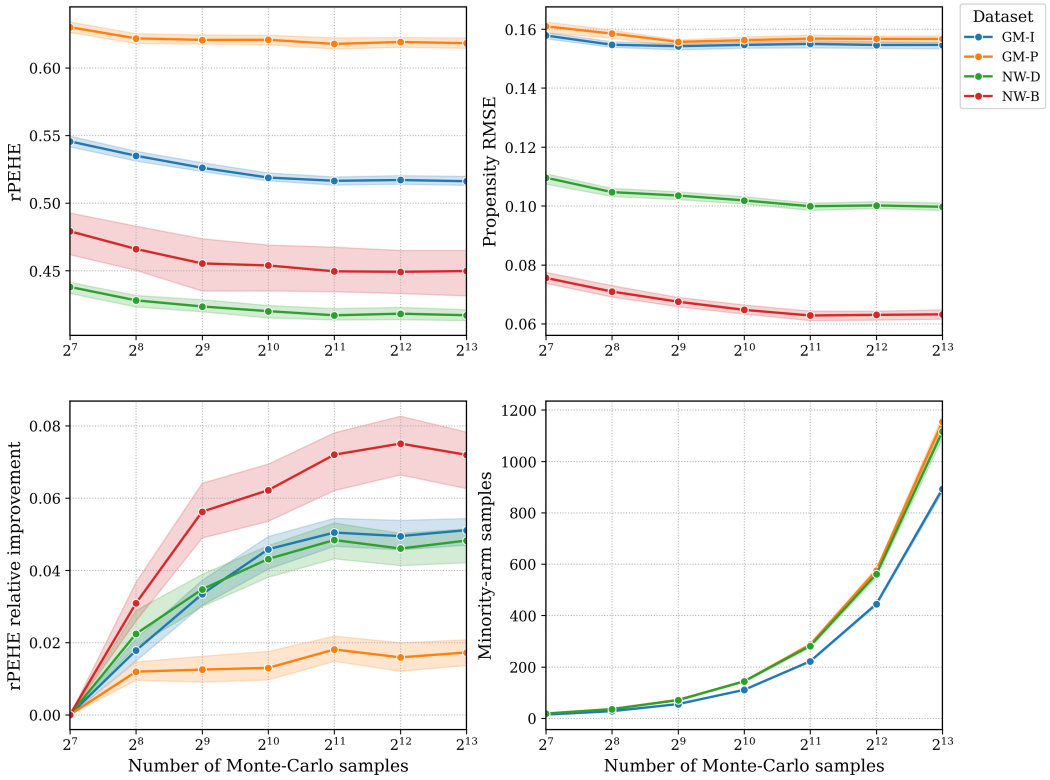


Figure 4: Sensitivity of JETO inference to the Monte Carlo budget K, averaged over 20 realizations per dataset. Top left: rPEHE. Top right: propensity-score RMSE. Bottom left: relative rPEHE improvement from $K = 128$. Bottom right: mean number of generated minority-arm samples for observations with $e(x) < 0.1$ or $e(x) > 0.9$. NW-B is omitted from the last panel because $e(x) = 0.5$. Shading indicates variability across realizations.

only small rPEHE improvements. We therefore use $K = 2048$ for final inference and $K = 512$ for validation.

We additionally compare Euler and classical fourth-order Runge–Kutta (RK4) integration on IHDP and NW-D over 30 independent realizations per dataset. The fitted model, Monte Carlo samples, and initial noise are held fixed across solver configurations. At equal step counts, RK4 consistently yields lower PEHE and rPEHE than Euler. Increasing RK4 from 10 to 20 or 50 steps produces essentially no further improvement while increasing inference time. Five-step RK4 is already close to the 10-step solution, but ten steps track the 50-step reference more closely, particularly on IHDP. Final inference therefore uses ten RK4 steps, whereas validation uses three steps to reduce repeated cost.

D.4 Flow Matching versus Diffusion

The joint JETO formulation is not intrinsically tied to Flow Matching. We therefore compare the main implementation with a diffusion-based variant that learns the same joint distribution $p(Y, W \mid X)$. The comparison uses 20 fresh paired realizations of GM-I and GM-P, distinct from the main benchmark realizations.

Both processes use the same broad MLP search space and are optimized independently using 100-trial Optuna TPE studies with the same random seed. The search covers network depth and width, batch size, learning rate, activation, normalization, dropout, weight decay, time embedding, Fourier features, and generated-state embedding. Both variants use standardized outcomes, hard treatment targets $-1/+1$, and the same treatment-loss weight. Flow Matching uses linear interpolation with uniform time sampling; diffusion uses a cosine DDPM schedule with epsilon prediction.

Computational budgets are matched by network evaluations. Validation uses three RK4 steps for Flow Matching, corresponding to 12 vector-field evaluations, and 12 ancestral denoising updates for diffusion; both use $K = 512$. Final inference uses ten RK4 steps (40 evaluations) or 40 diffusion updates, respectively, with $K = 2048$. Paired evaluation uses identical data splits, training seeds, and sampling seeds.

Table 10: Hyperparameter-search behavior of Flow Matching and diffusion. Values summarize validation rPEHE over 100 Optuna trials per process and dataset.

Dataset	Process	Best	Median	IQR	Trials > 10
GM-P	Flow Matching	**0.5859**	0.6482	0.0933	0/100
GM-P	Diffusion	0.6980	0.8577	66.9795	34/100
GM-I	Flow Matching	**0.5535**	0.6115	0.0694	0/100
GM-I	Diffusion	0.5843	0.7250	98.3274	36/100

Flow Matching remains stable throughout the search: all sampled configurations achieve validation rPEHE below one. Diffusion exhibits a much more irregular landscape, with 34–36% of trials producing rPEHE above ten and many configurations exceeding 100. This difference is visible in the first ten startup trials, which use identical hyperparameter configurations for both processes. Flow Matching ranges from 0.630 to 0.992 on GM-P and from 0.590 to 0.946 on GM-I, whereas diffusion ranges from 36.0 to 206.8 and from 53.8 to 189.4, respectively. The instability therefore cannot be attributed solely to diverging adaptive TPE proposals.

Table 11: Final rPEHE comparison between independently optimized Flow Matching and diffusion implementations of JETO. Paired differences are diffusion minus Flow Matching; positive values favor Flow Matching.

Dataset	Flow Matching	Diffusion	Relative increase	Mean paired difference	Paired difference 95% CI
GM-P	**0.5970**	0.7096	+18.9%	+0.1126	[0.0885, 0.1367]
GM-I	**0.5434**	0.6036	+11.1%	+0.0603	[0.0322, 0.0884]

Flow Matching achieves lower rPEHE on all 20 paired GM-P realizations and on 17 of 20 GM-I realizations. The same direction is observed for PEHE, potential-outcome reconstruction, CATE R^2, and propensity estimation. Mean propensity RMSE decreases from 0.1786 to 0.1429 on GM-P and from 0.1891 to 0.1616 on GM-I.

These results show that the joint formulation can be instantiated with either generative process, but the choice has a substantial practical effect. Under the matched search and inference budgets considered here, Flow Matching provides a smoother hyperparameter landscape and more accurate final CATE estimates. It is therefore retained as the default JETO implementation.

D.5 Propensity-Score Recovery

The density-recovery experiment in Section 6.3 also evaluates treatment propensity on the GM-P benchmark with increased mixture separation and a continuously relaxed treatment coordinate. We compare the Monte Carlo and density-integration estimators defined in Section 4.2 against the known propensity for the same low-, intermediate-, and high-propensity regions.

Table 12: Propensity recovery on GM-P, measured by RMSE_e in each propensity region. Entries are mean $\pm$ standard deviation; lower is better.

Estimator	Low	Middle	High
Monte Carlo	0.0533 ± 0.0055	**0.0932** ± 0.0117	0.0408 ± 0.0041
Density integral	**0.0488** ± 0.0049	0.0955 ± 0.0124	**0.0385** ± 0.0040

For numerical density evaluation, the relaxed treatment coordinate is integrated over $[-3, 0]$ for control and $[0, 3]$ for treatment using 48-point Gauss–Legendre quadrature. Class masses are obtained

by trapezoidal integration over the finite outcome grids and normalized by their sum to compensate for omitted probability mass. The Monte Carlo estimator uses $K = 2048$ joint samples per covariate vector.

The two estimators yield similar accuracy. Integration has slightly lower RMSE in the low- and high-propensity regions, whereas Monte Carlo sampling has a marginal advantage in the intermediate region. This agreement indicates that the density representation and generated samples encode similar treatment probabilities. Monte Carlo estimation also applies to the hard treatment codes used in the main benchmark, for which direct continuous density integration is unavailable.