Towards Reliable Simulation-Based Inference with Balanced Neural Ratio Estimation

Arnaud Delaunoy

Joeri Hermans

François Rozet

Antoine Wehenkel

Gilles Louppe

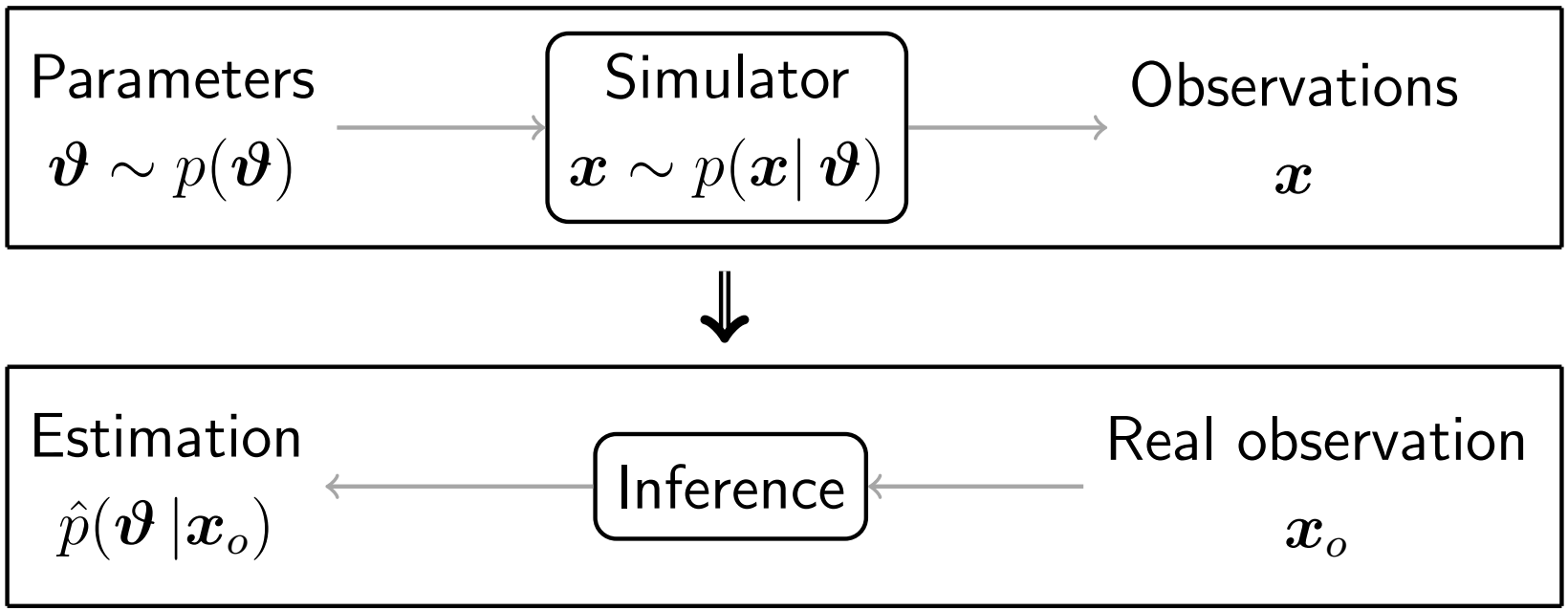
LIÈGE université

arXiv:2208.13624

Outline

1. What is simulation-based inference? + Some algorithms
2. Are those algorithms reliable / suitable for the scientific method?
3. Towards reliable simulation-based inference with balanced neural ratio estimation

Simulation-based inference


Parameters
$\boldsymbol{\vartheta} \sim p(\boldsymbol{\vartheta})$
Simulator
$\boldsymbol{x} \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$
Observations
$\boldsymbol{x}$
Estimation
$\hat{p}(\boldsymbol{\vartheta}\,|\boldsymbol{x}_o)$
Inference
Real observation
$\boldsymbol{x}_o$

Approximate Bayesian Computation (ABC)

1. Draw proposal parameters $\boldsymbol{\vartheta}_i$ from the prior $p(\boldsymbol{\vartheta})$.
2. Simulate synthetic observations for each parameter $\boldsymbol{x}_i \sim p(\boldsymbol{x}|\, \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i)$.
3. Accept parameters $\boldsymbol{\vartheta}_i$ if $d(\boldsymbol{x}_i, \boldsymbol{x}_o) < \epsilon$ for some distance d.
4. For $\epsilon \to 0$, accepted parameters are sampled from the posterior $p(\boldsymbol{\vartheta}\,|\boldsymbol{x}_o)$.

Neural methods

$$p(\boldsymbol{\vartheta}\,|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\,\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})}{p(\boldsymbol{x})}$$

- Neural Posterior Estimation (NPE): estimate $p(\boldsymbol{\vartheta}\,|\boldsymbol{x})$
- Neural Ratio Estimation (NRE): estimate $\frac{p(\boldsymbol{x}|\,\boldsymbol{\vartheta})}{p(\boldsymbol{x})}$
- Neural Likelihood Estimation (NLE): estimate $p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$

Neural Posterior/Likelihood Estimation

- **NPE**: Train a neural density estimator to approximate $p(\boldsymbol{\vartheta}\,|\boldsymbol{x})$ from data $(\boldsymbol{\vartheta}, \boldsymbol{x}) \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})$.
- **NLE**:
  1. Train a neural density estimator to approximate $p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$ from data $(\boldsymbol{\vartheta}, \boldsymbol{x}) \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})$.
  2. Use MCMC or VI to approximate $p(\boldsymbol{\vartheta}\,|\boldsymbol{x} = \boldsymbol{x}_o)$.

Neural density estimators are typically normalizing flows.

Neural Ratio Estimation (NRE)

Sample dataset and train a classifier

$(\boldsymbol{x}, \boldsymbol{\vartheta}) \sim p(\boldsymbol{x}, \boldsymbol{\vartheta})$	$y = 1$
$(\boldsymbol{x}, \boldsymbol{\vartheta}) \sim p(\boldsymbol{x})p(\boldsymbol{\vartheta})$	$y = 0$

The Bayes optimal classifier d^* can be expressed

$$d^*(\boldsymbol{\vartheta}, \boldsymbol{x}) = \frac{p(\boldsymbol{\vartheta}, \boldsymbol{x})}{p(\boldsymbol{\vartheta}, \boldsymbol{x}) + p(\boldsymbol{\vartheta})p(\boldsymbol{x})} \Leftrightarrow \frac{p(\boldsymbol{x}|\,\boldsymbol{\vartheta})}{p(\boldsymbol{x})} = \frac{d^*(\boldsymbol{\vartheta}, \boldsymbol{x})}{1 - d^*(\boldsymbol{\vartheta}, \boldsymbol{x})}.$$

Recover approximate posterior

$$(\boldsymbol{x}, \boldsymbol{\vartheta}) \longrightarrow \boxed{\text{Classifier}} \rightarrow \hat{d}(\boldsymbol{\vartheta}, \boldsymbol{x}) \rightarrow \hat{p}(\boldsymbol{\vartheta}\,|\boldsymbol{x}) = \frac{\hat{d}(\boldsymbol{\vartheta}, \boldsymbol{x})}{1 - \hat{d}(\boldsymbol{\vartheta}, \boldsymbol{x})} p(\boldsymbol{\vartheta})$$

Algorithm 1 Neural Ratio Estimation (NRE)

repeat

Sample from the joint $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta}, \boldsymbol{x}),\ y_i = 1\}_{i=1}^{n/2}$

Sample from the marginals $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta})p(\boldsymbol{x}),\ y_i = 0\}_{i=n/2+1}^{n}$

$\mathcal{L}[\hat{d}_\psi] = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i) + (1 - y_i)\log(1 - \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i))$

$\psi = \texttt{minimizer\_step}(\texttt{params=}\psi,\ \texttt{loss=}\mathcal{L}[\hat{d}_\psi])$

until convergence

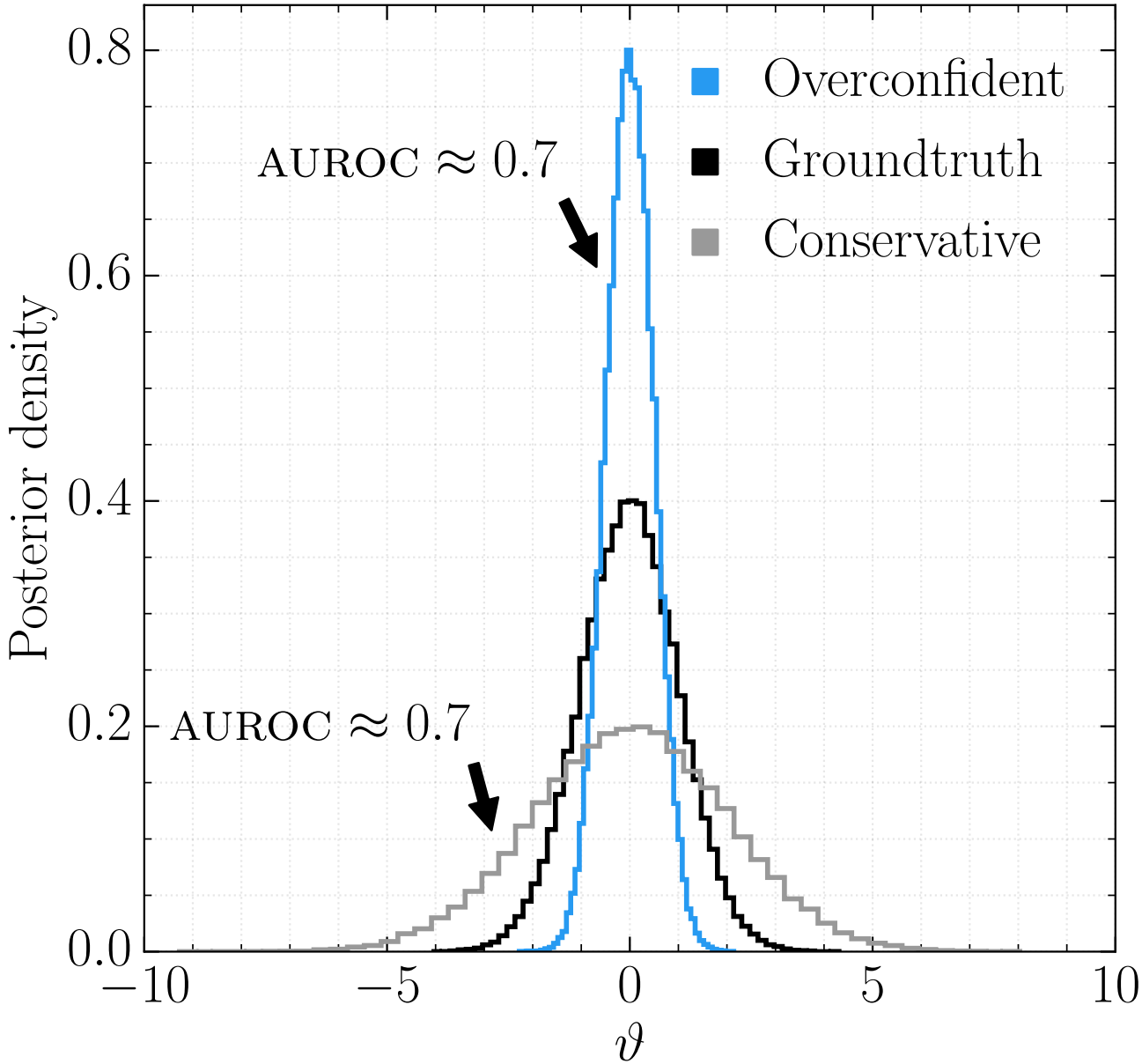
return $\hat{d}_\psi(\boldsymbol{\vartheta}, \boldsymbol{x})$.

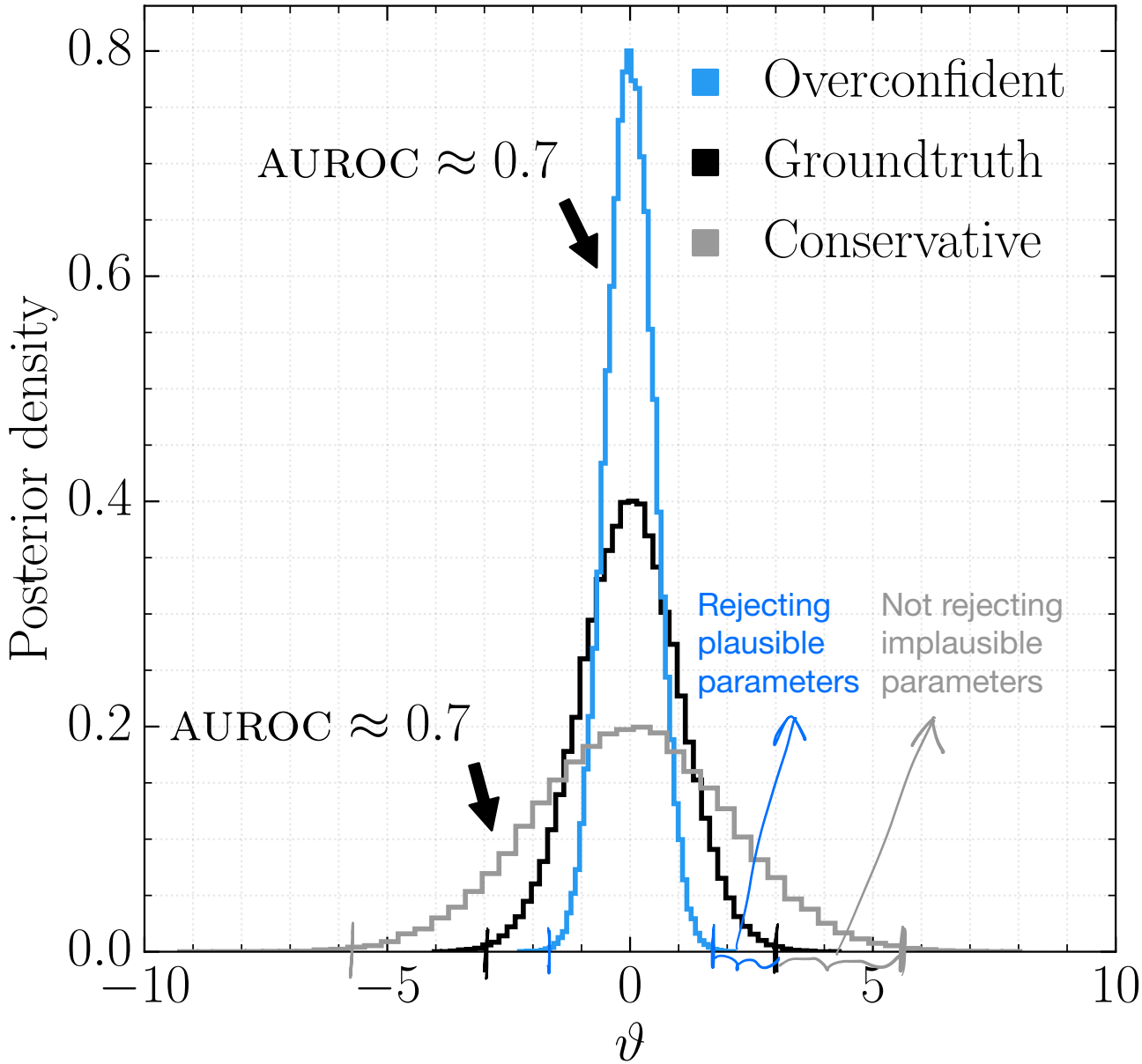
Sequential algorithms: Alternate between training and sampling for efficient sampling around a target $\boldsymbol{x}_o$.

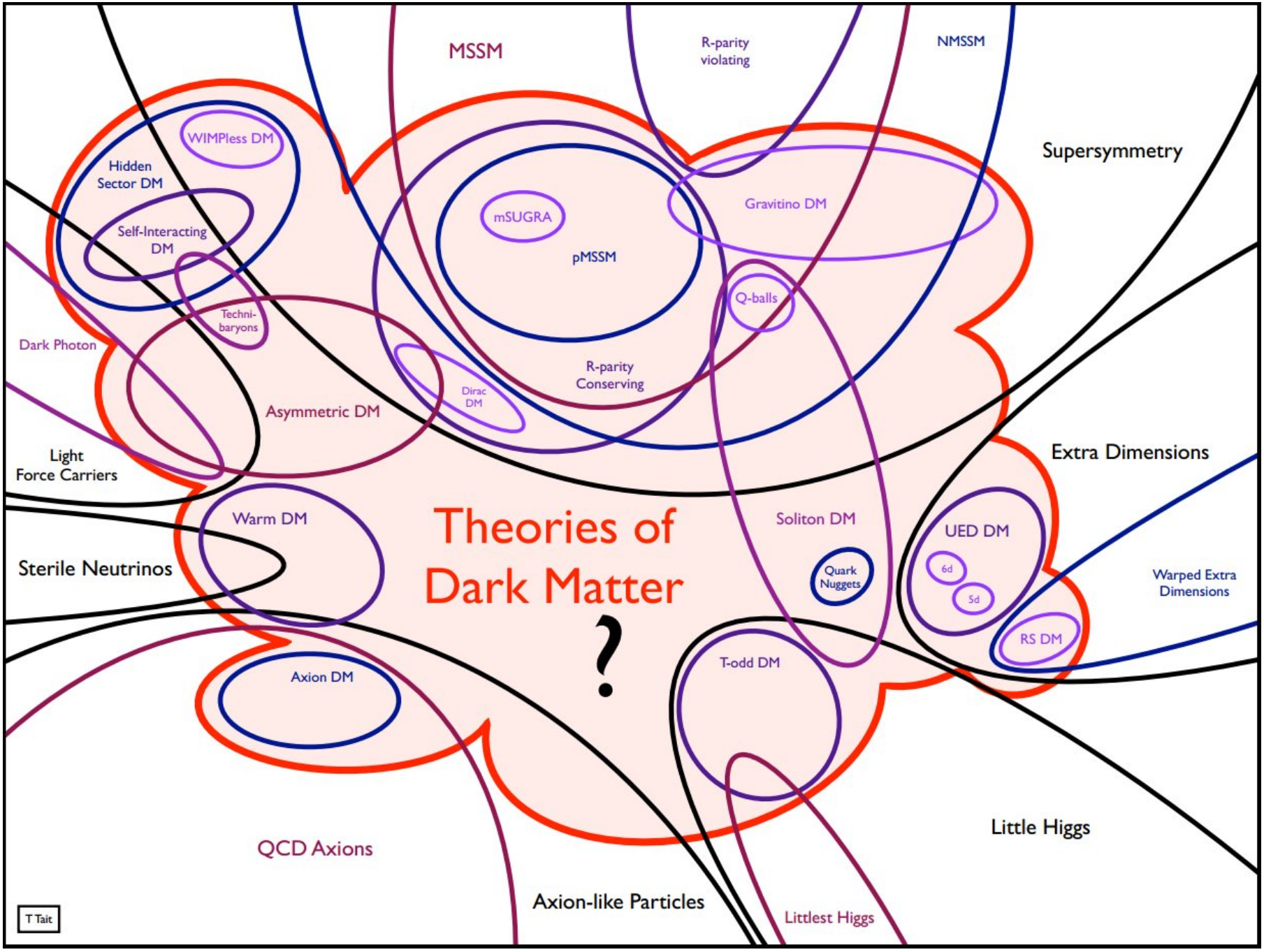
Algorithm 2 Sequential algorithm

Sample $\boldsymbol{\vartheta} \sim p(\boldsymbol{\vartheta}), \quad \boldsymbol{x} \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$
data $\leftarrow (\boldsymbol{\vartheta}, \boldsymbol{x})$
$\hat{p}_0(\boldsymbol{\vartheta}\,|\boldsymbol{x}) \leftarrow$ train(data)
for i in range(rounds) **do**
    Sample $\boldsymbol{\vartheta} \sim \hat{p}_i(\boldsymbol{\vartheta}\,|\boldsymbol{x} = \boldsymbol{x}_0), \quad \boldsymbol{x} \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$
    data $\leftarrow$ concatenate(data, $(\boldsymbol{\vartheta}, \boldsymbol{x})$)
    $\hat{p}_{i+1}(\boldsymbol{\vartheta}\,|\boldsymbol{x}) \leftarrow$ train(data)

Simulation-based inference reliability


Overconfident
Groundtruth
Conservative
AUROC ≈ 0.7
AUROC ≈ 0.7
Posterior density
0.0
0.2
0.4
0.6
0.8
−10
−5
0
5
10
ϑ


Overconfident
Groundtruth
Conservative
AUROC ≈ 0.7
AUROC ≈ 0.7
Rejecting plausible parameters
Not rejecting implausible parameters
Posterior density
0.0
0.2
0.4
0.6
0.8
−10
−5
0
5
10
ϑ


MSSM
R-parity violating
NMSSM
Supersymmetry
WIMPless DM
Hidden Sector DM
Self-Interacting DM
mSUGRA
pMSSM
Gravitino DM
Q-balls
Techni-baryons
Dark Photon
R-parity Conserving
Dirac DM
Asymmetric DM
Light Force Carriers
Extra Dimensions
Warm DM
Theories of Dark Matter
?
Soliton DM
UED DM
6d
5d
Quark Nuggets
Warped Extra Dimensions
Sterile Neutrinos
RS DM
T-odd DM
Axion DM
Little Higgs
QCD Axions
Axion-like Particles
Littlest Higgs
T Tait

Definition

The expected coverage is expressed

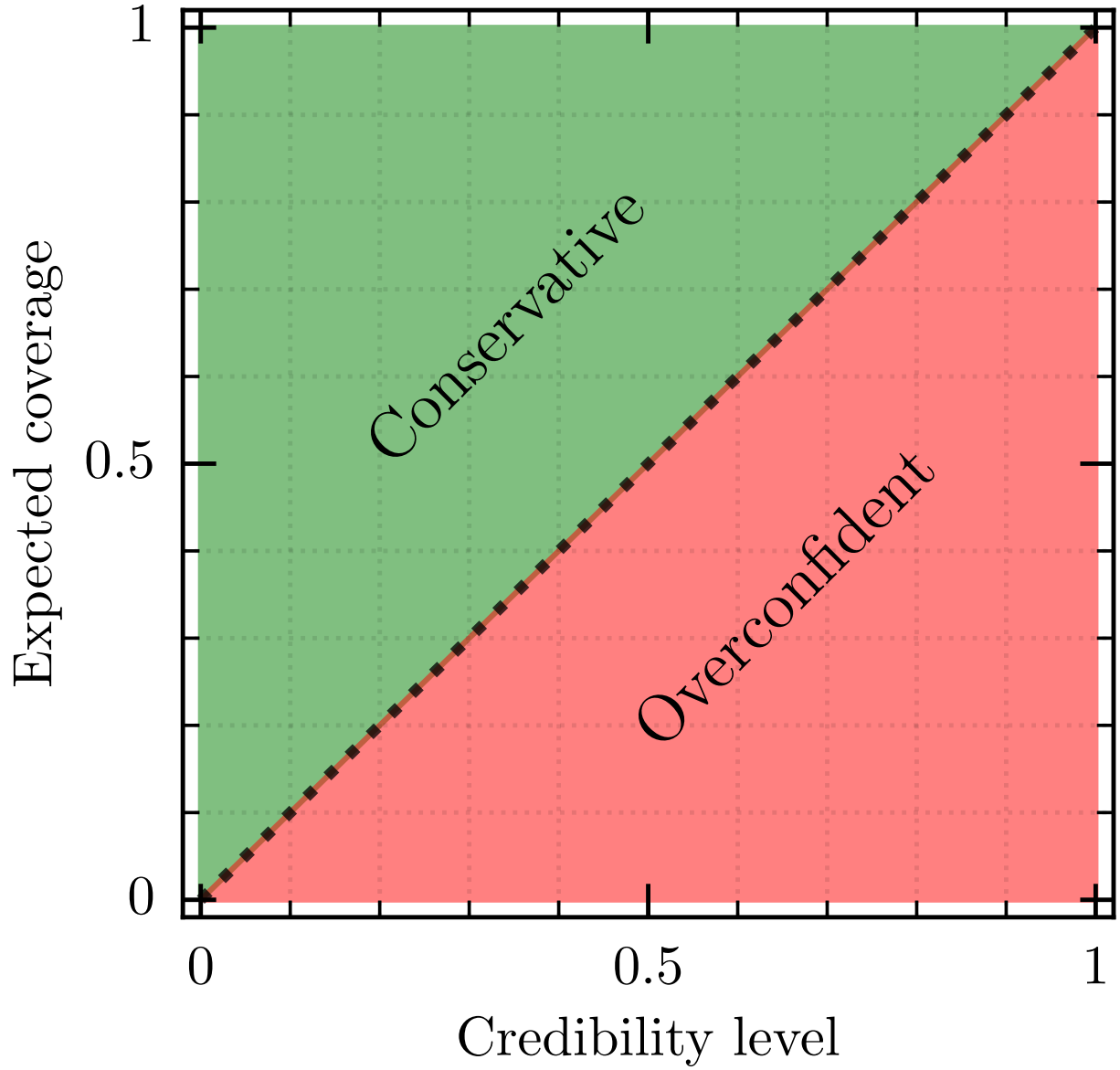
$$\text{expected coverage}(\hat{p}, \alpha) = \mathbb{E}_{p(\boldsymbol{\vartheta}, \boldsymbol{x})} \left[1 \left[\boldsymbol{\vartheta} \in \Theta_{\hat{p}(\boldsymbol{\vartheta} \,|\boldsymbol{x})}(1-\alpha) \right] \right],$$

where the function $\Theta_{\hat{p}(\boldsymbol{\vartheta} \,|\boldsymbol{x})}(1-\alpha)$ yields the $1-\alpha$ highest posterior density region of $\hat{p}(\boldsymbol{\vartheta} \,|\boldsymbol{x})$.

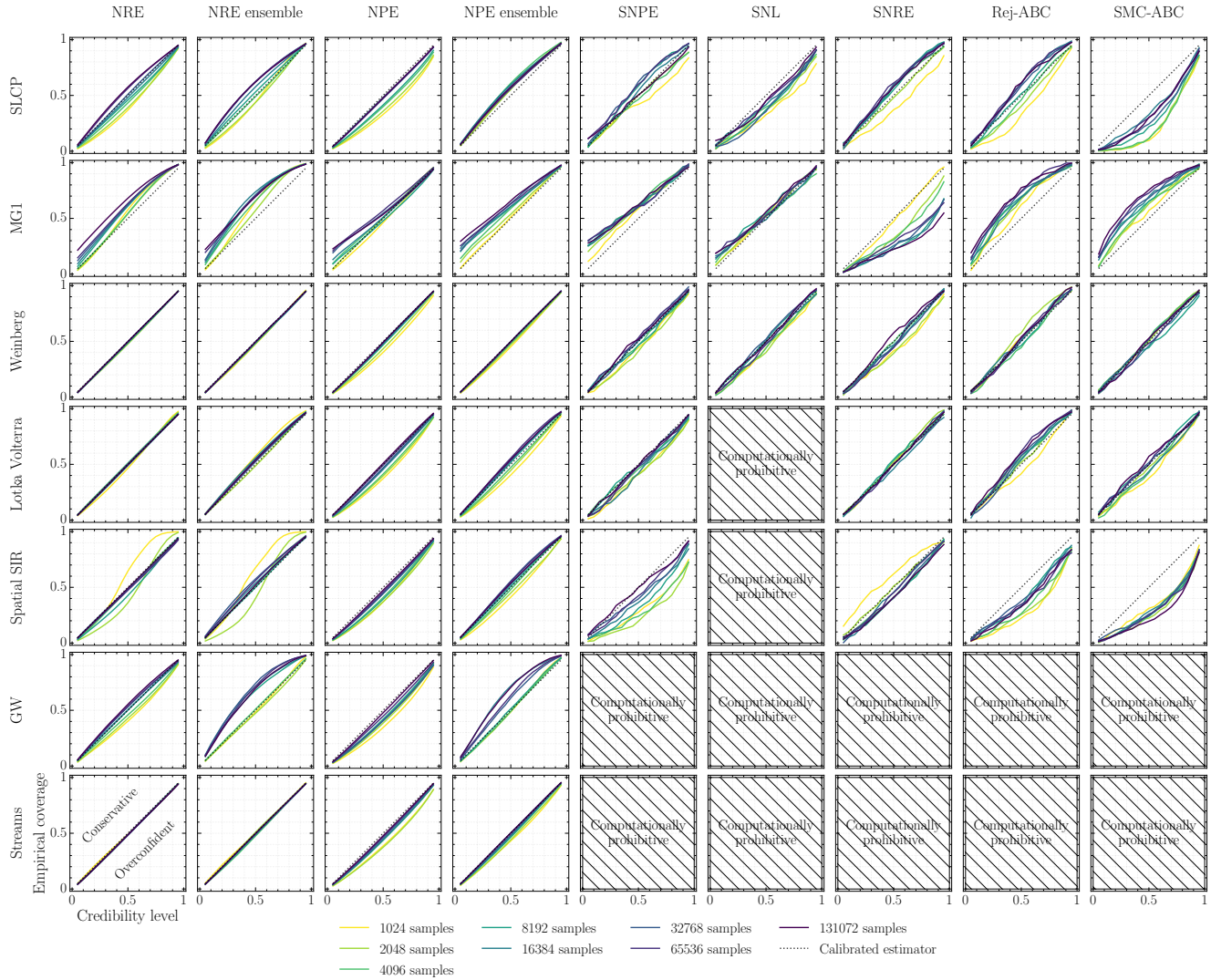
Definition

A **conservative model** is a model such that

$$\text{expected coverage}(\hat{p}, \alpha) \geq 1-\alpha, \quad \forall \alpha$$


1
0.5
0
Expected coverage
Conservative
Overconfident
0
0.5
1
Credibility level

Observation 1: All benchmarked algorithms may produce non-conservative posterior approximations.



Amortized (NRE, NPE, NLE):

1. Learn a model $\hat{p}(\boldsymbol{\vartheta} \,|\boldsymbol{x})$ valid for all $\boldsymbol{x}$.
2. Use this model to approximate the posterior for an observation $\hat{p}(\boldsymbol{\vartheta} \,|\boldsymbol{x} = \boldsymbol{x}_0)$.

Non-amortized (ABC, SNRE, SNPE, SNL):

1. Learn a model $\hat{p}(\boldsymbol{\vartheta} \,|\boldsymbol{x} = \boldsymbol{x}_0)$ for a given observation $\boldsymbol{x}_0$.
2. Repeat this procedure for every observation

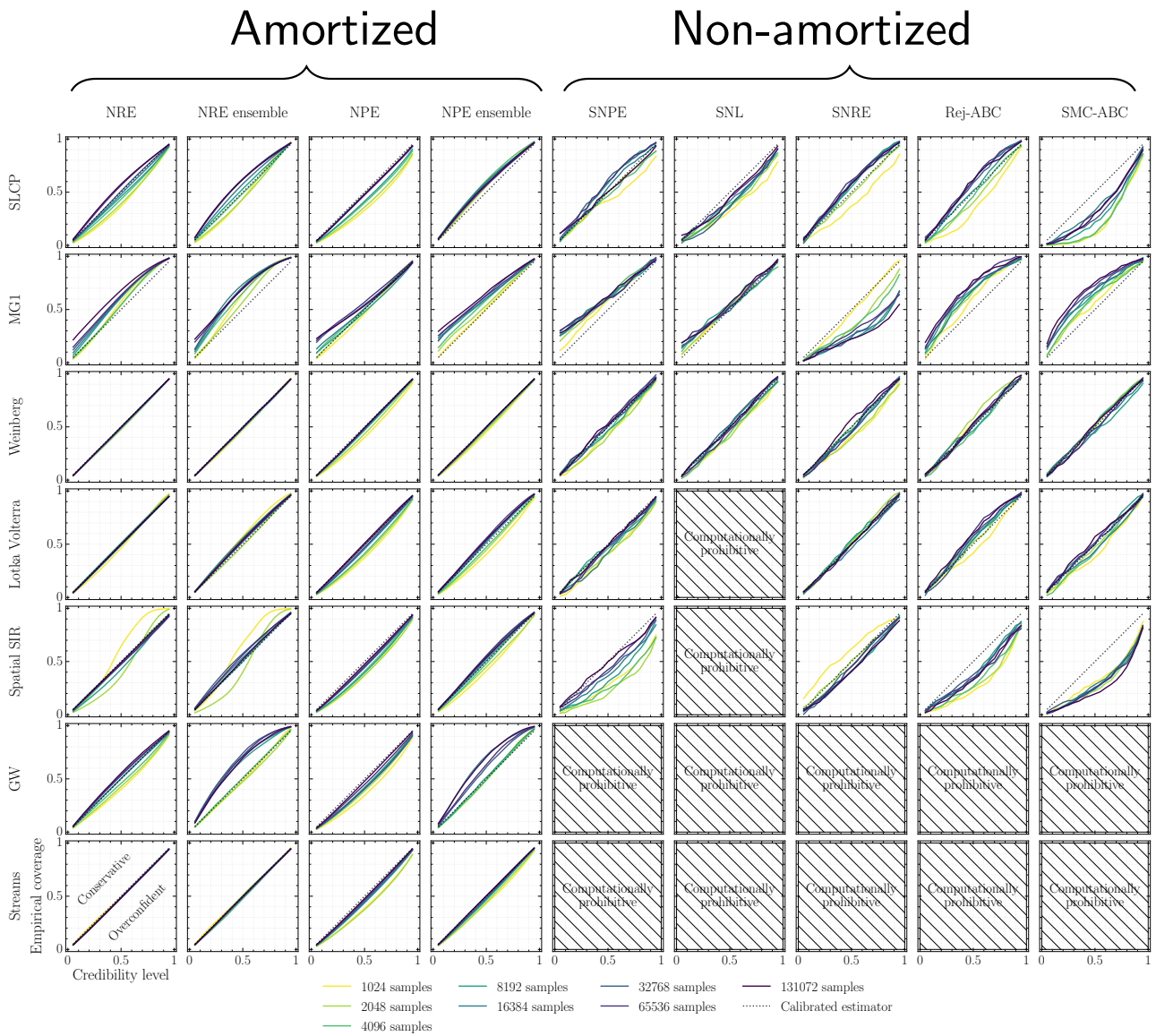
Sequential algorithm: Alternate between training and sampling for efficient sampling around the target $\boldsymbol{x}_o$.

Algorithm 3 Sequential algorithm

Sample $\boldsymbol{\vartheta} \sim p(\boldsymbol{\vartheta}), \quad \boldsymbol{x} \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$
data $\leftarrow (\boldsymbol{\vartheta}, \boldsymbol{x})$
$\hat{p}_0(\boldsymbol{\vartheta}\,|\boldsymbol{x}) \leftarrow$ train(data)
for i in range(rounds) **do**
    Sample $\boldsymbol{\vartheta} \sim \hat{p}_i(\boldsymbol{\vartheta}\,|\boldsymbol{x} = \boldsymbol{x}_0), \quad \boldsymbol{x} \sim p(\boldsymbol{x}|\,\boldsymbol{\vartheta})$
    data $\leftarrow$ concatenate(data, $(\boldsymbol{\vartheta}, \boldsymbol{x})$)
    $\hat{p}_{i+1}(\boldsymbol{\vartheta}\,|\boldsymbol{x}) \leftarrow$ train(data)

The model is only accurate around $\boldsymbol{x}_0 \rightarrow$ **non-amortized**!

Observation 2: Amortized approaches tend to be more conservative in contrast to non-amortized approaches.



Observation 3: Diagnosing non-amortized methods is computationally expensive.

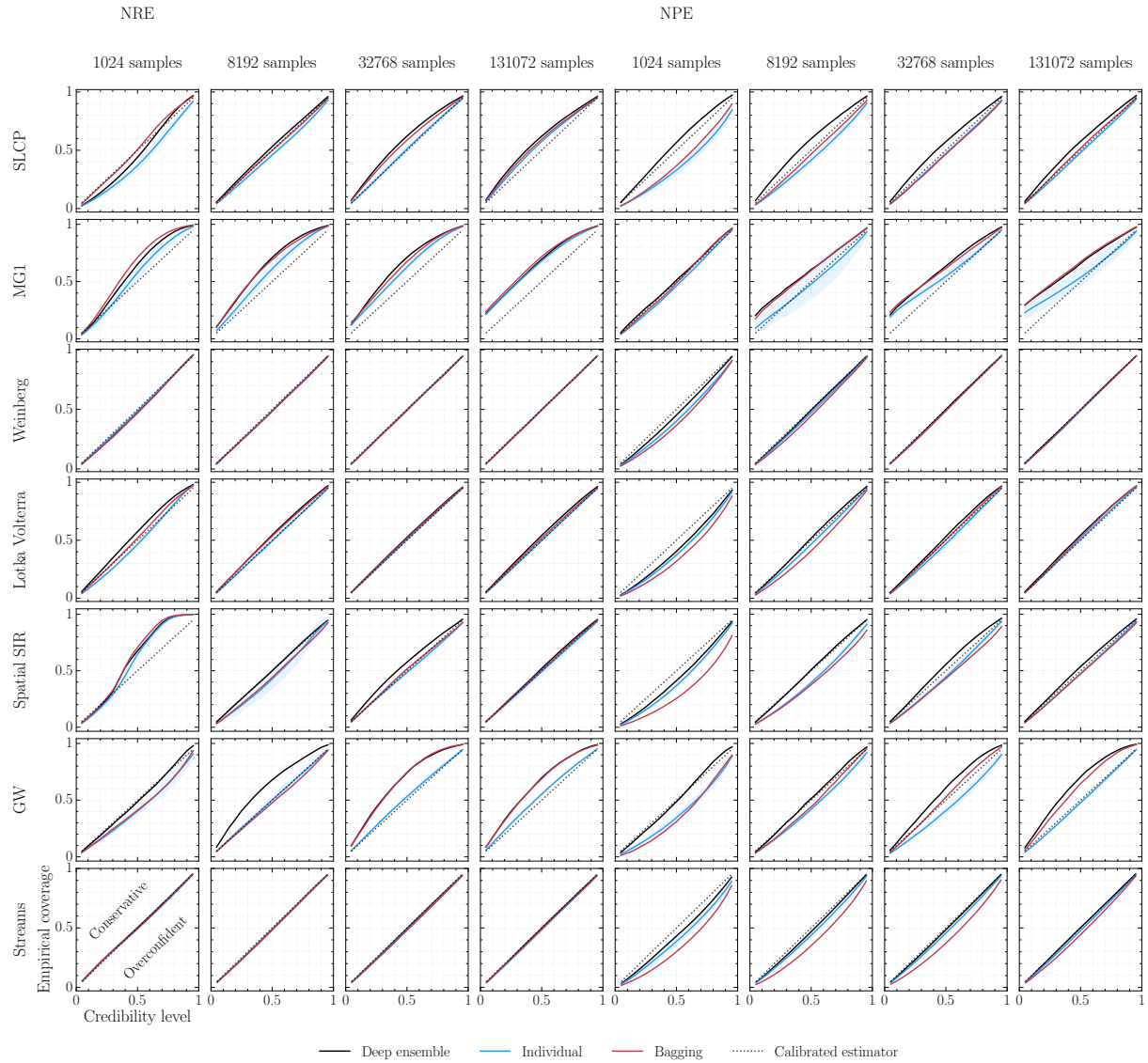
$$\text{expected coverage}(\hat{p}, \alpha) = \mathbb{E}_{p(\boldsymbol{\vartheta}, \boldsymbol{x})} \left[1 \left[\boldsymbol{\vartheta} \in \Theta_{\hat{p}(\boldsymbol{\vartheta} \mid \boldsymbol{x})}(1 - \alpha) \right] \right]$$

Intractable for non-amortized!

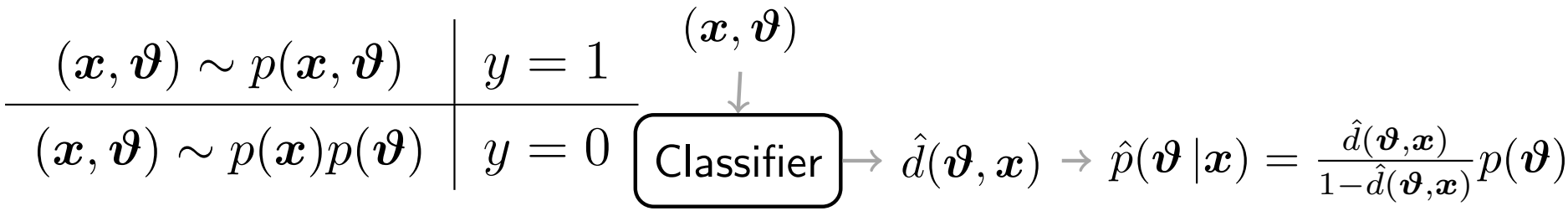
Recent work try to mitigate this issue:

- Miller, B. K., Cole, A., Forré, P., Louppe, G., Weniger, C. (2021). Truncated marginal neural ratio estimation. Advances in Neural Information Processing Systems, 34, 129-143.
- Deistler, M., Goncalves, P. J., Macke, J. H. (2022). Truncated proposals for scalable and hassle-free simulation-based inference. In Advances in Neural Information Processing Systems.

Observation 4: The expected coverage probability of an ensemble model is larger than the average individual model.



Towards reliable simulation-based inference with balanced neural ratio estimation



Algorithm 4 Neural Ratio Estimation (NRE)

repeat

Sample from the joint $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta}, \boldsymbol{x}),\ y_i = 1\}_{i=1}^{n/2}$

Sample from the marginals $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta})p(\boldsymbol{x}),\ y_i = 0\}_{i=n/2+1}^{n}$

$\mathcal{L}[\hat{d}_\psi] = -\frac{1}{n} \sum_{i=1}^{n} y_i \log \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i) + (1 - y_i) \log(1 - \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i))$

$\psi = \texttt{minimizer\_step}(\texttt{params=}\psi,\ \texttt{loss=}\mathcal{L}[\hat{d}_\psi])$

until convergence

return $\hat{d}_\psi(\boldsymbol{\vartheta}, \boldsymbol{x})$.

Idea: Restrict the classifier hypothesis space to more conservative models.

Definition

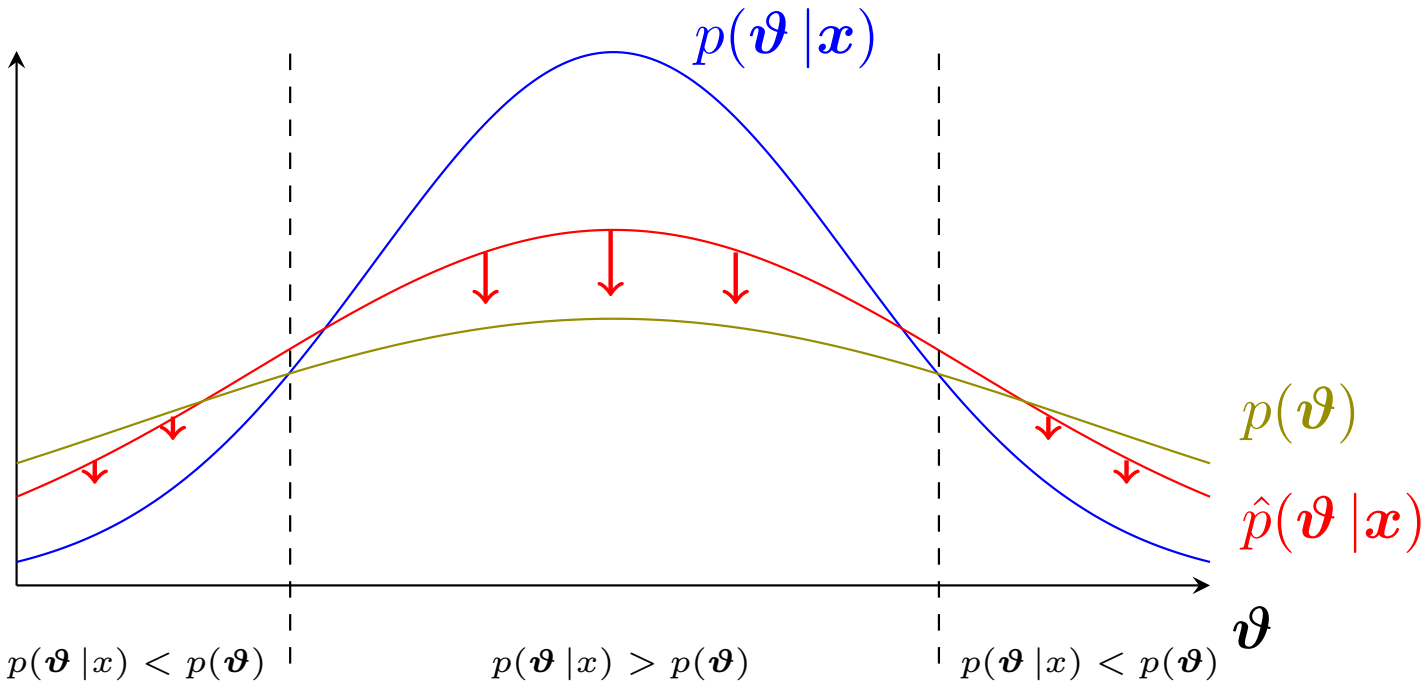
A classifier $\hat{d}$ is balanced if

$$\mathbb{E}_{p(\boldsymbol{\vartheta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] + \mathbb{E}_{p(\boldsymbol{\vartheta})p(\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] = 1.$$

Theorem 1

Any balanced classifier $\hat{d}$ *satisfies* $\mathbb{E}_{p(\boldsymbol{\vartheta},\boldsymbol{x})}\left[\dfrac{d^*(\boldsymbol{\vartheta},\boldsymbol{x})}{\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})}\right] \geq 1.$

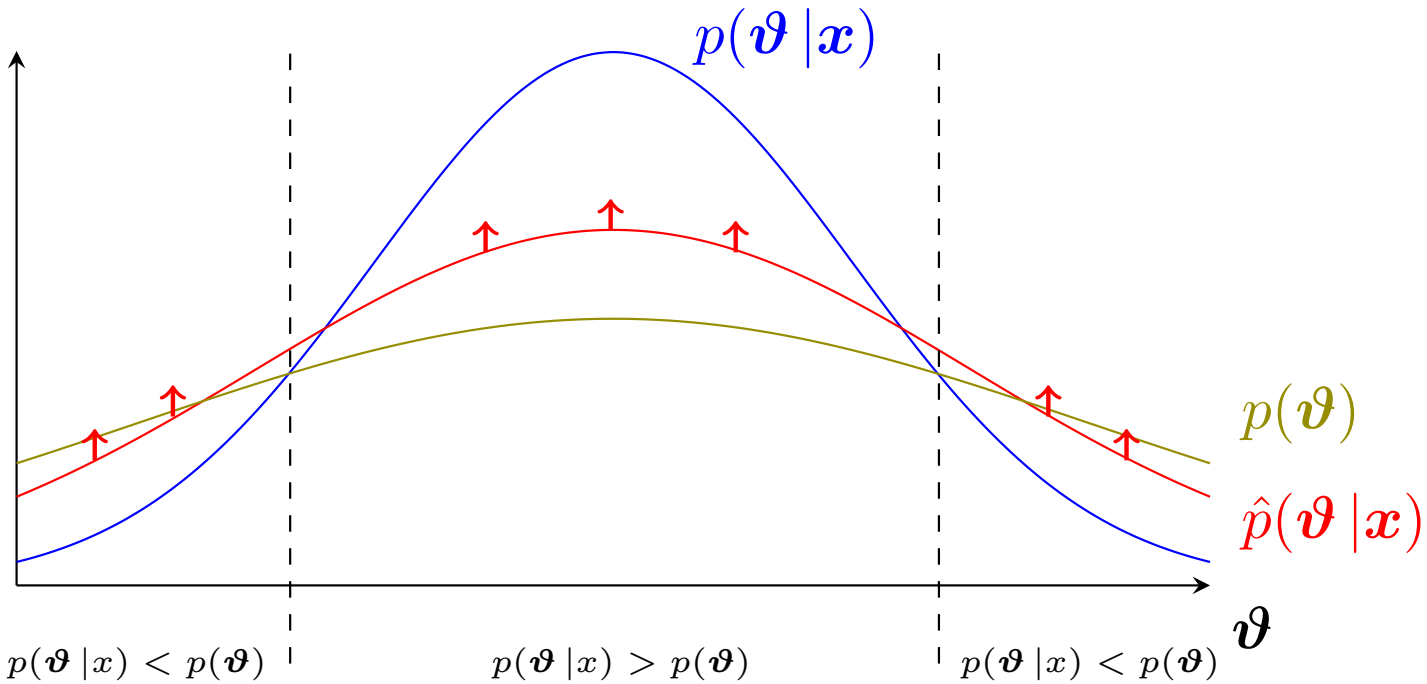
$$\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x}) \leq d^*(\boldsymbol{\vartheta},\boldsymbol{x}) \Leftrightarrow \frac{\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})} \leq \frac{d^*(\boldsymbol{\vartheta},\boldsymbol{x})}{1-d^*(\boldsymbol{\vartheta},\boldsymbol{x})} \Leftrightarrow \hat{p}(\boldsymbol{\vartheta}\,|\boldsymbol{x}) \leq p(\boldsymbol{\vartheta}\,|\boldsymbol{x})$$

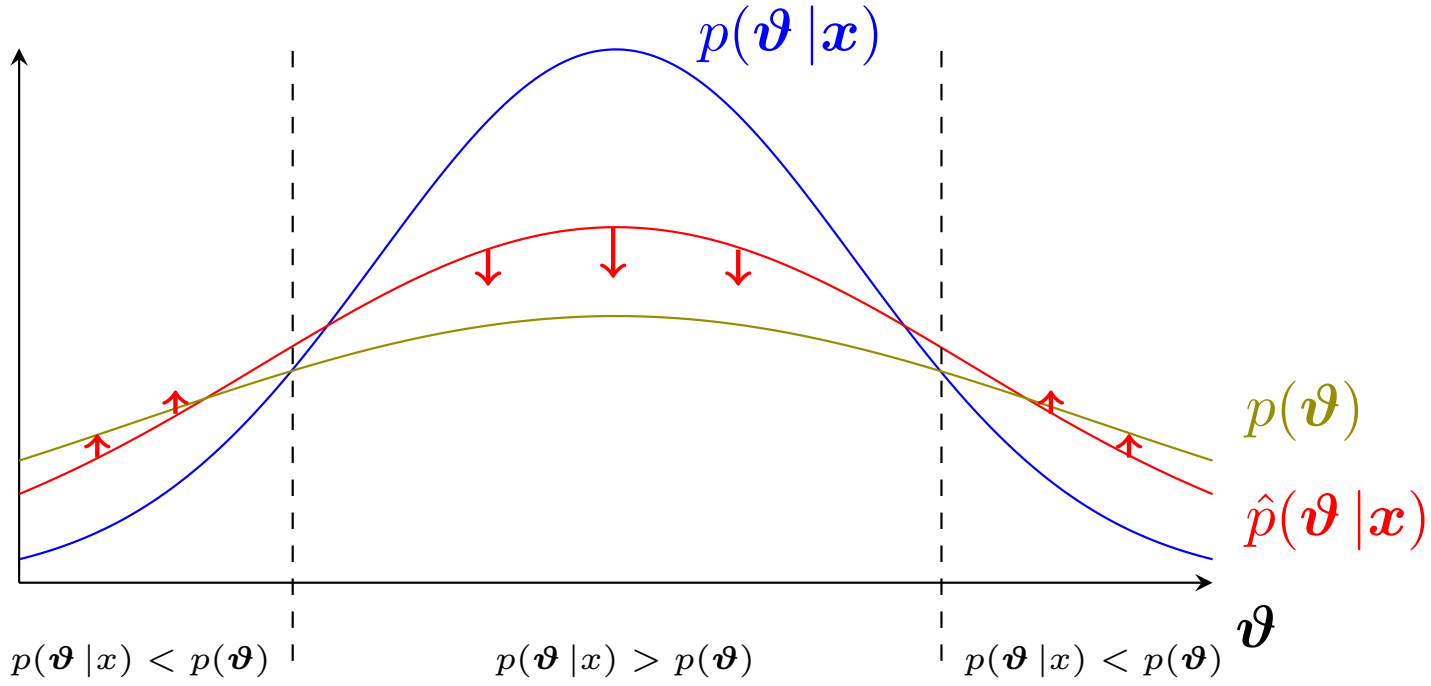


Theorem 2

Any balanced classifier $\hat{d}$ *satisfies* $\mathbb{E}_{p(\boldsymbol{\vartheta})p(\boldsymbol{x})}\left[\dfrac{1-d^*(\boldsymbol{\vartheta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})}\right] \geq 1.$

$$1-\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x}) \leq 1-d^*(\boldsymbol{\vartheta},\boldsymbol{x}) \Leftrightarrow \frac{\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})} \geq \frac{d^*(\boldsymbol{\vartheta},\boldsymbol{x})}{1-d^*(\boldsymbol{\vartheta},\boldsymbol{x})} \Leftrightarrow \hat{p}(\boldsymbol{\vartheta}\,|\boldsymbol{x}) \geq p(\boldsymbol{\vartheta}\,|\boldsymbol{x})$$





Theorem 3

The Bayes optimal classifier $d^*(\boldsymbol{\vartheta}, \boldsymbol{x})$ is balanced.

Balanced $\hat{d}$: $\mathbb{E}_{p(\boldsymbol{\vartheta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] + \mathbb{E}_{p(\boldsymbol{\vartheta})p(\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] = 1.$

Algorithm 5 Balanced Neural Ratio Estimation (BNRE)

repeat

Sample from the joint $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta},\boldsymbol{x}),\ y_i = 1\}_{i=1}^{n/2}$

Sample from the marginals $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta})p(\boldsymbol{x}),\ y_i = 0\}_{i=n/2+1}^{n}$

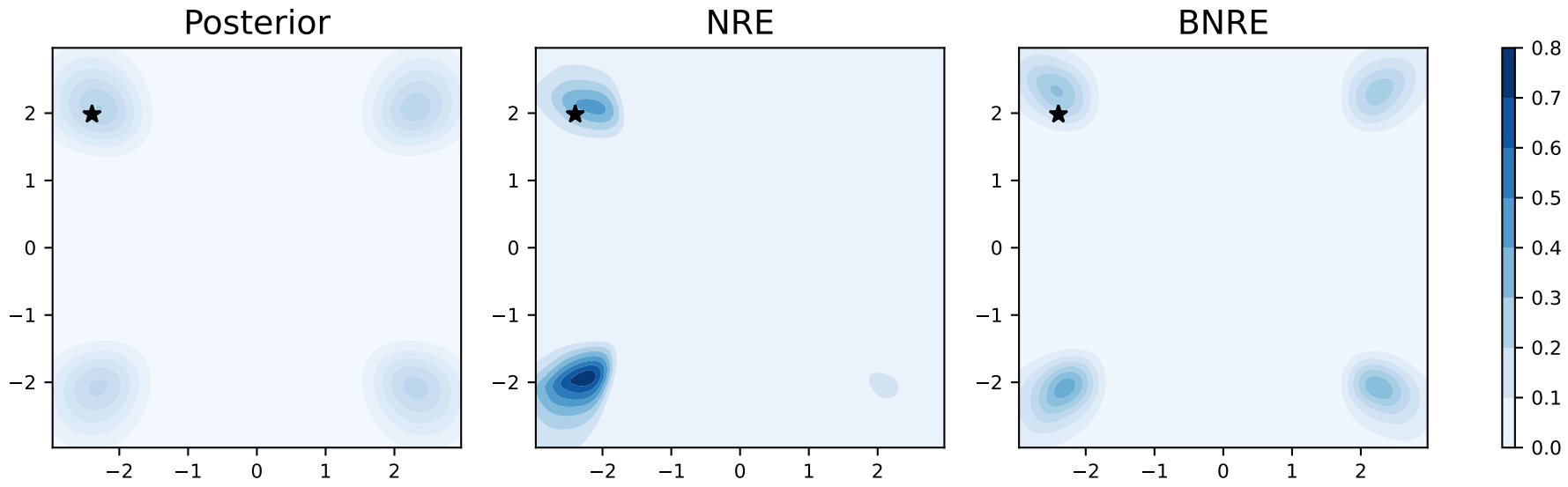
$\mathcal{L}[\hat{d}_\psi] = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{d}_\psi(\boldsymbol{\vartheta}_i,\boldsymbol{x}_i) + (1-y_i)\log(1-\hat{d}_\psi(\boldsymbol{\vartheta}_i,\boldsymbol{x}_i))$

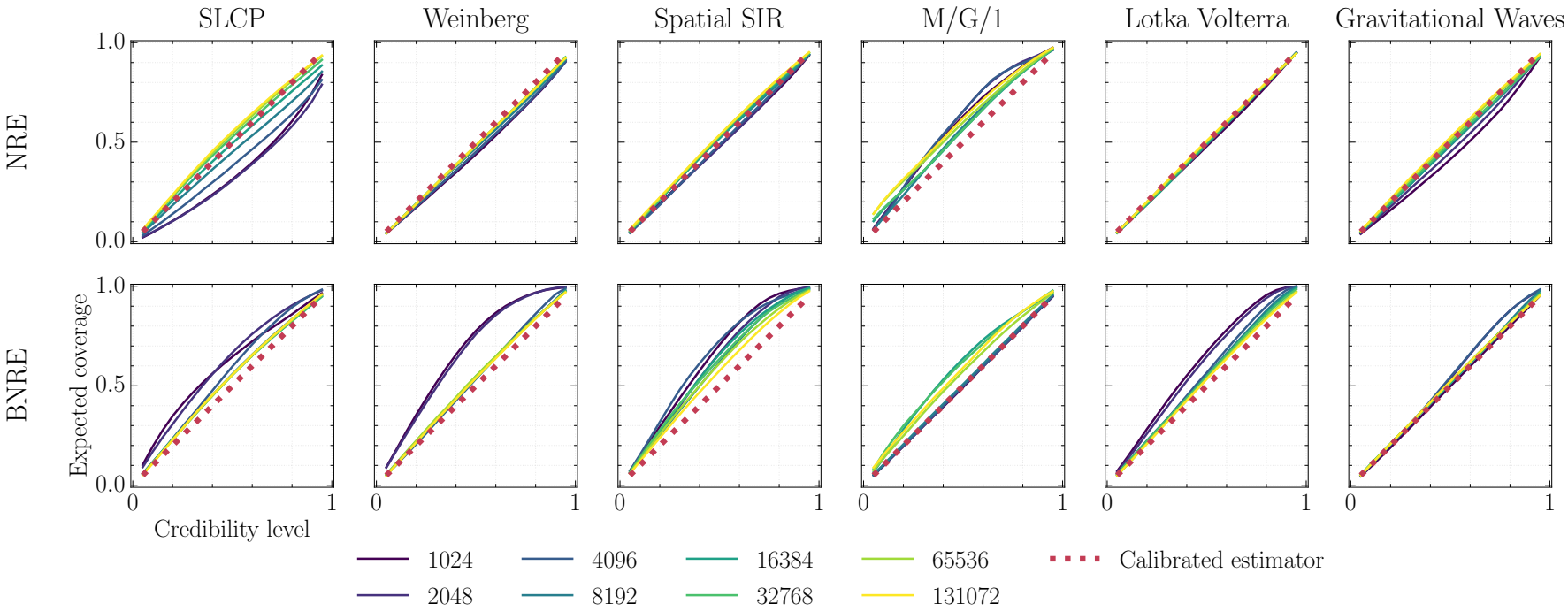
$\mathcal{B}[\hat{d}_\psi] = \frac{2}{n}\sum_{i=1}^{n/2} \hat{d}_\psi(\boldsymbol{\vartheta}_i,\boldsymbol{x}_i) + \frac{2}{n}\sum_{i=n/2+1}^{n} \hat{d}_\psi(\boldsymbol{\vartheta}_i,\boldsymbol{x}_i)$

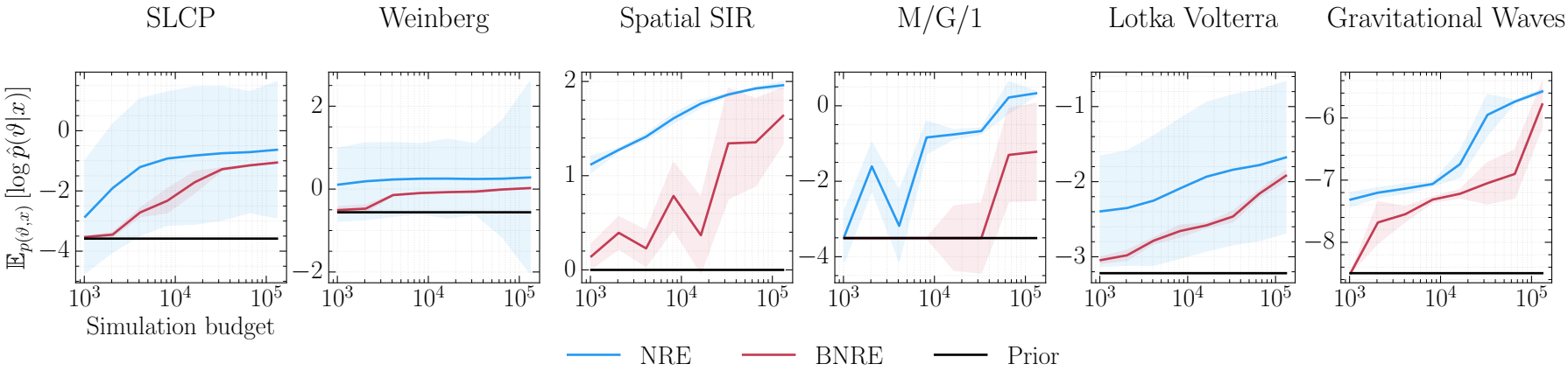
$\psi = \texttt{minimizer\_step}(\texttt{params=}\psi,\ \texttt{loss=}\mathcal{L}[\hat{d}_\psi] + \lambda(\mathcal{B}[\hat{d}_\psi] - 1)^2)$

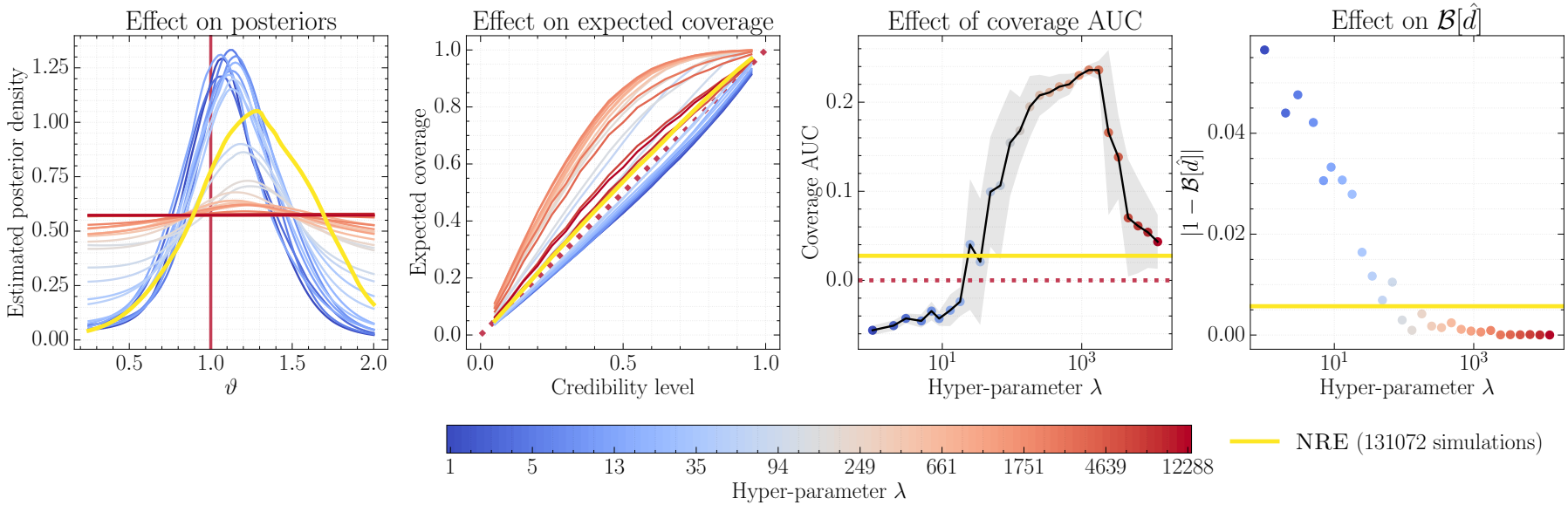
until convergence

return $\hat{d}_\psi(\boldsymbol{\vartheta},\boldsymbol{x})$.


Posterior
NRE
BNRE
2
1
0
−1
−2
−2
−1
0
1
2
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


SLCP
Weinberg
Spatial SIR
M/G/1
Lotka Volterra
Gravitational Waves
NRE
BNRE
Expected coverage
Credibility level
0.0
0.5
1.0
0
1
1024
2048
4096
8192
16384
32768
65536
131072
Calibrated estimator


SLCP
Weinberg
Spatial SIR
M/G/1
Lotka Volterra
Gravitational Waves
$\mathbb{E}_{p(\vartheta,x)}[\log \hat{p}(\vartheta|x)]$
Simulation budget
NRE
BNRE
Prior


Effect on posteriors
Estimated posterior density
ϑ
Effect on expected coverage
Expected coverage
Credibility level
Effect of coverage AUC
Coverage AUC
Hyper-parameter λ
Effect on $\mathcal{B}[\hat{d}]$
$|1 - \mathcal{B}[\hat{d}]|$
Hyper-parameter λ
1 5 13 35 94 249 661 1751 4639 12288
Hyper-parameter λ
NRE (131072 simulations)

- All benchmarked algorithms can produce non-conservative posterior approximations.
- Performing diagnostics to identify overconfident posterior approximations is crucial. The use of amortized algorithms or non-amortized ones that allow local diagnostics is then advised.
- Balanced Neural Ratio Estimation and ensembling constitute an immediately applicable solution to build more conservative approximate posteriors.
- BNRE should not be viewed as a way to obtain conservative posterior estimators with 100% reliability, but rather as a way to increase the reliability of the posterior estimators with minimal effort and no computational overhead.

Going further

- Extend to other algorithms
- better understanding of BNRE?
- Can we have guarantees?
- What if the simulator is misspecified?