DoubleGen: Debiased Generative Modeling of Counterfactuals

Alex Luedtke

Department of Health Care Policy
Harvard University

Joint work with

Kenji Fukumizu

Institute of Statistical Mathematics
Tokyo, Japan

Background on generative modeling

Generative modeling: a paradigm for generating synthetic data that looks like existing data

Underlies many of the recent advances in AI

- **Chatbots'** training begins by having them try to imitate a large corpus of text (internet text, books, etc.)
- **Image models** are trained to imitate large collections of images

Example: generating new faces

Generative models use **existing data**

Images from the Chicago Face Database (Ma et al., 2015).

Example: generating new faces

Generative models use **existing data** to generate **similar synthetic data**.

$\Downarrow$

Images from the Chicago Face Database (Ma et al., 2015).

Goal today

Goal today: present a general approach for generating synthetic **counterfactual data**

Goal today

Goal today: present a general approach for generating synthetic **counterfactual data**

In our example: generate counterfactual images of people **smiling**

Example of confounding in CelebA (Liu et al., 2015)

	Lipstick	Makeup	Female*	Earrings	No Beard	Blonde
Smiling	**56%**	**47%**	**65%**	**26%**	**88%**	**18%**
Not Smiling	38%	30%	52%	12%	79%	12%
Overall	**47%**	**38%**	**58%**	**19%**	**83%**	**15%**

*Perceived binary sex, as labeled by human annotators.

Example of confounding in CelebA (Liu et al., 2015)

	Lipstick	Makeup	Female	Earrings	No Beard	Blonde
Smiling	**56%**	**47%**	**65%**	**26%**	**88%**	**18%**
Not Smiling	38%	30%	52%	12%	79%	12%
Overall	**47%**	**38%**	**58%**	**19%**	**83%**	**15%**

When trained on only smiling faces, **generative models overrepresent some attributes**, failing to reflect **how the population would look if everyone smiled**.

Example: generating smiling faces

Ideally: Would intervene and collect a dataset of **counterfactual images**

Example: generating smiling faces

Ideally: Would intervene and collect a dataset of **counterfactual images**

Example: generating smiling faces

Ideally: Would intervene and collect a dataset of **counterfactual images**

Example: generating smiling faces

Ideally: Would intervene and collect a dataset of **counterfactual images**

- A generative model could then produce **synthetic counterfactual smiling images**

⇓

Oracle problem: all counterfactuals observed

For the moment, we suppose we have direct access **counterfactuals**.

- Dataset consists of $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Oracle problem: all counterfactuals observed

For the moment, we suppose we have direct access **counterfactuals**.

- Dataset consists of $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Generative modeling problem: learn a **transport map**

- Input: **noise** $U \sim \Pi$, for Π a known distribution
- Output: $\phi_{\mathbb{P}}(U)$, which has distribution $\mathbb{P}$

Oracle problem: all counterfactuals observed

For the moment, we suppose we have direct access **counterfactuals**.

- Dataset consists of $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Generative modeling problem: learn a **transport map**

- Input: **noise** $U \sim \Pi$, for Π a known distribution
- Output: $\phi_{\mathbb{P}}(U)$, which has distribution $\mathbb{P}$

Simple example: if $Y^\star$ is 1d, then can take:

- $\Pi = \mathrm{Uniform}[0, 1]$
- $\phi_{\mathbb{P}} = \mathbb{P}$'s inverse CDF, $F_{\mathbb{P}}^{-1}$

Example: autoregressive language models (Graves, 2013)

$Y^\star = (Y^\star(1), Y^\star(2), \ldots, Y^\star(d))$ is a sequence of tokens:

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

Tokenized sequence displayed as in https://platform.openai.com/tokenizer

Example: autoregressive language models (Graves, 2013)

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

A simple language model can be trained as follows:

1) **Statistical learning** to estimate

$$\theta_{\mathbb{P}} \in \underset{\theta}{\operatorname{argmin}} E_{\mathbb{P}}\left[\ell(\theta, Y^{\star})\right]$$

Example: autoregressive language models (Graves, 2013)

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

A simple language model can be trained as follows:

1) Statistical learning to estimate

$$\theta_{\mathbb{P}} \in \operatorname*{argmin}_{\theta} E_{\mathbb{P}}\left[\ell(\theta, Y^\star)\right],$$

where $\ell(\theta, y^\star) = -\log P_\theta\{y^\star(1)\} - \ldots$

Example: autoregressive language models (Graves, 2013)

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

A simple language model can be trained as follows:

1) **Statistical learning** to estimate

$$\theta_{\mathbb{P}} \in \operatorname*{argmin}_{\theta} E_{\mathbb{P}}\left[\ell(\theta, Y^\star)\right],$$

where $\ell(\theta, y^\star) = -\log P_\theta\{\, y^\star(1)\,\} - \log P_\theta\{\, y^\star(2) \mid y^\star(1)\,\}$

$$- \dots$$

Example: autoregressive language models (Graves, 2013)

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

A simple language model can be trained as follows:

1) **Statistical learning** to estimate

$$\theta_{\mathbb{P}} \in \operatorname*{argmin}_{\theta} E_{\mathbb{P}}\left[\ell(\theta, Y^\star)\right],$$

where $\ell(\theta, y^\star) = -\log P_\theta\{y^\star(1)\} - \log P_\theta\{y^\star(2) \mid y^\star(1)\}$

$$- \log P_\theta\{y^\star(3) \mid y^\star(2), y^\star(1)\} - \dots$$

Example: autoregressive language models (Graves, 2013)

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

A simple language model can be trained as follows:

1) **Statistical learning** to estimate

$$\theta_{\mathbb{P}} \in \operatorname*{argmin}_{\theta} E_{\mathbb{P}}\left[\ell(\theta, Y^\star)\right],$$

where $\ell(\theta, y^\star) = -\log P_\theta\{\, y^\star(1)\,\} - \log P_\theta\{\, y^\star(2) \mid y^\star(1)\,\}$
$$- \log P_\theta\{\, y^\star(3) \mid y^\star(2)\,,\, y^\star(1)\,\} - \dots$$

Example: autoregressive language models (Graves, 2013)

DoubleGen: Debiased Generative Modeling of Counterfactuals

(10948, 11757, 25, 18659, 72, 1882, 4140, 1799, 129776, 328, 32251, 69, 19106, 82)

A simple language model can be trained as follows:

1) **Statistical learning** to estimate

$$\theta_{\mathbb{P}} \in \underset{\theta}{\operatorname{argmin}}\, E_{\mathbb{P}}\left[\ell(\theta, Y^\star)\right],$$

where $\ell(\theta, y^\star) = -\log P_\theta\{\, y^\star(1)\,\} - \log P_\theta\{\, y^\star(2) \mid y^\star(1)\,\}$
$$- \log P_\theta\{\, y^\star(3) \mid y^\star(2)\,,\, y^\star(1)\,\} - \ldots$$

2) **Ancestral sampling** of tokens according to $P_{\widehat{\theta}}(\,\cdot \mid \cdot\,)$

Class of generative models considered today

	Outcome type ($Y^\star$)
Autoreg. model	$[k]^d$ (token seq.)
Diffusion model	$\mathbb{R}^d$ (e.g., image)
Flow matching	$\mathbb{R}^d$ (e.g., image)

Algorithm Oracle counterfactual generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Class of generative models considered today

	Outcome type ($Y^\star$)	Hypothesis ($\theta_{\mathbb{P}}$)	Loss (ℓ)
Autoreg. model	$[k]^d$ (token seq.)	next-token prob.	cross-entropy
Diffusion model	$\mathbb{R}^d$ (e.g., image)	score	denoising score matching
Flow matching	$\mathbb{R}^d$ (e.g., image)	vector field	velocity matching

Algorithm Oracle counterfactual generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework: hypothesis space, loss, sampler

1: **Risk minimization:** define θ_n via $R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^n \ell(\theta, Y_i^\star)$

Class of generative models considered today

	Outcome type ($Y^\star$)	Hypothesis ($\theta_{\mathbb{P}}$)	Loss (ℓ)	Sampler (τ)
Autoreg. model	$[k]^d$ (token seq.)	next-token prob.	cross-entropy	ancestral
Diffusion model	$\mathbb{R}^d$ (e.g., image)	score	denoising score matching	SDE solver
Flow matching	$\mathbb{R}^d$ (e.g., image)	vector field	velocity matching	ODE solver

Algorithm Oracle counterfactual generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework: hypothesis space, loss, sampler

1: **Risk minimization:** define θ_n via $R_n^\star(\theta) := \frac{1}{n} \sum_{i=1}^n \ell(\theta, Y_i^\star)$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Observed data

Dataset consists of n iid copies of $(X, A, Y) \sim P$ with

- $X =$ baseline covariates
- $A = a^\star \implies$ received intervention
- $Y =$ outcome

Suppose $\mathbb{P}$'s identifiable through the **G-formula**:

$$\mathbb{P}\{Y^\star \in \mathcal{Y}\} = \int P\{Y \in \mathcal{Y} \,|\, A = a^\star, X = x\}\, P_X(dx) \;\; \text{for all sets } \mathcal{Y}$$

Modifying oracle algorithm for factual problem

Algorithm Oracle counterfactual generative modeling generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm Oracle counterfactual generative modeling generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm Oracle counterfactual generative modeling generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: counterfactual data $Y_1^\star, Y_2^\star, \ldots, Y_n^\star \overset{\text{iid}}{\sim} \mathbb{P}$

Require: choice of generative modeling framework

Risk minimization: define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

return transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: factual data $(X_1, A_1, Y_1), (X_2, A_2, Y_2), \ldots, (X_n, A_n, Y_n) \overset{\text{iid}}{\sim} P$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: factual data $(X_1, A_1, Y_1), (X_2, A_2, Y_2), \ldots, (X_n, A_n, Y_n) \overset{\text{iid}}{\sim} P$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: factual data $(X_1, A_1, Y_1), (X_2, A_2, Y_2), \dots, (X_n, A_n, Y_n) \overset{\text{iid}}{\sim} P$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the empirical risk

$$R_n^\star(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, Y_i^\star)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: factual data $(X_1, A_1, Y_1), (X_2, A_2, Y_2), \ldots, (X_n, A_n, Y_n) \overset{\text{iid}}{\sim} P$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the AIPW* risk

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

*AIPW = augmented inverse probability weighted (Robins et al., 1994)

Modifying oracle algorithm for factual problem

Algorithm DoubleGen: Doubly robust generative modeling

Require: factual data $(X_1, A_1, Y_1), (X_2, A_2, Y_2), \ldots, (X_n, A_n, Y_n) \overset{\text{iid}}{\sim} P$

Require: choice of generative modeling framework

1: **Risk minimization:** define θ_n via the AIPW* risk

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

2: **return** transport map $\phi_n := \tau(\theta_n)$

*AIPW = augmented inverse probability weighted (Robins et al., 1994)

Estimating nuisances

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\left\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\right\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

Two nuisances must be estimated

[1] Zubizarreta (2015), Chernozhukov et al. (2021)

Estimating nuisances

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\left\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\right\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

Two nuisances must be estimated:

1) Inverse propensity:[1] stable balancing weights, Riesz regression, logistic regression

[1] Zubizarreta (2015), Chernozhukov et al. (2021)

Estimating nuisances

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\left\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\right\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

Two nuisances must be estimated:

1) **Inverse propensity:**[1] stable balancing weights, Riesz regression, logistic regression
2) **Outcome generative model:** conditional generative model, k-nearest neighbors

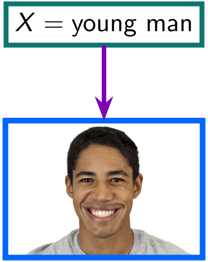
[1] Zubizarreta (2015), Chernozhukov et al. (2021)

Estimating nuisances

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\left\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\right\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

Two nuisances must be estimated:

1) **Inverse propensity:**[1] stable balancing weights, Riesz regression, logistic regression
2) **Outcome generative model:** conditional generative model, k-nearest neighbors



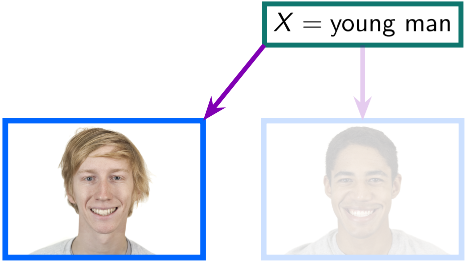
[1] Zubizarreta (2015), Chernozhukov et al. (2021)

Estimating nuisances

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

Two nuisances must be estimated:

1) **Inverse propensity:**[1] stable balancing weights, Riesz regression, logistic regression
2) **Outcome generative model:** conditional generative model, k-nearest neighbors



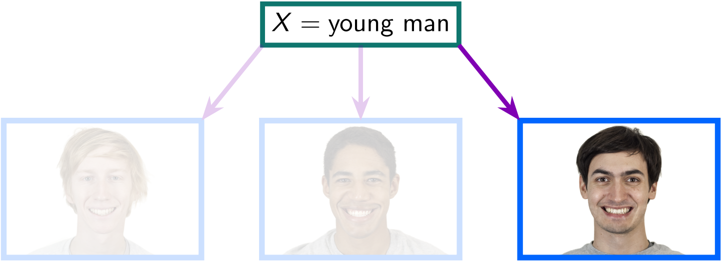
[1]Zubizarreta (2015), Chernozhukov et al. (2021)

Estimating nuisances

$$R_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\int \left[1(A_i = a^\star)\alpha_n(X_i)\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\} + \ell(\theta, \psi_n(u|X_i))\right]\Pi(du)$$

Two nuisances must be estimated:

1) **Inverse propensity:**[1] stable balancing weights, Riesz regression, logistic regression
2) **Outcome generative model:** conditional generative model, k-nearest neighbors



[1] Zubizarreta (2015), Chernozhukov et al. (2021)

Background and our objective

Oracle problem: all observations receive the intervention

Factual problem: some observations don't receive the intervention

Relationship to existing literature

Theoretical guarantees

Experiments

Discussion

Prior works on causal generative modeling

Iterative approaches: GANs (Kocaoglu et al., 2017), normalizing flows (Pawlowski et al., 2020), VAEs (Karimi et al., 2020), diffusion models (Chao et al., 2023)

Prior works on causal generative modeling

Covariates (X)

Iterative approaches: GANs (Kocaoglu et al., 2017), normalizing flows (Pawlowski et al., 2020), VAEs (Karimi et al., 2020), diffusion models (Chao et al., 2023)

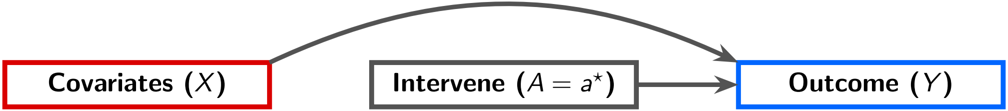
Prior works on causal generative modeling

Covariates (X)

Intervene ($A = a^\star$)

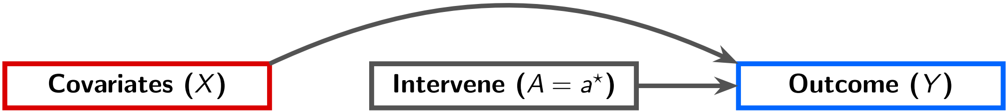
Iterative approaches: GANs (Kocaoglu et al., 2017), normalizing flows (Pawlowski et al., 2020), VAEs (Karimi et al., 2020), diffusion models (Chao et al., 2023)

Prior works on causal generative modeling



Iterative approaches: GANs (Kocaoglu et al., 2017), normalizing flows (Pawlowski et al., 2020), VAEs (Karimi et al., 2020), diffusion models (Chao et al., 2023)

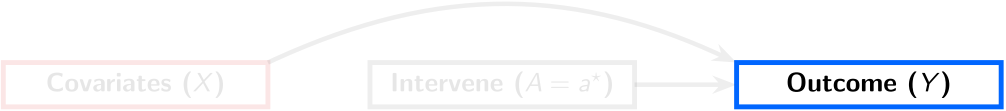
Prior works on causal generative modeling



Iterative approaches: GANs (Kocaoglu et al., 2017), normalizing flows (Pawlowski et al., 2020), VAEs (Karimi et al., 2020), diffusion models (Chao et al., 2023)

Joint generation: autoregressive flows (Khemakhem et al., 2021; Javaloy et al., 2024), variational graph autoencoders (Sanchez-Martin et al., 2021), diffusion models (Sanchez et al., 2022)

Prior works on causal generative modeling



Iterative approaches: GANs (Kocaoglu et al., 2017), normalizing flows (Pawlowski et al., 2020), VAEs (Karimi et al., 2020), diffusion models (Chao et al., 2023)

Joint generation: autoregressive flows (Khemakhem et al., 2021; Javaloy et al., 2024), variational graph autoencoders (Sanchez-Martin et al., 2021), diffusion models (Sanchez et al., 2022)

Direct approaches: inverse propensity weighting (Wu et al., 2024)

Contributions relative to existing causal generative modeling works

Unlike **DoubleGen**, existing approaches:

- Mostly only apply to **specific generative modeling paradigms**
- **Lack convergence guarantees**
- Are only **singly robust**

Prior works that use AIPW risk estimators

Conditional estimands

- **Average treatment effect (‘DR-learner’)** (van der Laan, 2006; van der Laan, 2013; Luedtke and van der Laan, 2016; Oprescu et al., 2019; Kennedy, 2023)
- **Classifier under selection bias** (Rotnitzky, Faraggi, et al., 2006)
- **Survival function** (Rubin et al., 2006)
- **Longitudinal mean** (Rotnitzky, Robins, et al., 2017; Luedtke, Sofrygin, et al., 2017)

General estimands

- **Ensemble learners** (van der Laan and Dudoit, 2003)
- **General learning algorithms** (Foster et al., 2023)
- **Stochastic gradient descent** (Yu et al., 2025)

Background and our objective

Oracle problem: all observations receive the intervention

Factual problem: some observations don't receive the intervention

Relationship to existing literature

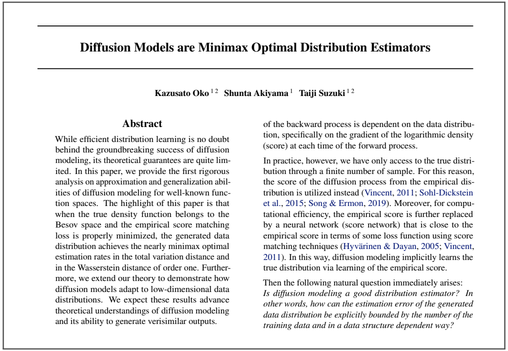
Theoretical guarantees

Experiments

Discussion

Main objective of theory

Make it as easy as possible to **port over existing results** from the generative modeling literature, with minimal modification



Diffusion Models are Minimax Optimal Distribution Estimators

Kazusato Oko [1 2] **Shunta Akiyama** [1] **Taiji Suzuki** [1 2]

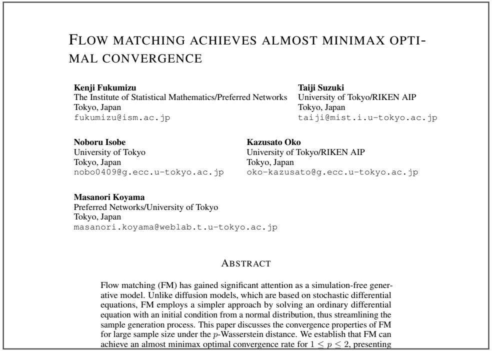
Abstract

While efficient distribution learning is no doubt behind the groundbreaking success of diffusion modeling, its theoretical guarantees are quite limited. In this paper, we provide the first rigorous analysis on approximation and generalization abilities of diffusion modeling for well-known function spaces. The highlight of this paper is that when the true density function belongs to the Besov space and the empirical score matching loss is properly minimized, the generated data distribution achieves the nearly minimax optimal estimation rates in the total variation distance and in the Wasserstein distance of order one. Furthermore, we extend our theory to demonstrate how diffusion models adapt to low-dimensional data distributions. We expect these results advance theoretical understandings of diffusion modeling and its ability to generate verisimilar outputs.

of the backward process is dependent on the data distribution, specifically on the gradient of the logarithmic density (score) at each time of the forward process.

In practice, however, we have only access to the true distribution through a finite number of sample. For this reason, the score of the diffusion process from the empirical distribution is utilized instead (Vincent, 2011; Sohl-Dickstein et al., 2015; Song & Ermon, 2019). Moreover, for computational efficiency, the empirical score is further replaced by a neural network (score network) that is close to the empirical score in terms of some loss function using score matching techniques (Hyvärinen & Dayan, 2005; Vincent, 2011). In this way, diffusion modeling implicitly learns the true distribution via learning of the empirical score.

Then the following natural question immediately arises: *Is diffusion modeling a good distribution estimator? In other words, how can the estimation error of the generated data distribution be explicitly bounded by the number of the training data and in a data structure dependent way?*



FLOW MATCHING ACHIEVES ALMOST MINIMAX OPTIMAL CONVERGENCE

Kenji Fukumizu
The Institute of Statistical Mathematics/Preferred Networks
Tokyo, Japan
fukumizu@ism.ac.jp

Taiji Suzuki
University of Tokyo/RIKEN AIP
Tokyo, Japan
taiji@mist.i.u-tokyo.ac.jp

Noboru Isobe
University of Tokyo
Tokyo, Japan
nobo0409@g.ecc.u-tokyo.ac.jp

Kazusato Oko
University of Tokyo/RIKEN AIP
Tokyo, Japan
oko-kazusato@g.ecc.u-tokyo.ac.jp

Masanori Koyama
Preferred Networks/University of Tokyo
Tokyo, Japan
masanori.koyama@weblab.t.u-tokyo.ac.jp

ABSTRACT

Flow matching (FM) has gained significant attention as a simulation-free generative model. Unlike diffusion models, which are based on stochastic differential equations, FM employs a simpler approach by solving an ordinary differential equation with an initial condition from a normal distribution, thus streamlining the sample generation process. This paper discusses the convergence properties of FM for large sample size under the p-Wasserstein distance. We establish that FM can achieve an almost minimax optimal convergence rate for $1 \leq p \leq 2$, presenting

Two-step analysis

Goal: Show the **true** and **estimated** counterfactual distributions are probably close:

$$\mathrm{Divergence}(\mathbb{P}, \mathbb{P}_{\theta_n}) \leq \mathrm{Rate}(n) \quad \text{w.h.p.}$$

Strategy:

1) **Bound divergence** by (transformation of) generalization error

$$\mathrm{GenError}(\theta) := \mathbb{E}_{\mathbb{P}}[\ell(\theta, Y^\star)] - \min_{\theta^\star} \mathbb{E}_{\mathbb{P}}[\ell(\theta^\star, Y^\star)]$$

2) **Bound generalization error**

Step 1: divergences already bounded in non-causal literature!

Prior works already showed that

$$\mathrm{Divergence}(\mathbb{P}, \mathbb{P}_\theta) \lesssim \mathrm{GenError}(\theta)^{\boldsymbol{b}} + \epsilon$$

Step 1: divergences already bounded in non-causal literature!

Prior works already showed that

$$\text{Divergence}(\mathbb{P}, \mathbb{P}_\theta) \lesssim \text{GenError}(\theta)^{b} + \epsilon$$

	Divergence	b	ϵ
Flow matching[1]	2-Wasserstein	$1/2$	0
Diffusion model[2]	Total variation	$1/2$	Trunc. error
Autoreg. language model[3]	KL divergence	1	0

[1]Benton et al. (2023), [2]Oko et al. (2023), [3]Definition of KL divergence

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n = \operatorname*{argmin}_{\theta \in \Theta} \mathrm{AIPW\,risk}(\theta; \alpha_n, \psi_n)$$

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n = \underset{\theta \in \Theta}{\text{argmin}} \underbrace{\mathrm{AIPW\,risk}(\theta; \alpha_n, \psi_n)}_{}$$

$$\frac{1}{n}\sum_{i=1}^{n} \int \left[1(A_i = a^\star)\alpha_n(X_i)\left\{\ell(\theta, Y_i) - \ell(\theta, \psi_n(u|X_i))\right\} + \ell(\theta, \psi_n(u|X_i))\right] \Pi(du)$$

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n \; = \; \operatorname*{argmin}_{\theta \in \Theta} \mathrm{AIPW\,risk}(\theta;\, \alpha_n,\, \psi_n)$$

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n = \operatorname*{argmin}_{\theta \in \Theta} \mathrm{AIPW\,risk}(\theta; \alpha_n, \psi_n)$$

Will relate it to one for the **oracle problem**:

$$\theta_n^\star = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \ell(\theta, Y_i^\star)$$

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n = \operatorname*{argmin}_{\theta \in \Theta} \mathrm{AIPW\,risk}(\theta;\, \alpha_n\,,\, \psi_n\,)$$

Will relate it to one for the **oracle problem**:

$$\theta_n^\star = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \ell(\theta, Y_i^\star)$$

Generalization bound (informal)**:** Under standard statistical learning conditions for the oracle problem, with probability at least $1 - 1/n$,

$$\mathrm{GenError}(\theta_n) \lesssim \inf_{\theta \in \Theta} \mathrm{GenError}(\theta) + \mathrm{Rate}(n, \mathrm{Size}(\Theta)) + \mathrm{Error}(\alpha_n)\,\mathrm{Error}(\psi_n)\ .$$

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n \;=\; \operatorname*{argmin}_{\theta\in\Theta} \mathrm{AIPW\,risk}(\theta;\, \alpha_n\,,\, \psi_n\,)$$

Will relate it to one for the **oracle problem**:

$$\theta_n^\star \;=\; \operatorname*{argmin}_{\theta\in\Theta} \tfrac{1}{n}\textstyle\sum_{i=1}^n \ell(\theta,\, Y_i^\star\,)$$

Generalization bound (informal)**:** Under standard statistical learning conditions for the oracle problem, with probability at least $1 - 1/n$,

$$\mathrm{GenError}(\theta_n) \lesssim \inf_{\theta\in\Theta} \mathrm{GenError}(\theta) + \mathrm{Rate}(n, \mathrm{Size}(\Theta)) + \mathrm{Error}(\alpha_n)\,\mathrm{Error}(\psi_n)\;\;.$$

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n \;=\; \underset{\theta\in\Theta}{\mathsf{argmin}}\, \mathrm{AIPW\,risk}(\theta;\, \alpha_n\,,\, \psi_n\,)$$

Will relate it to one for the **oracle problem**:

$$\theta_n^\star \;=\; \underset{\theta\in\Theta}{\mathsf{argmin}}\, \tfrac{1}{n}\textstyle\sum_{i=1}^n \ell(\theta,\, Y_i^\star\,)$$

Generalization bound (informal)**:** Under standard statistical learning conditions for the oracle problem, with probability at least $1 - 1/n$,

$$\mathrm{GenError}(\theta_n) \lesssim \inf_{\theta\in\Theta} \mathrm{GenError}(\theta) + \mathrm{Rate}(n, \mathrm{Size}(\Theta)) + \mathrm{Error}(\alpha_n)\,\mathrm{Error}(\psi_n)\;.$$

Leading terms match a **generalization bound for the oracle problem**.

Step 2: generalization bound

Provide generalization bound for empirical risk minimizer:

$$\theta_n = \operatorname*{argmin}_{\theta \in \Theta} \mathrm{AIPW\,risk}(\theta; \alpha_n, \psi_n)$$

Will relate it to one for the **oracle problem**:

$$\theta_n^\star = \operatorname*{argmin}_{\theta \in \Theta} \tfrac{1}{n} \textstyle\sum_{i=1}^n \ell(\theta, Y_i^\star)$$

Generalization bound (informal)**:** Under standard statistical learning conditions for the oracle problem, with probability at least $1 - 1/n$,

$$\mathrm{GenError}(\theta_n) \lesssim \inf_{\theta \in \Theta} \mathrm{GenError}(\theta) + \mathrm{Rate}(n, \mathrm{Size}(\Theta)) + \mathrm{Error}(\alpha_n)\,\mathrm{Error}(\psi_n)\,.$$

Leading terms match a **generalization bound for the oracle problem**.

Final term is **doubly robust**.

Total variation bound for DoubleGen diffusion models

Diffusion Models are Minimax Optimal Distribution Estimators

Kazusato Oko [1,2] **Shunta Akiyama** [1] **Taiji Suzuki** [1,2]

Following Oko et al., we give conditions under **DoubleGen diffusion models** with scores learned via a **neural network class** satisfy

$$\mathrm{TV}(\mathbb{P}, \mathbb{P}_{\theta_n}) \lesssim \log^{17/2}(n)\, n^{-\frac{s}{2s+d}} + \mathrm{Error}(\alpha_n)\, \mathrm{Error}(\psi_n)$$

with high probability.

Generating counterfactual smiling faces

	Lipstick	Makeup	Female	Earrings	No Beard	Blonde
Smiling	**56%**	**47%**	**65%**	**26%**	**88%**	**18%**
Not Smiling	38%	30%	52%	12%	79%	12%
Overall	**47%**	**38%**	**58%**	**19%**	**83%**	**15%**

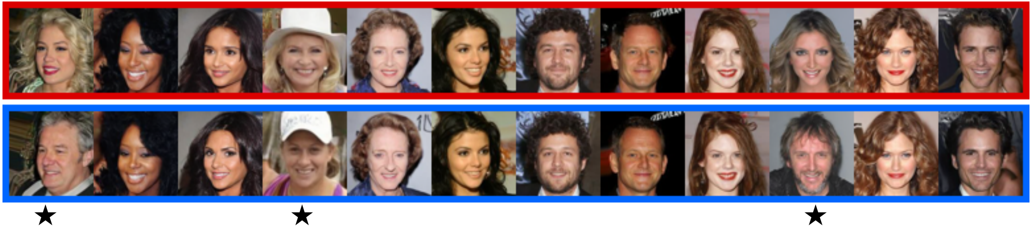
Trained two diffusion models with denoising score matching, using:

1) **Smiling instances** and a **standard loss**.

2) **All instances** and a **DoubleGen loss**.

Generating counterfactual smiling faces

	Lipstick	Makeup	Female	Earrings	No Beard	Blonde
Smiling	**56%**	**47%**	**65%**	**26%**	**88%**	**18%**
Not Smiling	38%	30%	52%	12%	79%	12%
Overall	**47%**	**38%**	**58%**	**19%**	**83%**	**15%**



Quantitative assessment: Fréchet and kernel ArcFace distances

		FAD ↓	KAD ↓
	Naïve	1.00	1.00
Both right	Plug-in	0.87	0.68
	IPW	0.88	0.71
	DoubleGen	**0.86**	**0.68**
Outcome wrong	Plug-in	1.90	2.17
	DoubleGen	**0.86**	**0.68**
Propensity wrong	IPW	0.93	0.71
	DoubleGen	**0.85**	**0.56**
Both wrong	DoubleGen	**1.01**	**0.79**

Quantitative assessment: Fréchet and kernel ArcFace distances

		FAD ↓	KAD ↓
	Naïve	1.00	1.00
Both right	Plug-in	0.87	0.68
	IPW	0.88	0.71
	DoubleGen	**0.86**	**0.68**
Outcome wrong	Plug-in	1.90	2.17
	DoubleGen	**0.86**	**0.68**
Propensity wrong	IPW	0.93	0.71
	DoubleGen	**0.85**	**0.56**
Both wrong	DoubleGen	**1.01**	**0.79**

DoubleGen typically

- **outperforms the naïve method** trained only with smiling instances

Quantitative assessment: Fréchet and kernel ArcFace distances

		FAD ↓	KAD ↓
	Naïve	1.00	1.00
Both right	Plug-in	0.87	0.68
	IPW	0.88	0.71
	DoubleGen	**0.86**	**0.68**
Outcome wrong	Plug-in	1.90	2.17
	DoubleGen	**0.86**	**0.68**
Propensity wrong	IPW	0.93	0.71
	DoubleGen	**0.85**	**0.56**
Both wrong	DoubleGen	**1.01**	**0.79**

DoubleGen typically

- **outperforms the naïve method** trained only with smiling instances
- **outperforms singly robust methods**

Generating counterfactual Amazon reviews (Hou et al., 2023)

Semi-synthetic experiment, with gold-standard counterfactual samples available from $\mathbb{P}$

- **Baseline covariates:** product category and other metadata
- **Intervention:** synthetic
  - lower propensity for some product categories: books, movies/TV, automotive

Generating counterfactual Amazon reviews (Hou et al., 2023)

Semi-synthetic experiment, with gold-standard counterfactual samples available from $\mathbb{P}$

- **Baseline covariates:** product category and other metadata
- **Intervention:** synthetic
  - lower propensity for some product categories: books, movies/TV, automotive
- **Outcome:** Amazon product review

```
5 stars: I am not sure what type of Keurig I have but this works great in
it! It sits up high enough so it does not get punctured like a regular
k-cup does.
```

```
3 stars: I am a big fan of Andrew Lloyd Webber's musicals. Cats contains
the very well-known song "Memory." Otherwise, there aren't many memorable
songs in this musical. It also is a revue, which means that there is no
real plot.
```

Autoregressive language model setup

We use **low-rank adaptation (LoRA)**[1] to finetune **Llama-3.2-1B**[2]

- 5.5M trainable parameters

[1]Hu et al. (2022), [2]Dubey et al. (2024)

Naïve approach and DoubleGen often generated similar reviews

5 stars: My son loves to use the game and can play for hours. Thanks for a fantastic app purchase!

5 stars: My son loves to use the game and can play for hours. Thanks for a fantastic game purchase.

5 stars: These are a must if you want to look great in a skirt. They are very durable. Will save me months and months of having to go buy new ones.

5 stars: These are a must if you want to look great in your shorts. They are very durable. Will save you money and time when it's time to order more.

Naïve approach underused the word 'book' (0.24% of reviews) DoubleGen used it with similar frequency as in test set (4.4%)

5 stars: This is amazing!!!! It's durable, easy to use, I love it and it came with all the batteries

5 stars: This book was amazing. The author took the time get to know and truly connect with both the characters.

3 stars: It's an OK quality mask. The design and the eye holes are nice. However, the straps on the back are not adjustable at all so it's hard to keep it on your face or to get the bottom part on straight.

3 stars: It's an OK book. The first and the last chapters are rather repetitive. The characters are interesting and likable.

Background and our objective

Oracle problem: all observations receive the intervention

Factual problem: some observations don't receive the intervention

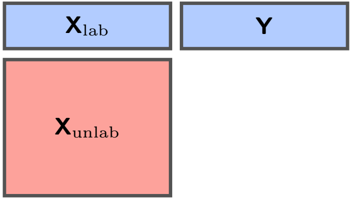
Relationship to existing literature

Theoretical guarantees

Experiments

Discussion

Equivalence with missing data problems



DoubleGen can be used to address outcomes missing at random:

- $A = a^\star \implies$ **outcome observed**
- $A \neq a^\star \implies$ **outcome missing**

AIPW risk estimator allows missing outcomes to be predicted by any algorithm

- E.g., a pretrained foundation model

Special case: MCAR outcomes from **prediction-powered inference***

*Angelopoulos et al. (2023)

Reduced-Entropy Sampling for Language Models

Language models often generate better text by **oversampling high-probability tokens***

Model then no longer targets counterfactual distribution $\mathbb{P}$

- Instead targets a lower-entropy variant

DoubleGen can still be used to estimate the transport maps for these schemes

*Caccia et al. (2018), Fan et al. (2018), Holtzman et al. (2019)

Extending DoubleGen to joint/conditional counterfactual sampling

Jointly with a subvector V of features X:

- Run the algorithm with modified outcome $Y' = (V, Y)$.

Conditionally on a subvector V of features X:

- Requires loss ℓ to depend on the condition v, as in text-to-image diffusion models*

In both cases, the analysis is nearly identical and yields a similar generalization bound.

*Saharia et al. (2022), Rombach et al. (2022)

Thank you!

References I

Angelopoulos, A. N. et al. (2023). "Prediction-powered inference". In: *Science* 382.6671, pp. 669–674.

Benton, J., G. Deligiannidis, and A. Doucet (2023). "Error bounds for flow matching methods". In: *arXiv preprint arXiv:2305.16860*.

Caccia, M. et al. (2018). "Language GANs falling short". In: *arXiv preprint arXiv:1811.02549*.

Chao, P. et al. (2023). "Modeling causal mechanisms with diffusion models for interventional and counterfactual queries". In: *arXiv preprint arXiv:2302.00860*.

Chernozhukov, V. et al. (2021). "Automatic debiased machine learning via Riesz regression". In: *arXiv preprint arXiv:2104.14737*.

Dubey, A. et al. (2024). "The llama 3 herd of models". In: *arXiv e-prints*, arXiv–2407.

Fan, A., M. Lewis, and Y. Dauphin (2018). "Hierarchical neural story generation". In: *arXiv preprint arXiv:1805.04833*.

Foster, D. J. and V. Syrgkanis (2023). "Orthogonal statistical learning". In: *The Annals of Statistics* 51.3, pp. 879–908.

References II

Graves, A. (2013). "Generating sequences with recurrent neural networks". In: *arXiv preprint arXiv:1308.0850*.

Holtzman, A. et al. (2019). "The curious case of neural text degeneration". In: *arXiv preprint arXiv:1904.09751*.

Hou, Y. et al. (2024). "Bridging Language and Items for Retrieval and Recommendation". In: *arXiv preprint arXiv:2403.03952*.

Hu, E. J. et al. (2022). "Lora: Low-rank adaptation of large language models.". In: *ICLR* 1.2, p. 3.

Javaloy, A., P. Sánchez-Martín, and I. Valera (2024). "Causal normalizing flows: from theory to practice". In: *Advances in Neural Information Processing Systems* 36.

Karimi, A.-H. et al. (2020). "Algorithmic recourse under imperfect causal knowledge: a probabilistic approach". In: *Advances in neural information processing systems* 33, pp. 265–277.

Kennedy, E. H. (2023). "Towards optimal doubly robust estimation of heterogeneous causal effects". In: *Electronic Journal of Statistics* 17.2, pp. 3008–3049.

References III

Khemakhem, I. et al. (2021). “Causal autoregressive flows”. In: *International conference on artificial intelligence and statistics*. PMLR, pp. 3520–3528.

Kocaoglu, M. et al. (2017). “Causalgan: Learning causal implicit generative models with adversarial training”. In: *arXiv preprint arXiv:1709.02023*.

Liu, Z. et al. (Dec. 2015). “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Luedtke, A. and K. Fukumizu (2025). “DoubleGen: Debiased Generative Modeling of Counterfactuals”. In: *arXiv preprint arXiv:2509.16842*.

Luedtke, A. R., O. Sofrygin, et al. (2017). “Sequential double robustness in right-censored longitudinal models”. In: *arXiv preprint arXiv:1705.02459*.

Luedtke, A. R. and M. J. van der Laan (2016). “Super-learning of an optimal dynamic treatment rule”. In: *The international journal of biostatistics* 12.1, pp. 305–332.

Ma, D. S., J. Correll, and B. Wittenbrink (2015). “The Chicago face database: A free stimulus set of faces and norming data”. In: *Behavior research methods* 47.4, pp. 1122–1135.

References IV

Oko, K., S. Akiyama, and T. Suzuki (2023). "Diffusion models are minimax optimal distribution estimators". In: *International Conference on Machine Learning*. PMLR, pp. 26517–26582.

Oprescu, M., V. Syrgkanis, and Z. S. Wu (2019). "Orthogonal random forest for causal inference". In: *International Conference on Machine Learning*. PMLR, pp. 4932–4941.

Pawlowski, N., D. Coelho de Castro, and B. Glocker (2020). "Deep structural causal models for tractable counterfactual inference". In: *Advances in neural information processing systems* 33, pp. 857–869.

Robins, J. M., A. Rotnitzky, and L. P. Zhao (1994). "Estimation of regression coefficients when some regressors are not always observed". In: *Journal of the American statistical Association* 89.427, pp. 846–866.

Rombach, R. et al. (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695.

References V

Rotnitzky, A., D. Faraggi, and E. Schisterman (2006). "Doubly robust estimation of the area under the receiver-operating characteristic curve in the presence of verification bias". In: *Journal of the American Statistical Association* 101.475, pp. 1276–1288.

Rotnitzky, A., J. Robins, and L. Babino (2017). "On the multiply robust estimation of the mean of the g-functional". In: *arXiv preprint arXiv:1705.08582*.

Rubin, D. and M. J. van der Laan (2006). "Doubly robust censoring unbiased transformations". In.

Saharia, C. et al. (2022). "Palette: Image-to-image diffusion models". In: *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10.

Sanchez, P. and S. A. Tsaftaris (2022). "Diffusion causal models for counterfactual estimation". In: *arXiv preprint arXiv:2202.10166*.

Sanchez-Martin, P., M. Rateike, and I. Valera (2021). "Vaca: Design of variational graph autoencoders for interventional and counterfactual queries". In: *arXiv preprint arXiv:2110.14690*.

van der Laan, M. J. (2013). "Targeted Learning of an Optimal Dynamic Treatment, and Statistical Inference for its Mean Outcome". In.

References VI

van der Laan, M. J. and S. Dudoit (2003). "Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples". In.

van der Laan, M. (2006). "Statistical inference for variable importance". In: *International Journal of Biostatistics* 2.1, Article–2.

VanderWeele, T. J. and M. A. Hernan (2013). "Causal inference under multiple versions of treatment". In: *Journal of causal inference* 1.1, pp. 1–20.

Wu, S. et al. (2024). "Counterfactual generative models for time-varying treatments". In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3402–3413.

Yu, F. et al. (2025). "Stochastic Gradients under Nuisances". In: *Advances in Neural Information Processing Systems*.

Zubizarreta, J. R. (2015). "Stable weights that balance covariates for estimation with incomplete outcome data". In: *Journal of the American Statistical Association* 110.511, pp. 910–922.

Extra Slides

Identification conditions

1) **Positivity:** $P(A = a^\star \mid X) > 0$ a.s.
2) **Ignorability:** $Y^\star \perp A \mid X$
3) **Consistency:** $Y = Y^\star$ whenever $A = A^\star$

Violation of conditions: multiple versions of treatment

If there are multiple versions of treatment, then **G-formula** instead identifies

- counterfactual distribution under a **stochastic intervention** (VanderWeele et al., 2013)