Julia Linhart
3rd year PhD Student at Inria Saclay (MIND)
Alexandre Gramfort and Pedro L.C. Rodrigues

PHYSTAT-SBI
Munich, May 15th, 2024

L-C2ST: Local Validation Diagnostics for Posterior Approximations in Simulation-Based Inference

arXiv > stat > arXiv:2306.03580

Statistics > Machine Learning

[Submitted on 6 Jun 2023]

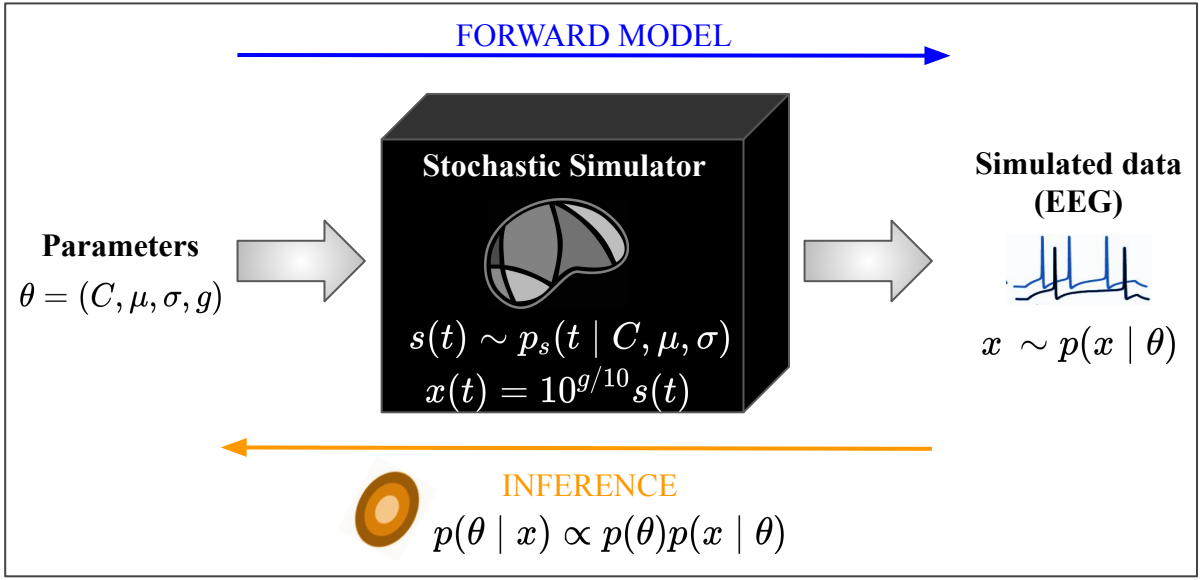
L-C2ST: Local Diagnostics for Posterior Approximations in Simulation-Based Inference

Julia Linhart, Alexandre Gramfort, Pedro L. C. Rodrigues

1. **Context & motivation**
2. **Method**
3. **Numerical illustration**

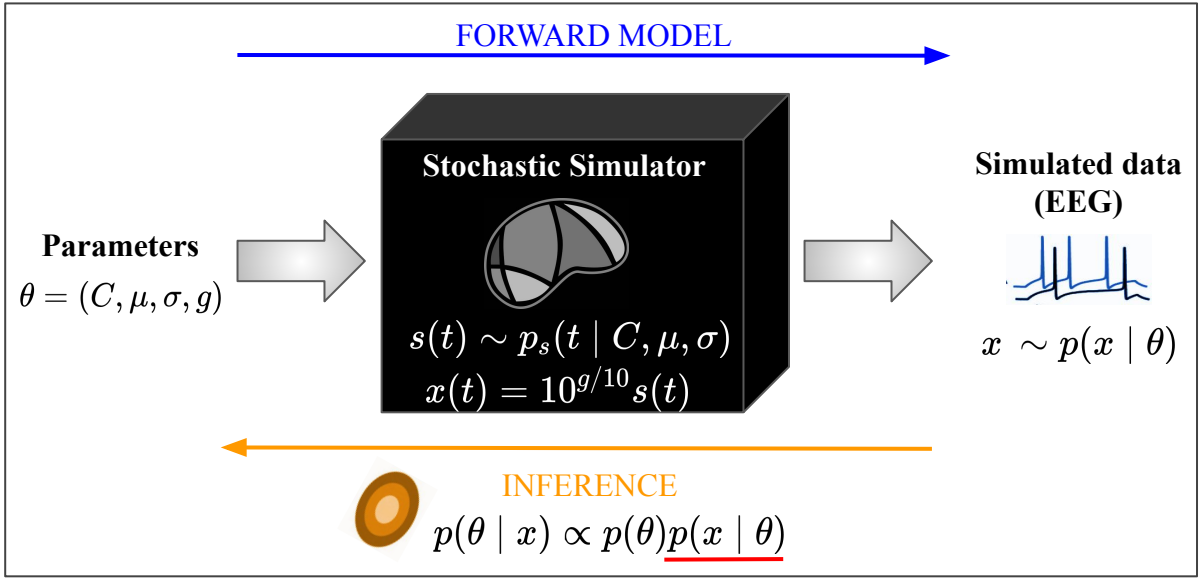
Inverting a simulator

➢ **GOAL:** understand and explain complex systems from experimental data *(population genetics, astrophysics, cosmology, neuroscience, ...)*



Inverting a simulator

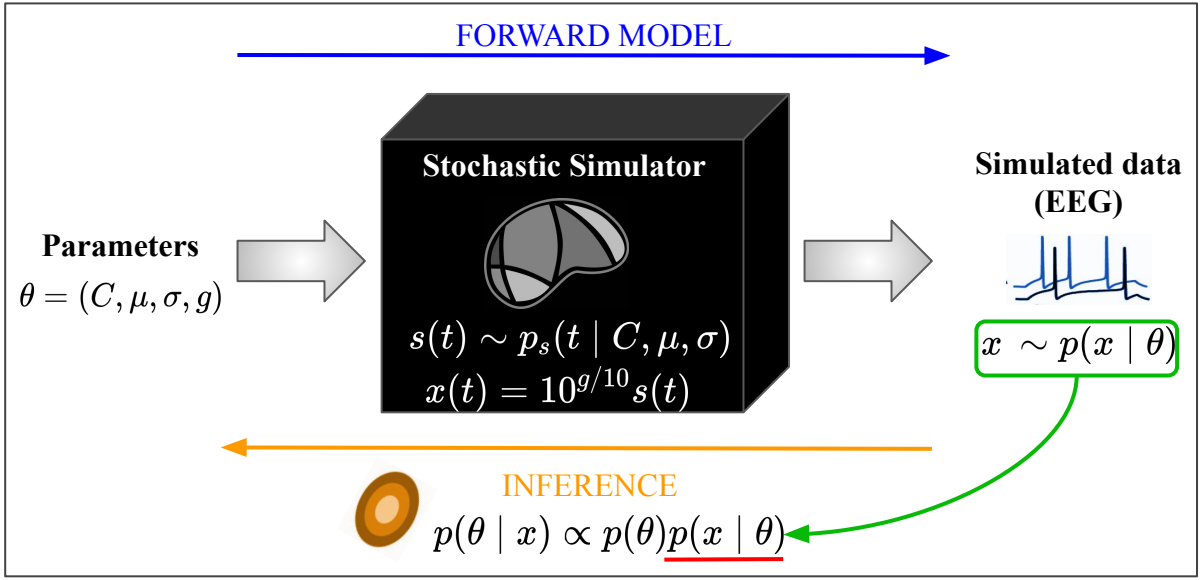
➢ **GOAL:** understand and explain complex systems from experimental data *(population genetics, astrophysics, cosmology, neuroscience, ...)*



✘ Intractable likelihood
→ NO MCMC

Inverting a simulator

➢ **GOAL:** understand and explain complex systems from experimental data *(population genetics, astrophysics, cosmology, neuroscience, ...)*

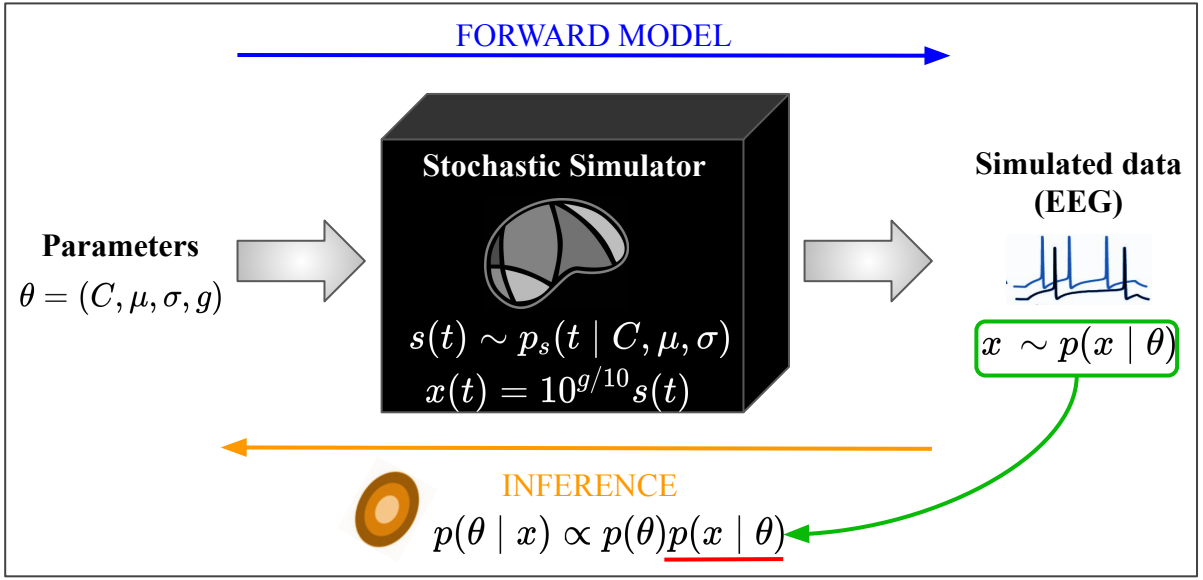


✗ Intractable likelihood → NO MCMC

✓ *Implicit* likelihood via simulations

Inverting a simulator

➢ **GOAL:** understand and explain complex systems from experimental data *(population genetics, astrophysics, cosmology, neuroscience, ...)*



✗ Intractable likelihood → NO MCMC

✓ *Implicit* likelihood via simulations

SBI: Simulation-Based Inference

SBI: Simulation-Based Inference (Cranmer et al., 2020)

Goal: estimate the posterior $p(\theta \mid x) \propto p(\theta)p(x \mid \theta)$ using simulated data from the joint p.d.f.

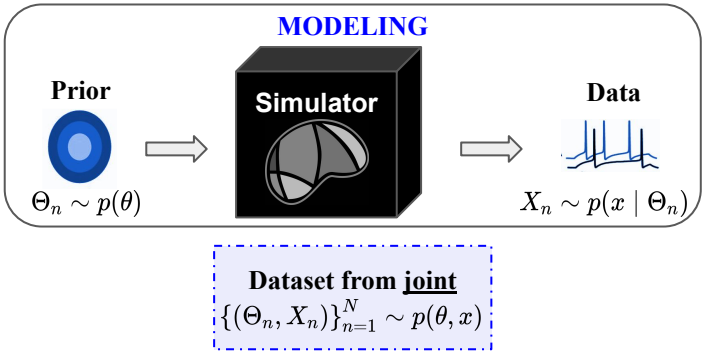


Figure 1: Typical SBI workflow.

SBI: Simulation-Based Inference (Cranmer et al., 2020)

Goal: estimate the posterior $p(\theta \mid x) \propto p(\theta)p(x \mid \theta)$ using simulated data from the joint p.d.f.

✓ Powerful inference algorithms using deep generative models (DGM) trained on data from the joint p.d.f.

> Neural Posterior Estimation (NPE, Greenberg et al., 2019) *with Normalizing Flows trained via maximum likelihood estimation.*
>
> $$\mathcal{L}^N(\phi) = -\frac{1}{N}\sum_{n=1}^{N} \log q_\phi(\Theta_n \mid X_n)$$

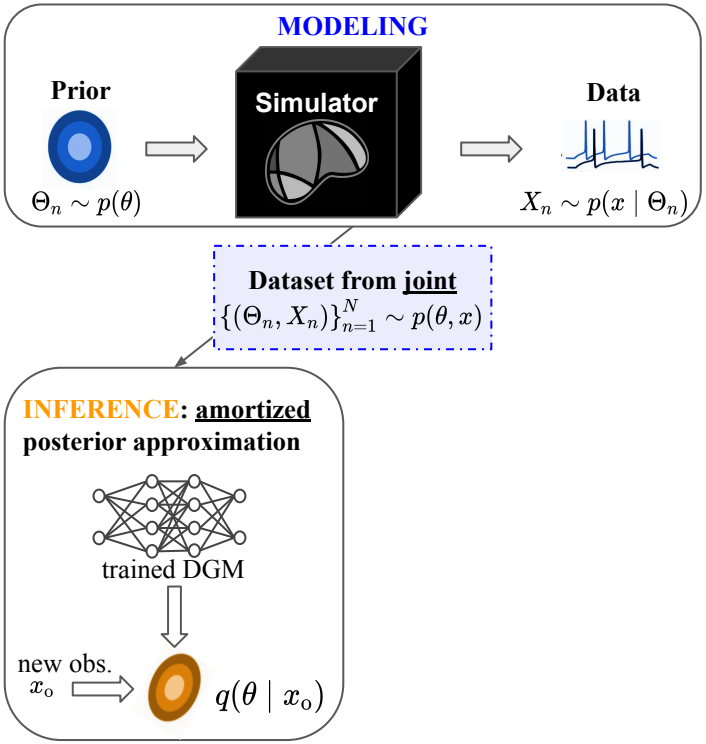


Figure 1: Typical SBI workflow.

SBI: Simulation-Based Inference (Cranmer et al., 2020)

Goal: estimate the posterior $p(\theta \mid x) \propto p(\theta)p(x \mid \theta)$ using simulated data from the joint p.d.f.

✓ Powerful inference algorithms using deep generative models (DGM) trained on data from the joint p.d.f.

✗ Validation is a challenge in *real scenarios* with unknown target posterior *(i.e. not benchmarks)*

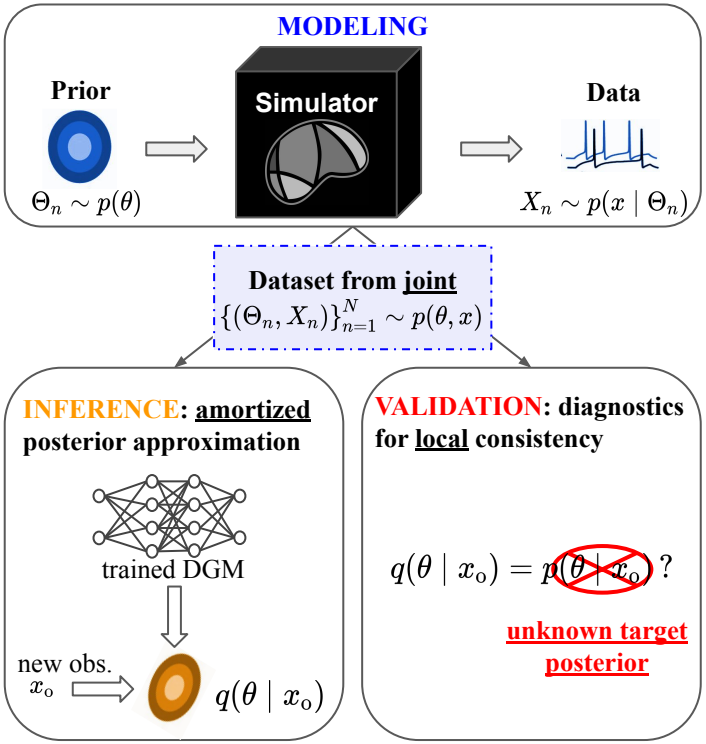
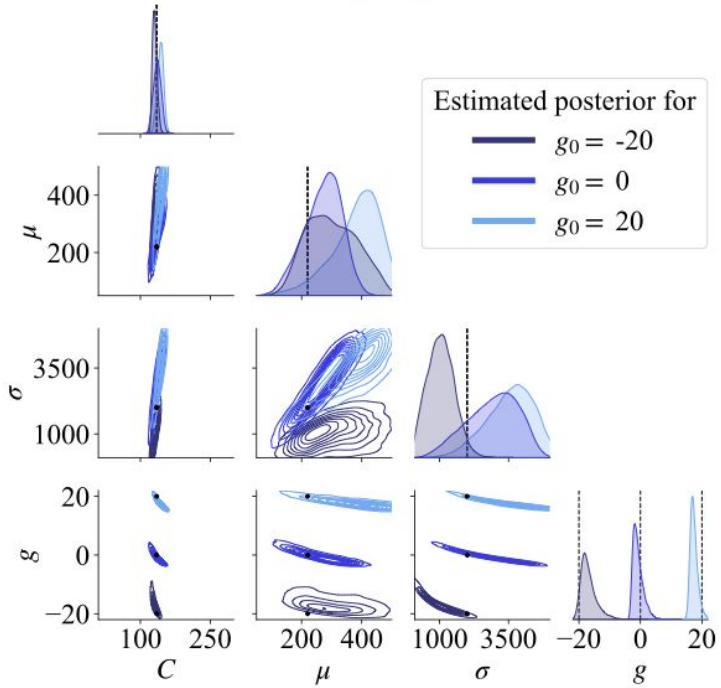


Figure 1: Typical SBI workflow.

Example in computational neuroscience

Inference of the JRNMM[1] posterior $p(\theta \mid x_o)$:

$$\theta = (C, \mu, \sigma, g) \in \mathbb{R}^4,\ x \in \mathbb{R}^{33}$$
$$x_o = \text{Simulator}(C_o, \mu_o, \sigma_o, g_o)$$
$$g_o \in [-10, 0, 10],\ \text{rest is fixed}$$



Neural Posterior Estimation (NPE) of the JRNMM[1] posterior. Pairplots of the inferred posteriors for three different observations simulated via the JRNMM with *varying gain parameter*.

[1](Jansen and Rit, 1995)

Example in computational neuroscience

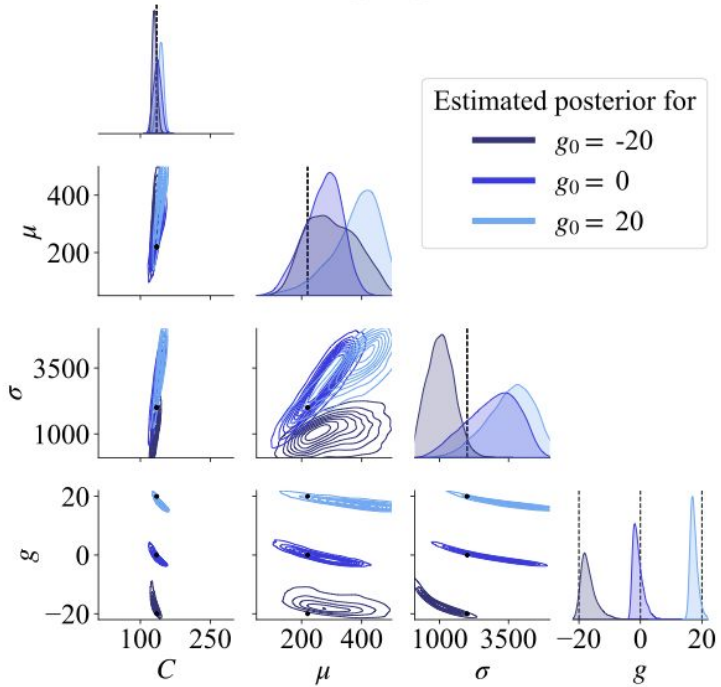
Inference of the JRNMM[1] posterior $p(\theta \mid x_o)$:

$$\theta = (C, \mu, \sigma, g) \in \mathbb{R}^4,\ x \in \mathbb{R}^{33}$$
$$x_o = \text{Simulator}(C_o, \mu_o, \sigma_o, g_o)$$
$$g_o \in [-10, 0, 10],\ \text{rest is fixed}$$

✓ Precise estimation for (C, g) when $g_o = 0$

✗ Small bias for (C, g) when $g_o \neq 0$

✗ Big variance and bias for (μ, σ)



Neural Posterior Estimation (NPE) of the JRNMM[1] posterior. Pairplots of the inferred posteriors for three different observations simulated via the JRNMM with *varying gain parameter*.

[1](Jansen and Rit, 1995)

Validating SBI algorithms

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at x***$_o$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = p(\theta \mid x_o), \quad \forall \theta \in \mathbb{R}^m .$$

Validating SBI algorithms **with classifiers**

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at*** $\boldsymbol{x_o}$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : \; q(\theta \mid x_o) = p(\theta \mid x_o), \quad \forall \theta \in \mathbb{R}^m \,. \tag{1}$$

Reformulation of (1) as a binary classification problem. *Consider two balanced classes:*

$$\Theta \mid (C = 0) \sim q(\theta \mid x_o) \quad vs. \quad \Theta \mid (C = 1) \sim p(\theta \mid x_o) \tag{2}$$

Theorem (Local consistency and binary classification). *The null hypothesis holds if, and only if, the optimal Bayes classifier cannot distinguish between q and p:*

$$\mathcal{H}_0(x_o) \text{ holds} \iff d^\star_{x_o}(\theta) = \mathbb{P}(C = 1 \mid \Theta = \theta; x_o) = \frac{1}{2}, \quad \forall \theta \in \mathbb{R}^m \,. \tag{3}$$

Validating SBI algorithms **with classifiers**

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at x_o*** *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = p(\theta \mid x_o), \quad \forall \theta \in \mathbb{R}^m . \tag{1}$$

Reformulation of (1) as a binary classification problem. *Consider two balanced classes:*

$$\Theta \mid (C = 0) \sim q(\theta \mid x_o) \quad vs. \quad \Theta \mid (C = 1) \sim p(\theta \mid x_o) \tag{2}$$

Theorem (Local consistency and binary classification). *The null hypothesis holds if, and only if, the optimal Bayes classifier cannot distinguish between q and p:*

$$\mathcal{H}_0(x_o) \text{ holds} \iff d^\star_{x_o}(\theta) = \mathbb{P}(C = 1 \mid \Theta = \theta; x_o) = \frac{1}{2}, \quad \forall \theta \in \mathbb{R}^m . \tag{3}$$

Proof. The optimal Bayes classifier predicts the class with highest probability according to

$$d^\star_{x_o}(\theta) = \mathbb{P}(C = 1 \mid \theta; x_o) = \frac{p(\theta \mid x_o)}{p(\theta \mid x_o) + q(\theta \mid x_o)}, \quad \mathbb{P}(C = 0 \mid \Theta = \theta; x_o) = 1 - d^\star_{x_o}(\theta) .$$

Validating SBI algorithms **with classifiers**

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at*** $\boldsymbol{x_o}$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = p(\theta \mid x_o), \quad \forall \theta \in \mathbb{R}^m . \tag{1}$$

Reformulation of (1) as a binary classification problem. *Consider two balanced classes:*

$$\Theta \mid (C = 0) \sim q(\theta \mid x_o) \quad vs. \quad \Theta \mid (C = 1) \sim p(\theta \mid x_o)$$

Classifier Two-Sample Tests (C2ST) (Lopez-Paz et al., 2016). Hypothesis test for distribution equality (1) based on the *accuracy* of a binary classifier trained on equally as many samples from q and p.

- ➢ currently the most *powerful and flexible* approach (scales to high-dimensional, non-Euclidean data)
- ➢ many applications: statistical independence and sample quality tests, noise contrastive estimation, GANs, density ratio estimation algorithms (e.g. NRE, Hermans et al., 2021), etc.
- ➢ used to *benchmark SBI-algorithms* on toy-models (Lueckmann et al., 2021)

Validating SBI algorithms **with classifiers**

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at*** $\boldsymbol{x_o}$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = \cancel{p(\theta \mid x_o)}, \quad \forall \theta \in \mathbb{R}^m . \tag{1}$$

unknown target posterior

Reformulation of (1) as a binary classification problem. *Consider two balanced classes:*

$$\Theta \mid (C = 0) \sim q(\theta \mid x_o) \quad vs. \quad \cancel{\Theta \mid (C = 1) \sim p(\theta \mid x_o)}$$

Classifier Two-Sample Tests (C2ST) (Lopez-Paz et al., 2016). Hypothesis test for distribution equality (1) based on the *accuracy* of a binary classifier trained on equally as many samples from q and p.

- ➢ currently the most *powerful and flexible* approach (scales to high-dimensional, non-Euclidean data)
- ➢ many applications: statistical independence and sample quality tests, noise contrastive estimation, GANs, density ratio estimation algorithms (e.g. NRE, Hermans et al., 2021), etc.
- ➢ used to *benchmark SBI-algorithms* on toy-models (Lueckmann et al., 2021)

Validating SBI algorithms

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at x***$_o$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = \cancel{p(\theta \mid x_o)}, \quad \forall \theta \in \mathbb{R}^m .$$

unknown target posterior

Validating SBI algorithms

Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at x***$_o$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = \cancel{p(\theta \mid x_o)}, \quad \forall \theta \in \mathbb{R}^m .$$

unknown target posterior

✓ Standard approaches (SBC) use data from the joint and evaluate the estimator ***in expectation*** over x

$$\mathbb{E}_x[q(\theta \mid x) = p(\theta \mid x)]$$

Validating SBI algorithms

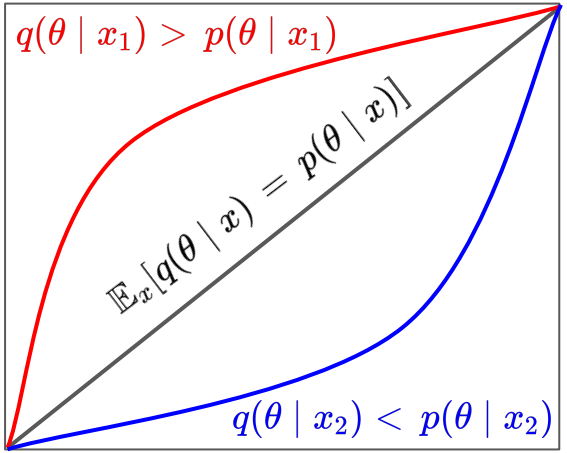
Definition (Local consistency). *A conditional density estimator* ***q*** *is* ***locally consistent at*** $\boldsymbol{x_o}$ *with the true posterior* ***p*** *if, and only if, the following null hypothesis holds:*

$$\mathcal{H}_0(x_o) : q(\theta \mid x_o) = \cancel{p(\theta \mid x_o)}, \quad \forall \theta \in \mathbb{R}^m .$$

unknown target posterior

✓ Standard approaches (SBC) use data from the joint and evaluate the estimator ***in expectation*** over x

✗ ***No local evaluation***
→ *possibly false conclusions, less interpretable*



Validating SBI algorithms

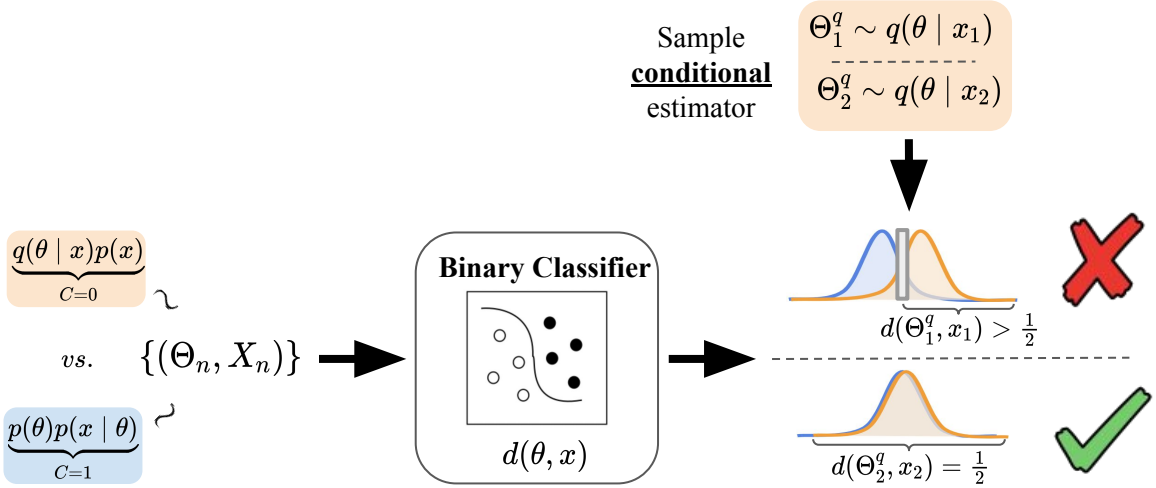
- **C2ST is useless** in *real* SBI applications: requires samples from the true posterior !

- Standard approaches (SBC, Talts et al., 2018) evaluate the estimator *in expectation* over x
  - ✓ use data from the joint
  - ✗ *no local evaluation: possibly false conclusions, less interpretable*

- Attempts to *make them local: local*-HPD (Zhao et al., 2021), *local*-multi-PIT (Linhart et al., 2022)
  - ✓ use data from the joint
  - ✓ allow for local evaluation at any observation
  - ✗ *not practical: many hyperparameters, computationally too expensive*

Make C2ST local using only data from the joint!

L-C2ST: **Local** Classifier Two-Sample Tests



→ no need for true posterior samples
→ powerful and efficient
→ necessary and *sufficient conditions* for local consistency
→ easy to implement and interpret

L-C2ST: **Local** Classifier Two-Sample Tests

Theorem (Local consistency and single class evaluation). *If the classifier* $\boldsymbol{d}$ *is* optimal *and* $N_v \to +\infty$,

$$\hat{t}_{\mathrm{MSE}_0} = \frac{1}{N_v} \sum_{n=1}^{N_v} \left(d(\Theta_n^q, x_\mathrm{o}) - \frac{1}{2} \right)^2 = 0, \quad \Theta_n^q \sim q(\theta \mid x_\mathrm{o})$$

is a necessary and sufficient condition for the local consistency of $\boldsymbol{q}$ *at* x_o.

L-C2ST: **Local** Classifier Two-Sample Tests

Theorem (Local consistency and single class evaluation). *If the classifier* $\boldsymbol{d}$ *is* optimal *and* $N_v \to +\infty$,

$$\hat{t}_{\mathrm{MSE}_0} = \frac{1}{N_v}\sum_{n=1}^{N_v}\left(d(\Theta_n^q, x_o) - \frac{1}{2}\right)^2 = 0, \qquad \Theta_n^q \sim q(\theta \mid x_o)$$

is a necessary and sufficient condition for the local consistency of $\boldsymbol{q}$ *at* x_o.

Proof. Let d be a classifier trained to distinguish between the joint distributions. It converges to the optimal Bayes classifier:

$$d^\star(\theta, x) = \mathbb{P}(C = 1 \mid \theta; x) = \frac{p(\theta, x)}{p(\theta, x) + q(\theta \mid x)p(x)} = \frac{p(\theta \mid x)\cancel{p(x)}}{p(\theta \mid x)\cancel{p(x)} + q(\theta \mid x)p(x)} = d_x^\star(\theta)\,. \tag{4}$$

Let x_o be a fixed observation. For $\Theta_n^q \sim q(\theta \mid x_o)$, we have that

$$\hat{t}_{\mathrm{MSE}_0} = \frac{1}{N_v}\sum_{n=1}^{N_v}\left(d_{x_o}^\star(\Theta_n^q) - \frac{1}{2}\right)^2 \xrightarrow[N_v \to \infty]{} \int \left(d_{x_o}^\star(\theta) - \frac{1}{2}\right)^2 q(\theta \mid x_o)\mathrm{d}\theta\,. \tag{5}$$

q being a p.d.f. we can conclude

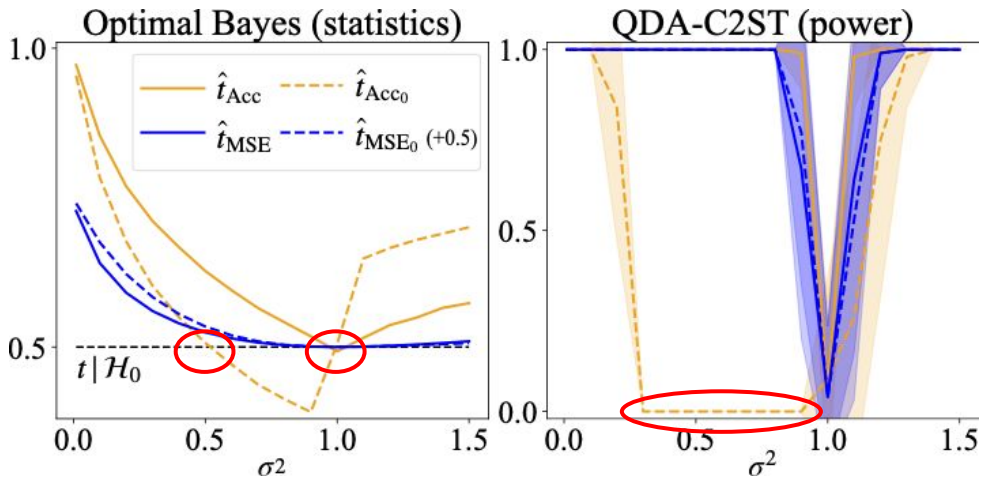
$$\boxed{\lim_{N_v \to \infty} \hat{t}_{\mathrm{MSE}_0}(f^\star, x_o) = 0 \iff d_{x_o}^\star(\theta) = \frac{1}{2}, \forall\theta \underbrace{\iff}_{(3)} \mathcal{H}_0(x_o) \text{ holds} : q(\theta \mid x_o) = p(\theta \mid x_o), \forall\theta}$$

L-C2ST: **Local** Classifier Two-Sample Tests

Theorem (Local consistency and single class evaluation). *If the classifier* $\boldsymbol{d}$ *is* optimal *and* $N_v \to +\infty$,

$$\hat{t}_{\mathrm{MSE}_0} = \frac{1}{N_v} \sum_{n=1}^{N_v} \left(d(\Theta_n^q, x_o) - \frac{1}{2} \right)^2 = 0, \quad \Theta_n^q \sim q(\theta \mid x_o)$$

is a necessary and sufficient condition for the local consistency of $\boldsymbol{q}$ *at* x_o.



$p = \mathcal{N}(0, \mathbf{I}_2) \quad vs. \quad q = \mathcal{N}(0, \sigma^2 \mathbf{I}_2)$

$\hat{t}_{\mathrm{MSE}_0}$: single-class MSE
✓ close to *oracle*
$p = q$ only at $\sigma^2 = 1$

$\hat{t}_{\mathrm{Acc}_0}$: single class accuracy
✗ different from *oracle*
insufficient for $p = q$
→ fails to detect when $p \neq q$

L-C2ST: **Local** Classifier Two-Sample Tests

Algorithm 1: ℓ-C2ST – training the classifier on data from the joint distribution

Input: posterior estimator q; calibration data $\mathcal{D}_{\text{cal}} = \{\Theta_n, X_n\}_{n=1}^{N_{\text{cal}}}$; classifier f
Output: estimate d of the class probabilities

```
/* Construct classification training set
```
for $n = 1, \ldots, N_{\text{cal}}$ **do**
  $\Theta_n^q \sim q(\theta \mid X_n)$
  $W_{2n} = (\Theta_n^q, X_n)$; $C_{2n} = 0$ /* Sample from $q(\theta \mid x)p(x)$
  $W_{2n+1} = (\Theta_n, X_n)$; $C_{2n+1} = 1$ /* Sample from $p(\theta, x)$
$\mathcal{D} \leftarrow \{W_n, C_n\}_{n=1}^{2N_{\text{cal}}}$

```
/* Get estimate d of the class probabilities
```
Train the classifier f on $\mathcal{D}$
$d \leftarrow f_{\text{probability}}$
return d

Algorithm 2: ℓ-C2ST – evaluating the test statistic for any x_o

Input: Observation x_o; estimate d obtained in Algorithm 3
Output: test statistic $\hat{t}_{\text{MSE}_0}(x_o)$

```
/* Evaluate the classifier
```
Generate N_v samples $\Theta_n^q \sim q(\theta \mid x_o)$
Get predicted probabilities predicted probabilities $d(\Theta_n^q, x_o)$

```
/* Compute the test statistic
```
$\hat{t}_{\text{MSE}_0}(x_o) \leftarrow \frac{1}{N_v} \sum_n \left(d(\Theta_n^q, x_o) - \frac{1}{2}\right)^2$
return $\hat{t}_{\text{MSE}_0}(x_o)$

➢ Requires sampling from the approximation (can be hard and depends on x)

“better” version for **Normalizing Flows** !

Experiment 1: Benchmark example

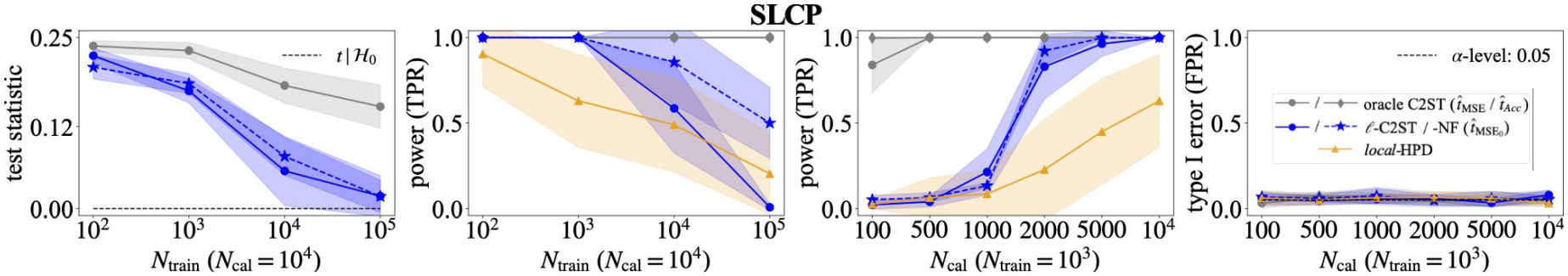


Figure 3: Power analysis on SLCP from `sbibm`[1] *(N_v=10 000 and 100 null trials).*

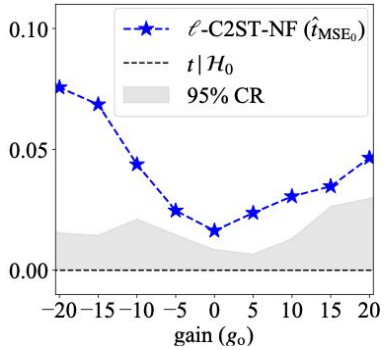
$\boxed{\theta \in \mathbb{R}^5,\ x \in \mathbb{R}^8}$

- ✓ Test statistic reflects convergence of NPE (plot 1)
- ✓ Type I error is controlled at 0.05 (plot 4)
- ✓ ℓ-C2ST(**-NF**) is more powerful than *local*-HPD (plots 2,3)

[1] https://github.com/sbi-benchmark/sbibm

Experiment 2: a *real* neuroscience simulator

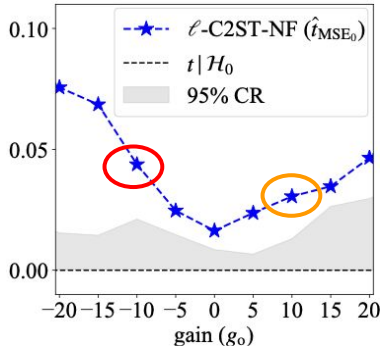
$\theta = (C, \mu, \sigma, g) \in \mathbb{R}^4$, $x \in \mathbb{R}^{33}$
$x_o = \text{Simulator}(C_o, \mu_o, \sigma_o, g_o)$
$g_o \in [-20, 20]$, rest is fixed



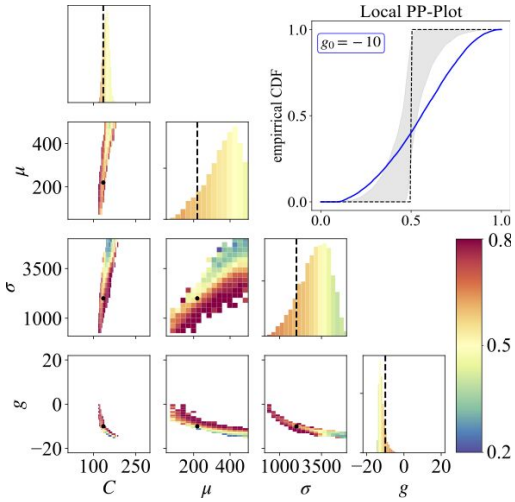
local behavior

Experiment 2: a *real* neuroscience simulator

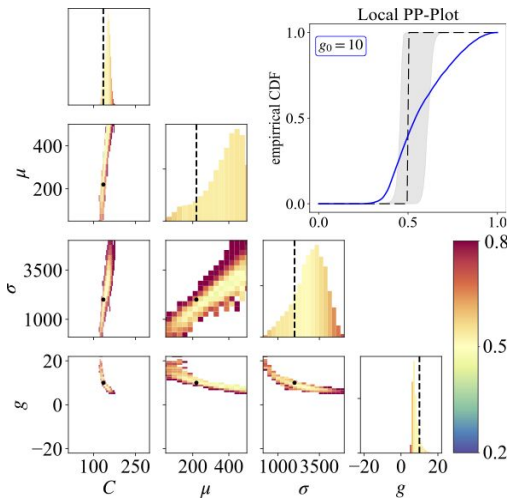
$\theta = (C, \mu, \sigma, g) \in \mathbb{R}^4$, $x \in \mathbb{R}^{33}$
$x_o = \text{Simulator}(C_o, \mu_o, \sigma_o, g_o)$
$g_o \in [-20, 20]$, rest is fixed



local behavior



BIAS



OVERDISPERSION

Conclusion

L-C2ST: Local Diagnostics for Posterior Approximations in Simulation-Based Inference

★ *local evaluation* at any given observation using *only data from the joint*
★ necessary and *sufficient conditions* for local consistency
★ *easy to implement* and *interpretable* diagnostics: *where and how does my estimator fail ?*

Paper: https://arxiv.org/abs/2306.03580 Code: https://github.com/JuliaLinhart/lc2st

Experimental results:

1. (sbi benchmark) → *powerful and efficient* validation method, win over competitors
2. (neuroscience application) → interesting *insights on local failure-modes* of SBI algorithms

Perspectives: *Calibration* of the classifier, *local correction* of the posterior estimator

References

Pedro Luiz Coelho Rodrigues, Thomas Moreau, Gilles Louppe, and Alexandre Gramfort. HNPE: Leveraging Global Parameters for Neural Posterior Estimation. In *NeurIPS 2021*, Sydney (Online), Australia, December 2021. URL https://hal.science/hal-03139916.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences (PNAS)*, 117:30055–30062, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1912789117.

Ben H. Jansen and Vincent G. Rit. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological Cybernetics 1995 73:4*, 73:357–366, 9 1995. ISSN 1432-0770. doi: 10.1007/BF00199471.

Julia Linhart, Alexandre Gramfort, and Pedro L. C. Rodrigues. Validation diagnostics for SBI algorithms based on normalizing flows, 2022.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *5th International Conference on Learning Representations, ICLR 2017*, 10 2016. doi: 10.48550/arxiv.1610.06545.

References

Jan-Matthis Lueckmann, Jan Boelts, David S Greenberg, Pedro J Gonçalves, and Jakob H Macke. Benchmarking simulation-based inference. *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (PMLR)*, 130:343–351, 4 2021. doi: 10.48550/arxiv.2101.04653.

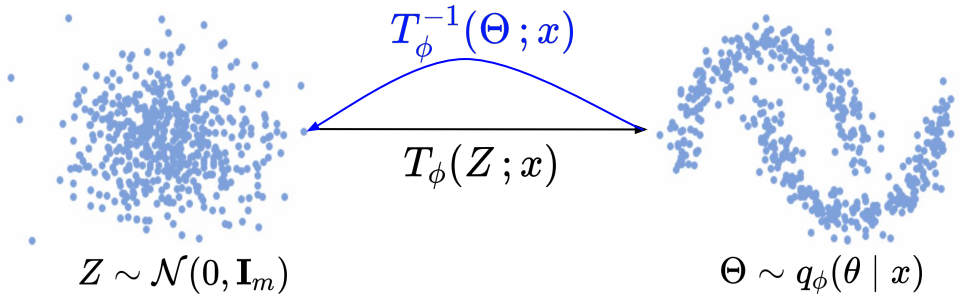
Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating bayesian inference algorithms with simulation-based calibration. 4 2018. doi: 10.48550/arxiv.1804.06788.

David Zhao, Niccolò Dalmasso, Rafael Izbicki, and Ann B. Lee. Diagnostics for conditional density models and bayesian inference algorithms. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1830–1840. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/zhao21b.html.

Appendix 1: L-C2ST-**NF**

Definition (Normalizing Flows, Papamakarios et. al, 2021). *A normalizing flow* q_ϕ *is a conditional density estimator that (for every x) defines a map between a* Gaussian base distribution $u(z) = \mathcal{N}(0, \mathbf{I}_m)$ *and any complex target distribution* $p(\theta \mid x)$ *via a* bijective transformation $T_\phi(.\,; x)$ *with Jacobian* $J_{T_\phi}(.\,; x)$*:*

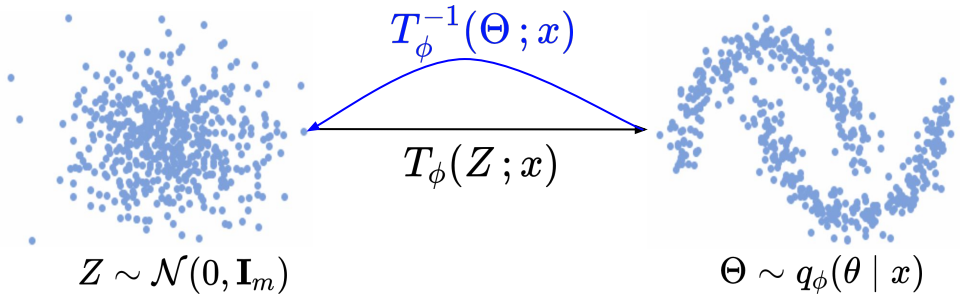
$$q_\phi(\theta \mid x) = u(z)\left|\det J_{T_\phi^{-1}}(\theta; x)\right|, \quad z = T_\phi^{-1}(\theta\,; x), \quad \forall \theta \in \mathbb{R}^m \tag{6}$$



Appendix 1: L-C2ST-**NF**

Theorem (Local consistency and normalizing flows). *Given a posterior approximation q_ϕ based on a normalizing flow, its local consistency at x_o can be characterized by*

$$\mathcal{H}_0(x_o) : \; q_\phi(\theta \mid x_o) = p(\theta \mid x_o) \iff \mathcal{N}(0, \mathbf{I}_m) = p\Big(T_\phi^{-1}(\theta\,; x) \mid x_o\Big), \quad \forall \theta \in \mathbb{R}^m$$



Appendix 1: L-C2ST-**NF**

Theorem (Local consistency and normalizing flows). *Given a posterior approximation q_ϕ based on a normalizing flow, its local consistency at x_o can be characterized by*

$$\mathcal{H}_0(x_o) : q_\phi(\theta \mid x_o) = p(\theta \mid x_o) \iff \mathcal{N}(0, \mathbf{I}_m) = p\Big(T_\phi^{-1}(\theta\,; x) \mid x_o\Big), \quad \forall \theta \in \mathbb{R}^m$$

<u>New binary classification task:</u>

$$(Z, X) \mid (C = 0) \sim \mathcal{N}(0, \mathbf{I}_m)p(x) \quad vs. \quad (Z, X) \mid (C = 1) \sim p\Big(T_\phi^{-1}(\theta; x), x\Big)$$

- The Gaussian distribution is *independent of x and q_ϕ*!
- More *efficient and exact hypothesis test*: the null distribution can be precomputed on large datasets and re-used for every new estimator of the same task *(see Algorithm 4 in Appendix 2)*

Appendix 1: L-C2ST-**NF**

Algorithm 1: ℓ-C2ST – training the classifier on data from the joint distribution

Input: posterior estimator q; calibration data $\mathcal{D}_{\text{cal}} = \{\Theta_n, X_n\}_{n=1}^{N_{\text{cal}}}$; classifier f

Output: estimate d of the class probabilities

```
/* Construct classification training set
for n = 1, ..., N_cal do
    Θ_n^q ~ q(θ | X_n)
    W_2n = (Θ_n^q, X_n); C_2n = 0 /* Sample from q(θ | x)p(x)
    W_2n+1 = (Θ_n, X_n); C_2n+1 = 1 /* Sample from p(θ, x)
D ← {W_n, C_n}_{n=1}^{2N_cal}
/* Get estimate d of the class probabilities
Train the classifier f on D
d ← f_probability
return d
```

```
/* Construct classification training set
for n in 1, ..., N_cal do
    Z_n ~ N(0, I_m); Z_n^q = T_φ^{-1}(Θ_n; X_n) /* inverse NF-transformation
    W_2n = (Z_n, X_n); C_2n = 0
    W_2n+1 = (Z_n^q, X_n); C_2n+1 = 1
```

- ➢ Does not require sampling from the approximation
- ➢ Does not depend on x under null

Appendix 2: Algorithms (L-C2ST)

Algorithm 1: ℓ-C2ST – training the classifier on data from the joint distribution

Input: posterior estimator q; calibration data $\mathcal{D}_{\text{cal}} = \{\Theta_n, X_n\}_{n=1}^{N_{\text{cal}}}$; classifier f; number of samples $N_{\mathcal{H}}$ from the distribution under the null hypothesis

Output: estimate d of the class probabilities; estimates $\{d_1, \ldots, d_{N_{\mathcal{H}}}\}$ under the null hypothesis

```
/* Construct classification training set                                  */
```

for $n = 1, \ldots, N_{\text{cal}}$ **do**

- $\Theta_n^q \sim q(\theta \mid X_n)$
- $W_{2n} = (\Theta_n^q, X_n)$; $C_{2n} = 0$ `/* Sample from` $q(\theta \mid x)p(x)$ `*/`
- $W_{2n+1} = (\Theta_n, X_n)$; $C_{2n+1} = 1$ `/* Sample from` $p(\theta, x)$ `*/`

$\mathcal{D} \leftarrow \{W_n, C_n\}_{n=1}^{2N_{\text{cal}}}$

```
/* Get estimate d of the class probabilities                              */
```

Train the classifier f on $\mathcal{D}$

$d \leftarrow f_{\text{probability}}$

```
/* Estimate d under the null hypothesis via permutation procedure         */
```

for $h = 1, \ldots N_{\mathcal{H}}$ **do**

- Randomly permute labels C_n in $\mathcal{D}$
- Train the classifier f on new $\mathcal{D}$
- $d_h \leftarrow f_{\text{probability}}$

return d; $\{d_1, \ldots, d_{N_{\mathcal{H}}}\}$

Algorithm 2: ℓ-C2ST – evaluating test statistics and p-values for any x_o

Input: Observation x_o; estimates d and $\{d_1, \ldots, d_{N_{\mathcal{H}}}\}$ obtained in Algorithm 1

Output: test statistic $\hat{t}_{\text{MSE}_0}(x_o)$; p-value $\hat{p}(x_o)$

Generate N_v samples $\Theta_n^q \sim q(\theta \mid x_o)$ with predicted probabilities $d(\Theta_n^q, x_o)$ and $d_h(\Theta_n^q, x_o)$

```
/* Compute test statistics                                                */
```

$\hat{t}_{\text{MSE}_0}(x_o) \leftarrow \frac{1}{N_v} \sum_n \left(d(\Theta_n^q, x_o) - \frac{1}{2}\right)^2$

for $h = 1, \ldots, N_{\mathcal{H}}$ **do**

- $\hat{t}_h(x_o) \leftarrow \frac{1}{N_v} \sum_n \left(d_h(\Theta_n^q, x_o) - \frac{1}{2}\right)^2$

```
/* Compute p-value                                                        */
```

$\hat{p}(x_o) \leftarrow \frac{1}{N_{\mathcal{H}}} \sum_h \mathbb{I}\left(\hat{t}_h(x_o) > \hat{t}_{\text{MSE}_0}(x_o)\right)$

return $\hat{t}_{\text{MSE}_0}(x_o)$, $\hat{p}(x_o)$

Appendix 2: Algorithms (L-C2ST-NF)

Algorithm 3: ℓ-C2ST-NF – training the classifier on the joint distribution

Input: NF posterior estimator q_ϕ; calibration data $\mathcal{D}_{\text{cal}} = \{\Theta_n, X_n\}_{n=1}^{N_{\text{cal}}}$; classifier f
Output: estimate d of the class probabilities
`/* Construct classification training set */`
for n *in* $1, \ldots, N_{\text{cal}}$ **do**
  $Z_n \sim \mathcal{N}(0, I_m)$; $Z_n^q = T_\phi^{-1}(\Theta_n; X_n)$ `/* inverse NF-transformation */`
  $W_{2n} = (Z_n, X_n)$; $C_{2n} = 0$
  $W_{2n+1} = (Z_n^q, X_n)$; $C_{2n+1} = 1$
$\mathcal{D} \leftarrow \{W_n, C_n\}_{n=1}^{2N_{\text{cal}}}$
`/* Get estimate` d `of the class probabilities */`
Train the classifier f on $\mathcal{D}$
$d \leftarrow f_{\text{probability}}$
return d

Algorithm 4: ℓ-C2ST-NF – precompute the null distribution for any estimator

Input: calibration data $\mathcal{D}_{\text{cal}} = \{\Theta_n, X_n\}_{n=1}^{N_{\text{cal}}}$; classifier f; number of null samples $N_{\mathcal{H}}$
Output: estimates $\{d_1, \ldots, d_{N_{\mathcal{H}}}\}$ of the class probabilities under the null
for h *in* $1, \ldots N_{\mathcal{H}}$ **do**
  `/* Construct classification training set */`
  Sample $Z_n \sim \mathcal{N}(0, I_m)$ for $n = 1, \ldots, 2N_{\text{cal}}$
  $\mathcal{D} \leftarrow \{(Z_{2n}, X_n), 0\}_{n=1}^{N_{\text{cal}}} \cup \{(Z_{2n+1}, X_n), 1\}_{n=1}^{N_{\text{cal}}}$
  `/* Get estimate` d `of the class probabilities */`
  Train the classifier f on $\mathcal{D}$
  $d_h \leftarrow f_{\text{probability}}$
return $\{d_1, \ldots, d_{N_{\mathcal{H}}}\}$

Appendix 2: Algorithms (PP-plots with L-C2ST)

Algorithm 5: ℓ-C2ST – local PP-plots for any x_o

Input: evaluation data set $\mathcal{D}_{v0}$; an observation x_o; estimate d of the class probabilities; estimates $\{d_1, \ldots, d_{N_\mathcal{H}}\}$ under the null; grid $\mathcal{G}$ of PP-levels in (0,1); significance level α

Output: empirical CDF-values $\{\hat{F}(l; x_o)\}_{l \in \mathcal{G}}$ of predicted class-0 probabilities; $(1-\alpha)$-confidence bands $\{L_l(x_o), U_l(x_o)\}_{l \in \mathcal{G}}$

Predict class-0 probabilities $\{d_0(v, x_0) = 1 - d(v, x_o)\}_{v \in \mathcal{D}_{v0}}$ `/* d is an estimate of class 1 */`

for l *in* $\mathcal{G}$ **do**

- `/* Compute empirical CDFs at` l `*/`
- $\hat{F}(l; x_o) \leftarrow \frac{1}{N_{v0}} \sum \mathbb{I}_{d_0(v, x_0) \leq l}$
- **for** $h = 1, \ldots, N_\mathcal{H}$ **do**
  - $\hat{F}_h(l; x_o) \leftarrow \frac{1}{N_{v0}} \sum \mathbb{I}_{d_0(v, x_0) \leq l}$
- `/* Compute confidence bands at` l `*/`
- $L_l(x_o), U_l(x_o) \leftarrow q_{\frac{\alpha}{2}}(\{\hat{F}_h(l; x_o)\}_{h=1}^{N_\mathcal{H}}), q_{1-\frac{\alpha}{2}}(\{\hat{F}_h(l; x_o)\}_{h=1}^{N_\mathcal{H}})$ `/* quantiles */`

return $\{\hat{F}(l; x_o)\}_{l \in \mathcal{G}}, L_l(x_o), U_l(x_o)\}_{l \in \mathcal{G}}$

Appendix 3: Benchmark Examples (Lueckmann et al., 2021)

SLCP

A challenging inference task designed to have a simple likelihood and a complex posterior. The prior is uniform over five parameters $\boldsymbol{\theta}$ and the data are a set of four two-dimensional points sampled from a Gaussian likelihood whose mean and variance are nonlinear functions of $\boldsymbol{\theta}$:

Prior	$\mathcal{U}(-\mathbf{3}, \mathbf{3})$
Simulator	$\mathbf{x}\|\boldsymbol{\theta} = (\mathbf{x}_1, \ldots, \mathbf{x}_4),\ \mathbf{x}_i \sim \mathcal{N}(\mathbf{m}_{\boldsymbol{\theta}}, \mathbf{S}_{\boldsymbol{\theta}}),$ where $\mathbf{m}_{\boldsymbol{\theta}} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix},\ \mathbf{S}_{\boldsymbol{\theta}} = \begin{bmatrix} s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2 \end{bmatrix},\ s_1 = \theta_3^2, s_2 = \theta_4^2, \rho = \tanh \theta_5$
Dimensionality	$\boldsymbol{\theta} \in \mathbb{R}^5, \mathbf{x} \in \mathbb{R}^8$
References	Papamakarios et al. (2019b); Greenberg et al. (2019); Hermans et al. (2020) Durkan et al. (2020)

Appendix 4: Runtimes for Experiment 1 (BENCHMARK)

	$N_{\text{cal}} = 5\ 000$				$N_{\text{cal}} = 10\ 000$			
N_{train}	10^2	10^3	10^4	10^5	10^2	10^3	10^4	10^5
oracle C2ST	**5.47**	**4.52**	5.95	7.56	**16.36**	**18.03**	23.3	15.33
ℓ-C2ST	5.92	5.09	**1.78**	1.81	27.62	34.06	**17.9**	**3.65**
ℓ-C2ST-NF	6.98	6.84	6.38	**1.72**	43.99	25.01	18.56	7.62
local-HPD	282.19	282.85	279.38	282.5	956.91	938.21	682.92	530.45

Table 1: Run-time (in seconds) to compute the test statistic for the `SLCP` task (mean over observations). C2ST has close to constant run-time for fixed N_{cal}. Local methods become faster with increasing N_{train} and ℓ-C2ST(-NF) stays comparable to the *oracle* C2ST, even for $N_{\text{cal}} = 10\ 000$. While the amortization cost of ℓ-C2ST is not an issue, *local*-HPD is always at least 30 times slower.

Appendix 5: On the Cross-Entropy loss of L-C2ST

The theoretical cross-entropy loss function to distinguish between data (Θ, X) from class $C = 1$ and class $C = 0$ is defined by

$$l_{\mathrm{CE}}(d) := -\frac{1}{2}\mathbb{E}_{(\Theta,X)|C=1}\left[\log\left(d(\Theta, X)\right)\right] - \frac{1}{2}\mathbb{E}_{(\Theta,X)|C=0}\left[\log\left(1 - d(\Theta, X)\right)\right] \quad . \tag{20}$$

Note that the we have equal marginals $X \mid (C = 1) \sim p(x) = q(x) = X \mid (C = 0)$.[6] This allows us to take the expectation over X and approximate (20) via Monte-Carlo for only one set of conditioning observations $\{X_n\}_{n=1}^{N_{\mathrm{cal}}}$, and with data points Θ_n and Θ_n^q respectively associated to class $C = 1$ and $C = 0$ for a given X_n:

$$\begin{aligned} l_{\mathrm{CE}}(d) &= \mathbb{E}_X\left[-\frac{1}{2}\mathbb{E}_{\Theta|X,C=1}\left[\log\left(d(\Theta, X)\right)\right] - \frac{1}{2}\mathbb{E}_{\Theta|X,C=0}\left[\log\left(1 - d(\Theta, X)\right)\right]\right] \\ &\approx -\frac{1}{2N_{\mathrm{cal}}}\sum_{n=1}^{N_{\mathrm{cal}}} \log\left(d(\Theta_n, X_n)\right) + \log\left(1 - d(\Theta_n^q, X_n)\right) \quad . \end{aligned} \tag{21}$$

[6] The joint distributions $p(\theta, x)$ and $q(\theta, x)$ are both modeled using samples $X \sim p(x \mid \Theta)$ obtained from the prior $\Theta \sim p(\theta)$, which implies that the marginals $p(x)$ and $q(x)$ are both defined by $\int p(x \mid \theta)p(\theta)\mathrm{d}\theta$.