Targeted simulation–based inference for efficient posterior marginal estimation

Antoine Luciano (Université Paris-Dauphine), Ryan Giordano (University of California, Berkeley)
(Equal contribution joint first authors)

Efficient Approximate Bayesian Inference Workshop (BIRS, March 2025)

High–dimensional Bayesian causal inference [Ben-Michael et al., 2023]

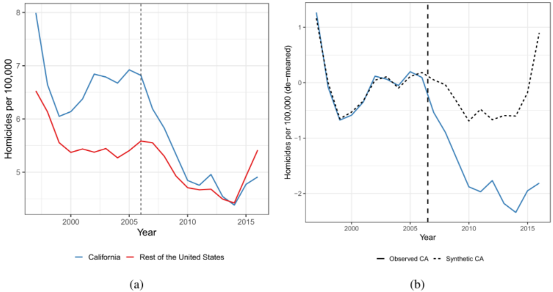


FIG. 1. *Annual homicide rate per* 100,000, 1997 *through* 2016. *The dotted line is* 2006, *the year the APPS program was launched.* (*a*) *California and the rest of the United States.* (*b*) *California and Synthetic California, matched based on the pre-intervention years* 1997–2006, *de-meaned.*

In 2006, California instituted the "Armed and Prohibited Persons System" (APPS) system to actively reclaim guns from prohibited carriers. **Did the APPS reduce homicides?**

High–dimensional Bayesian causal inference [Ben-Michael et al., 2023]

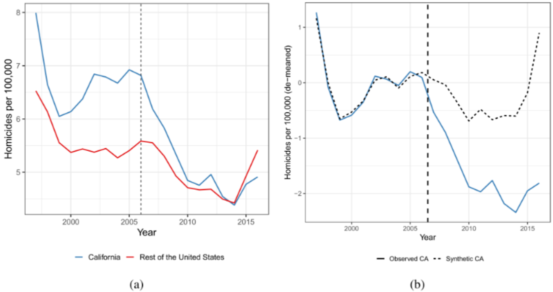


FIG. 1. *Annual homicide rate per* 100,000, 1997 *through* 2016. *The dotted line is* 2006, *the year the APPS program was launched. (a) California and the rest of the United States. (b) California and Synthetic California, matched based on the pre-intervention years* 1997–2006, *de-meaned.*

In 2006, California instituted the “Armed and Prohibited Persons System” (APPS) system to actively reclaim guns from prohibited carriers. **Did the APPS reduce homicides?**

Bayesian techniques can model counterfactuals [Oganisian and Roy, 2021, Li et al., 2023].

By modeling control–condition gun violence in all fifty states as a Gaussian process, we can produce a Bayesian belief about a “counterfactual California” [Ben-Michael et al., 2023].

High–dimensional Bayesian causal inference

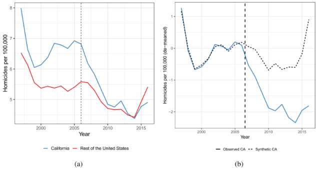


FIG. 1. *Annual homicide rate per* 100,000, 1997 *through* 2016. *The dotted line is* 2006, *the year the APPS program was launched. (a) California and the rest of the United States. (b) California and Synthetic California, matched based on the pre-intervention years* 1997–2006, *de-meaned.*

$$x = \text{Data (homicide counts for 50 states and 19 years)}$$

$$\theta = \text{Parameters (GP loadings and values, hyperparameters)}$$

$$\pi(\theta) = \text{Prior (complex but closed form density)}$$

$$\pi(x|\theta) = \text{Poisson likelihood given GP values}$$

$$\phi := f(\theta) = \text{Counterfactual effect of AAPS in CA}$$

High–dimensional Bayesian causal inference

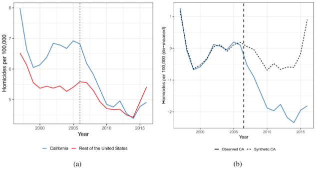


FIG. 1. *Annual homicide rate per* 100,000, 1997 *through* 2016. *The dotted line is* 2006, *the year the APPS program was launched. (a) California and the rest of the United States. (b) California and Synthetic California, matched based on the pre-intervention years* 1997–2006, *de-meaned.*

$$x = \text{Data (homicide counts for 50 states and 19 years)}$$

$$\theta = \text{Parameters (GP loadings and values, hyperparameters)}$$

$$\pi(\theta) = \text{Prior (complex but closed form density)}$$

$$\pi(x|\theta) = \text{Poisson likelihood given GP values}$$

$$\phi := f(\theta) = \text{Counterfactual effect of AAPS in CA}$$

Problem: We only really care about $\pi(\phi|x_{\text{obs}})$, a scalar posterior.

But θ is very high dimensional ($> 50 \times 19 \times K$ for K independent GP components)!

High–dimensional Bayesian causal inference

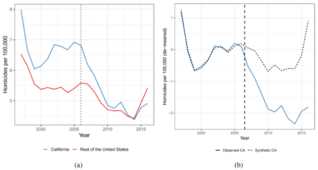


FIG. 1. *Annual homicide rate per* 100,000, 1997 *through* 2016. *The dotted line is* 2006, *the year the APPS program was launched. (a) California and the rest of the United States. (b) California and Synthetic California, matched based on the pre-intervention years* 1997–2006, *de-meaned.*

$$x = \text{Data (homicide counts for 50 states and 19 years)}$$

$$\theta = \text{Parameters (GP loadings and values, hyperparameters)}$$

$$\pi(\theta) = \text{Prior (complex but closed form density)}$$

$$\pi(x|\theta) = \text{Poisson likelihood given GP values}$$

$$\phi := f(\theta) = \text{Counterfactual effect of AAPS in CA}$$

Problem: We only really care about $\pi(\phi|x_{\text{obs}})$, a scalar posterior.

But θ is very high dimensional ($> 50 \times 19 \times K$ for K independent GP components)!

Almost all standard MCMC and VI techniques estimate the full $\pi(\theta|x)$ to get $\pi(\phi|x_{\text{obs}})$.

$\Rightarrow$ High complexity, high computational cost, slow model–building iteration. Hard to do good Bayesian causal inference! [Oganisian and Roy, 2021, Li et al., 2023]

Talk outline

View low–dimensional marginal estimation is a *likelihood–free inference problem*:

$$\pi(\phi, x_{\text{obs}}) = \underbrace{\int_{\theta : f(\theta) = \phi} \pi(\theta, x_{\text{obs}}) d\theta}_{\text{intractable}}$$

Talk outline

View low–dimensional marginal estimation is a *likelihood–free inference problem*:

$$\pi(\phi, x_{\text{obs}}) = \underbrace{\int_{\theta : f(\theta) = \phi} \pi(\theta, x_{\text{obs}}) d\theta}_{\text{intractable}}$$

- Contrast two very different existing techniques:
  - Approximate Bayesian computation (ABC)
  - Neural ratio estimation (NRE)
  - In practice, each depends on an arbitrary threshold parameter...in opposite ways.

Talk outline

View low–dimensional marginal estimation is a *likelihood–free inference problem*:

$$\pi(\phi, x_{\text{obs}}) = \underbrace{\int_{\theta: f(\theta)=\phi} \pi(\theta, x_{\text{obs}}) d\theta}_{\text{intractable}}$$

- Contrast two very different existing techniques:
  - Approximate Bayesian computation (ABC)
  - Neural ratio estimation (NRE)
  - In practice, each depends on an arbitrary threshold parameter...in opposite ways.
- **Our contribution:** NRE–ABC, a synthesis that stands to provide the best of both methods.

Method 1: Approximate Bayesian Computation (ABC)

Ingredients for likelihood–free inference:

- Observed data, x_{obs}.
- The ability to simulate $\phi, x \sim \pi(\phi, x)$:
  1. Draw $\theta \sim \pi(\theta)$ (high–dimensional)
  2. Draw $x \sim \pi(x|\theta)$
  3. Set $\phi = f(\theta)$ (scalar–valued)
  4. Only keep (ϕ, x) (discard θ)
- A thresholded "data similarity kernel," $K_\epsilon(x) = \mathbb{I}\left(\|x - x_{\text{obs}}\| \leq \epsilon\right)$

Method 1: Approximate Bayesian Computation (ABC)

Ingredients for likelihood–free inference:

- Observed data, x_{obs}.
- The ability to simulate $\phi, x \sim \pi(\phi, x)$:
  1. Draw $\theta \sim \pi(\theta)$ (high–dimensional)
  2. Draw $x \sim \pi(x|\theta)$
  3. Set $\phi = f(\theta)$ (scalar–valued)
  4. Only keep (ϕ, x) (discard θ)
- A thresholded "data similarity kernel," $K_\epsilon(x) = \mathbb{I}\left(\|x - x_{\text{obs}}\| \leq \epsilon\right)$

An ABC algorithm: [Tavaré et al., 1997, Beaumont et al., 2002]

Draw $(\phi_n, x_n) \overset{iid}{\sim} \pi(\phi, x)$, and keep only if $K_\epsilon(x_n) = 1$.

We use ϕ_n to approximate $\hat{\pi}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}}) \approx \pi^\star_{\text{ABC};\epsilon}(\phi|x_{\text{obs}}) \propto \int \pi(\phi, x) K_\epsilon(x) dx$.

Method 1: Approximate Bayesian Computation (ABC)

Ingredients for likelihood–free inference:

- Observed data, x_{obs}.
- The ability to simulate $\phi, x \sim \pi(\phi, x)$:
  1. Draw $\theta \sim \pi(\theta)$ (high–dimensional)
  2. Draw $x \sim \pi(x|\theta)$
  3. Set $\phi = f(\theta)$ (scalar–valued)
  4. Only keep (ϕ, x) (discard θ)
- A thresholded "data similarity kernel," $K_\epsilon(x) = \mathbb{I}\left(\|x - x_{\text{obs}}\| \leq \epsilon\right)$

An ABC algorithm: [Tavaré et al., 1997, Beaumont et al., 2002]

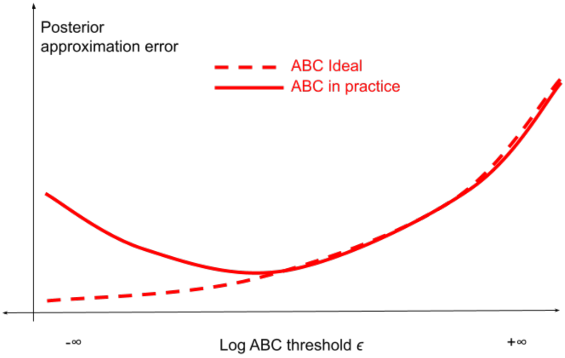
Draw $(\phi_n, x_n) \overset{iid}{\sim} \pi(\phi, x)$, and keep only if $K_\epsilon(x_n) = 1$.

We use ϕ_n to approximate $\hat{\pi}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}}) \approx \pi^\star_{\text{ABC};\epsilon}(\phi|x_{\text{obs}}) \propto \int \pi(\phi, x) K_\epsilon(x) dx$.

If threhsolding only keeps $x = x_{\text{obs}}$, then $\pi^\star_{\text{ABC};\epsilon}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$.

But in practice, $\pi^\star_{\text{ABC};\epsilon}(\phi|x_{\text{obs}}) \neq \pi(\phi|x_{\text{obs}})$ for any nonzero ϵ and practical norm $\|\cdot\|$.

Effect of thresholding on ABC



Method 2: Neural ratio estimation (NRE)

A NRE algorithm: [Cranmer et al., 2016, Hermans et al., 2020]

1. Repeat many times:
   1.1 Draw $\theta \sim \pi(\theta)$.
   1.2 Draw $y \sim \text{Bernoulli}(0.5)$.
   1.3 If $y = 1$:
      - Draw $x \sim \pi(x|\theta)$
   1.4 Else if $y = 0$:
      - Draw $\theta' \sim \pi(\theta')$
      - Draw $x \sim \pi(x|\theta')$
   1.5 Set $\phi = f(\theta)$.
   1.6 Keep only (ϕ, x) (discard θ)

Method 2: Neural ratio estimation (NRE)

A NRE algorithm: [Cranmer et al., 2016, Hermans et al., 2020]

1. Repeat many times:
    1.1 Draw $\theta \sim \pi(\theta)$.
    1.2 Draw $y \sim \text{Bernoulli}(0.5)$.
    1.3 If $y = 1$:
      - Draw $x \sim \pi(x|\theta)$
    1.4 Else if $y = 0$:
      - Draw $\theta' \sim \pi(\theta')$
      - Draw $x \sim \pi(x|\theta')$
    1.5 Set $\phi = f(\theta)$.
    1.6 Keep only (ϕ, x) (discard θ)
2. Train an ML classifier on the samples $((\phi, x), y)$ to learn the odds ratio $\hat{r}(\phi, x)$.

Method 2: Neural ratio estimation (NRE)

A NRE algorithm: [Cranmer et al., 2016, Hermans et al., 2020]

1. Repeat many times:
   1.1 Draw $\theta \sim \pi(\theta)$.
   1.2 Draw $y \sim \text{Bernoulli}(0.5)$.
   1.3 If $y = 1$:
      - Draw $x \sim \pi(x|\theta)$
   1.4 Else if $y = 0$:
      - Draw $\theta' \sim \pi(\theta')$
      - Draw $x \sim \pi(x|\theta')$
   1.5 Set $\phi = f(\theta)$.
   1.6 Keep only (ϕ, x) (discard θ)
2. Train an ML classifier on the samples $((\phi, x), y)$ to learn the odds ratio $\hat{r}(\phi, x)$.

Posterior estimator: $$\hat{\pi}_{\text{NRE}}(\phi|x_{\text{obs}}) \propto \overbrace{\pi(\phi)}^{\text{Known (or easily estimated)}} \underbrace{\hat{r}(\phi, x_{\text{obs}})}_{\text{Computable from classifier}}$$

Method 2: Neural ratio estimation (NRE)

A NRE algorithm: [Cranmer et al., 2016, Hermans et al., 2020]

1. Repeat many times:
   1.1 Draw $\theta \sim \pi(\theta)$.
   1.2 Draw $y \sim \text{Bernoulli}(0.5)$.
   1.3 If $y = 1$:
      - Draw $x \sim \pi(x|\theta)$
   1.4 Else if $y = 0$:
      - Draw $\theta' \sim \pi(\theta')$
      - Draw $x \sim \pi(x|\theta')$
   1.5 Set $\phi = f(\theta)$.
   1.6 Keep only (ϕ, x) (discard θ)
2. Train an ML classifier on the samples $((\phi, x), y)$ to learn the odds ratio $\hat{r}(\phi, x)$.

$$\text{Posterior estimator:} \quad \hat{\pi}_{\text{NRE}}(\phi|x_{\text{obs}}) \propto \overbrace{\pi(\phi)}^{\text{Known (or easily estimated)}} \underbrace{\hat{r}(\phi, x_{\text{obs}})}_{\text{Computable from classifier}}$$

Why does this make sense?

The optimal odds ratio is

$$r^\star(\phi, x) = \frac{\pi(\phi, x|y=1)\mathbb{P}(y=1)}{\pi(\phi, x|y=0)\mathbb{P}(y=0)} = \frac{\pi(\phi, x)}{\pi(\phi)\pi(x)} = \frac{\pi(\phi|x)}{\pi(\phi)}.$$

Therefore $\pi^\star_{\text{NRE}}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$.

Method 2: Neural ratio estimation (NRE)

A NRE algorithm: [Cranmer et al., 2016, Hermans et al., 2020]

1. Repeat many times:
   1.1 Draw $\theta \sim \pi(\theta)$.
   1.2 Draw $y \sim \text{Bernoulli}(0.5)$.
   1.3 If $y = 1$:
      - Draw $x \sim \pi(x|\theta)$
   1.4 Else if $y = 0$:
      - Draw $\theta' \sim \pi(\theta')$
      - Draw $x \sim \pi(x|\theta')$
   1.5 Set $\phi = f(\theta)$.
   1.6 Keep only (ϕ, x) (discard θ)
2. Train an ML classifier on the samples $((\phi, x), y)$ to learn the odds ratio $\hat{r}(\phi, x)$.

Posterior estimator: $$\hat{\pi}_{\text{NRE}}(\phi|x_{\text{obs}}) \propto \overbrace{\pi(\phi)}^{\text{Known (or easily estimated)}} \underbrace{\hat{r}(\phi, x_{\text{obs}})}_{\text{Computable from classifier}}$$

Why does this make sense?

The optimal odds ratio is

$$r^\star(\phi, x) = \frac{\pi(\phi, x|y=1)\mathbb{P}(y=1)}{\pi(\phi, x|y=0)\mathbb{P}(y=0)} = \frac{\pi(\phi, x)}{\pi(\phi)\pi(x)} = \frac{\pi(\phi|x)}{\pi(\phi)}.$$

Therefore $\pi^\star_{\text{NRE}}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$.

Method 2: Neural ratio estimation (NRE)

Problem: We have found that typically $\hat{r}(\phi, x_{\text{obs}}) \neq r^\star(\phi, x_{\text{obs}})$, even for "easy" posteriors.

Method 2: Neural ratio estimation (NRE)

Problem: We have found that typically $\hat{r}(\phi, x_{\text{obs}}) \neq r^\star(\phi, x_{\text{obs}})$, even for "easy" posteriors.

Why? Educated conjecture: ML algorithms come with *marginal guarantees*!

If the prior $\pi(\theta)$ is dispersed, then $\pi(x)$ is dispersed.

Most training data look very little like x_{obs}, and the classifier is trying to do well on all of them.

Method 2: Neural ratio estimation (NRE)

Problem: We have found that typically $\hat{r}(\phi, x_{\text{obs}}) \neq r^{\star}(\phi, x_{\text{obs}})$, even for "easy" posteriors.

Why? Educated conjecture: ML algorithms come with *marginal guarantees*!

If the prior $\pi(\theta)$ is dispersed, then $\pi(x)$ is dispersed.

Most training data look very little like x_{obs}, and the classifier is trying to do well on all of them.

Idea: Use the ABC simulations to learn an odds ratio $\hat{r}_{\epsilon}(\phi, x)$.

This should focus the classifier on (ϕ, x) that matter for computing $\pi(\phi|x_{\text{obs}})$.

Call the corresponding estimator $\pi^{\star}_{\text{NRE};\epsilon}(\phi|x_{\text{obs}})$.

Method 2: Neural ratio estimation (NRE)

Problem: We have found that typically $\hat{r}(\phi, x_{\text{obs}}) \neq r^\star(\phi, x_{\text{obs}})$, even for "easy" posteriors.

Why? Educated conjecture: ML algorithms come with *marginal guarantees*!

If the prior $\pi(\theta)$ is dispersed, then $\pi(x)$ is dispersed.

Most training data look very little like x_{obs}, and the classifier is trying to do well on all of them.

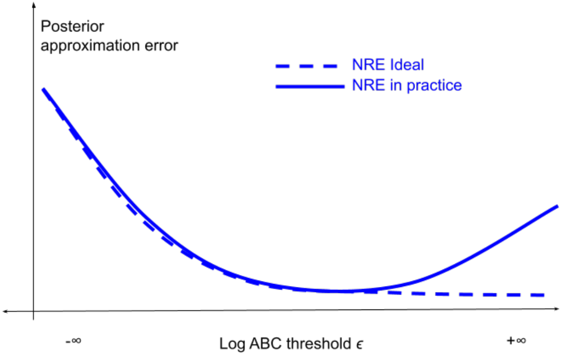
Idea: Use the ABC simulations to learn an odds ratio $\hat{r}_\epsilon(\phi, x)$.

This should focus the classifier on (ϕ, x) that matter for computing $\pi(\phi|x_{\text{obs}})$.

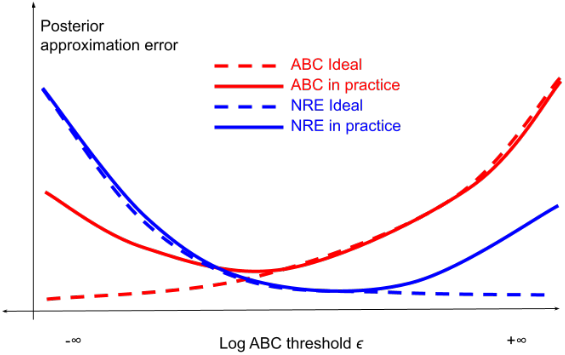
Call the corresponding estimator $\pi^\star_{\text{NRE};\epsilon}(\phi|x_{\text{obs}})$.

Problem: Censoring the training data means the classifier learns the wrong odds ratio. The smaller ϵ, the worse it is.

Thresholding has different effects



Thresholding has different effects



ABC-NRE

We propose the **NRE–ABC estimator**:

$$\hat{\pi}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) \propto \overbrace{\hat{\pi}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}})}^{\text{Estimate from ABC}} \underbrace{\hat{r}_{\epsilon}(\phi, x_{\text{obs}})}_{\text{Classifier trained on ABC samples}}$$

ABC-NRE

We propose the **NRE–ABC estimator**:

$$\hat{\pi}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) \propto \overbrace{\hat{\pi}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}})}^{\text{Estimate from ABC}} \underbrace{\hat{r}_{\epsilon}(\phi, x_{\text{obs}})}_{\text{Classifier trained on ABC samples}}$$

Our one theoretical result:

When the classifier is trained on the ABC samples, we show that $\pi^{\star}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$, irrespective of ϵ.

ABC-NRE

We propose the **NRE–ABC estimator**:

$$\hat{\pi}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) \propto \overbrace{\hat{\pi}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}})}^{\text{Estimate from ABC}} \underbrace{\hat{r}_{\epsilon}(\phi, x_{\text{obs}})}_{\text{Classifier trained on ABC samples}}$$

Our one theoretical result:

When the classifier is trained on the ABC samples, we show that $\pi^{\star}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$, irrespective of ϵ.

Our combined estimator can safely use *intermediate values of* ϵ for which *both* $\pi^{\star}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}})$ and $\pi^{\star}_{\text{NRE};\epsilon}(\phi|x_{\text{obs}})$ are biased, but for which $\pi^{\star}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$.

ABC-NRE

We propose the **NRE–ABC estimator**:

$$\hat{\pi}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) \propto \overbrace{\hat{\pi}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}})}^{\text{Estimate from ABC}} \underbrace{\hat{r}_{\epsilon}(\phi, x_{\text{obs}})}_{\text{Classifier trained on ABC samples}}$$

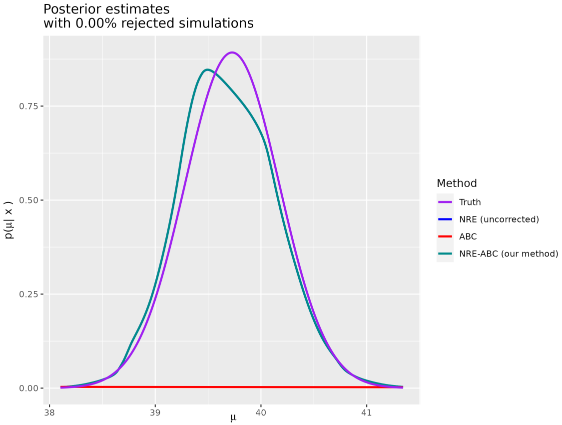
Our one theoretical result:

When the classifier is trained on the ABC samples, we show that $\pi^{\star}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$, irrespective of ϵ.

Our combined estimator can safely use *intermediate values of* ϵ for which *both* $\pi^{\star}_{\text{ABC};\epsilon}(\phi|x_{\text{obs}})$ and $\pi^{\star}_{\text{NRE};\epsilon}(\phi|x_{\text{obs}})$ are biased, but for which $\pi^{\star}_{\text{NRE–ABC};\epsilon}(\phi|x_{\text{obs}}) = \pi(\phi|x_{\text{obs}})$.

We can choose ϵ (and $\|\cdot\|$) to balance computational costs from simulation and classifier training, without worrying about finding a "sweet spot."

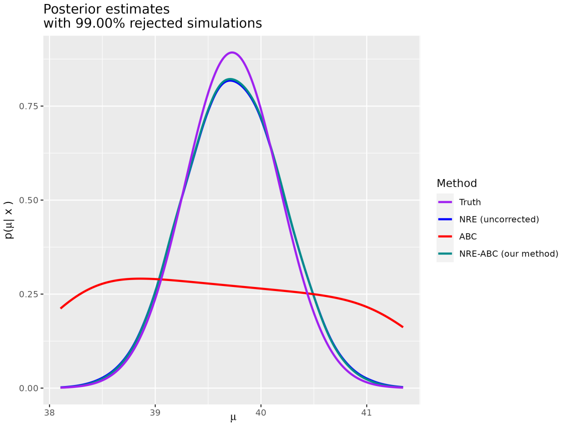
Univariate normal simulation



These plots show simulation results for the model $x_n \overset{iid}{\sim} \mathcal{N}(\mu, 1)$ with $N = 5$ data points.

We used the prior $\mu \sim \mathcal{N}\left(0, 20^2\right)$ and a true $\mu_0 = 40$.

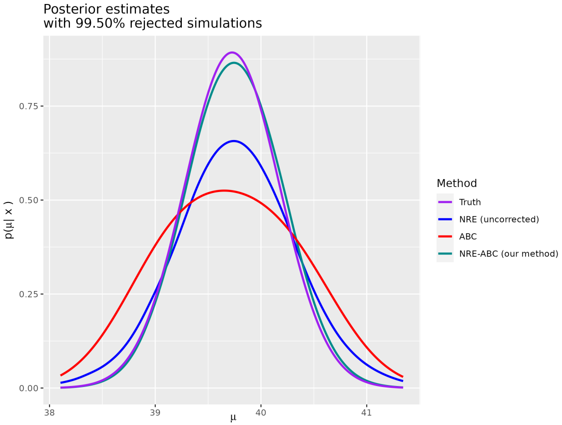
Univariate normal simulation



These plots show simulation results for the model $x_n \overset{iid}{\sim} \mathcal{N}(\mu, 1)$ with $N = 5$ data points.

We used the prior $\mu \sim \mathcal{N}\left(0, 20^2\right)$ and a true $\mu_0 = 40$.

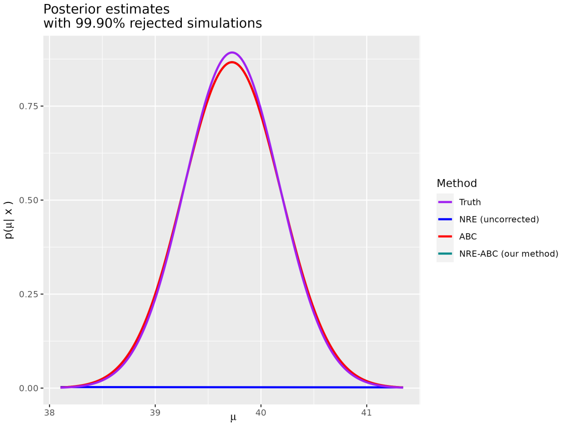
Univariate normal simulation



These plots show simulation results for the model $x_n \overset{iid}{\sim} \mathcal{N}(\mu, 1)$ with $N = 5$ data points.

We used the prior $\mu \sim \mathcal{N}\left(0, 20^2\right)$ and a true $\mu_0 = 40$.

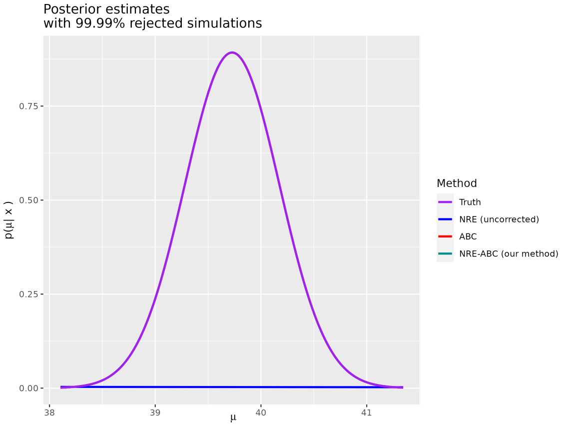
Univariate normal simulation



These plots show simulation results for the model $x_n \overset{iid}{\sim} \mathcal{N}(\mu, 1)$ with $N = 5$ data points.

We used the prior $\mu \sim \mathcal{N}\left(0, 20^2\right)$ and a true $\mu_0 = 40$.

Univariate normal simulation



These plots show simulation results for the model $x_n \overset{iid}{\sim} \mathcal{N}(\mu, 1)$ with $N = 5$ data points.

We used the prior $\mu \sim \mathcal{N}(0, 20^2)$ and a true $\mu_0 = 40$.

ABC-NRE

This is early work! There is a lot left to do.

- Run on higher–dimensional and real–world problems (ongoing)
- Compare with other truncation methods [Miller et al., 2021]
- Improve neural net architecture and systematically compare compute cost with MCMC
- Use ML to learn the ABC norm for thresholding
- Diagnostics with simulation–based calibration (SBC) [Talts et al., 2020].
  - Side note: improving the statistical power of SBC was the original motivation for this project!

Arxiv post coming soon!

References

M. Beaumont, W. Zhang, and D. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4): 2025–2035, 2002.

E. Ben-Michael, D. Arbour, A. Feller, A. Franks, and S. Raphael. Estimating the effects of a California gun control program with multitask Gaussian processes. *The Annals of Applied Statistics*, 17(2):985–1016, 2023.

K. Cranmer, J. Pavez, and G. Louppe. Approximating Likelihood Ratios with Calibrated Discriminative Classifiers, March 2016.

J. Hermans, V. Begy, and G. Louppe. Likelihood-free MCMC with Amortized Approximate Ratio Estimators. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4239–4248. PMLR, November 2020.

F. Li, P. Ding, and F. Mealli. Bayesian causal inference: A critical review. *Philosophical Transactions of the Royal Society A*, 381(2247):20220153, 2023.

B. Miller, A. Cole, P. Forré, G. Louppe, and C. Weniger. Truncated marginal neural ratio estimation. *Advances in Neural Information Processing Systems*, 34:129–143, 2021.

A. Oganisian and J. Roy. A practical introduction to Bayesian estimation of causal effects: Parametric and nonparametric approaches. *Statistics in medicine*, 40(2):518–551, 2021.

S. Talts, M. Betancourt, D. Simpson, A. Vehtari, and A. Gelman. Validating Bayesian Inference Algorithms with Simulation-Based Calibration, October 2020.

S. Tavaré, D. Balding, R. Griffiths, and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2): 505–518, 1997.