

Arbitrary Marginal Neural Ratio Estimation for Likelihood-free Inference

Auteur : Rozet, François

Promoteur(s) : Louppe, Gilles

Faculté : Faculté des Sciences appliquées

Diplôme : Master : ingénieur civil en science des données, à finalité spécialisée

Année académique : 2020-2021

URI/URL : <https://github.com/francois-rozet/amnre>; <http://hdl.handle.net/2268.2/12993>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.



UNIVERSITY OF LIÈGE
SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

Arbitrary Marginal Neural Ratio Estimation for Likelihood-free Inference

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Data Science and Engineering

Author
François ROZET

Advisor
Pr. Gilles LOUPPE

Academic year 2020-2021

Abstract

In many areas of science, computer simulators are used to describe complex real-world phenomena. These simulators are stochastic forward models, meaning that they randomly generate synthetic realizations according to input parameters. A common task for scientists is to use such models to infer the parameters given observations. Due to their complexity, the likelihoods – essential for inference – implicitly defined by these simulators are typically not tractable. Consequently, scientists have relied on “likelihood-free” methods to perform parameter inference. In this thesis, we build upon one of these methods, the neural ratio estimation (NRE) of the likelihood-to-evidence (LTE) ratio, to enable inference over arbitrary subsets of the parameters. Called arbitrary marginal neural ratio estimation (AMNRE), this novel method is easy to use, efficient and can be implemented with basic neural network architectures. Through a series of experiments, we demonstrate the applicability of AMNRE and find it to be competitive with baseline methods, despite using a fraction of the computing resources. We also apply AMNRE to the challenging problem of parameter inference of binary black hole systems from gravitational waves observation and obtain promising results. As a complement to this contribution, we discuss the problem of overconfidence in predictive models and propose regularization methods to induce uncertainty in neural predictions.

Acknowledgments

I would like to thank my advisor Professor Gilles LOUPPE for introducing me to the field of simulation-based inference and, more generally, to the world of research. I am grateful for his availability and valuable feedback throughout this work and look forward to our future collaborations.

I would also like to thank Antoine WEHENKEL for the time he devoted to me and my thesis. In particular, his help in the early stage of my work was decisive.

Finally, I want to express my gratitude to my family and friends who supported me during my thesis and, more generally, my studies. I am especially grateful to my parents for all the sacrifices they made for me and without which nothing would have been possible.

Contents

1	Introduction	1
1.1	Problem statement	1
2	Frontier	3
2.1	Traditional estimation	3
2.2	Neural density estimation	4
2.2.1	Normalizing flows	4
2.2.2	Neural posterior estimation	5
2.3	Neural ratio estimation	6
2.3.1	Likelihood-to-reference ratio	7
2.3.2	Likelihood-to-evidence ratio	7
2.4	Feature imputation	8
2.5	Summary and discussion	9
3	Methods	11
3.1	Marginal NRE and NPE	11
3.1.1	Shared embedding	11
3.2	Arbitrary marginal NRE	12
3.2.1	Masking strategy	13
4	Experiments	15
4.1	Quality assessment	15
4.1.1	Receiver operating characteristic	15
4.1.2	Earth mover’s distance	16
4.1.3	Calibration	18
4.2	Simulators	19
4.3	Experimental protocol	21
4.4	Results	24
5	Conclusion	38
	Acronyms	40
	Bibliography	42
A	Additional figures	49
B	Autoregressive flows	63
C	Overconfidence	65
D	Consistency optimization	69

Chapter 1

Introduction

In many areas of science, computer simulators are used to describe complex phenomena like high energy particle interactions [1], compact binary coalescence events [2] or neuronal ion-channel dynamics [3, 4]. These simulators are *stochastic forward models* or *probabilistic programs*, meaning that they randomly generate synthetic realizations according to input parameters. A common task for scientists is to use such models to perform *statistical inference* [5, 6] of the parameters given one or more observations. Unfortunately, due to their complexity, the likelihoods – essential for parameter inference – implicitly defined by these simulators are typically not tractable. The problem of statistical inference under intractable likelihoods is commonly referred to as *likelihood-free* inference (LFI) or *simulation-based* inference (SBI) and is a rapidly expanding field of research [7].

Formally, a stochastic forward model takes a vector of parameters $\theta \in \Theta$ as input, samples internally a series $z \in \mathcal{Z}$ of latent variables $z_i \sim p(z_i|\theta, z_{<i})$ and finally produces a realization $x \in \mathcal{X} \sim p(x|\theta, z)$ as output, thereby defining an implicit likelihood $p(x|\theta)$. This likelihood generally is *intractable* as it corresponds to

$$p(x|\theta) = \int_{\mathcal{Z}} p(x, z|\theta) dz = \int_{\mathcal{Z}} p(x|\theta, z) \prod_i p(z_i|\theta, z_{<i}) dz, \quad (1.1)$$

the integral of the joint likelihood $p(x, z|\theta)$ over *all* possible trajectories through the latent space \mathcal{Z} . Moreover, in Bayesian inference, we are interested in the posterior

$$p(\theta|x^*) = \frac{p(x^*|\theta)p(\theta)}{p(x^*)} = \frac{p(x^*|\theta)p(\theta)}{\int_{\Theta} p(x^*|\theta')p(\theta') d\theta'} \quad (1.2)$$

for some observation(s) x^* and assuming a prior $p(\theta)$, which not only involves the potentially intractable likelihood $p(x^*|\theta)$ but also an integral over the parameter space Θ . For simulators with high-dimensional parameter spaces, this is a second source of intractability, leading to even more challenging problems.

Consequently, instead of using the true likelihood to perform inference, scientists have relied on “likelihood-free” surrogate models $\hat{p}(x|\theta)$ of the likelihood or $\hat{p}(\theta|x)$ of the posterior, covered in Chapter 2.

1.1 Problem statement

Domain scientists are not always interested in the full set of simulator parameters at once. In particular, when interpreting posterior predictions, they generally study several small parameter subsets, especially singletons and pairs, while ignoring the others. This applies to all fields of science: particle physics, astronomy, climatology, biology, medicine, etc

[8–13]. For example, in the context of high energy particle interactions, physicists might want to infer the mass of particles, regardless of their spin or charge.

Formally, a subspace $\Theta_a \leq \Theta$ of the parameter space is of interest, while the complement subspace $\Theta_b : \Theta_a \times \Theta_b = \Theta$ is unobserved. The *marginal posterior estimation* (MPE) task is then to estimate the marginal posterior

$$p(\theta_a|x^*) = \int_{\Theta_b} p(\theta|x^*) d\theta_b \quad (1.3)$$

for some observation(s) x^* . For this task, current LFI methods resort to numerical integration of a surrogate model $\hat{p}(\theta|x^*)$ of the full posterior, which is computationally expensive if Θ_b is large. For domain scientists, the computation time introduced by this approach is inconvenient, especially if several subspaces Θ_a are studied.

A naive solution to get rid of numerical integration is to learn a surrogate $\hat{p}(\theta_a|x^*)$ by considering θ_b as part of the latent variables. If we are interested in a single or a few predetermined subspaces, this is perfectly reasonable. However, if we need to choose *arbitrarily* the subspace at inference time, *i.e.* arbitrary MPE, this solution is not viable anymore as there exists an exponential number ($2^{|\Theta|} - 1$) of marginal posteriors. In this thesis, we focus on the development of a method able to estimate, without numerical marginalization, the marginal posterior $p(\theta_a|x^*)$ over *any* parameter subspace Θ_a .

This study applies to all simulators as we consider them to be *black boxes*, meaning that we do not have access to any information besides the realizations x , like the latent variables z_i , the latent likelihoods $p(z_i|\theta, z_{<i})$ or the conditional likelihood $p(x|\theta, z)$, which can be leveraged to improve inference [14, 15].

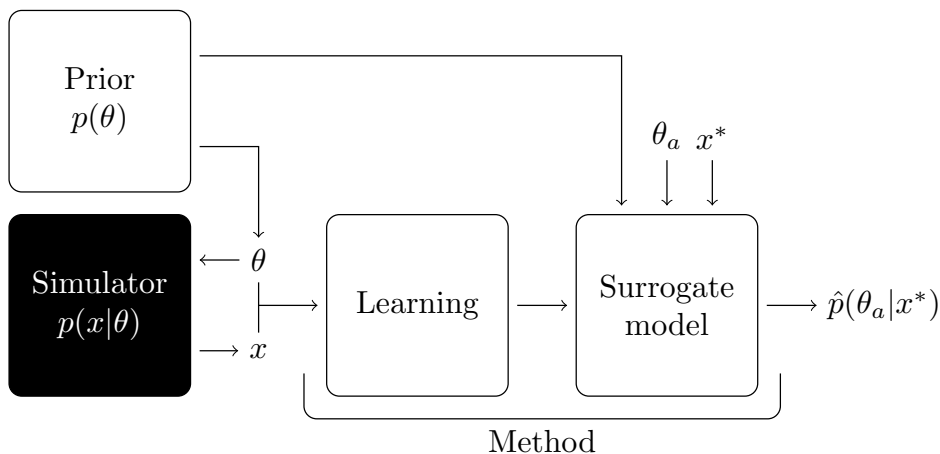


Figure 1.1. The objective of this thesis is to develop a method able to perform arbitrary MPE of any black box simulator, assuming a prior. This objective requires the design of a suitable surrogate model and learning procedure.

Chapter 2

Frontier

This chapter acts both as a background introduction and a literature review. First, we take a look at traditional and recent LFI posterior estimation methods, comparing their origins, strengths and limitations. Then, we cover the topic of feature imputation, closely related to the problem at hand.

Note. The name of this chapter is a tribute to “The frontier of simulation-based inference” by Cranmer et al. [7], a remarkable introduction to SBI and its challenges from which the chapter’s structure is inspired.

2.1 Traditional estimation

For decades, the most widespread approaches to Bayesian inference were *approximate Bayesian computation* (ABC) methods [16–18]. These methods approximate the posterior by *comparing* simulated realizations x with the observation x^* . More precisely, in the ABC rejection algorithm – the simplest form of ABC – parameters θ are drawn from the prior and realizations x are generated with the simulator using these parameters. If x is sufficiently close to x^* , θ is retained, and otherwise rejected. This condition of sufficient proximity is formalized as

$$\rho(x, x^*) \leq \epsilon, \quad (2.1)$$

where $\rho : \mathcal{X}^2 \mapsto \mathbb{R}_+$ is a distance metric and $\epsilon \in \mathbb{R}_+$ is an arbitrary tolerance.

After many iterations, the set of retained parameters is representative of the prior weighted by the probability that (2.1) is satisfied, leading to an approximate version of the posterior $p(\theta|x^*)$. In the limit of $\epsilon \rightarrow 0$, this approximate posterior becomes exact, but the acceptance probability of samples vanishes for continuous realizations, especially if high-dimensional. Thus, to increase *sample efficiency*, ABC often relies on hand-crafted low-dimensional summary statistics $s(x)$ to perform the comparison. The *quality of inference* is tied to how well those statistics retain information about the parameters θ . Consequently, ABC presents a trade-off between sample efficiency and inference quality. Moreover, since x^* is directly used in the rejection process, inference for different observations requires repeating the entire algorithm.

An alternative to ABC is to approximate the likelihood $p(x|\theta)$ of simulated data with classical *density estimation* (DE) techniques like histograms or kernel density estimation [19, 20]. Statistical inference then proceeds as if the likelihood was tractable. The main advantage over ABC is that the computational cost of the simulation and likelihood estimation stages can be *amortized* over several experiments with different observations, making it particularly well-suited for problems with many independent and identically

distributed (i.i.d.) observations, like high energy particle interactions in CERN’s Large Hadron Collider.

Nonetheless, like ABC, classical DE presents a trade-off between sample efficiency and inference quality, as it also scales poorly to high-dimensional data spaces. Similarly to the k -nearest neighbors algorithm [21–23], both ABC and classical DE are said to suffer from the *curse of dimensionality*¹: the required number of samples increases exponentially with the dimension of the data, in the worst case.

2.2 Neural density estimation

Fortunately, the relatively recent advances in *deep learning* (DL) [25] allow to handle much higher-dimensional data without loss of quality [26], leading to an increasingly popular use of *neural networks* (NNs) for DE [12, 27–33].

2.2.1 Normalizing flows

One class of these *neural density estimation* (NDE) techniques are *normalizing flows* (NFs) [34–41], in which a random variable u with simple distribution p_u (*e.g.* uniform or multi-variate Gaussian) is mapped to the sample space through an *invertible and differentiable* transformation $x = g(u)$. The sample distribution is then given by the change-of-variables formula

$$p(x) = p_u(f(x)) \left| \det J_f(x) \right| \quad (2.2a)$$

$$= p_u(f(x)) \left| \det J_g(f(x)) \right|^{-1}, \quad (2.2b)$$

where f is the inverse of g and $J_f = \frac{\partial f}{\partial x}$ denotes the Jacobian of f . Therefore, with the right transformation g and a base distribution p_u , we can construct any tractable distribution and sample from it, under reasonable assumptions [42, 43]. However, constructing arbitrarily complex bijections is not trivial, especially while keeping the Jacobian determinant tractable. The approach of NFs is to leverage the fact that a composition of invertible functions is itself invertible. Indeed, if g_1, g_2, \dots, g_n are bijective functions,

$$g = g_1 \circ g_2 \circ \dots \circ g_n \quad (2.3)$$

is also bijective, with inverse

$$f = f_N \circ \dots \circ f_2 \circ f_1, \quad (2.4)$$

and the Jacobian determinant takes the form

$$\det J_f(x) = \prod_{i=1}^n \det J_{f_i}(z_{i-1}), \quad (2.5)$$

where $z_i = f_i(z_{i-1})$ and $z_0 = x$. Thus, a sequence of simple differentiable bijective transformations can be stacked together to construct more complex transformations which are, in turn, invertible and differentiable. In this sequence, the probability density is said to “flow” from an irregular distribution $p(x)$ towards a simpler, more “normal” form $p_u(u)$, hence the name “normalizing flow” [34].

¹The expression was originally employed by Bellman [24] in the field of dynamic programming.

If the transformations have adjustable parameters, we obtain a mechanism to construct new *families* of distributions

$$q_\phi(x) = p_u(f_\phi(x)) \left| \det J_{f_\phi}(x) \right| \quad (2.6)$$

whose parameters ϕ can be *optimized* to approximate an unknown distribution $p(x)$ by maximizing the likelihood of i.i.d. samples $x_i \sim p(x)$. That is

$$\phi^* = \arg \max_{\phi} \prod_i q_\phi(x_i) = \arg \max_{\phi} \sum_i \log q_\phi(x_i) = \arg \max_{\phi} \mathbb{E}_{p(x)} [\log q_\phi(x)]. \quad (2.7)$$

This stacking of differentiable parametric functions is reminiscent of a NN and, as such, is generally trained using *stochastic gradient descent* (SGD) [44–47] optimization techniques within *automatic differentiation* (AD) [48–51] frameworks.

NFs extend naturally to the task of estimating a *conditional* density $p(x|y)$ by conditioning the transformations of x with y , *i.e.* $x = g_\phi(u|y)$ and $u = f_\phi(x|y)$, which allows to construct and train families of conditional distributions

$$q_\phi(x|y) = p_u(f_\phi(x|y)) \left| \det J_{f_\phi}(x|y) \right|. \quad (2.8)$$

2.2.2 Neural posterior estimation

For neural posterior estimation (NPE), we approximate $p(\theta|x)$ with a conditional distribution family $q_\phi(\theta|x)$. As in (2.7), the parameters ϕ are optimized by maximizing the expected log-density over the implicit joint distribution $p(\theta, x) = p(\theta)p(x|\theta)$, *i.e.*

$$\phi^* = \arg \max_{\phi} \mathbb{E}_{p(\theta, x)} [\log q_\phi(\theta|x)]. \quad (2.9)$$

With this approach, the trained surrogate posterior is amortized, meaning that, even if the training stage requires a lot of samples, inference itself is simulation-free and can be repeated several times with different observations.

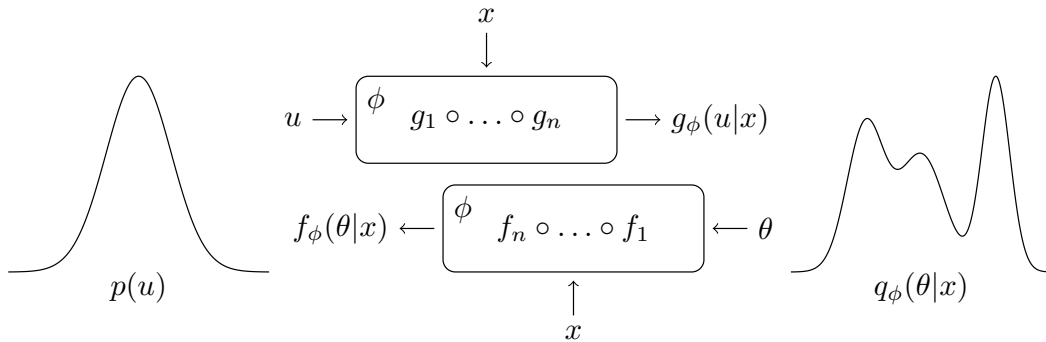


Figure 2.1. Illustration of the conditional flow architecture for NPE.

This is not the case of *sequential* neural posterior estimation (SNPE) methods [27–29], a special type of NPE drawing inspiration from sequential Monte Carlo (SMC) techniques [52–54]. The rationale is that, if we are ultimately interested in the posterior at a specific observation x^* , drawing parameters θ from the prior $p(\theta)$ is wasteful as parameters with low(er) posterior density $p(\theta|x^*)$ are less informative [29].

Instead, SNPE iteratively refines a *proposal* distribution $\tilde{p}(\theta)$ to be more and more informative about $p(\theta|x^*)$ and that replaces the prior during training (see Algorithm 1). Unfortunately, optimizing (2.9) on samples drawn from a proposal no longer yields the target posterior. Papamakarios et al. [27], Lueckmann et al. [28] and Greenberg et al. [29] differ primarily in how they tackle this problem, mainly by defining an alternative loss function \mathcal{L} to minimize.

Algorithm 1 SNPE with per-round proposal updates [12, 29]

Input: simulator with (implicit) likelihood $p(x|\theta)$, observation x^* , prior $p(\theta)$, conditional distribution family $q_\phi(\theta|x)$, loss function \mathcal{L} , empty buffer \mathcal{D} , number of simulations N , number of rounds R

```

1  $\tilde{p}(\theta) \leftarrow p(\theta)$ 
2 for  $1 \dots R$  do
3   for  $1 \dots N$  do
4     sample  $\theta \sim \tilde{p}(\theta)$ 
5     simulate  $x \sim p(x|\theta)$ 
6     store  $(\theta, x)$  in  $\mathcal{D}$ 
7    $\phi \leftarrow \arg \min_{\phi} \sum_{(\theta, x) \in \mathcal{D}} \mathcal{L}(\theta, q_\phi(\theta|x))$ 
8    $\tilde{p}(\theta) \leftarrow q_\phi(\theta|x^*)$ 
9 return  $q_\phi(\theta|x)$ 
```

SNPE is a typical example of *active learning* in that it simulates realizations for parameters which are expected to increase our knowledge the most, thereby improving sample efficiency over standard NPE for *single* observation inference.

2.3 Neural ratio estimation

As shown by Cranmer et al. [55], NNs can also be trained to approximate the *likelihood ratio*

$$r(x|\theta_0, \theta_1) = \frac{p(x|\theta_1)}{p(x|\theta_0)} \quad (2.10)$$

between two parameters θ_0 and θ_1 , traditionally used for hypothesis testing [56, 57], hence the name *neural ratio estimation* (NRE). To do so, a classifier network $d : \mathcal{X} \mapsto [0, 1]$ is trained to *discriminate* realizations $x \sim p(x|\theta_0)$, labeled $y = 0$, from equally sampled realizations $x \sim p(x|\theta_1)$, labeled $y = 1$. Indeed, for this task, the decision function

$$d^*(x) = p(y = 1|x) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)} \quad (2.11)$$

which models the optimal Bayes classifier [55] leads to the likelihood ratio via

$$r(x|\theta_0, \theta_1) = \frac{d^*(x)}{1 - d^*(x)}. \quad (2.12)$$

2.3.1 Likelihood-to-reference ratio

In the context of parameter inference, we are interested in the likelihood ratio between arbitrary hypotheses. A solution proposed by Cranmer et al. [55] is to condition the classifier with $\theta \sim p(\theta)$ and train $d(x|\theta)$ to distinguish pairs $(\theta, x) \sim p(\theta, x)$ from pairs $(\theta, x) \sim p(\theta)p(x|\theta_r)$, where θ_r is a fixed reference hypothesis. In this setting, the decision function modeling the optimal Bayes classifier is

$$d^*(x|\theta) = \frac{p(x|\theta)}{p(x|\theta) + p(x|\theta_r)}, \quad (2.13)$$

thereby defining the *likelihood-to-reference* (LTR) [58] ratio

$$r(x|\theta) = r(x|\theta_r, \theta) = \frac{d^*(x|\theta)}{1 - d^*(x|\theta)}. \quad (2.14)$$

The LTR ratio gives access to the likelihood ratio between arbitrary hypotheses as

$$r(x|\theta_0, \theta_1) = \frac{r(x|\theta_1)}{r(x|\theta_0)}. \quad (2.15)$$

However, Thomas et al. [59] point out that the choice of reference hypothesis θ_r has a significant effect on the approximation quality. For a realization x with null or numerically negligible likelihoods $p(x|\theta)$ and $p(x|\theta_r)$, the evaluation of the LTR ratio is numerically undefined.

2.3.2 Likelihood-to-evidence ratio

Assuming a prior $p(\theta)$, Hermans et al. [58] propose to train the classifier at discriminating between (θ, x) pairs from the joint distribution $p(\theta, x)$ and pairs from the marginal model $p(\theta)p(x)$. That is

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{p(\theta, x)p(\theta')} [\mathcal{L}(d_{\phi}(\theta, x)) + \mathcal{L}(1 - d_{\phi}(\theta', x))], \quad (2.16)$$

where \mathcal{L} is a *strictly proper scoring rule* (SPSR) [60, 61]. A scoring rule is a measure of the accuracy of probabilistic predictions. A SPSR is a scoring rule that is *uniquely* optimized by the true probabilities. Especially, for binary classification of events A and B , a SPSR satisfies

$$\arg \min_q p(A)\mathcal{L}(q) + p(B)\mathcal{L}(1 - q) = \frac{p(A)}{p(A) + p(B)}. \quad (2.17)$$

Popular examples of strictly proper scoring rules are the Brier/quadratic score $\mathcal{L}(p) = (1 - p)^2$ and the negative log-likelihood (NLL) $\mathcal{L}(p) = -\log p$. Based on these properties, Hermans et al. [58] demonstrate that the optimal discriminator for their task is

$$d^*(\theta, x) = \frac{p(\theta, x)}{p(\theta, x) + p(\theta)p(x)}, \quad (2.18)$$

from which follows the *likelihood-to-evidence* (LTE) [58] ratio²

$$r(\theta, x) = \frac{d^*(\theta, x)}{1 - d^*(\theta, x)} = \frac{p(\theta, x)}{p(\theta)p(x)} = \frac{p(x|\theta)}{p(x)} = \frac{p(\theta|x)}{p(\theta)}. \quad (2.19)$$

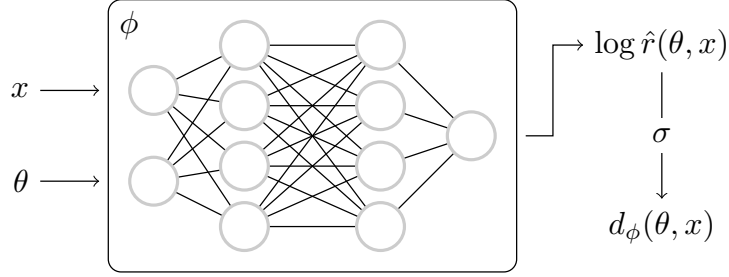


Figure 2.2. Illustration of the classifier architecture for NRE of the LTE ratio.

Unlike the LTR ratio, the LTE ratio is always numerically defined, as it is only ever evaluated where the marginal model $p(\theta)p(x)$ is strictly positive.

In practice, to prevent numerical stability issues when $d(\theta, x) \rightarrow 1$, it is the approximate log-ratio $\log \hat{r}(\theta, x)$ that is extracted from the NN and the class prediction is recovered as

$$\sigma(\log \hat{r}(\theta, x)) = \frac{1}{1 + \exp(-\log \hat{r}(\theta, x))} = d(\theta, x). \quad (2.20)$$

Like NPE, NRE of the LTE ratio gives direct access to an amortized surrogate posterior in the form of $\hat{p}(\theta|x) = \hat{r}(\theta, x)p(\theta)$. However, unlike NPE, we cannot directly sample from this surrogate, meaning that an additional stage is necessary for inference. In their work, Hermans et al. [58] apply Markov chain Monte Carlo (MCMC) [62, 63] sampling (see Algorithm 2) to their surrogate posterior, which is asymptotically ($T \rightarrow \infty$) exact.

Algorithm 2 Metropolis-Hastings MCMC sampling [62]

Input: function $f(x) \propto p(x)$, transition distribution $q(x'|x)$, initial sample x_0 , number of steps T , burn-in period B

Output: Markov chain of $p(x)$

```

1 for  $t = 1 \dots T$  do
2   sample  $x' \sim q(x'|x_{t-1})$ 
3    $\alpha \leftarrow \frac{f(x')}{f(x_{t-1})} \frac{q(x_{t-1}|x')}{q(x'|x_{t-1})}$ 
4   sample  $u \sim \mathcal{U}(0, 1)$ 
5   if  $u \leq \alpha$  then
6      $x_t \leftarrow x'$ 
7   else
8      $x_t \leftarrow x_{t-1}$ 
9 return  $\{x_t\}_{t=B}^T$ 

```

2.4 Feature imputation

The problem of posterior inference is closely related to the one of *feature imputation* (FI), *i.e.* the process of replacing missing data features by probable substitutes given the features that remain observable. However, unlike parameters and observation in posterior

²Hermans et al. [58] use the same notation $r(x|\theta)$ for the LTR and LTE ratios. To prevent any confusion, in this work, we use $r(\theta, x)$ to denote the LTE ratio.

inference, the sets of missing and observed features are not fixed, making FI a more general problem from which we can take inspiration for our problem of arbitrary MPE (see Section 1.1).

Widespread approaches to neural data generation are *variational auto-encoders* (VAEs) [64] and *generative adversarial networks* (GANs) [65]. Both methods attempt to learn a data distribution $p(x)$, or $p(x|y)$ in the case of conditional generation [66, 67], and allow sampling from this distribution. The generative adversarial imputation net (GAIN) [68] and VAE with *arbitrary conditioning* (VAEAC) [69] are FI methods. Precisely, the authors consider the problem of learning *all* conditional distributions $p(x|x_o)$, where $x_o \subseteq x$ is a subset of observable features in $x \in \mathcal{X}$.

To this end, both methods introduce a binary mask $b \in \{0, 1\}^{|\mathcal{X}|}$ that describes which features are observed. This *trick* allows to condition the generative network with respect to any subset of observed features instead of training a different one for each combination. Eventually, the only conditional distribution that is learned is $p(x|x_b, b)$, where $x_b = (x_i : b_i = 1)$.

Generalizing this concept, Belghazi et al. [70] introduce a second mask $r \in \{0, 1\}^{|\mathcal{X}|}$ declaring the features we request a substitute of. Their GAN, dubbed the neural conditioner (NC), is therefore able to sample from the arbitrary conditional *and* marginal distribution $p(x_r|x_b, b, r)$, where $x_r = (x_i : r_i = 1)$. Interestingly, the authors demonstrate empirically that the NC generalizes to mask pairs (b, r) never or barely encountered during training, suggesting a form of *continuity* across conditional/marginal distributions, essential to neural approximators [26]. This result also implies that distinct networks trained for specific conditionals/marginals would not be necessarily better at their task than a single, potentially larger, NC.

2.5 Summary and discussion

NPE and NRE NPE and NRE are the two main DL approaches to posterior estimation. They both give access to an amortized surrogate of the posterior, which can be evaluated and sampled from. However, sampling from an NRE model requires MCMC sampling, which could be computationally expensive.

On the other hand, because they are constrained to be invertible, NFs (NPE) often require involved architectures, while classifier networks (NRE) can be as simple as multi-layer perceptrons (MLPs) [26, 71]. This difference results in generally faster NRE models, even taking MCMC sampling into account. There is also a difference in *inductive bias*, as some NF transformations, especially coupling [72] and autoregressive [39, 73] ones, impose a certain structure upon the modeled distribution [34, 74].

Masking What should be remembered from FI methods is the idea of conditioning the network with masks in order to learn a single model of all conditional and marginal distributions. Li et al. [75] take inspiration from this idea to develop an arbitrary conditional normalizing flow (ACFlow) that can deal with the variable dimensionality of arbitrary conditionals and marginals, which was previously infeasible in flow models.

Due to the constraint of invertibility, dealing with this variable dimensionality requires specially designed transformations and propagation of the masks through the flow. Con-

versely, because there is no such constraint in GANs, Belghazi et al. [70] condition the NC by providing the masks as inputs. Since it does not present that constraint either, a NRE classifier could be similarly conditioned to tackle arbitrary conditional or marginal problems.

Chapter 3

Methods

Following the discussion of Section 2.5, our approach to arbitrary MPE is to apply the “masking” technique of the NC [70] to NRE and, in particular, to the amortized LTE ratio estimators of Hermans et al. [58]. The result is a novel arbitrary marginal NRE (AMNRE) method, which we describe in this chapter, alongside two comparison baselines.

3.1 Marginal NRE and NPE

As mentioned in Section 1.1, if we are interested in a few predetermined subspaces, a reasonable solution would be to train a *distinct* surrogate marginal posterior $\hat{p}(\theta_a|x)$ for each parameter subspace Θ_a . Recycling notations from Section 2.4, let $a \in \{0, 1\}^{|\Theta|} = \Omega$ representing a subspace $\Theta_a \leq \Theta$ such that $\theta_a = (\theta_i : a_i = 1) \in \Theta_a$.

The first approach we consider for this task is to train a (distinct) marginal NRE (MNRE) classifier $d_\phi(\theta_a, x)$, for all masks a in the set of masks of interest $A \subseteq \Omega$. This approach is proposed by Hermans et al. [58] and used by Delaunoy et al. [76]. Adapting (2.16), we have

$$\phi_a^* = \arg \min_{\phi} \mathbb{E}_{p(\theta, x)p(\theta')} [\mathcal{L}(d_\phi(\theta_a, x)) + \mathcal{L}(1 - d_\phi(\theta'_a, x))], \quad (3.1)$$

which allows to estimate the marginal LTE ratio with

$$\hat{r}(\theta_a, x) = \frac{d_{\phi_a^*}(\theta_a, x)}{1 - d_{\phi_a^*}(\theta_a, x)} \quad (3.2)$$

and, subsequently, the marginal posterior density as $\hat{p}(\theta_a|x) = \hat{r}(\theta_a, x)p(\theta_a)$. For simplicity, we assume that the parameters are independently drawn, *i.e.* that $p(\theta) = \prod_i p(\theta_i)$, which gives direct access to the marginal prior

$$p(\theta_a) = \prod_{i:a_i=1} p(\theta_i). \quad (3.3)$$

Our second approach is to train a marginal NPE (MNPE) distribution family $q_\phi(\theta_a|x)$ as a surrogate for $p(\theta_a|x)$, from which we can sample and evaluate the density. Similarly to (2.9), the optimal distribution parameters are retrieved as

$$\phi_a^* = \arg \max_{\phi} \mathbb{E}_{p(\theta, x)} [\log q_\phi(\theta_a|x)]. \quad (3.4)$$

3.1.1 Shared embedding

For simulators with high-dimensional or structured realizations like text, images or time series, it is common to use a neural *embedding* $h : \mathcal{X} \mapsto \mathcal{Y}$ to (pre-)process the realization

x into a vector of features $y = h(x)$ [33, 76]. This abstraction of the realizations' structure enables simpler architectures for the estimator network, like MLPs [71] or residual networks [77].

In the case of MNRE and MNPE, if the realizations are difficult to process, each marginal posterior estimator needs the full capacity (width, depth and architecture) to do so, even if it is to perform partially the same computations. To lower the capacity needed by the estimators, we propose to *share* an embedding $h_\psi(x)$ among them and train them altogether. Doing so, the objectives of MNRE and MNPE respectively become

$$\psi^*, \{\phi_a^*\} = \arg \min_{\psi, \{\phi_a\}} \mathbb{E}_{p(\theta, x)p(\theta')} \left[\sum_a \mathcal{L}(d_{\phi_a}(\theta_a, h_\psi(x)) + \mathcal{L}(1 - d_{\phi_a}(\theta'_a, h_\psi(x))) \right] \quad (3.5)$$

$$\psi^*, \{\phi_a^*\} = \arg \min_{\psi, \{\phi_a\}} \mathbb{E}_{p(\theta, x)} \left[\sum_a \log q_{\phi_a}(\theta_a | h_\psi(x)) \right], \quad (3.6)$$

where $\{\phi_a\}$ is a shorthand for $\{\phi_a \mid a \in A\}$, the set of trainable parameters of the estimators.

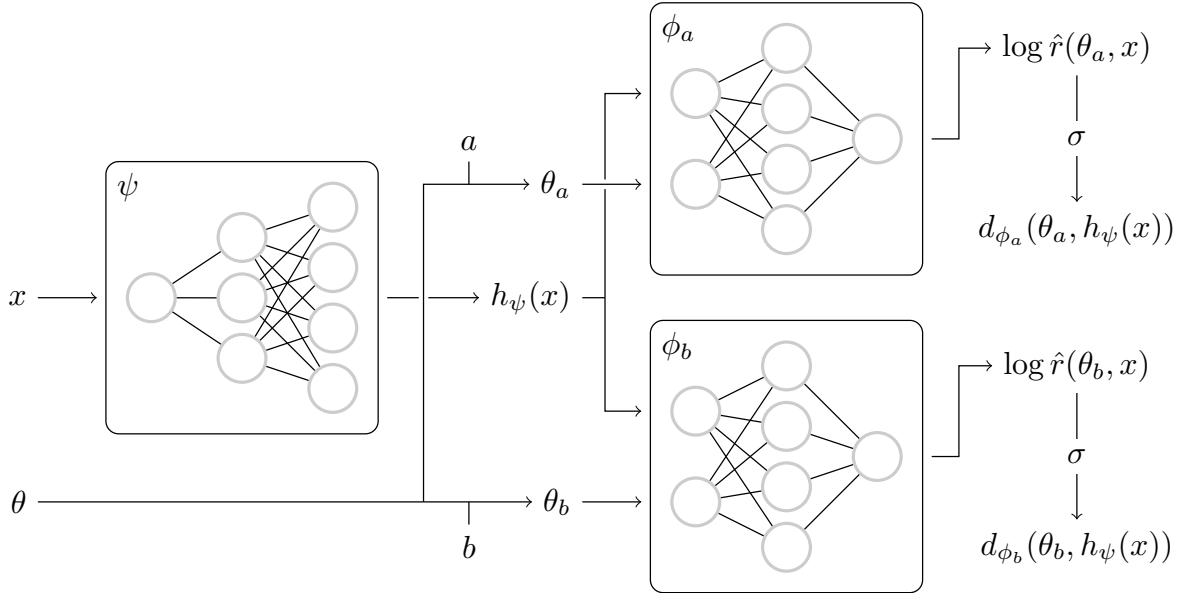


Figure 3.1. Illustration of the classifiers' architecture for MNRE with a shared embedding, considering two masks $a, b \in A$.

3.2 Arbitrary marginal NRE

Even with a shared embedding, training a distinct model for each of the $2^{|\Theta|-1}$ parameter subspaces is not reasonably feasible. Instead, we would like a single model to learn *all* the marginal posteriors $p(\theta_a|x)$. Taking inspiration from the NC [70], we propose to condition a NRE classifier of the LTE ratio [58] with an additional binary mask $a \in \Omega$ that indicates which parameters are provided.

During training, the masks are randomly sampled from a distribution $p(a)$, such that the optimization problem becomes

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{p(\theta, x)p(\theta')} \mathbb{E}_{p(a)} \left[\mathcal{L}(d_\phi(\theta_a, x, a)) + \mathcal{L}(1 - d_\phi(\theta'_a, x, a)) \right], \quad (3.7)$$

where \mathcal{L} is a SPSR (see Section 2.3.2). Reformulating this objective, we have

$$\begin{aligned}
L &= \iiint_{\Theta \times \mathcal{X} \times \Theta} p(\theta, x) p(\theta') \sum_{a \in \Omega} p(a) \left[\mathcal{L}(d(\theta_a, x, a)) + \mathcal{L}(1 - d(\theta'_a, x, a)) \right] d\theta dx d\theta' \\
&= \iint_{\Theta \times \mathcal{X}} \sum_{a \in \Omega} p(a) \left[p(\theta, x) \mathcal{L}(d(\theta_a, x, a)) + p(\theta) p(x) \mathcal{L}(1 - d(\theta_a, x, a)) \right] d\theta dx \\
&= \sum_{a \in \Omega} p(a) \iint_{\Theta_a \times \mathcal{X}} \underbrace{\left[p(\theta_a, x) \mathcal{L}(d(\theta_a, x, a)) + p(\theta_a) p(x) \mathcal{L}(1 - d(\theta_a, x, a)) \right]}_{\ell(d(\theta_a, x, a))} d\theta_a dx,
\end{aligned}$$

which is minimized only if each term $\ell(d(\theta_a, x, a))$ is itself minimized. Then, and since \mathcal{L} satisfies (2.17), the decision function that models the optimal AMNRE classifier is

$$\begin{aligned}
d^*(\theta_a, x, a) &= \arg \min_q \ell(q) \\
&= \arg \min_q p(\theta_a, x) \mathcal{L}(q) + p(\theta_a) p(x) \mathcal{L}(1 - q) \\
&= \frac{p(\theta_a, x)}{p(\theta_a, x) + p(\theta_a) p(x)}.
\end{aligned} \tag{3.8}$$

As desired, an AMNRE classifier gives access to an estimator

$$\hat{r}(\theta_a, x|a) = \frac{d_{\phi^*}(\theta_a, x, a)}{1 - d_{\phi^*}(\theta_a, x, a)} \tag{3.9}$$

of *all* marginal LTE ratios and an estimator $\hat{p}(\theta_a|x, a) = \hat{r}(\theta_a, x|a)p(\theta_a)$ of all marginal posterior densities.

In terms of network architectures, like NRE, AMNRE does not have any particular requirements, with the notable exception of the variable input size of θ_a . To make the method more convenient, in practice, θ_a is replaced by the element-wise product $\theta \cdot a$, carrying the same information at fixed size.

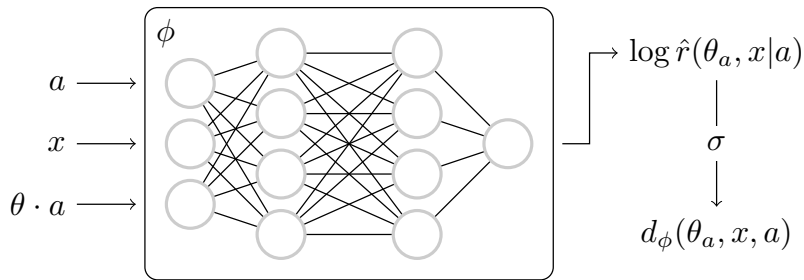


Figure 3.2. Illustration of the classifier architecture for AMNRE.

3.2.1 Masking strategy

The mask distribution is an important part of AMNRE's training. If some masks $a \in \Omega$ have a small probability $p(a)$ to be selected, it is likely that the estimator will not model their respective marginal posteriors as well as other, more frequent masks. On the other hand, as mentioned in Section 2.4, Belghazi et al. [70] demonstrate that the NC generalizes to never or barely encountered masks. To check if this property is shared by AMNRE, we consider two masking strategies, with very different distributions $p(a)$ over Ω .

Uniform masking All (non-empty) masks have the same probability

$$p(a) = \frac{1}{2^{|\Theta|} - 1} \quad (3.10)$$

to be selected. This strategy is nearly equivalent to randomly masking each parameter θ_i according to a Bernoulli distribution of probability 0.5. Hence, the average mask size (number of unmasked parameters) is slightly over $\frac{|\Theta|}{2}$.

Poisson masking The mask size $|a|$ is selected according to a Poisson distribution

$$\text{Pois}_\lambda(k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (3.11)$$

to favor low-dimensional subspaces. Formally,

$$p(a) = \binom{|\Theta|}{|a|}^{-1} \begin{cases} 0 & \text{if } |a| = 0 \\ \text{Pois}_\lambda(|a| - 1) & \text{if } |a| < |\Theta| \\ \sum_{k=|a|}^{\infty} \text{Pois}_\lambda(k - 1) & \text{else} \end{cases}, \quad (3.12)$$

where $\lambda = 1$. Here, the average mask size is slightly under $\lambda + 1$.

Chapter 4

Experiments

In this chapter, we attempt to demonstrate and compare the applicability of our methods to the problem of MPE. To this end, we design and perform a series of experiments on the methods and discuss their results for simulators of different complexities. The experiments focus on the accuracy achievable by the methods themselves, rather than by the trained models. To mitigate the differences in *inductive bias* due to network architecture, we mainly use simple networks (MLPs), with large capacities (depth and width) to ensure sufficient expressiveness. We also allocate large simulation budgets to the methods, as accuracy of the approximation is preferred over sample efficiency, from a scientific point of view.

Note. The implementation of the methods, simulators and experiments is made available at <https://github.com/francois-rozet/ammre>. The majority of the code is written in Python and the neural networks are built and trained using the PyTorch [49, 78] automatic differentiation framework. We also rely on nflows [79] to implement NF networks and matplotlib [80] to display results graphically.

4.1 Quality assessment

Because the likelihood is by definition intractable, in LFI, it is usually challenging to guarantee an accurate surrogate posterior, which is mandatory before drawing any sort of scientific conclusions based on its predictions. In this section, we review the tools we use to assess and compare the accuracy of our models, either quantitatively or qualitatively.

4.1.1 Receiver operating characteristic

A widespread indicator of a binary classifier’s performance is the *receiver operating characteristic* (ROC) curve, which is obtained by plotting the false positive rate (FPR) of the classifier against its true positive rate (TPR), at various threshold settings. If the classifier is unable to discriminate between the two classes, the FPR and TPR are equal at any threshold and the ROC curve is diagonal. Accordingly, the higher the performance of the classifier, the more the curve deviates from the diagonal and the larger the *area under the curve* (AUC) is. Hence, the AUC of the ROC curve, or ROC AUC, is a measure of the quality of classifiers.

For our NRE-based methods, constructing the ROC curve is straightforward as the trained model is already a LTE classifier. For NPE-based methods, the surrogate marginal posterior(s) $\hat{p}(\theta_a|x)$ must first be transformed into a decision function ranging from 0 to 1.

Specifically, since $\hat{r}(\theta_a, x) = \frac{\hat{p}(\theta_a|x)}{p(\theta_a)}$ is an estimator of the LTE ratio,

$$d(\theta_a, x) = \frac{\hat{r}(\theta_a, x)}{1 + \hat{r}(\theta_a, x)} = \left(1 + \frac{p(\theta_a)}{\hat{p}(\theta_a|x)}\right)^{-1} \quad (4.1)$$

is a decision function approximating the optimal LTE classifier (see Section 2.3.2). Therefore, we can construct ROC curves of both NRE and NPE-based models for the same discrimination task, which allows to compare their performances.

Adversarial ROC

In their work, Hermans et al. [58] point out that if a surrogate LTE ratio $\hat{r}(\theta, x)$ is exact, a classifier would not be able to distinguish samples from the likelihood $p(x|\theta^*)$ and the reweighted evidence model $\hat{r}(\theta^*, x)p(x)$, for arbitrary parameters θ^* . Thus, the discriminative performance of a sufficiently powerful classifier on the latter task is an indicator of the exactness of $\hat{r}(\theta^*, x)$. In this case, the ROC curve and ROC AUC are not used to assess the performance of the classifier but rather the difficulty of its task: discriminating between a distribution and an approximation thereof. In some sense, the surrogate ratio is an *adversary* for the classifier.

Generalizing this concept, we propose to train a classifier $c_\phi(\theta_a, x)$ at discriminating between the joint distribution $p(\theta_a, x)$ and the reweighted marginal model $\hat{p}(\theta_a, x) = \hat{r}(\theta_a, x)p(\theta_a)p(x)$, that is,

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{p(\theta_a, x)} [\mathcal{L}(c_\phi(\theta_a, x))] + \mathbb{E}_{\hat{p}(\theta_a, x)} [\mathcal{L}(1 - c_\phi(\theta_a, x))], \quad (4.2)$$

which allows to assess the quality of $\hat{r}(\theta_a, x)$ on the full parameter space, instead of a single point θ^* . In practice, we sample from $p(\theta_a)p(x)$ and reweight the samples by $\hat{r}(\theta_a, x)$, *i.e.*

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{p(\theta, x)p(\theta')} [\mathcal{L}(c_\phi(\theta_a, x)) + \hat{r}(\theta'_a, x)\mathcal{L}(1 - c_\phi(\theta'_a, x))], \quad (4.3)$$

which is mathematically equivalent. We recognize here an objective similar to the one of MNRE, indicating that an *adversarial classifier* can be trained using almost the same routines as a MNRE classifier.

4.1.2 Earth mover's distance

Like the Kullback-Leibler (KL) divergence [81], the *earth mover's distance* (EMD) [82] is a measure of the distance between two distributions. Specifically, the EMD is the minimum *cost* of transforming a distribution $p(x)$ over a space \mathcal{X} into another distribution $q(x)$. Formally, if $\gamma(x, y)$ describes the density moved from x to y and $c(x, y)$ is the price for moving density from x to y ,

$$\text{EMD}(p, q) = \inf_{\gamma \in \Gamma(p, q)} \iint_{\mathcal{X} \times \mathcal{X}} \gamma(x, y) c(x, y) \, dx \, dy \quad (4.4)$$

where $\Gamma(p, q)$ is the collection of all joint distributions $\gamma(x, y)$ with marginals $p(x)$ and $q(y)$. The price $c : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$ is a *metric* over the space \mathcal{X} , *i.e.* a function that defines a concept of distance between the members of \mathcal{X} . The EMD is a more powerful tool to compare distributions than the KL divergence, as it takes the geometry of the probability

space into account. However, this power comes at a cost, as finding the best transport plan γ^* comes down to solving an *optimal transportation* (OT) [42] problem over the metric space (\mathcal{X}, c) , which is computationally expensive in a general metric space.

Cuturi [83] propose an entropic regularization of the transportation problem, turning it into a (strictly) convex problem that can be solved with matrix scaling algorithms, including Sinkhorn-Knopp’s fixed point iteration [84] algorithm which is known to have linear convergence [85]. Importantly, the optimal solution γ_λ of this regularized transportation problem, where λ is an adjustable parameter, is guaranteed to converge to the optimal transport plan γ^* as $\lambda \rightarrow \infty$ [83]. Therefore, the *dual-Sinkhorn divergence* [83] over the metric space (\mathcal{X}, c)

$$D_\lambda(p, q) = \iint_{\mathcal{X} \times \mathcal{X}} \gamma_\lambda(x, y) c(x, y) \, dx \, dy \quad (4.5)$$

satisfies

$$\text{EMD}(p, q) = \lim_{\lambda \rightarrow \infty} D_\lambda(p, q). \quad (4.6)$$

In our experiments, we use the dual-Sinkhorn divergence¹ as a computationally cheap but accurate approximation of the EMD between distribution histograms.

Histograms

Histograms are a convenient and versatile numerical representation of continuous distributions. They allow to marginalize distributions easily, obtain intrinsic quantities of the distributions, like their entropy, or measure the difference between distributions, for instance using the EMD. In one or two dimensions, they also help to understand the distributions through visualization of the modes and confidence regions.

For these reasons, in our experiments, we make extensive use of histograms. In particular, for low-dimensional parameter subspaces Θ_a , we cut the subspace into uniformly spaced grid cells/bins (*e.g.* $100 \times 100 \times \dots$ cells) and evaluate the surrogate marginal posterior(s) $\hat{p}(\theta_a|x)$ at the center of each of these cells, thereby creating a histogram of the surrogate distribution.

Unfortunately, this routine is not tractable for high(er)-dimensional subspaces, as the number of cells grows exponentially with the dimensionality of the subspace. In this case, we sample a (large) population of points from the surrogate and build the histogram by counting the number of points within each bin. Still, it is not tractable either to store a value for each of the bins.

Instead, we store a value only for bins that contain at least one point, assuming the others are empty. This kind of data structure is referred to as a *sparse* array and allows to store and manipulate very large but almost empty arrays. `PyTorch` already implements sparse arrays, but does not provide tools to compute histograms in multiple dimensions. Conversely, `NumPy` [86] and `SciPy` [87] give access to such tools but only implement sparse matrices, *i.e.* 2-d arrays. Furthermore, because our neural surrogates are hosted on graphics processing units (GPUs), their samples are as well and it is preferable to process them on-device with GPU-acceleration, which is not supported by `NumPy`.

¹We select $\lambda = 100$ as D_λ typically approximates the EMD with accuracy when λ exceeds 50 [83].

Because of these limitations, we decided to re-implement the histogram routines of `NumPy` within the `PyTorch` framework. The result is `torchist`², a small `Python` package to build and manipulate dense and sparse histograms, with full support for GPU-acceleration. The package also provides routines (functions) to compute the entropy, KL divergence and EMD (as the dual-Sinkhorn divergence) of dense and sparse histograms.

4.1.3 Calibration

Given an observation x^* , a q -credible region is a subset $S \subseteq \Theta$ into which parameters θ sampled from the posterior $p(\theta|x^*)$ have a probability q to fall. Formally,

$$q = \mathbb{E}_{p(\theta|x^*)} [\mathbb{1}(\theta \in S)] = \int_S p(\theta|x^*) d\theta. \quad (4.7)$$

Among all the q -credible regions, the smallest, *i.e.* the one with the fewest members, is the region that contains the parameters of highest posterior density, sometimes called the highest posterior density region (HPDR).

In their work, Delaunoy et al. [76] use an amortized surrogate posterior $p(\theta|x)$ to predict the 0.5 and 0.9-HPDRs for realizations $x^* \sim p(x|\theta^*)$ and check whether they contain the true parameters $\theta^* \sim p(\theta)$ or not. If the expected surrogate posterior is consistent with the prior, *i.e.* if

$$p(\theta) \approx \mathbb{E}_{p(x)} [\hat{p}(\theta|x)], \quad (4.8)$$

θ^* should be in the two predicted regions 50 % and 90 % of the time, respectively. It is equivalent to state that the smallest probability q such that the q -HPDR contains θ^* ,

$$q = \int_{\Theta} \hat{p}(\theta|x^*) \mathbb{1}[\hat{p}(\theta|x^*) \geq \hat{p}(\theta^*|x^*)] d\theta, \quad (4.9)$$

should be *uniformly* distributed over $[0, 1]$ for pairs $(\theta^*, x^*) \sim p(\theta, x)$. We can assess visually that q is correctly distributed by plotting it against its empirical cumulative density function (CDF), which should be diagonal. If not, the surrogate is not correctly *calibrated* with respect to the prior.

This calibration test can also be performed with the percentile rank p of the true parameters θ^* in the surrogate posterior, instead of q , since

$$p = \int_{\Theta} \hat{p}(\theta|x^*) \mathbb{1}[\hat{p}(\theta|x^*) < \hat{p}(\theta^*|x^*)] d\theta = 1 - q. \quad (4.10)$$

In our experiments, we apply this latter version of the test to our amortized surrogate marginal posteriors $\hat{p}(\theta_a|x)$. To compute the percentile of θ_a^* , we can either 1) sample a large number of parameters θ_a from the surrogate $\hat{p}(\theta_a|x^*)$ and calculate the proportion that satisfies the condition $\hat{p}(\theta_a|x^*) \leq \hat{p}(\theta_a^*|x^*)$ or 2) compute an accurate histogram of $\hat{p}(\theta_a|x^*)$ and integrate the region within which the same condition is satisfied.

Since we cannot sample from our NRE-based surrogates without MCMC sampling, we select the second option, which can only be performed for low-dimensional subspaces Θ_a . As the construction of the histogram has to be repeated for numerous pairs (θ_a^*, x^*) , this calibration test is very expensive in practice. Due to time and resource constraints, we only consider 1-dimensional marginal surrogates for this test.

²<https://github.com/francois-rozet/torchist>

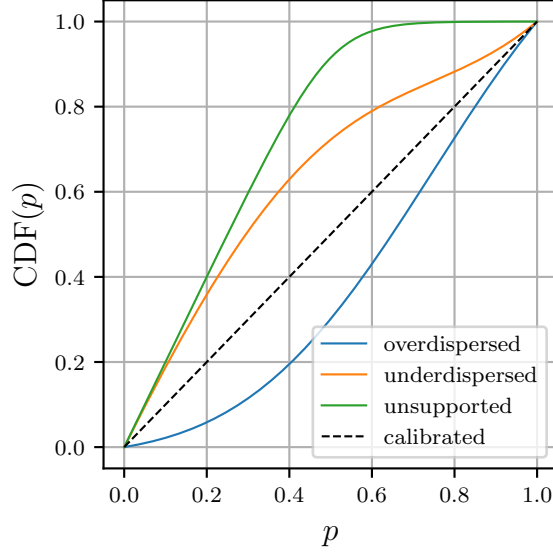


Figure 4.1. This figure demonstrates the visual inspection of the percentiles’ CDF for the calibration test of a surrogate posterior $\hat{p}(\theta|x)$ presented in Section 4.1.3. If the CDF lies close to the diagonal, the surrogate is well calibrated. When the surrogate is hesitant or overdispersed, unlikely parameters are ranked too high, likely parameters too low and the CDF is under the diagonal. Conversely, when the surrogate is too confident or underdispersed, the CDF is over the diagonal. Finally, when some percentiles are severely under-represented, the CDF is flat for them. For instance, when the true posterior barely supports the HPDR of the surrogate, the CDF is almost horizontal for high percentiles.

Note. This calibration test is actually a special case of *simulation-based calibration* (SBC) [88] where the random variable $f(\theta)$ is chosen as the estimated posterior density $\hat{p}(\theta|x^*)$. The reader is invited to consult the work of Talts et al. [88] for a more in-depth analysis.

4.2 Simulators

Simple likelihood and complex posterior Papamakarios et al. [30] introduce a toy simulator where $\theta \in \mathbb{R}^5$ parametrizes a 2-d multivariate Gaussian from which four points are independently sampled to construct a realization x . The generative process is

$$\begin{aligned} \theta_i &\sim \mathcal{U}(-3, 3) \quad \text{for } i = 1, \dots, 5 \\ \mu &= (\theta_1, \theta_2) \\ s_1 &= \theta_3^2, \quad s_2 = \theta_4^2, \quad \rho = \tanh(\theta_5) \\ \Sigma &= \begin{pmatrix} s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2 \end{pmatrix} \\ x &= (z_1, \dots, z_4) \quad \text{where } z_j \sim \mathcal{N}(\mu, \Sigma), \end{aligned}$$

for which the likelihood $p(x|\theta) = \prod_j p(z_j|\theta)$ is tractable. Despite its *simple likelihood*, the simulator has a *complex posterior* (SLCP) with four symmetric modes due to the squaring of θ_3 and θ_4 . SLCP is a non-trivial posterior estimation benchmark that allows to retrieve the ground-truth (GT) posterior through MCMC sampling [62, 63] of the likelihood.

Hodgkin-Huxley In neuroscience, the Hodgkin-Huxley (HH) model [3] is a widespread non-linear mechanistic model of neural dynamics for which numerous parameter inference methods have been proposed.

In particular, Gonçalves et al. [12] successfully apply SNPE (see Section 2.2.2) to infer the posterior over the eight parameters $\theta = (g_{\text{Na}}, g_{\text{K}}, g_{\text{l}}, g_{\text{M}}, \tau_{\text{max}}, -V_T, \sigma, -E_l)$ of a HH simulator given summary statistics x of the electro-physiological recording (number of spikes, mean and standard deviation of the resting potential, mean, standard deviation, skewness and kurtosis of the voltage). The prior distribution of the parameters is considered uniform, *i.e.* $\theta \sim \mathcal{U}(b_l, b_h)$, between the parameter space boundaries $b_l = (0.5, 10^{-4}, 10^{-4}, 10^{-4}, 50, 40, 10^{-4}, 35)$ and $b_h = (80, 15, 0.6, 0.6, 3000, 90, 0.15, 100)$.

As we do not have any knowledge in the domain, we borrow the HH simulator from the official code³ released by Gonçalves et al. [12] and use the implementation as a black box.

Gravitational waves In recent years, the observations of gravitational waves (GW) from compact binary coalescences (CBCs) have had a massive impact on our understanding of the Universe, partly thanks to inference of the systems' parameters. To obtain posterior samples, the LIGO/Virgo collaboration (LVC) currently applies MCMC or nested sampling [89, 90] algorithms to involved physical models of the likelihood of emitted waves [91–93]. With these approaches, posterior calculation typically takes days for binary black hole (BBH) mergers and has to be repeated from scratch for each observation, like ABC (see Section 2.1).

With the primary intent to perform fast(er) inference, Green et al. [32] apply NPE (see Section 2.2.2) over the full 15-dimensional set of precessing quasi-circular BBH parameters, conditioned on GW observations from the LVC detectors. They evaluate their network on data surrounding the first recorded gravitational-wave event, GW150914, and demonstrate that the approach performs inference in close agreement with conventional sampling methods.

As they evaluate their network on real observations, the generation of realistic waveforms is a key part of their work. The processing of the waveforms is also important as it radically changes the representation of the realization fed to the network. Indeed, Green et al. [32] compress the frequency-domain waveforms to a reduced basis corresponding to the first 100 components of a singular value decomposition (SVD).

For our experiments, we borrow the waveform simulator and processing pipelines from the official code⁴ released by the authors, but make a few modifications:

- The 15 parameters of the simulator are detector-frame masses (m_1, m_2) , reference phase ϕ_c , time of coalescence t_c , luminosity distance d_L , spin magnitudes (a_1, a_2) , spin angles $(\theta_1, \theta_2, \phi_{12}, \phi_{JL})$, inclination angle θ_{JN} , polarization angle ψ and sky location (α, δ) . All parameters are independent, with the exception of the masses that satisfy $10 M_\odot \leq m_2 \leq m_1 \leq 80 M_\odot$. To obtain a completely independent prior, we replace m_2 by the mass ratio

$$q = \frac{m_2}{m_1} \in [0.125, 1]. \quad (4.11)$$

³https://github.com/mackelab/IdentifyMechanisticModels_2020

⁴<https://github.com/stephengreen/lfi-gw>

- Although a prior uniform in volume, $d_L^3 \sim \mathcal{U}(100^3, 1000^3)$, would be more physical, we adopt a prior uniform in distance, $d_L \sim \mathcal{U}(100, 1000)$, to better cover the parameter space.
- We use 128 SVD components instead of 100 in the waveform reduced basis. It should be noted that 1) frequency-domain waveforms are represented by complex-valued numbers and 2) an observation corresponds to two waveforms from two geographically distant detectors (H1 and L1). Hence, a single realization is composed of 512 real-valued numbers.
- Because waveform generation is costly to perform in real time, Green et al. [32] sample “intrinsic” parameters and save associated waveform polarizations h_+ and h_\times , in advance of training. At train time, they sample “extrinsic” parameters, project h_+ and h_\times onto detectors and add Gaussian noise. In our implementation, we sample all parameters at once to generate and process the waveforms ahead of training. The noise is also added during training to prevent overfitting.

	SLCP	HH	GW
Tractable likelihood	Yes	No	No
Parameters	5	8	15
Realization size	8	7	512

Table 4.1. Summary of the simulators’ characteristics.

4.3 Experimental protocol

In this section, we present in details the experiments we perform. In the following sections, if an experiment differs from what is presented here, it will be explicitly stated.

Datasets For each simulator, we use three fixed datasets of pairs $(\theta, x) \sim p(\theta, x)$ to train, validate and test the methods, respectively. SLCP and HH have an additional training set to train adversarial classifiers (see Section 4.1.1). The sizes of the datasets are provided in Table 4.2.

	SLCP	HH	GW
Training set	1 048 576	1 048 576	4 194 304
Validation set	131 072	131 072	131 072
Testing set	131 072	131 072	131 072
Adversarial set	1 048 576	1 048 576	—

Table 4.2. Dataset sizes for each simulator.

Methods The considered methods are NRE, NPE, MNRE, MNPE and AMNRE. In the case of MNRE and MNPE, the subspaces of interest are all one and two-dimensional subspaces. For SLCP, we add the full space $\Theta_a = \Theta$ of parameters, for a total of 16 subspaces of interest. For HH, we also consider the full space, as well as the subset

$(g_{\text{Na}}, g_{\text{K}}, g_{\text{I}}, -V_T)$ of parameters, for a total of 38 subspaces of interest. We do not use a shared embedding by default. For AMNRE, we use the uniform masking strategy.

Architectures For NRE-based methods, the NN is a MLP with 7 hidden layers of 256 neurons and ELU [94] activation functions. For NPE-based methods, the NF is a MAF [39] (see Appendix B) with 7 transformations⁵, each parametrized by a MLP with 3 hidden layers of 128 neurons and ReLU [95] activation functions, and a unit Gaussian base distribution. For adversarial classifiers, the NN is a MLP with 11 hidden layers of 512 neurons and ELU [94] activation functions.

Statistic	NRE classifier	NPE flow	Adversarial classifier
Architecture	MLP	MAF	MLP
Parameters [–]	464 385	369 222	2 896 897
Evaluation rate [batch/s]	830 ± 13	195 ± 7	415 ± 4
Sampling rate [batch/s]	–	34 ± 2	–

Table 4.3. Various estimator statistics for the SLCP simulator. Rates evaluated with batches of 1024 elements on a single GTX 1080Ti GPU.

In all architectures, a static up-front layer standardizes the input parameters θ_i with respect to their mean and variance in the training set.

Training All models are optimized with the AdamW [46, 47] stochastic optimization algorithm. At each epoch, the batches are built by sampling without replacement from the training set. The number of batches per epoch is 256, the batch size is 1024, the weight decay is 10^{-3} and the initial learning rate is 10^{-3} . We apply a “Reduce On Plateau” scheduling of the learning rate, that is, we reduce the learning rate by a factor 2 each time the loss on the validation set has not decreased for 7 consecutive epochs. The training stops when the learning rate reaches 10^{-6} or lower.

Hyperparameter	Default
Optimizer	AdamW
Weight decay	10^{-3}
Batches per epoch	256
Batch size	1024
Initial learning rate	10^{-3}
Scheduling	Reduce On Plateau
Reduce factor	2
Patience	7
Stopping learning rate	10^{-6}

Table 4.4. Default training hyperparameters.

⁵MAFs with a larger number of transformations presented convergence issues during our experiments and more expressive parameter networks did not improve the quality of estimations. These issues are further discussed in Chapter 5.

In the case of NRE-based methods, the independent parameters θ' are obtained by shifting circularly ($i \leftarrow i + 1$ and $n \leftarrow 1$) the batch of parameters θ . For AMNRE, each element in the batch has a different mask, sampled from the mask distribution. For MNRE and MNPE, all estimators are trained at once (see Figure 3.1), even without shared embedding. Finally, the NLL $\mathcal{L}(p) = -\log p$ is chosen as SPSR in NRE’s objective (2.16).

Training is repeated 5 times, leading to 5 model instances for each method and each simulator. Each instance is evaluated separately, before aggregating the results.

Evaluation All evaluation tests are performed on the testing set. The ROC and adversarial ROC curves are built for a few subspaces of different sizes.

As mentioned in Section 4.1.2, high-dimensional histograms are built from sampled populations. For NPE-based methods, we sample $2^{22} = 4\,194\,304$ points from the surrogate (marginal) posteriors. For NRE-based methods, the population is sampled with MCMC sampling (see Algorithm 2). To guarantee a representative population, we generate 4192 independent Markov chains of $T = 16\,384$ steps with a burn-in period of $B = 8192$ steps and a Gaussian transition distribution⁶. The same settings are used when sampling from the tractable likelihood of SLCP to retrieve the ground-truth posterior (see Section 4.2). In both low and high-dimensional subspaces Θ_a , dimensions are discretized into 100 uniformly spaced bins.

For histograms that are not built from populations, we compute the total probability $P = \sum_i p_i$. Ideally, this value should be 1, as it indicates the probability of θ_a being in Θ_a . For the following tests, the bins are normalized by P , *i.e.* $p_i \leftarrow \frac{p_i}{P}$.

When the ground-truth posterior is available (SLCP), we measure the accuracy of our surrogate marginal posteriors as the EMD between their histogram and the marginalized histogram of the ground-truth posterior. We also evaluate the consistency of surrogates over different subspaces as the EMD between their histograms, marginalized on their common subspace.

For the calibration test, the empirical CDF is built from percentiles computed in histograms of 256 bins, for 8192 pairs (θ^*, x^*) of the testing set.

⁶The Gaussian distribution is centered around the previous step x_{t-1} with a small standard deviation. Precisely, the standard deviations are selected to be 2 % of the parameter space dimensions. For instance, in SLCP, where $\theta_i \in [-3, 3]$, the standard deviation is $0.02 \times (3 + 3) = 0.12$.

4.4 Results

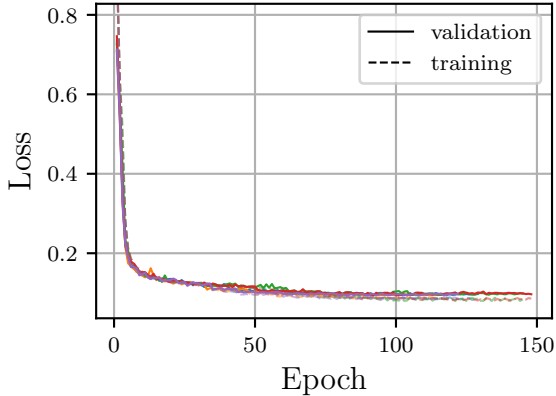
In this section, we present and discuss the results of our experiments.

Simple likelihood and complex posterior

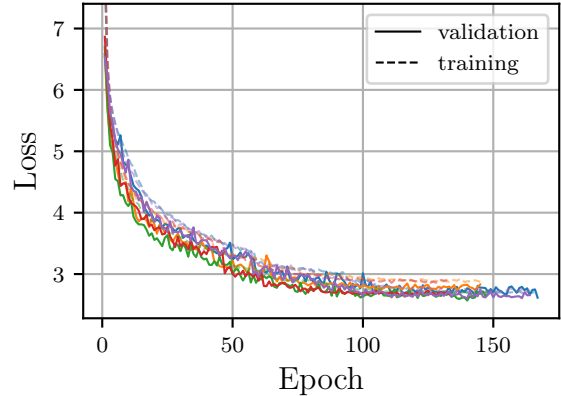
As can be observed in Figure 4.2, all models seem to converge towards an optimum consistent over the 5 instances, with the exception of MNPE for which some instances fail to reach the same optimum as the others.

A second observation is that training and validation losses stay close to each other, indicating little overfitting. However, the three NRE-based methods demonstrate a stronger correlation between the training and validation losses than NPE and MNPE, which could be due to the *supervised* nature of NRE’s task. This observation also indicates that the training set is representative of the validation set.

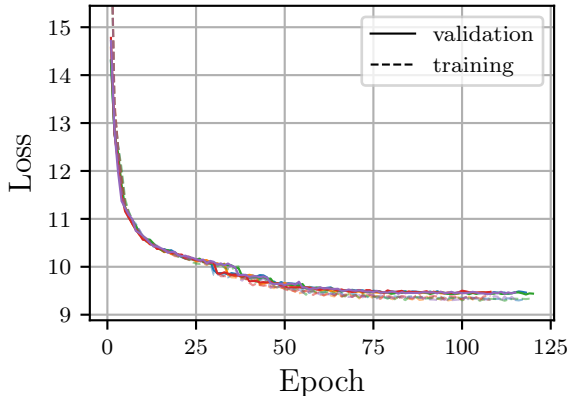
We also notice that AMNRE’s loss is higher than the one of NRE, but lower than the one of MNRE, averaged over the subspaces. This is not surprising as NRE classifier can use the full set of parameters to perform discrimination, while only half (2.5) of those are provided to AMNRE classifier on average (see Section 3.2.1). Similarly, due to our selection of subspaces of interest (mostly 1-d and 2-d), less than two parameters are provided to MNRE classifiers, on average.



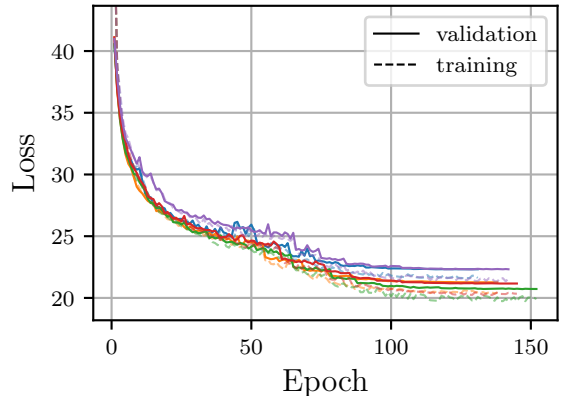
(a) NRE



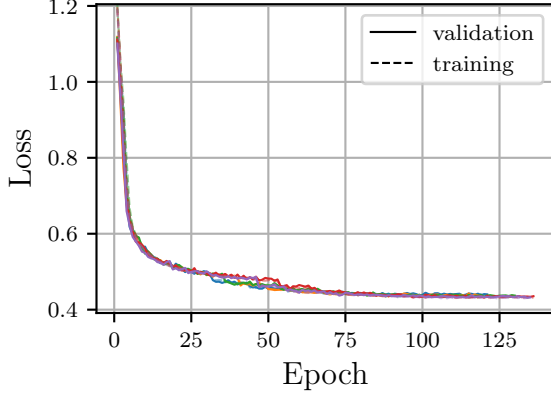
(b) NPE



(c) MNRE (16 subspaces)



(d) MNPE (16 subspaces)

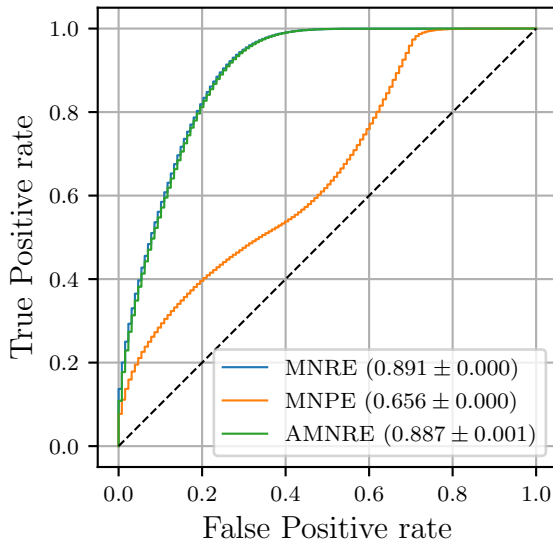


(e) AMNRE

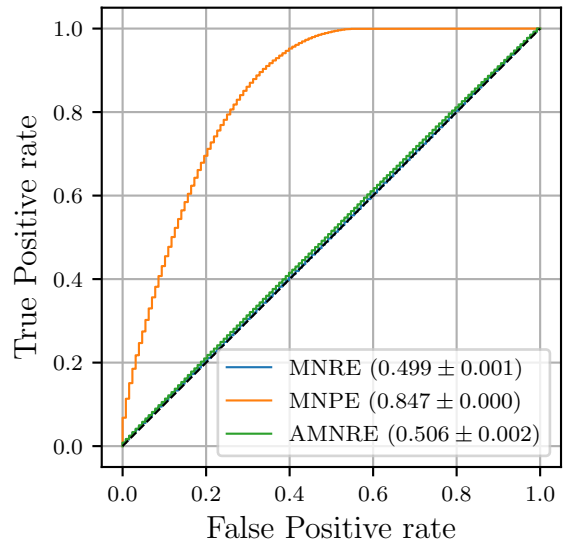
Figure 4.2. Mean training and validation losses of SLCP surrogate models. Each color corresponds to a different model instance. All methods converge without clear signs of overfitting.

This interpretation is confirmed by the ROC curves (see Figure 4.3) of the classifiers, which get closer to the upper left corner as the number of provided parameters increases. We note that AMNRE’s performance, measured by the AUC, is very close to the performance of MNRE, despite using only one network for all subspaces. MNPE also performs similarly to MNRE, with the exception of the parameter subset θ_3 , for which it is significantly worse.

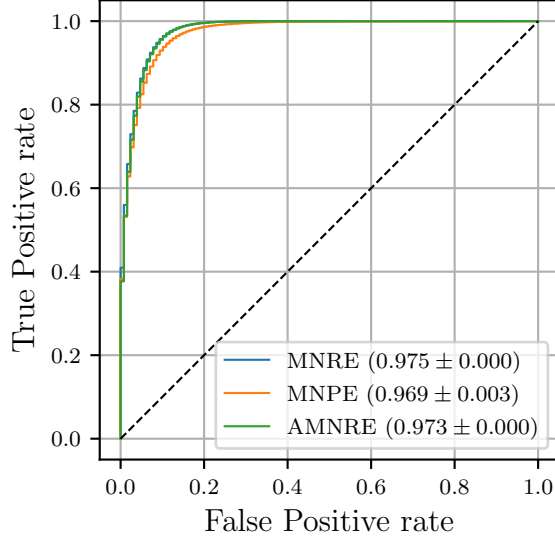
This is explained by the inability of a MAF to model one-dimensional distributions more complex than its base distribution, as mentioned in Appendix B. Since the base distribution is a unit Gaussian, the flow is unable to model the two modes of θ_3 posterior. As expected, adversarial classifiers detect this limitation, which translates into high adversarial AUC (see Figure 4.3b). Conversely, for MNRE and AMNRE, adversarial classifiers are not able to discriminate, which is manifested by their diagonal ROC curves.



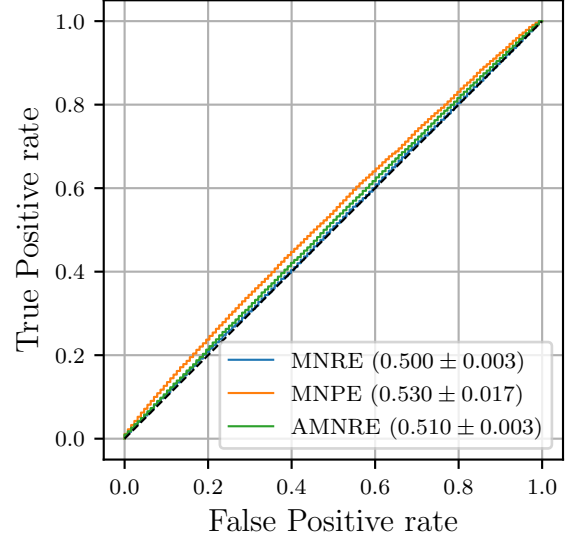
(a) Subset θ_3



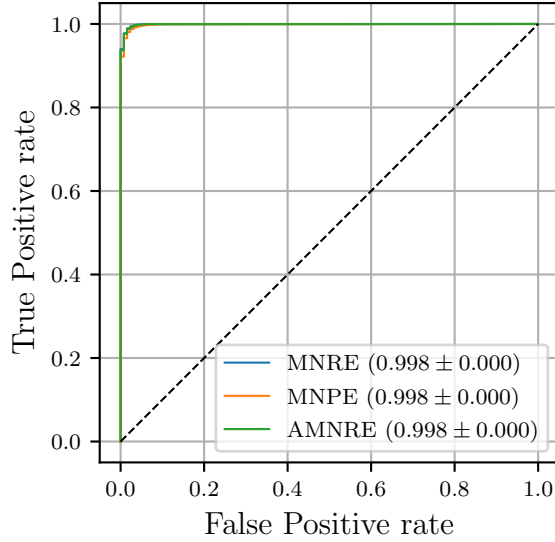
(b) Subset θ_3 , adversarial



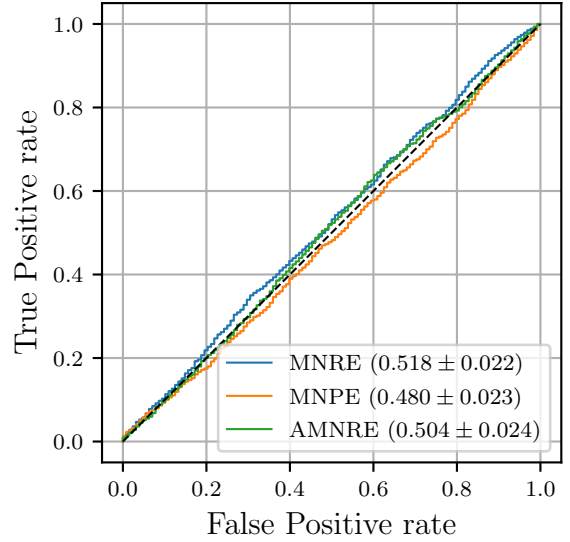
(c) Subset (θ_3, θ_4)



(d) Subset (θ_3, θ_4) , adversarial



(e) Full set θ



(f) Full set θ , adversarial

Figure 4.3. ROC and adversarial ROC curves of SLCP surrogate models. Curves are averaged over the model instances. The AUC's mean and standard deviation are given in the legend. For the full set θ , MNRE (resp. MNPE) is equivalent to NRE (resp. NPE). The performance of classifiers increases with the number of provided parameters. Adversarial classifiers are not able to detect significant inaccuracies.

For multi-dimensional subspaces, MAF is not limited anymore and MNPE catches up with MNRE and AMNRE. Particularly, adversarial classifiers are not able to detect significant discrepancies between the full surrogate posteriors and the ground-truth posterior. To some extent, this is supported by visual inspection of 1-d and 2-d HPDRs, for a realization of the testing set (see Figure 4.4). Unfortunately, because these surrogates are 5-dimensional distributions, it is difficult to analyze them further.

Nevertheless, for low-dimensional surrogate marginal posteriors, we can build accurate histograms and measure the differences with the marginalized ground-truth posterior.

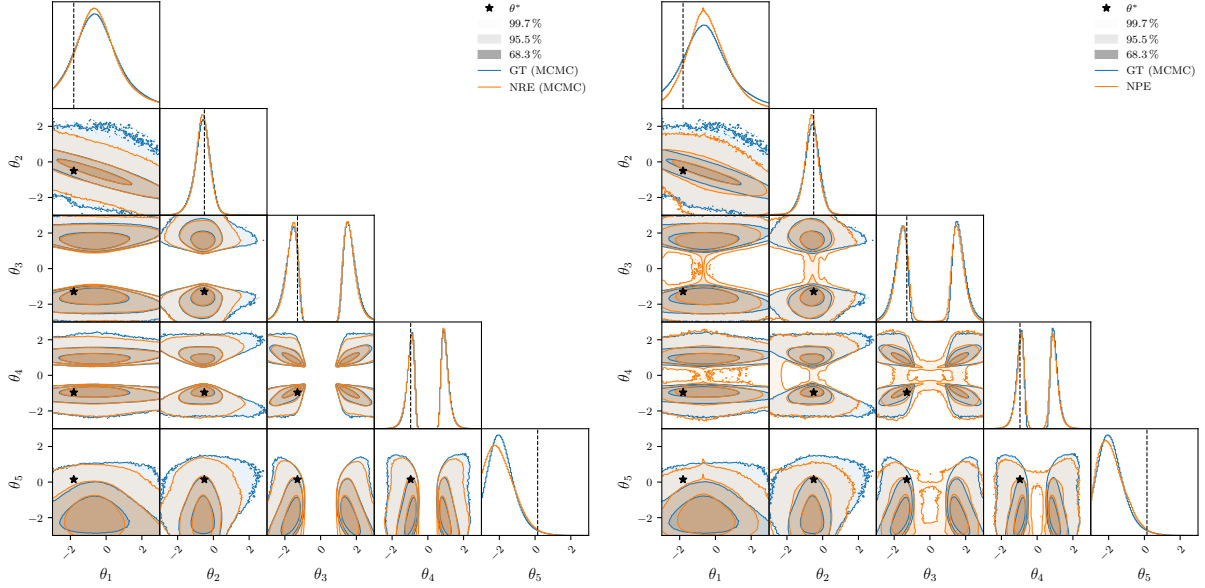


Figure 4.4. Ground-truth posterior against NRE (left) and NPE (right) surrogate SLCP posteriors, marginalized in one and two-dimensional subspaces, for a realization x^* of the testing set. Density is averaged over the model instances. Contours represent the 68.3%, 95.5 % and 99.7 % HPDRs. Stars represent the true parameters θ^* of the realization. (M)NRE and (M)NPE approximate correctly the structure of the ground-truth posterior. (M)NPE present some density leaks outside of the ground-truth posterior support.

First, let us consider the total probability of these histograms. We observe in Figure 4.5 that all marginal methods present high variance in the total probability of their histograms. This could indicate 1) density leaks outside of the subspace Θ_a , 2) spikes or dips in the density that are lost during the discretization, 3) an estimator that is not a probability measure, *i.e.* that does not integrate to 1, or a combination thereof. Although, for MNPE, only the first two options are possible as, by construction, a NF always defines a probability measure.

Unlike MNRE and AMNRE, MNPE has a tendency to underestimate the total probability, especially for the θ_3 and θ_4 parameters, for which MAF cannot model the two modes, as discussed previously. This particularly stands out in the second plot of Figure 4.5, where the EMD to the ground-truth θ_3 and θ_4 posteriors is up to five times larger for MNPE than for MNRE. Considering all the subsets, we observe that MNRE and AMNRE diverges from the ground-truth very similarly, *i.e.* with close EMD mean and variance, while MNPE is generally less accurate, especially for 1-d and multi-modal posteriors (*e.g.* (θ_2, θ_3) or (θ_1, θ_4) subsets).

If there are perceivable differences between the ground-truth and the surrogates, there are also differences between the surrogates themselves. Having consistent surrogate marginal posteriors, *i.e.* surrogates that are close when marginalized on their common subspace, is one of the challenges of MPE. For instance, the surrogate posteriors over (θ_1, θ_2) and (θ_2, θ_3) should be equivalent when projected onto θ_2 . As explained in Section 4.3, we use the EMD between (marginalized) histograms to quantify this consistency. Doing so for all 1-d and 2-d subspaces, we obtain a consistency *matrix*, which can support interesting discussions.

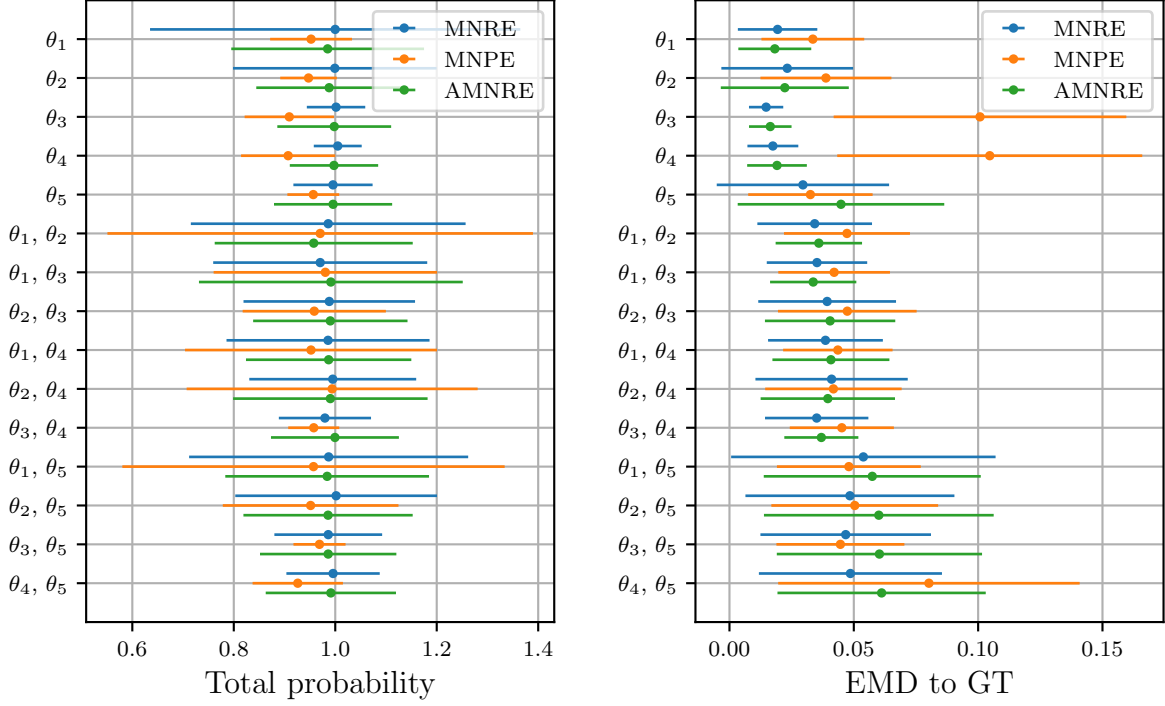
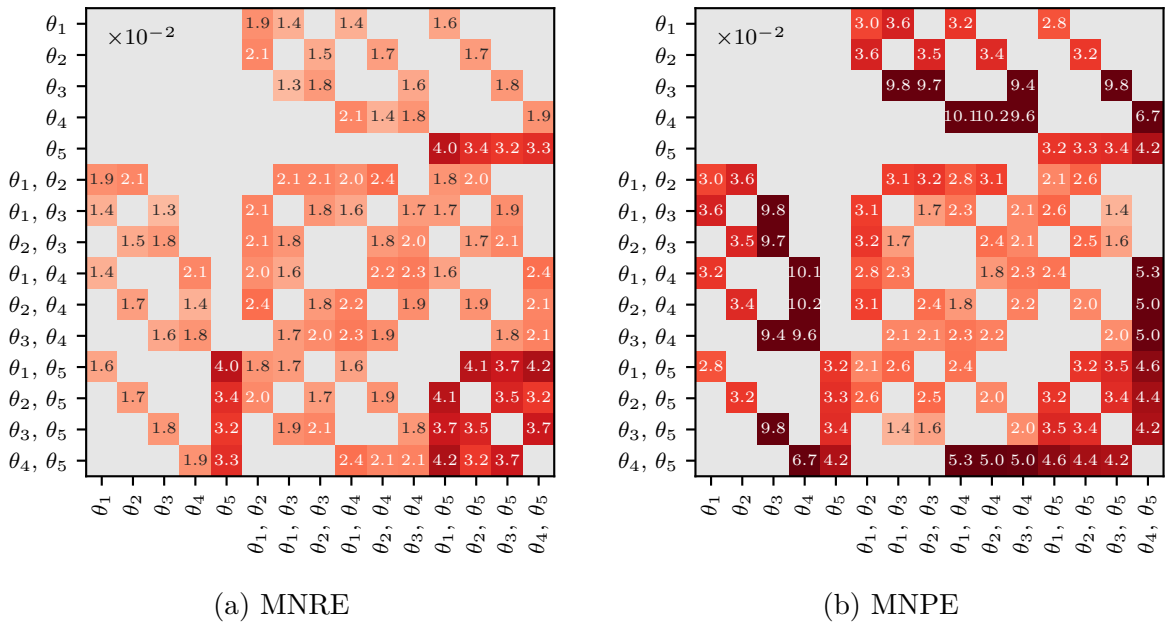
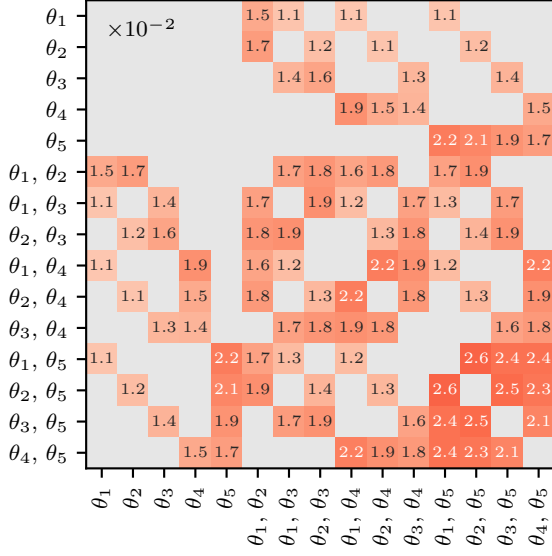


Figure 4.5. Total probability (left) and EMD to the marginalized ground-truth posterior (right) of 1-d and 2-d surrogate marginal SLCP posterior histograms. The bars represent the quantity mean and standard deviation over 64 realizations from the testing set and the model instances. MNPE tend to underestimate slightly the total probability, on average. All methods have high total probability variance. MNPE diverges more from the ground-truth than MNRE and AMNRE, on average.

In Figure 4.6a, we observe that MNRE surrogate marginal posteriors for subsets containing θ_5 are less consistent with each others than subsets containing other parameters. The same disparity appears for AMNRE, although less significantly. This could indicate

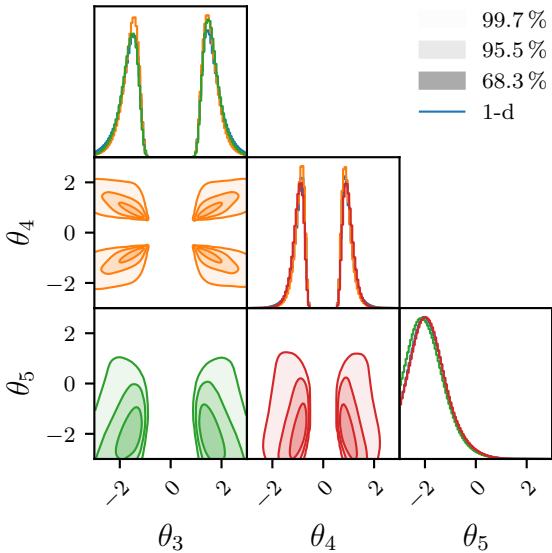




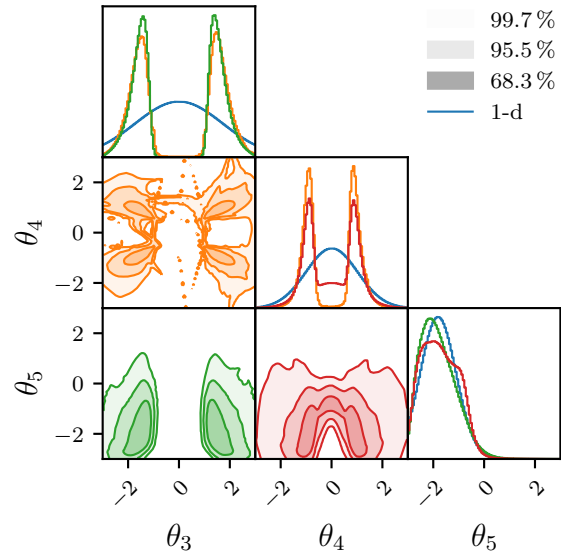
(c) AMNRE

Figure 4.6. EMD between 1-d and 2-d surrogate marginal SLCP posterior histograms. Values are averaged over 64 realizations from the testing set and the model instances. Darker colors indicate larger average EMDs. Cells are greyed out at the intersection of two marginal posteriors that are equal or that do not have a common subspace. AMNRE surrogates are more consistent than MNRE's. MNPE has consistency issues.

that, among the five parameters of SLCP, θ_5 is the hardest to infer. This hypothesis is supported by Figure 4.5, as the EMD to the marginalized ground-truth posterior is significantly higher, on average, for subsets containing θ_5 .



(a) MNRE



(b) MNPE

Figure 4.7. 1-d and 2-d surrogate marginal posteriors over a subset of SLCP's parameters, for a realization of the testing set. Density is averaged over the model instances. Projections of the 2-d surrogates onto their 1-d subspaces are drawn with the same color. MNPE surrogates present artifacts in low density regions.

In the consistency matrix of MNPE, we notice once again the limitations of MAF for the θ_3 and θ_4 parameters, as their surrogates are heavily inconsistent with other 2-d surrogates. MNPE seems to also have difficulties with the (θ_4, θ_5) subset, but not (θ_3, θ_5) . Since SLCP is symmetric with respect to θ_3 and θ_4 , this difference is likely due to model instances not reaching the optimum during training (see Figure 4.2d). The said inconsistencies can be visualized in Figure 4.7, for a realization of the testing set.

In the same figure, we notice that MNPE poorly models the low density regions of the posteriors. We do not believe this phenomenon to be due solely to the MAF architecture, as the problem also occurs for simple, Gaussian-like posteriors (see Figure A.2). In our opinion, the *unsupervised* nature of NF training is partially accountable: where NRE extracts knowledge from both high and low density samples during training, NPE only leverages high density samples.

Overall, for SLCP, MNRE and AMNRE are fairly equivalent. The former is slightly more accurate, but the latter has more consistent surrogates. In line with the findings of Belghazi et al. [70] (see Section 2.4), this suggests a form of continuity across the marginal posteriors that AMNRE takes advantage of. In conclusion of this section, for SLCP, AMNRE definitely enables consistent arbitrary MPE. However, SLCP is toy simulator with few parameters and, thus, not a sufficient benchmark.

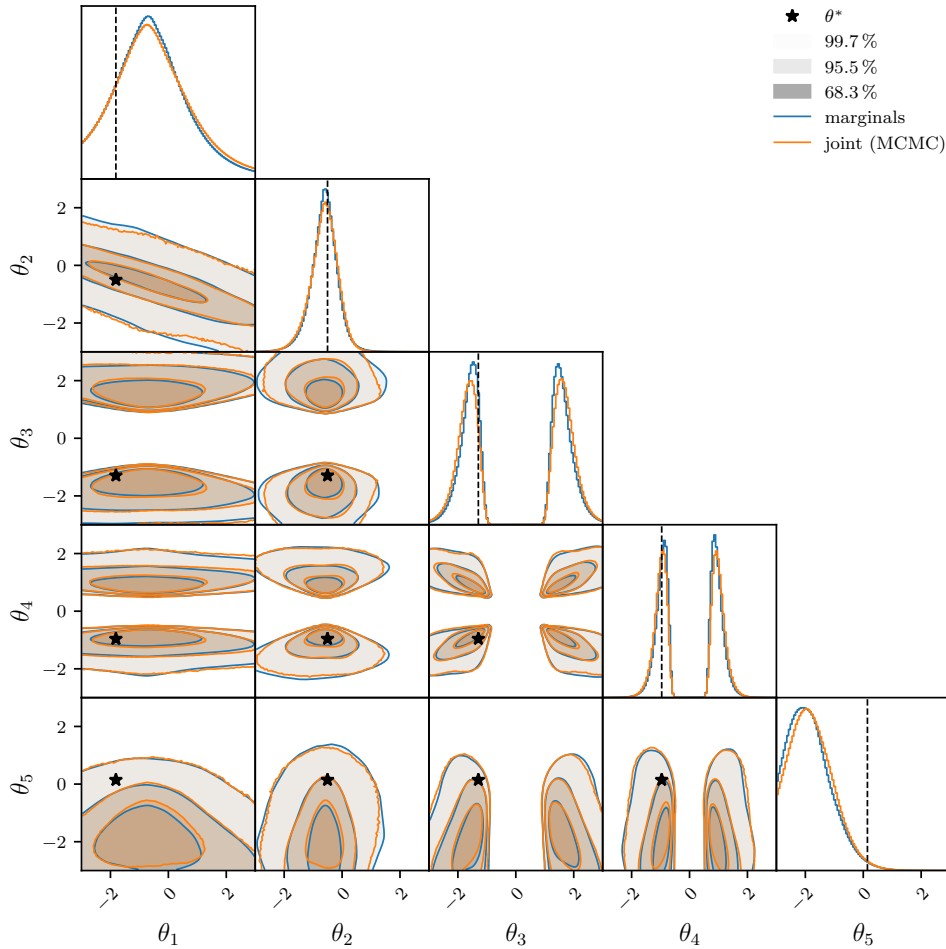


Figure 4.8. AMNRE surrogate joint SLCP posterior against 1-d and 2-d surrogates, for a realization of the testing set. Density is averaged over the model instances. AMNRE performs consistent arbitrary MPE for the SLCP simulator.

Hodgkin-Huxley

Thanks to its higher-dimensional parameter space and more complex generative process, the HH simulator is a more challenging benchmark than SLCP. So much so that NRE and NPE do not agree on the structure of the posterior (see Figure 4.9), which will be discussed later on.

With HH, we wish to evaluate two additional (sub-)methods: MNRE with a shared embedding and AMNRE with Poisson masking (see Chapter 3). In particular, for the former, we introduce, as embedding, an MLP with 7 hidden layers of 256 neurons and ELU [94] activation functions which processes the realization into a vector of 256 features. Additionally, we reduce the capacity of the classifiers to 3 hidden layers of 64 neurons. To prevent any ambiguity, in figures and tables, MNRE with shared embedding and AMNRE

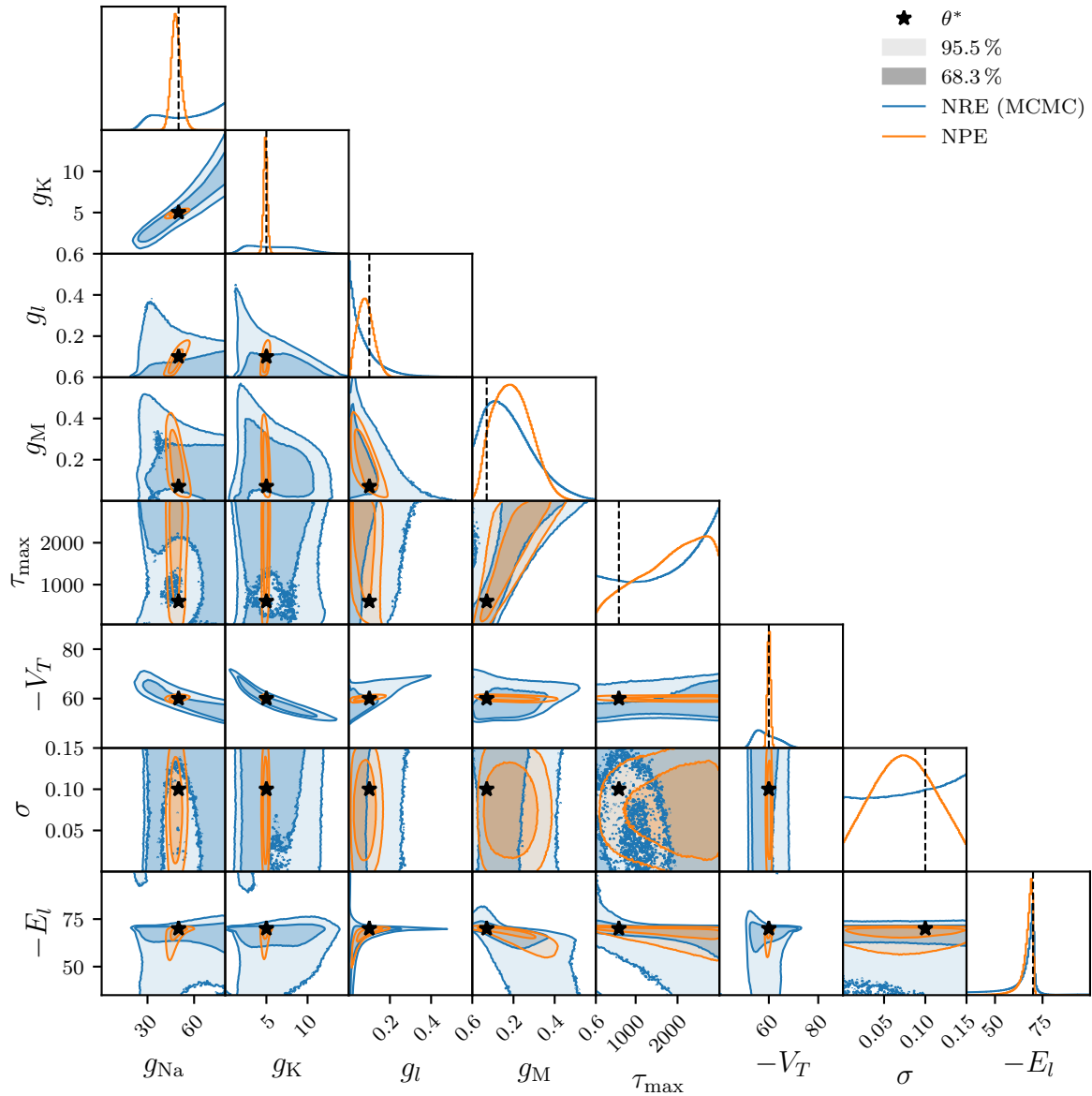


Figure 4.9. NRE against NPE surrogate HH posteriors, marginalized in one and two-dimensional subspaces, for a realization of reference [12]. Density is averaged over the model instances. NRE and NPE do not agree on the posterior structure. NPE’s approximation is significantly less dispersed than NRE’s.

with Poisson masking are denoted MNRE-s and AMNRE-p, respectively. AMNRE with uniform masking, which is the default, is sometimes denoted AMNRE-u.

Like for SLCP, all methods converge without signs of overfitting (see Figure A.4). The performance of classifiers, measured by their ROC AUC, also increases with the number of provided parameters (see Figure A.5). However, unlike what we observed in the previous section, adversarial classifiers are able to detect inaccuracies in the surrogate marginal posteriors. Especially, all methods seem to perform worse in higher-dimensional subspaces.

A reasonable explanation is that, for this simulator, the support of the joint distribution $p(\theta, x)$ is very small with respect to the marginal model $p(\theta)p(x)$. Then, the regions in

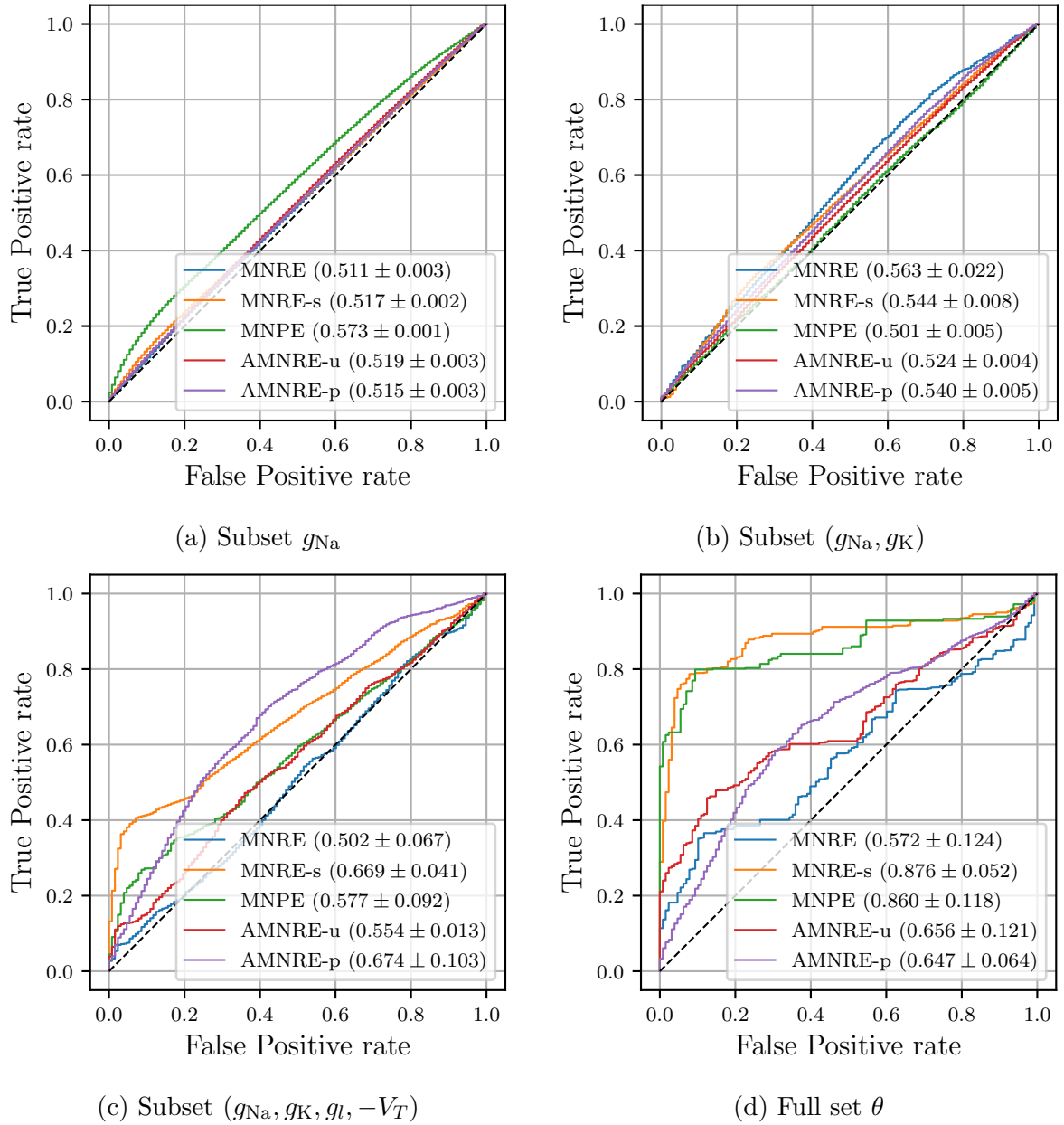


Figure 4.10. Adversarial ROC curves of HH surrogate models. Curves are averaged over the model instances. Adversarial classifiers detect more and more inaccuracies as the number of provided parameters increases. The AUC variance also increases.

which the models should struggle, *i.e.* where $r(\theta, x) \approx 1$, are rare and do not impact the training objective significantly. Consequently, NRE classifiers do not have enough incentive to learn correctly the ratio in these regions and NPE flows do not stumble upon low(ish) density samples enough to model correctly the associated regions. Fortunately, adversarial classifiers are not affected by this phenomenon as their task is not to discriminate between high and low density samples but rather to spot *out-of-distribution* (OOD) samples in the reweighted marginal model $\hat{p}(\theta_a, x)$.

Nevertheless, in low-dimensional subspaces Θ_a , the distribution $p(\theta_a, x)$ is much more dispersed than $p(\theta, x)$, which allows to apply successfully our marginal methods. As can be seen in Figures 4.11 and A.8, MNRE, MNPE and AMNRE seem to agree with the marginalized surrogate posterior of NPE (*cf.* Figure 4.9). However, we observe that the

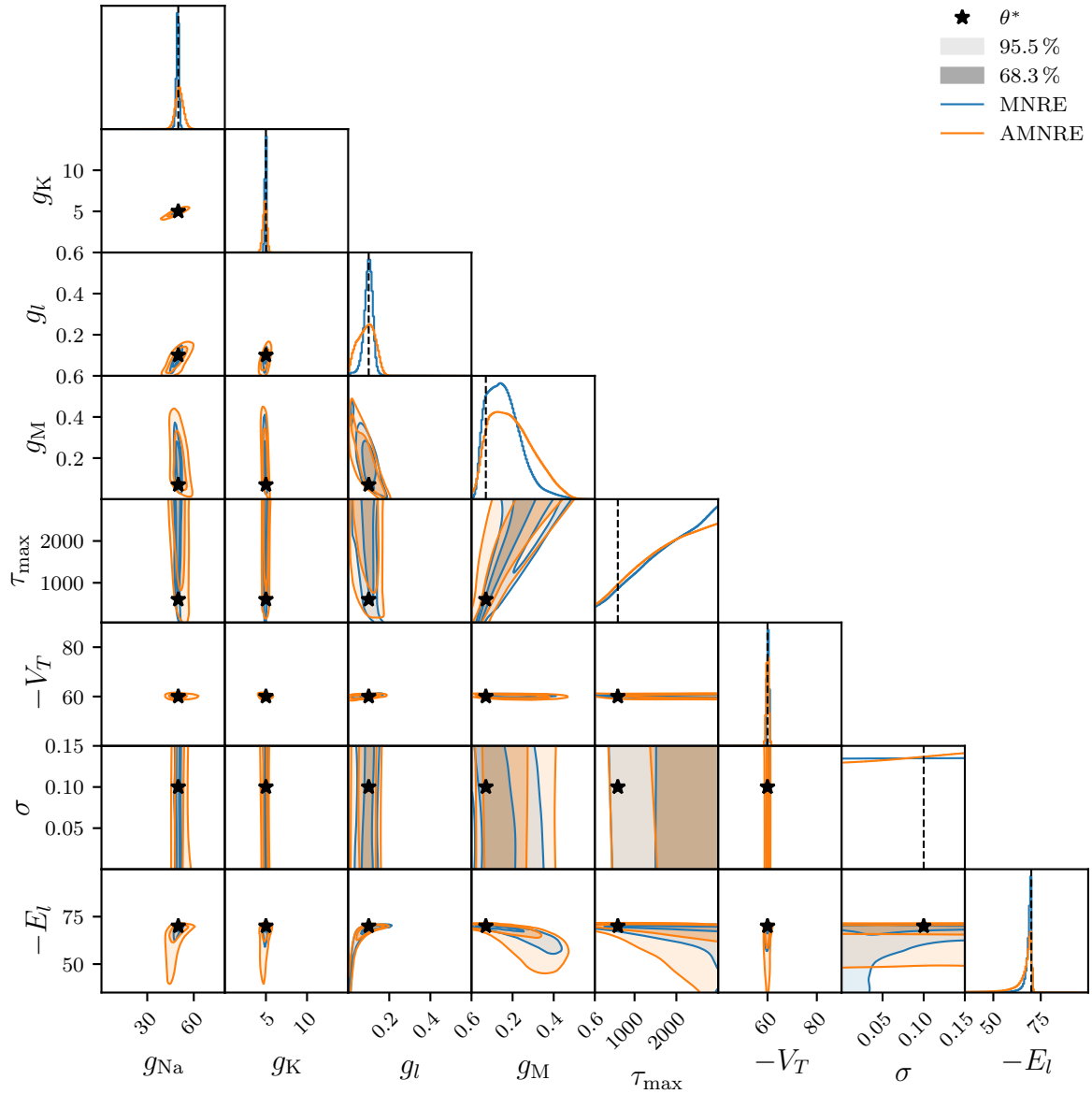


Figure 4.11. MNRE against AMNRE 1-d and 2-d surrogate marginal HH posteriors, for a realization of reference. Density is averaged over the model instances. MNRE and AMNRE surrogates have a similar structure, but the former is more confident, *i.e.* less dispersed, than the latter.

HPDRs of MNRE surrogates are more narrow than those of AMNRE and MNPE, which could either indicate that MNRE is overconfident or that AMNRE is underconfident.

Presented in Section 4.1.3, calibration tests allow to detect miscalibrated posterior approximations, including overconfident and underconfident ones. In Figure 4.12, we observe that the percentile CDFs of MNRE surrogates are above the diagonal, indicating overconfident/underdispersed predictions. Conversely, AMNRE’s CDFs are closer to the diagonal, which corresponds to better calibrated surrogates, with the exception of the $-E_l$ parameter.

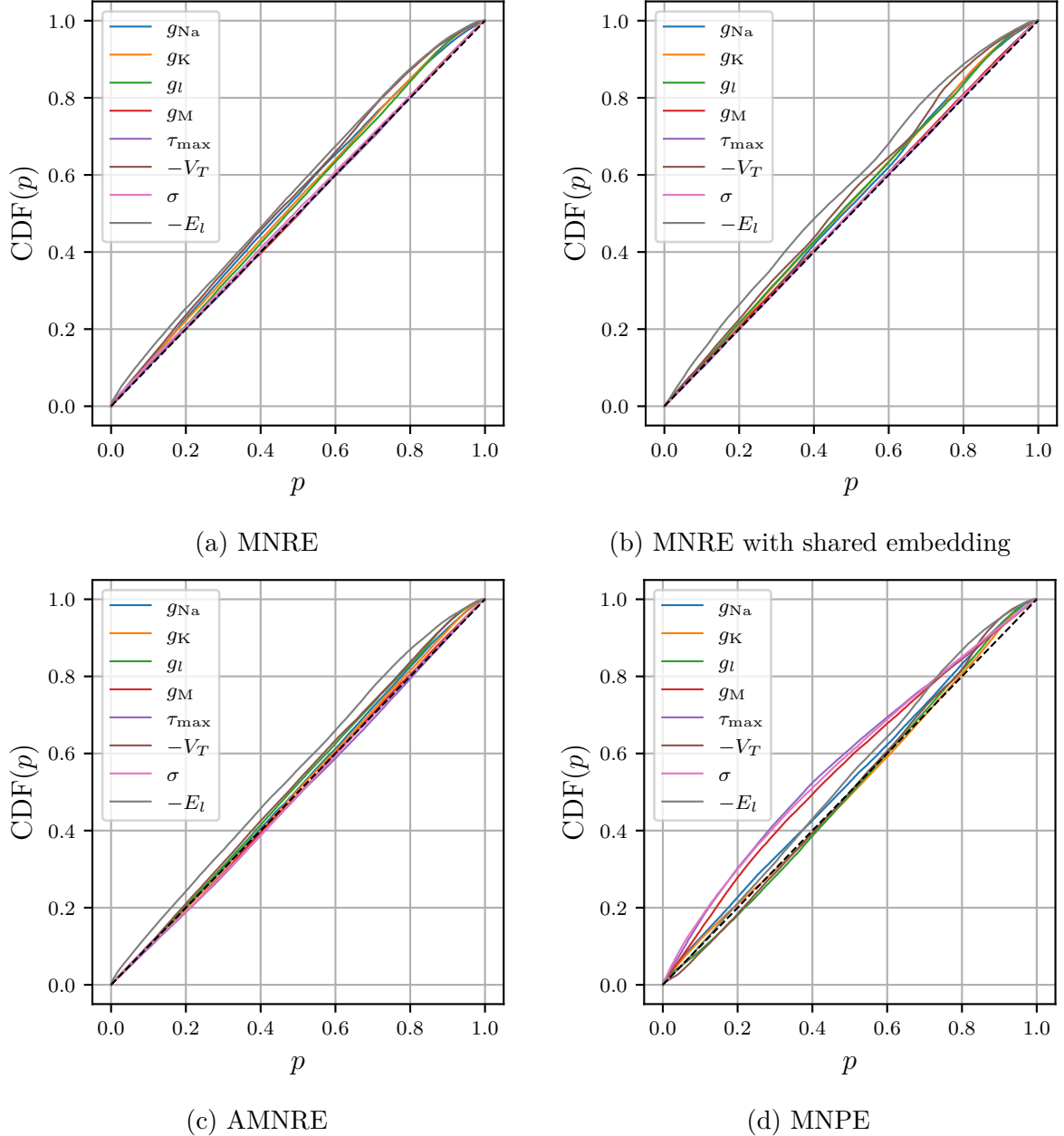


Figure 4.12. Calibration tests of 1-d surrogate marginal HH posteriors. Percentile CDFs are averaged over the model instances. All models are almost calibrated. MNRE surrogates are less well calibrated than AMNRE’s. Using a shared embedding (MNRE) or does not seem to alter significantly the calibration. MNPE surrogates for the g_M , τ_{max} and σ parameters are miscalibrated.

It should be kept in mind that the calibration test does not judge of the accuracy of approximations, but, rather, of their consistency with the prior. In particular, a surrogate that is equal to the prior, *i.e.* $\hat{p}(\theta|x) = p(\theta)$, would be perfectly calibrated. Therefore, this test allows us to say that AMNRE surrogates are better calibrated than MNRE’s, but not that they are more accurate. The overconfidence issue and how to alleviate it using regularizing loss functions is discussed in Appendix C. For MNPE, we observe a strong miscalibration for the g_M , τ_{\max} and σ parameters. As for θ_3 and θ_4 in SLCP, this is due to the limitations of MAF, which cannot model non-Gaussian 1-d distributions.

Concerning the variants of MNRE and AMNRE, we note that sharing an embedding among smaller classifiers does not deteriorate or improve significantly the accuracy of MNRE in low-dimensional subspaces. However, the smaller classifiers are not expressive enough to model the higher-dimensional marginal posteriors, as demonstrated by their adversarial ROC curves (see Figure 4.10). In addition to slightly faster convergence (see Figure A.4) and lightweight architecture (a tenth of the parameters), an unsuspected benefit of sharing an embedding is to have more consistent surrogates (see Figure A.7). In particular, MNRE with shared embedding has more consistent surrogates than AMNRE, which is itself more consistent than MNRE without shared embedding.

The opposite happens for AMNRE with Poisson masking as it significantly deteriorates the consistency of AMNRE surrogates, although it does not seem to affect their accuracy in low dimension (see Figures 4.10 and A.10). In high dimension, the results are mixed as adversarial classifiers consider AMNRE-p less accurate than AMNRE-u for the subset $(g_{Na}, g_K, g_l, -V_T)$, but more accurate for the full set θ . Therefore, it is difficult to draw any conclusions, especially with the large variability of the AUCs. Nevertheless, we note that, even though AMNRE-p has mostly access to one or two parameters during training, it learns how to discriminate when provided more parameters. Consequently, AMNRE could be trained with data in which a lot of features are missing, and still approximate correctly the full posterior.

Overall, the HH benchmark allows to demonstrate that MPE methods, and especially MNRE and AMNRE, are able to model low-dimensional marginal posteriors accurately, even when the full posterior is hard to approximate. However, it also demonstrates that further work is needed to enable accurate arbitrary MPE over all subspaces for challenging simulators.

Gravitational waves

As explained in Section 4.2, traditional methods used by the LVC (LIGO/Virgo collaboration) typically take days to evaluate the posterior of a single GW observation. Green et al. [32] demonstrate that NPE performs posterior inference over the full 15-dimensional set of BBH parameters in close agreement with these sampling methods, for a fraction of the time (minutes).

Similarly, we attempt to demonstrate that AMNRE performs inference of the 1-d and 2-d marginal posteriors in agreement with sampling methods. Because the realizations are more structured than in SLCP and HH, we introduce an embedding network to (pre)process the realizations into vectors of 512 features. This embedding is a ResNet [77] consisting of 10 residual blocks of 2 layers with 512 neurons and ELU [94] activation functions. We also increase the capacity of the main network to 11 hidden layers of 512

neurons. This architecture is small (a fifteenth of the parameters) in comparison to the flow used by Green et al. [32].

For training, the number of batches per epoch is increased to 1024, the initial learning rate is reduced to 2×10^{-4} and the scheduling patience increased to 11. Three model instances are trained, instead of five. The total training procedure, for each instance, takes about 2 hours on a single GTX 1080Ti GPU. All model instances converge to a common optimum, without signs of overfitting (see Figure 4.13).

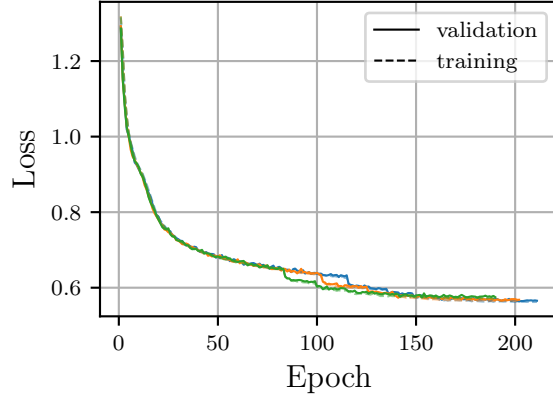


Figure 4.13. Mean training and validation losses of AMNRE surrogate models for GW. All instances converge without signs of overfitting.

Like Green et al. [32], we evaluate our model(s) on the first recorded gravitational-wave event, GW150914, which takes less than a second for all 1-d and 2-d surrogate marginal posteriors combined. As reference, we use the posterior samples produced by Bilby [92, 93] with the `dynesty` [90, 96] nested sampler. The sampler setup is borrowed from the official implementation of Green et al. [32], although the prior is modified to be consistent with our prior. It took 3 days for Bilby to complete the posterior inference of GW150914.

As can be seen in Figure 4.14, AMNRE surrogate marginal posteriors share the same structure as the marginalized posterior inferred by Bilby. For some parameter subsets, especially those containing the mass m_1 and the mass ratio q , the predictions are not accurate or, rather, not confident enough. However, for other parameters, including the luminosity distance d_L and sky location (α, δ) , the surrogates are in close agreement with Bilby.

Overall, these results are clearly less accurate than the ones of Green et al. [32]. Nevertheless, it should be reminded that our network has only a fifteenth of the parameters of theirs and take 2 hours instead of 6 days to be trained. Our results could likely be improved with a larger network and/or embedding. The sampling of extrinsic parameters at train time could also improve accuracy, although it breaks the assumption of a black box simulator. Eventually, even though these results are not impressive with respect to state-of-the-art methods, they are a promising demonstration of the applicability of AMNRE in challenging scientific settings.

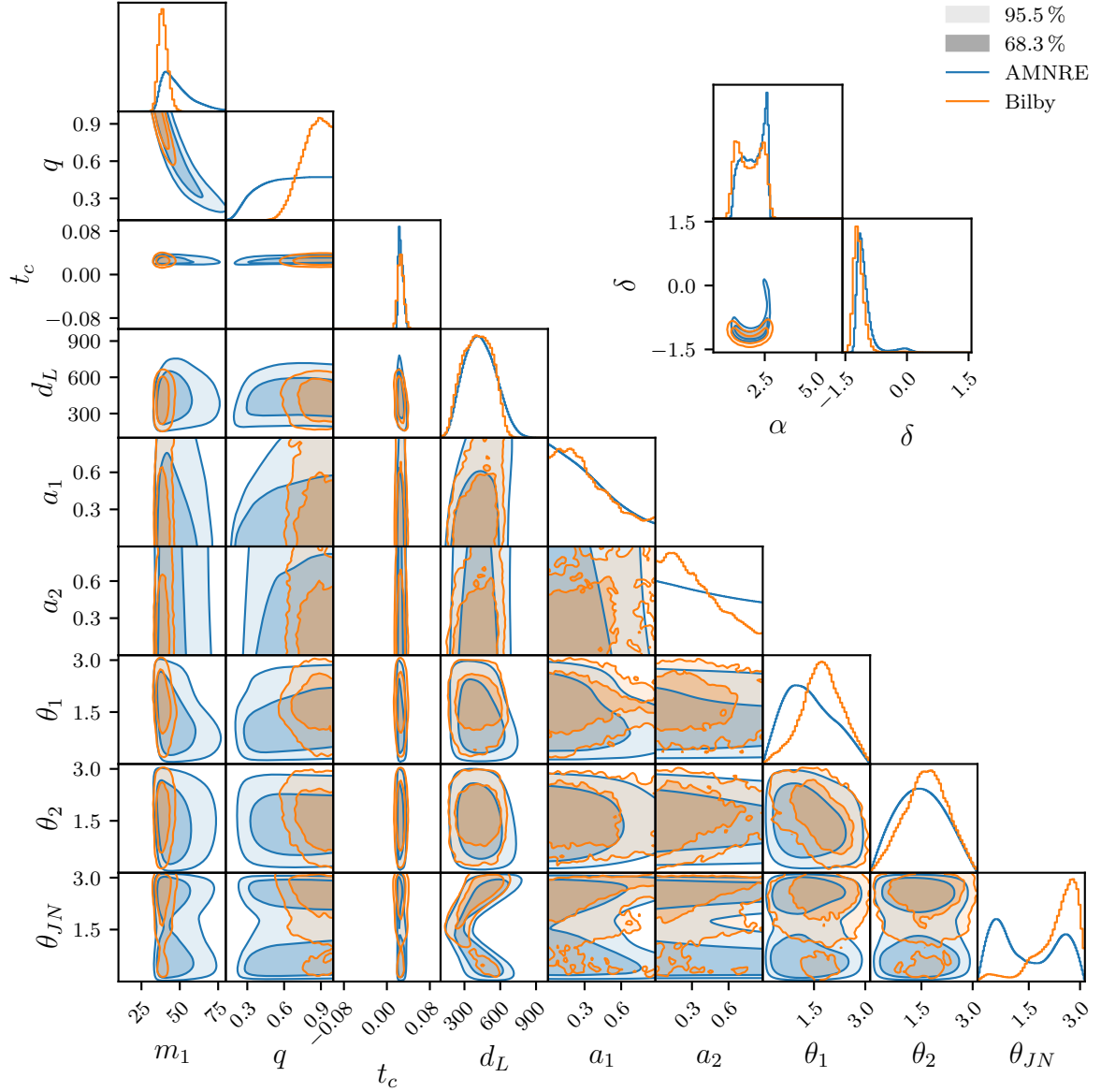


Figure 4.14. AMNRE 1-d and 2-d surrogate marginal posteriors against marginalized Bilby posterior samples over a subset of the parameters, for the GW150914 observation. Density is averaged over the model instances. Most surrogates share the same structure as Bilby’s predictions. The surrogates of the mass m_1 and mass ratio q have too much variance, *i.e.* are underconfident, but predict the correct modes. The surrogate of the inclination angle θ_{JN} has a secondary mode. The surrogates of the coalescence time t_c , luminosity distance d_L and sky location (α, δ) are in close agreement with Bilby.

Chapter 5

Conclusion

The central contribution of this work is the novel AMNRE method, which enables integration-less amortized MPE over arbitrary parameter subspaces. The method is easy to use, efficient and can be implemented with basic network architectures, like MLPs.

We demonstrate through a series of experiments the applicability of the method and compare it to two baselines, MNRE and MNPE, in which a different estimator is trained for each subspaces of interest. The results indicate that AMNRE is competitive with the baselines, even though it uses only one estimator for all subspaces. In particular, AMNRE learns accurate surrogate marginal posteriors over low-dimensional subspaces. The surrogates of AMNRE also seem to be more consistent with each others than MNRE’s or MNPE’s. However, we note that sharing an embedding between MNRE classifiers improves greatly their consistency, measured as the EMD between distribution histograms. The evaluation of the EMD, as well as the construction of histograms, is performed using `torchist`, a `Python` package to build and manipulate histograms within the `PyTorch` framework, which we developed for this thesis. Also based on histograms, a calibration test inspired by coverage tests of credible regions is used to assess if surrogate posteriors are consistent with respect to the prior. We find that some estimators, notably MNRE classifiers, are overconfident in their predictions.

For high-dimensional subspaces, the results are mixed as AMNRE performs well for the SLCP simulator but poorly for the more challenging HH simulator. We formulate and motivate the hypothesis that the latter result is due to the low entropy of HH’s posterior, which makes the discrimination task of NRE classifiers too “easy”. To diagnose this problem, we introduce a novel adversarial ROC test that is able to detect differences between the implicit joint distribution of the simulator and an approximation thereof. We detect, with this test, that neither (M)NRE nor AMNRE were able to model the full HH posterior.

Finally, we apply AMNRE to the challenging problem of BBH parameter inference from GW observation and obtain promising results, paving the way for convenient and efficient parameter inference for domain scientists.

Limitations

The main limitation of our work is the lack of comparison with alternative methods. To the best of our knowledge, ACFlow [75] is currently the only alternative that enables truly arbitrary MPE. During our work, we attempted several times to use the official implementation¹ of Li et al. [75]. Unfortunately, we were not able to adapt it to our use

¹<https://github.com/lupalab/ACFlow>

as 1) we are not familiar with the framework used by the authors (`TensorFlow`) and 2) the implementation is centered around the paper’s experiments, *i.e.* not modular.

A second limitation of our experiments is the absence of hyperparameter optimization. Especially, we use very simple network architectures with large capacities and allocate large simulation budgets. These are important aspects of our methods that we do not study. In particular, concerning MNPE, a lot of issues were linked to the MAF architecture. We suspect that more powerful NF architectures like *neural spline flows* (NSFs) [40] or *unconstrained monotonic neural networks* (UMNNs) [41] could lead to better results, although they are slower than MAF. The impact of small simulation budgets on the approximation quality is also an aspect worth inspecting.

Because NRE does not work correctly for the HH simulator, we are not able to assess properly the performance of AMNRE in high dimension. Consequently, most of our experiments focus on one and two-dimensional marginal posteriors. These are important from the point of view of domain scientists, but do not cover the full capacity of AMNRE. Using another simulator, with a more dispersed posterior, could have allowed more compelling results.

Finally, some quality assessment tools, like the consistency matrix or the calibration test, can be applied to larger than one and two-dimensional subspaces, but, because they are computationally expensive, we only apply them to the latter. With a better planning of our experiments and a more systematic approach to research, we could have presented more supporting results.

Perspectives

In this work, we sometimes scratch the surface of deep problems without actually investigating them. Here follows a list of problems, related or not to AMNRE, for which we suggest a direction of investigation.

- As mentioned previously, we observe poor behaviors of NRE-based methods for the HH simulator. Since adversarial classifiers are able to detect these behaviors, a promising idea would be to use an adversarial classifier not as diagnostic of the ratio estimator but as a correction.
- We notice on several occasions that (M)NPE tends to model poorly the low density regions of the posterior. An interesting idea would be to apply the discrimination objective of NRE to a NPE flow, reformulated as a LTE ratio estimator (see Section 4.1.1). This should lead to a NPE model that is accurate both in high and low density regions.
- The overconfidence of some surrogates is also a problem we detect. We discuss and propose regularization methods to alleviate this problem in Appendix C.
- In our experiments, the consistency of surrogates is used as a quality assessment tool. In Appendix D, we consider to impose the consistency by minimizing measures of deviation between the surrogates.

Generally, all the limitations of our work could also be the subject of future work.

Acronyms

ABC	approximate Bayesian computation	3, 4, 20
ACFlow	arbitrary conditional normalizing flow	9, 38
AD	automatic differentiation	5
AMNRE	arbitrary marginal neural ratio estimation	11, 13, 21–39, 51, 52, 54, 57, 58, 61, 62
AUC	area under the curve	15, 16, 25, 26, 32, 35
BBH	binary black hole	20, 35, 38
CBC	compact binary coalescence	20
CDF	cumulative density function	18, 19, 23, 34, 49, 66
DE	density estimation	3, 4
DL	deep learning	4, 9
ELU	exponential linear unit	22, 31, 35
EMD	earth mover’s distance	16–18, 23, 27–29, 38, 58
FI	feature imputation	8, 9
FL	focal loss	65–68
FPR	false positive rate	15
GAIN	generative adversarial imputation net	9
GAN	generative adversarial network	9, 10
GPU	graphics processing unit	17, 18, 22
GT	ground-truth	19
GW	gravitational waves	20, 21, 35, 36, 38
HH	Hodgkin-Huxley	20, 21, 31–35, 38, 39, 52–54, 58–62, 66–68
HPDR	highest posterior density region	18, 19, 26, 27, 34, 66, 68
i.i.d.	independent and identically distributed	4, 5
KL	Kullback-Leibler	16, 18
LFI	likelihood-free inference	1–3, 15
LTE	likelihood-to-evidence	7, 8, 11–13, 15, 16, 39
LTR	likelihood-to-reference	7, 8
LVC	LIGO/Virgo collaboration	20, 35
MADE	masked autoencoder for distribution estimation	63
MAF	masked autoregressive flow	22, 25–27, 30, 35, 39, 63, 64
MCMC	Markov chain Monte Carlo	8, 9, 18, 20, 23
MLP	multi-layer perceptron	9, 12, 15, 22, 31, 38
MNPE	marginal neural posterior estimation	11, 12, 21, 23–30, 33–35, 38, 39, 49, 50, 52, 59

MNRE	marginal neural ratio estimation	11, 12, 16, 21, 23–35, 38, 50–52, 54–56, 58–60, 65, 66, 68, 69
MPE	marginal posterior estimation	2, 9, 11, 15, 27, 30, 35, 38
NC	neural conditioner	9–13
NDE	neural density estimation	4
NF	normalizing flow	4, 5, 9, 15, 22, 27, 30, 39, 63
NLL	negative log-likelihood	7, 23, 65–67
NN	neural network	4–6, 8, 22, 63, 64, 69
NPE	neural posterior estimation	5, 6, 8, 9, 11, 15, 16, 20–24, 26, 27, 30, 31, 33, 35, 39
NRE	neural ratio estimation	6, 8–13, 15, 16, 18, 21–24, 26, 27, 30, 31, 33, 38, 39, 52, 66
NSF	neural spline flow	39
OOD	out-of-distribution	33
OT	optimal transportation	17
PL	peripheral loss	65–68
ReLU	rectified linear unit	22
ROC	receiver operating characteristic	15, 16, 23, 25, 26, 32, 35, 38, 53
SBC	simulation-based calibration	19
SBI	simulation-based inference	1, 3
SGD	stochastic gradient descent	5
SLCP	simple likelihood and complex posterior	19, 21–23, 25–32, 35, 38, 49–51
SMC	sequential Monte Carlo	5
SNPE	sequential neural posterior estimation	5, 6, 20
SPSR	strictly proper scoring rule	7, 13, 23, 66
SVD	singular value decomposition	20, 21
TPR	true positive rate	15
UMNN	unconstrained monotonic neural network	39
VAE	variational auto-encoder	9
VAEAC	variational auto-encoder with arbitrary conditioning	9

Bibliography

- [1] M Clemencic et al. “The LHCb simulation application, Gauss: design, evolution and experience”. In: *Journal of Physics: Conference Series*. Vol. 331. 3. IOP Publishing. 2011, p. 032023 (page 1).
- [2] LIGO Scientific Collaboration et al. “LALSuite: LIGO Scientific Collaboration Algorithm Library Suite”. In: *Astrophysics Source Code Library* (2020), ascl–2012 (page 1).
- [3] Alan L Hodgkin and Andrew F Huxley. “A quantitative description of membrane current and its application to conduction and excitation in nerve”. In: *The Journal of physiology* 117.4 (1952), pp. 500–544 (pages 1, 20).
- [4] CE Dangerfield, David Kay, and Kevin Burrage. “Stochastic models and simulation of ion channel dynamics”. In: *Procedia Computer Science* 1.1 (2010), pp. 1587–1596 (page 1).
- [5] George Casella and Roger L Berger. “Statistical inference”. Cengage Learning, 2021 (page 1).
- [6] George EP Box and George C Tiao. “Bayesian inference in statistical analysis”. Vol. 40. John Wiley & Sons, 2011 (page 1).
- [7] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. “The frontier of simulation-based inference”. In: *Proceedings of the National Academy of Sciences* (2020) (pages 1, 3).
- [8] Sara Bolognesi et al. “Spin and parity of a single-produced resonance at the LHC”. In: *Physical Review D* 86.9 (2012), p. 095031 (page 2).
- [9] Benjamin P Abbott et al. “Improved analysis of GW150914 using a fully spin-precessing waveform model”. In: *Physical Review X* 6.4 (2016), p. 041014 (page 2).
- [10] Nabila Aghanim et al. “Planck 2018 results-VI. Cosmological parameters”. In: *Astronomy & Astrophysics* 641 (2020), A6 (page 2).
- [11] Antti Solonen et al. “Efficient MCMC for climate model parameter estimation: Parallel adaptive chains and early rejection”. In: *Bayesian Analysis* 7.3 (2012), pp. 715–736 (page 2).
- [12] Pedro J Gonçalves et al. “Training deep neural density estimators to identify mechanistic models of neural dynamics”. In: *Elife* 9 (2020), e56261 (pages 2, 4, 6, 20, 31).
- [13] Fabian Fröhlich et al. “Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model”. In: *Cell systems* 7.6 (2018), pp. 567–579 (page 2).
- [14] Markus Stoye et al. “Likelihood-free inference with an improved cross-entropy estimator”. In: *arXiv preprint arXiv:1808.00973* (2018) (page 2).

- [15] Johann Brehmer et al. “Mining gold from implicit models to improve likelihood-free inference”. In: *Proceedings of the National Academy of Sciences* 117.10 (2020), pp. 5242–5249 (page 2).
- [16] Donald B Rubin. “Bayesianly justifiable and relevant frequency calculations for the applied statistician”. In: *The Annals of Statistics* (1984), pp. 1151–1172 (page 3).
- [17] Mark A Beaumont, Wenyang Zhang, and David J Balding. “Approximate Bayesian computation in population genetics”. In: *Genetics* 162.4 (2002), pp. 2025–2035 (page 3).
- [18] Scott A Sisson, Yanan Fan, and Mark Beaumont. “Handbook of approximate Bayesian computation”. CRC Press, 2018 (page 3).
- [19] Peter Whittle. “On the smoothing of probability density functions”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), pp. 334–343 (page 3).
- [20] Emanuel Parzen. “On estimation of a probability density function and mode”. In: *The annals of mathematical statistics* 33.3 (1962), pp. 1065–1076 (page 3).
- [21] Evelyn Fix and Joseph Lawson Hodges. “Discriminatory analysis. Nonparametric discrimination: Consistency properties”. In: *International Statistical Review/Revue Internationale de Statistique* 57.3 (1989), pp. 238–247 (page 4).
- [22] Naomi S Altman. “An introduction to kernel and nearest-neighbor nonparametric regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185 (page 4).
- [23] Piotr Indyk and Rajeev Motwani. “Approximate nearest neighbors: towards removing the curse of dimensionality”. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998, pp. 604–613 (page 4).
- [24] Richard Bellman. “Dynamic programming”. In: *Science* 153.3731 (1966), pp. 34–37 (page 4).
- [25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444 (page 4).
- [26] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366 (pages 4, 9).
- [27] George Papamakarios and Iain Murray. “Fast ε -free inference of simulation models with bayesian conditional density estimation”. In: *Advances in neural information processing systems*. 2016, pp. 1028–1036 (pages 4–6).
- [28] Jan-Matthis Lueckmann et al. “Flexible statistical inference for mechanistic models of neural dynamics”. In: *arXiv preprint arXiv:1711.01861* (2017) (pages 4–6).
- [29] David Greenberg, Marcel Nonnenmacher, and Jakob Macke. “Automatic posterior transformation for likelihood-free inference”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2404–2414 (pages 4–6).
- [30] George Papamakarios, David Sterratt, and Iain Murray. “Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 837–848 (pages 4, 19).

- [31] George Papamakarios. “Neural density estimation and likelihood-free inference”. In: *arXiv preprint arXiv:1910.13233* (2019) (page 4).
- [32] Stephen R Green and Jonathan Gair. “Complete parameter inference for GW150914 using deep learning”. In: *Machine Learning: Science and Technology* 2.3 (2021), 03LT01 (pages 4, 20, 21, 35, 36).
- [33] Maximilian Dax et al. “Real-time gravitational-wave science with neural posterior estimation”. In: *arXiv preprint arXiv:2106.12594* (2021) (pages 4, 12).
- [34] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. “Normalizing flows: An introduction and review of current methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020) (pages 4, 9).
- [35] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel recurrent neural networks”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1747–1756 (page 4).
- [36] Aaron van den Oord et al. “Conditional image generation with pixelfcn decoders”. In: *arXiv preprint arXiv:1606.05328* (2016) (page 4).
- [37] Diederik P Kingma and Prafulla Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: *arXiv preprint arXiv:1807.03039* (2018) (page 4).
- [38] George Papamakarios et al. “Normalizing flows for probabilistic modeling and inference”. In: *arXiv preprint arXiv:1912.02762* (2019) (page 4).
- [39] George Papamakarios, Theo Pavlakou, and Iain Murray. “Masked autoregressive flow for density estimation”. In: *arXiv preprint arXiv:1705.07057* (2017) (pages 4, 9, 22, 63).
- [40] Conor Durkan et al. “Neural spline flows”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 7511–7522 (pages 4, 39).
- [41] Antoine Wehenkel and Gilles Louppe. “Unconstrained monotonic neural networks”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 1545–1555 (pages 4, 39).
- [42] Cédric Villani. “Topics in optimal transportation”. 58. American Mathematical Soc., 2003 (pages 4, 17).
- [43] Vladimir Igorevich Bogachev, Aleksandr Viktorovich Kolesnikov, and Kirill Vladimirovich Medvedev. “Triangular transformations of measures”. In: *Sbornik: Mathematics* 196.3 (2005), p. 309 (page 4).
- [44] Léon Bottou. “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of COMPSTAT’2010*. Springer, 2010, pp. 177–186 (page 5).
- [45] Ilya Sutskever et al. “On the importance of initialization and momentum in deep learning”. In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147 (page 5).
- [46] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014) (pages 5, 22).
- [47] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017) (pages 5, 22).

- [48] Atilim Gunes Baydin et al. “Automatic differentiation in machine learning: a survey”. In: *Journal of machine learning research* 18 (2018) (page 5).
- [49] Adam Paszke et al. “Automatic differentiation in pytorch”. In: (2017) (pages 5, 15).
- [50] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536 (page 5).
- [51] Robert Hecht-Nielsen. “Theory of the backpropagation neural network”. In: *Neural networks for perception*. Elsevier, 1992, pp. 65–93 (page 5).
- [52] Scott A Sisson, Yanan Fan, and Mark M Tanaka. “Sequential monte carlo without likelihoods”. In: *Proceedings of the National Academy of Sciences* 104.6 (2007), pp. 1760–1765 (page 5).
- [53] Gareth W Peters, Yanan Fan, and Scott A Sisson. “On sequential Monte Carlo, partial rejection control and approximate Bayesian computation”. In: *Statistics and Computing* 22.6 (2012), pp. 1209–1222 (page 5).
- [54] Fernando V Bonassi and Mike West. “Sequential Monte Carlo with adaptive weights for approximate Bayesian computation”. In: *Bayesian Analysis* 10.1 (2015), pp. 171–187 (page 5).
- [55] Kyle Cranmer, Juan Pavez, and Gilles Louppe. “Approximating likelihood ratios with calibrated discriminative classifiers”. In: *arXiv preprint arXiv:1506.02169* (2015) (pages 6, 7).
- [56] Jerzy Neyman and Egon Sharpe Pearson. “IX. On the problem of the most efficient tests of statistical hypotheses”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231.694-706 (1933), pp. 289–337 (page 6).
- [57] Quang H Vuong. “Likelihood ratio tests for model selection and non-nested hypotheses”. In: *Econometrica: Journal of the Econometric Society* (1989), pp. 307–333 (page 6).
- [58] Joeri Hermans, Volodimir Begy, and Gilles Louppe. “Likelihood-free MCMC with Amortized Approximate Ratio Estimators”. In: *arXiv preprint arXiv:1903.04057* (2019) (pages 7, 8, 11, 12, 16).
- [59] Owen Thomas et al. “Likelihood-free inference by ratio estimation”. In: *arXiv preprint arXiv:1611.10242* (2016) (page 7).
- [60] Tilmann Gneiting and Adrian E Raftery. “Strictly proper scoring rules, prediction, and estimation”. In: *Journal of the American statistical Association* 102.477 (2007), pp. 359–378 (page 7).
- [61] Edgar C Merkle and Mark Steyvers. “Choosing a strictly proper scoring rule”. In: *Decision Analysis* 10.4 (2013), pp. 292–304 (page 7).
- [62] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: (1970) (pages 8, 19).
- [63] Ming-Hui Chen, Qi-Man Shao, and Joseph G Ibrahim. “Monte Carlo methods in Bayesian computation”. Springer Science & Business Media, 2012 (pages 8, 19).

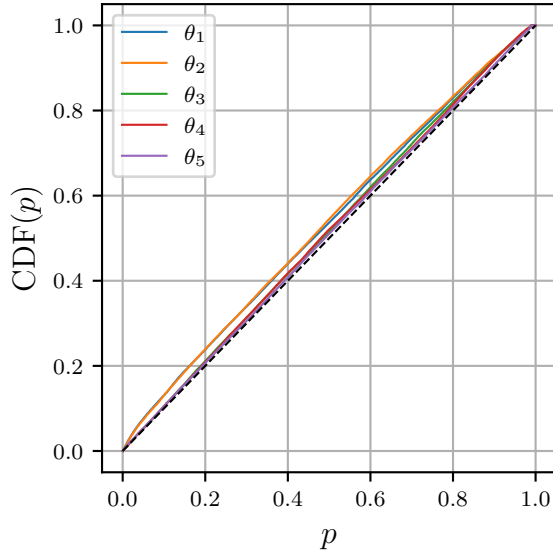
- [64] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013) (page 9).
- [65] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014) (page 9).
- [66] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. “Learning structured output representation using deep conditional generative models”. In: *Advances in neural information processing systems* 28 (2015), pp. 3483–3491 (page 9).
- [67] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014) (page 9).
- [68] Jinsung Yoon, James Jordon, and Mihaela Schaar. “Gain: Missing data imputation using generative adversarial nets”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5689–5698 (page 9).
- [69] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. “Variational autoencoder with arbitrary conditioning”. In: *arXiv preprint arXiv:1806.02382* (2018) (page 9).
- [70] Mohamed Ishmael Belghazi et al. “Learning about an exponential amount of conditional distributions”. In: *arXiv preprint arXiv:1902.08401* (2019) (pages 9–13, 30).
- [71] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386 (pages 9, 12).
- [72] Laurent Dinh, David Krueger, and Yoshua Bengio. “Nice: Non-linear independent components estimation”. In: *arXiv preprint arXiv:1410.8516* (2014) (page 9).
- [73] Durk P Kingma et al. “Improved variational inference with inverse autoregressive flow”. In: *Advances in neural information processing systems* 29 (2016), pp. 4743–4751 (page 9).
- [74] Alessio Spantini, Daniele Bigoni, and Youssef Marzouk. “Inference via low-dimensional couplings”. In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2639–2709 (page 9).
- [75] Yang Li, Shoaib Akbar, and Junier B Oliva. “ACFlow: Flow Models for Arbitrary Conditional Likelihoods”. In: *Proceedings of Machine Learning Research* 119 (2020) (pages 9, 38).
- [76] Arnaud Delaunoy et al. “Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization”. In: *arXiv preprint arXiv:2010.12931* (2020) (pages 11, 12, 18).
- [77] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (pages 12, 35).
- [78] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037 (page 15).
- [79] Conor Durkan et al. “nflows: normalizing flows in PyTorch”. 2020. URL: <https://github.com/bayesiains/nflows> (page 15).

- [80] John D Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in science & engineering* 9.03 (2007), pp. 90–95 (page 15).
- [81] Solomon Kullback and Richard A Leibler. “On information and sufficiency”. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86 (page 16).
- [82] Yossi Rubner, Leonidas J Guibas, and Carlo Tomasi. “The earth mover’s distance, multi-dimensional scaling, and color-based image retrieval”. In: *Proceedings of the ARPA image understanding workshop*. Vol. 661. 1997, p. 668 (page 16).
- [83] Marco Cuturi. “Sinkhorn distances: Lightspeed computation of optimal transport”. In: *Advances in neural information processing systems* 26 (2013), pp. 2292–2300 (page 17).
- [84] Richard Sinkhorn. “Diagonal equivalence to matrices with prescribed row and column sums”. In: *The American Mathematical Monthly* 74.4 (1967), pp. 402–405 (page 17).
- [85] Philip A Knight. “The Sinkhorn–Knopp algorithm: convergence and applications”. In: *SIAM Journal on Matrix Analysis and Applications* 30.1 (2008), pp. 261–275 (page 17).
- [86] Charles R Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362 (page 17).
- [87] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature methods* 17.3 (2020), pp. 261–272 (page 17).
- [88] Sean Talts et al. “Validating Bayesian inference algorithms with simulation-based calibration”. In: *arXiv preprint arXiv:1804.06788* (2018) (page 19).
- [89] John Skilling. “Nested sampling for general Bayesian computation”. In: *Bayesian analysis* 1.4 (2006), pp. 833–859 (page 20).
- [90] Edward Higson et al. “Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation”. In: *Statistics and Computing* 29.5 (2019), pp. 891–913 (pages 20, 36).
- [91] John Veitch et al. “Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library”. In: *Physical Review D* 91.4 (2015), p. 042003 (page 20).
- [92] Gregory Ashton et al. “BILBY: a user-friendly Bayesian inference library for gravitational-wave astronomy”. In: *The Astrophysical Journal Supplement Series* 241.2 (2019), p. 27 (pages 20, 36).
- [93] IM Romero-Shaw et al. “Bayesian inference for compact binary coalescences with BILBY: Validation and application to the first LIGO–Virgo gravitational-wave transient catalogue”. In: *Monthly Notices of the Royal Astronomical Society* 499.3 (2020), pp. 3295–3319 (pages 20, 36).
- [94] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015) (pages 22, 31, 35).
- [95] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Icml*. 2010 (page 22).

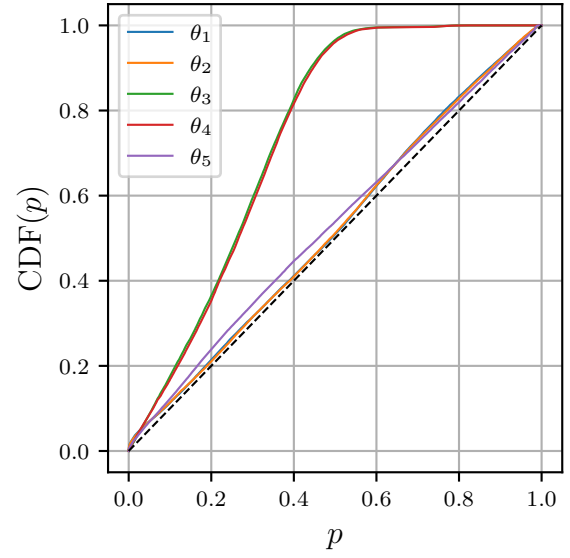
- [96] Joshua S Speagle. “dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences”. In: *Monthly Notices of the Royal Astronomical Society* 493.3 (2020), pp. 3132–3158 (page 36).
- [97] Mathieu Germain et al. “Made: Masked autoencoder for distribution estimation”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 881–889 (page 63).
- [98] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *arXiv preprint arXiv:1612.01474* (2016) (page 65).
- [99] Dan Hendrycks and Kevin Gimpel. “A baseline for detecting misclassified and out-of-distribution examples in neural networks”. In: *arXiv preprint arXiv:1610.02136* (2016) (page 65).
- [100] Gabriel Pereyra et al. “Regularizing neural networks by penalizing confident output distributions”. In: *arXiv preprint arXiv:1701.06548* (2017) (page 65).
- [101] Chuan Guo et al. “On calibration of modern neural networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1321–1330 (page 65).
- [102] Jishnu Mukhoti et al. “Calibrating deep neural networks using focal loss”. In: *arXiv preprint arXiv:2002.09437* (2020) (page 65).
- [103] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988 (page 65).

Appendix A

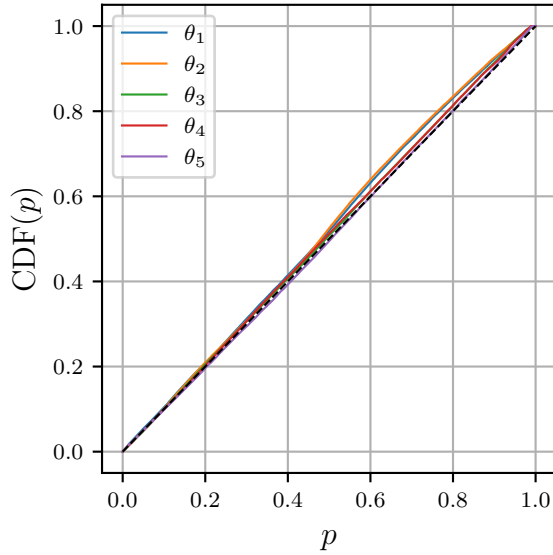
Additional figures



(a) MNRE



(b) MNPE



(c) AMNRE

Figure A.1. Calibration tests of 1-d surrogate marginal SLCP posteriors. CDFs are averaged over the model instances. MNPE presents strong miscalibration of θ_3 and θ_4 surrogates.

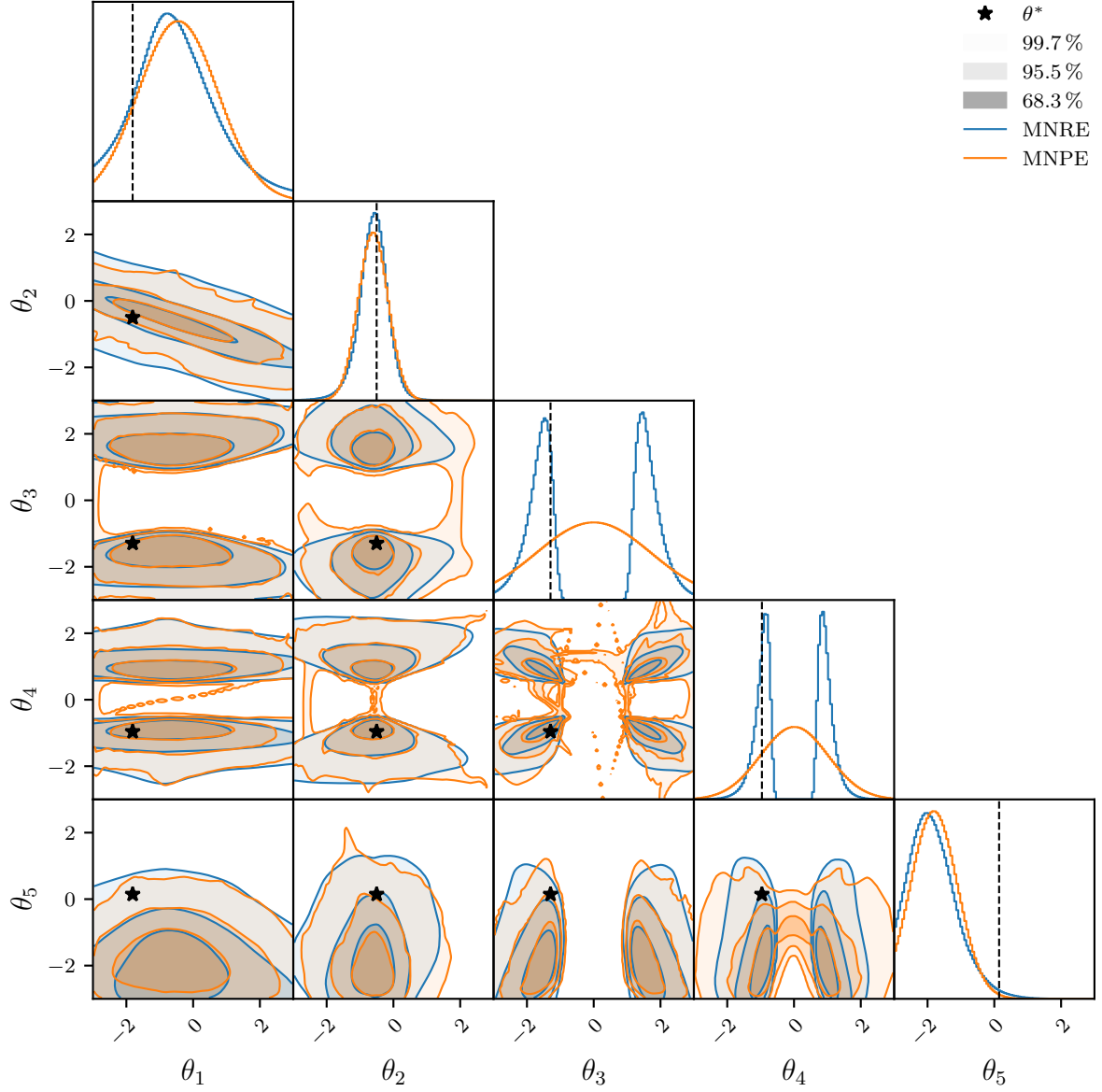


Figure A.2. MNRE against MNPE 1-d and 2-d surrogate marginal SLCP posteriors, for a realization of the testing set. Density is averaged over the model instances. MNPE surrogates present artifacts in low density regions.

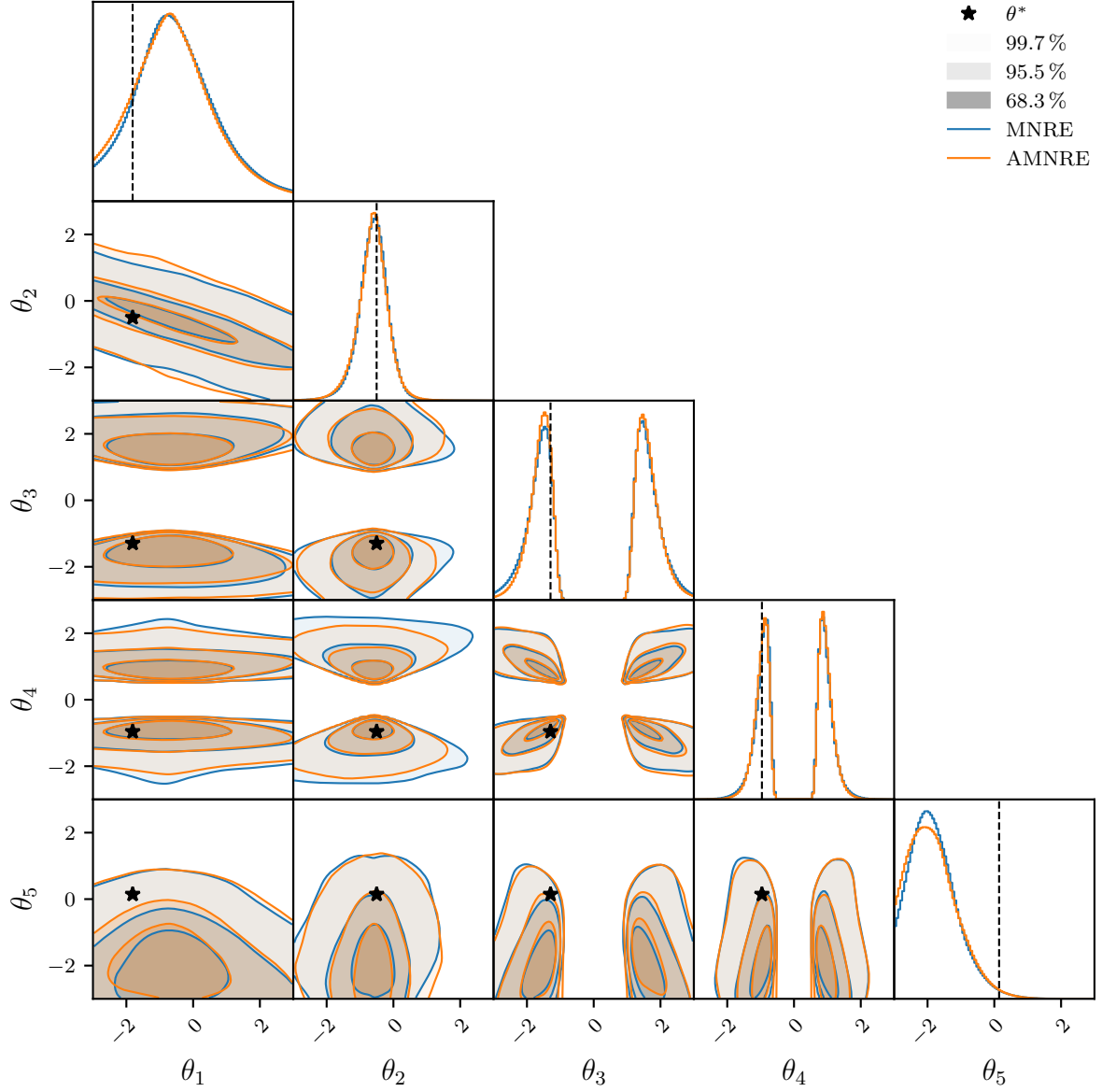
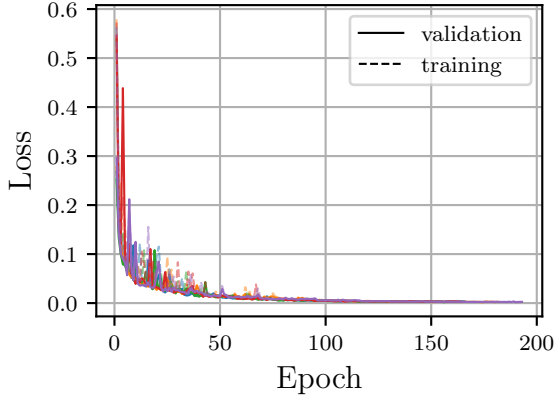
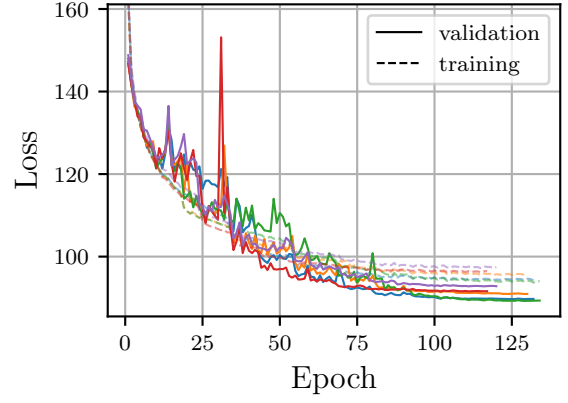


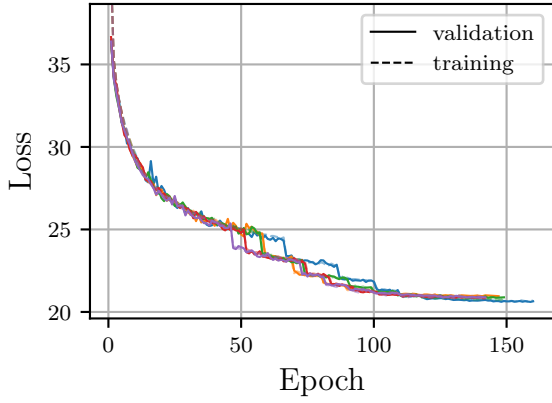
Figure A.3. MNRE against AMNRE 1-d and 2-d surrogate marginal SLCP posteriors, for a realization of the testing set. Density is averaged over the model instances. MNRE and AMNRE surrogates share sensibly the same structure.



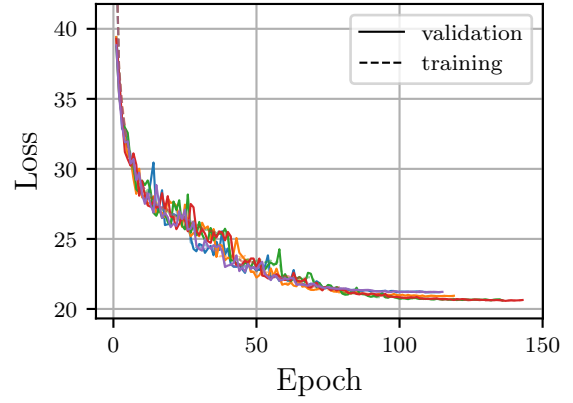
(a) NRE



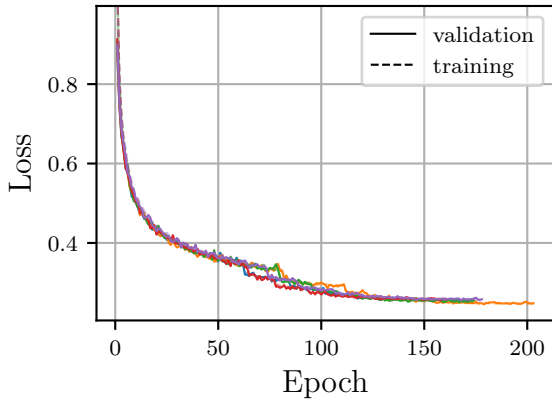
(b) MNPE (38 subspaces)



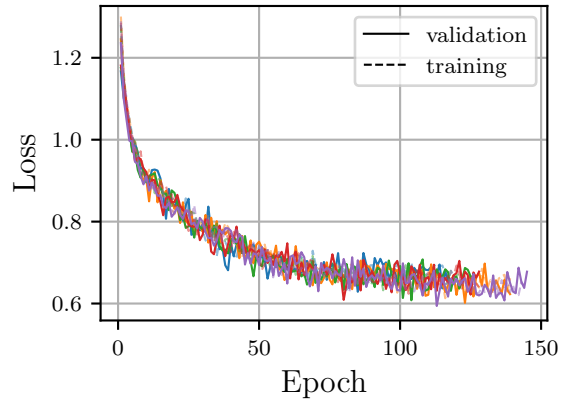
(c) MNRE (38 subspaces)



(d) MNRE with shared embedding

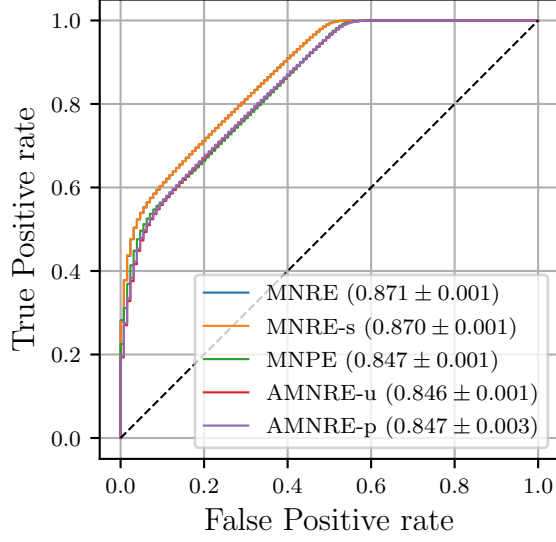


(e) AMNRE with uniform masking

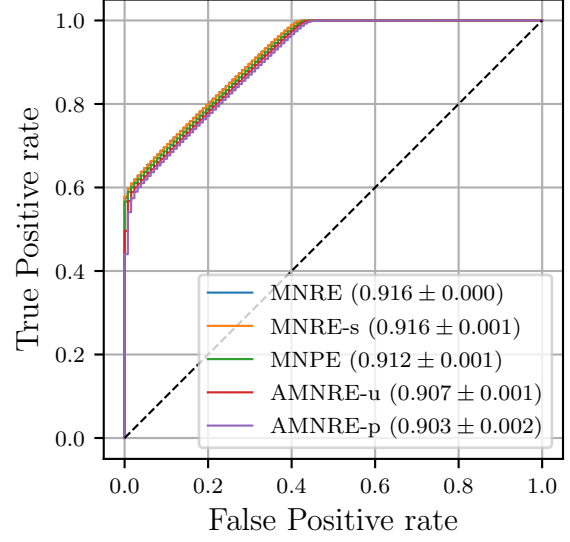


(f) AMNRE with Poisson masking

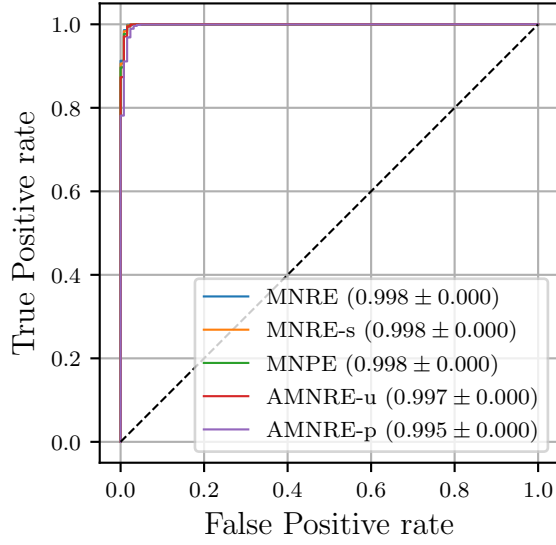
Figure A.4. Mean training and validation losses of HH surrogate models. All methods converge without signs of overfitting. NRE almost reaches a null loss. Sharing an embedding among smaller classifiers does not hinder MNRE's convergence. AMNRE with Poisson masking is significantly less stable.



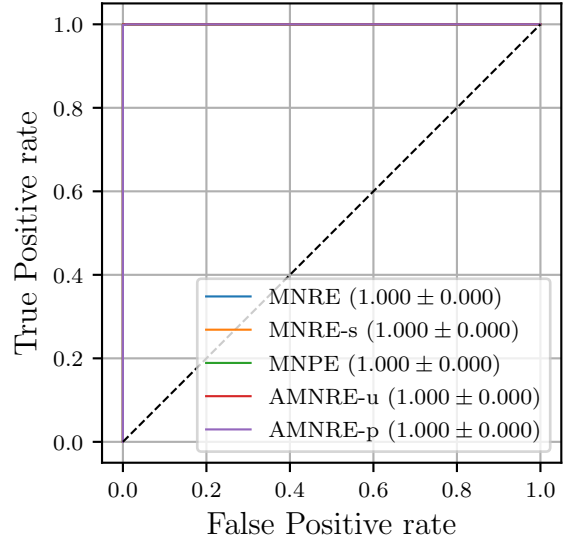
(a) Subset g_{Na}



(b) Subset (g_{Na}, g_K)



(c) Subset $(g_{Na}, g_K, g_l, -V_T)$



(d) Full set θ

Figure A.5. ROC curves of HH surrogate models. Curves are averaged over the model instances. The performance of classifiers increases with the number of provided parameters.

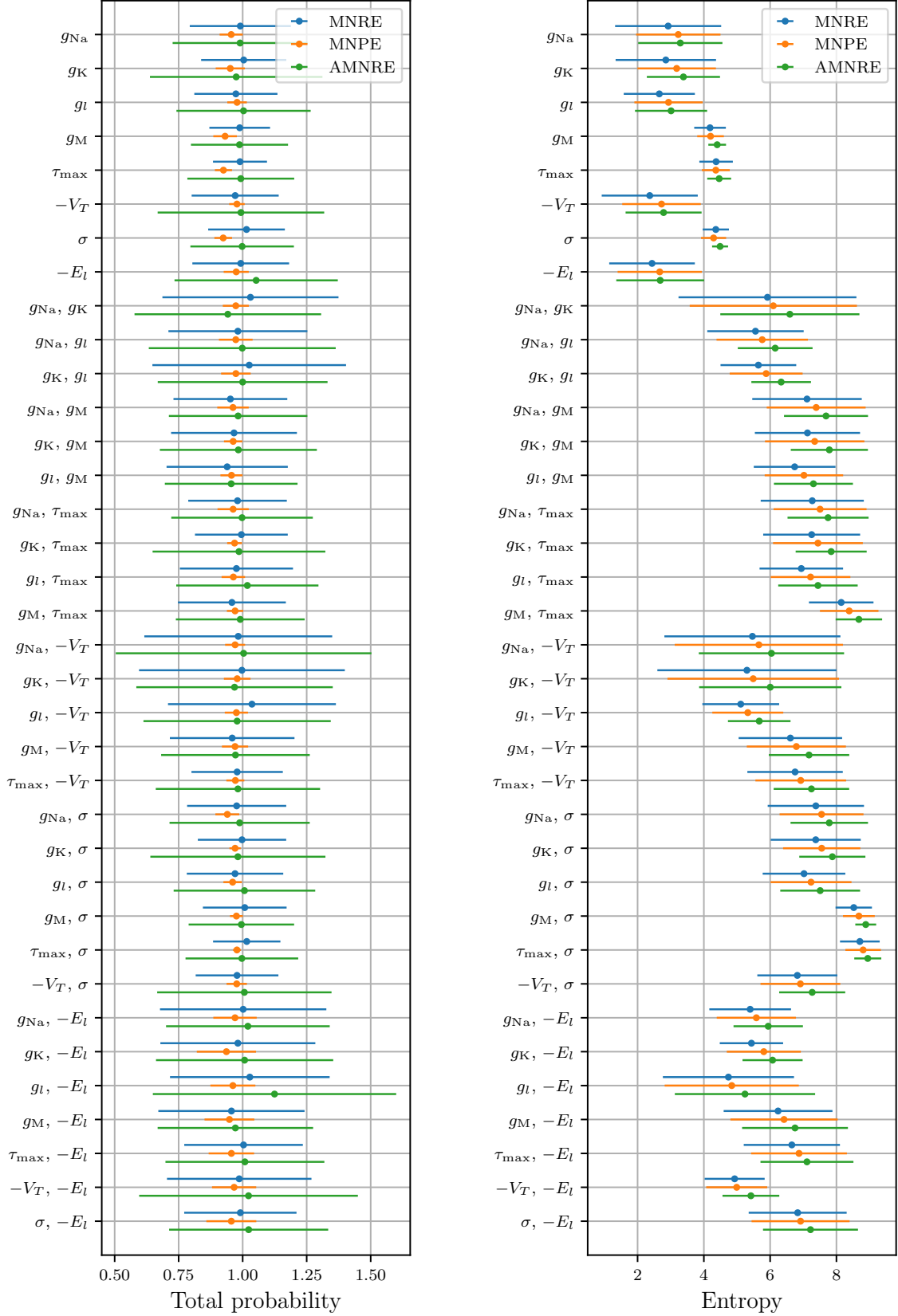
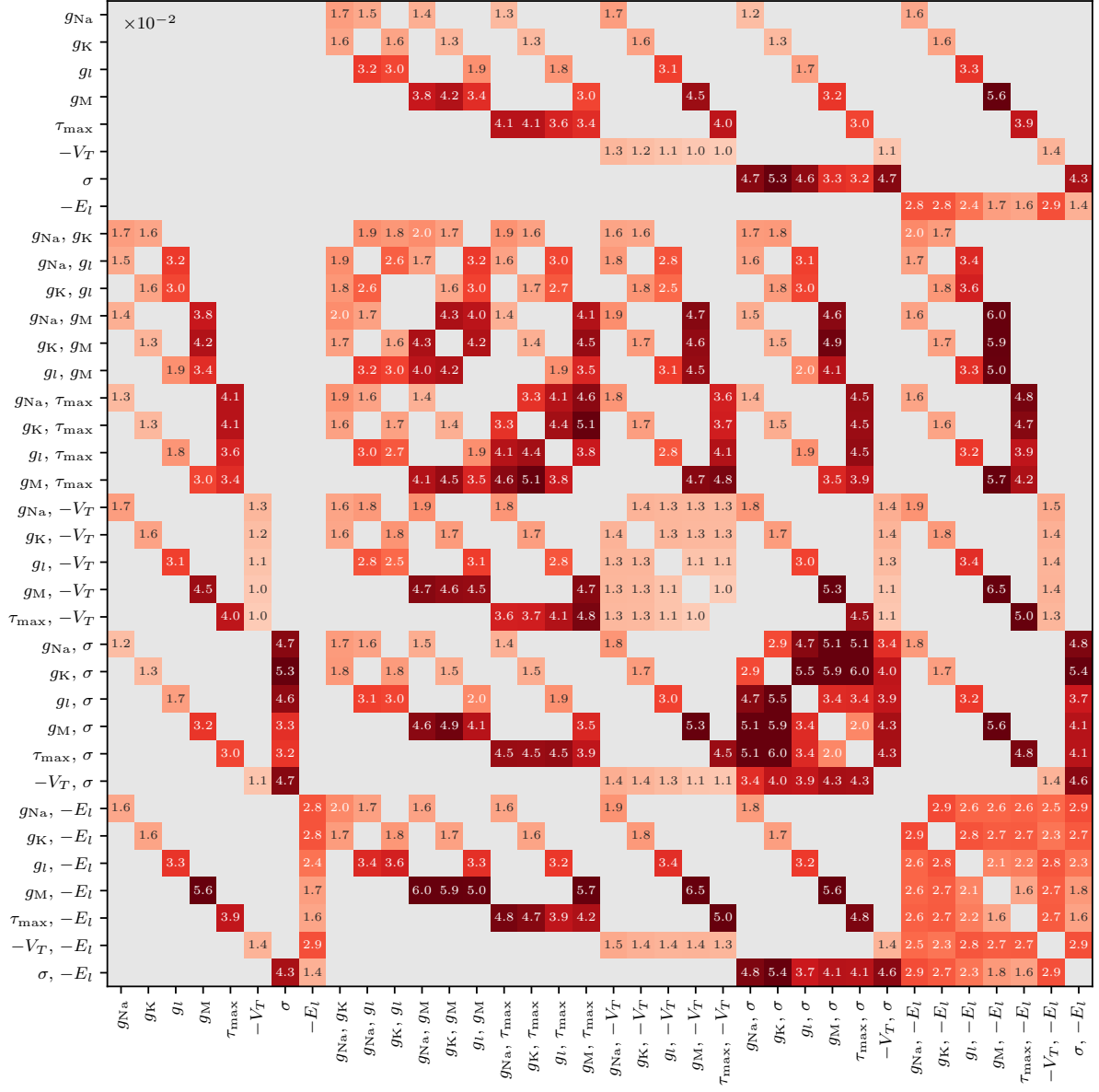
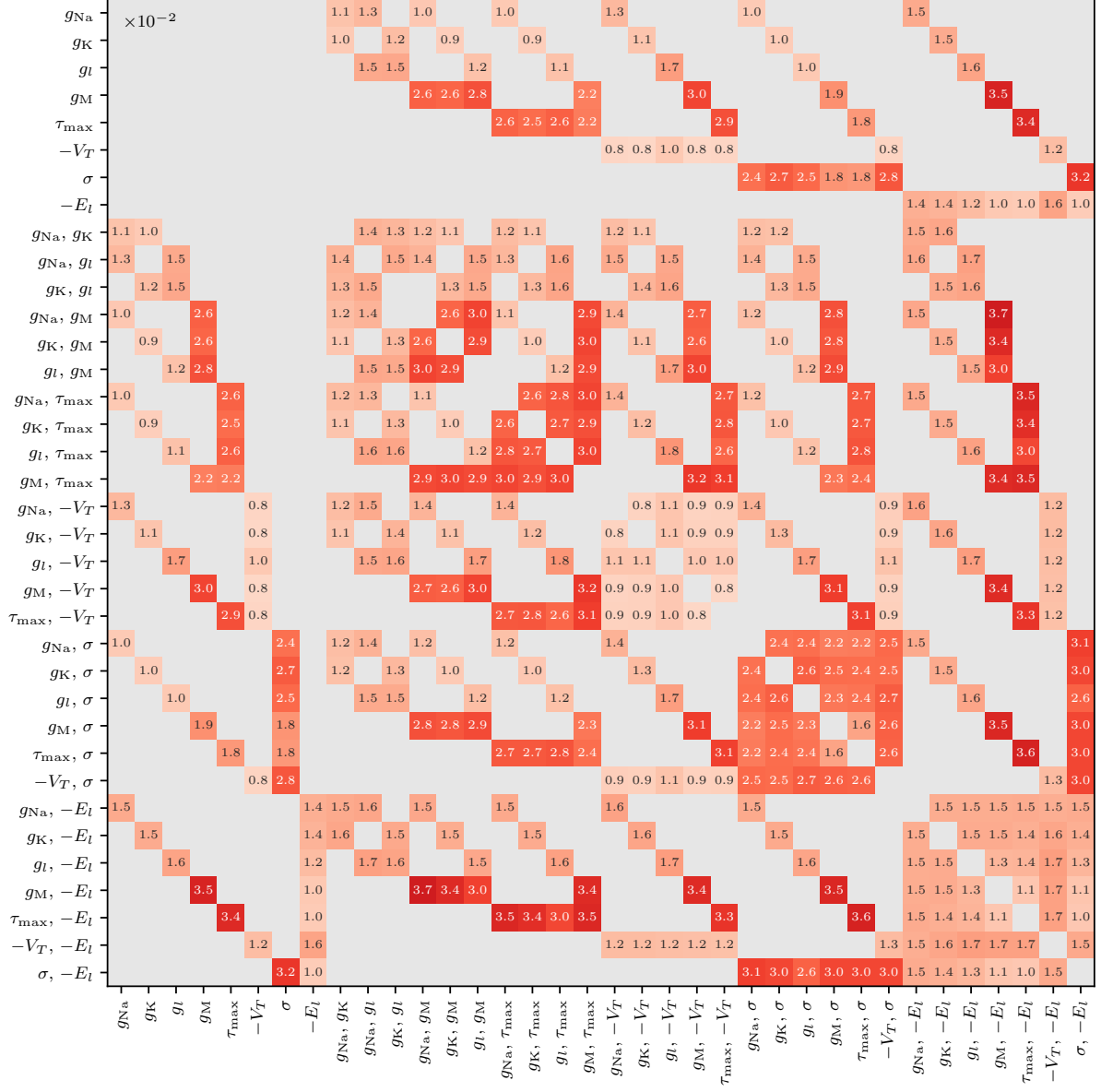


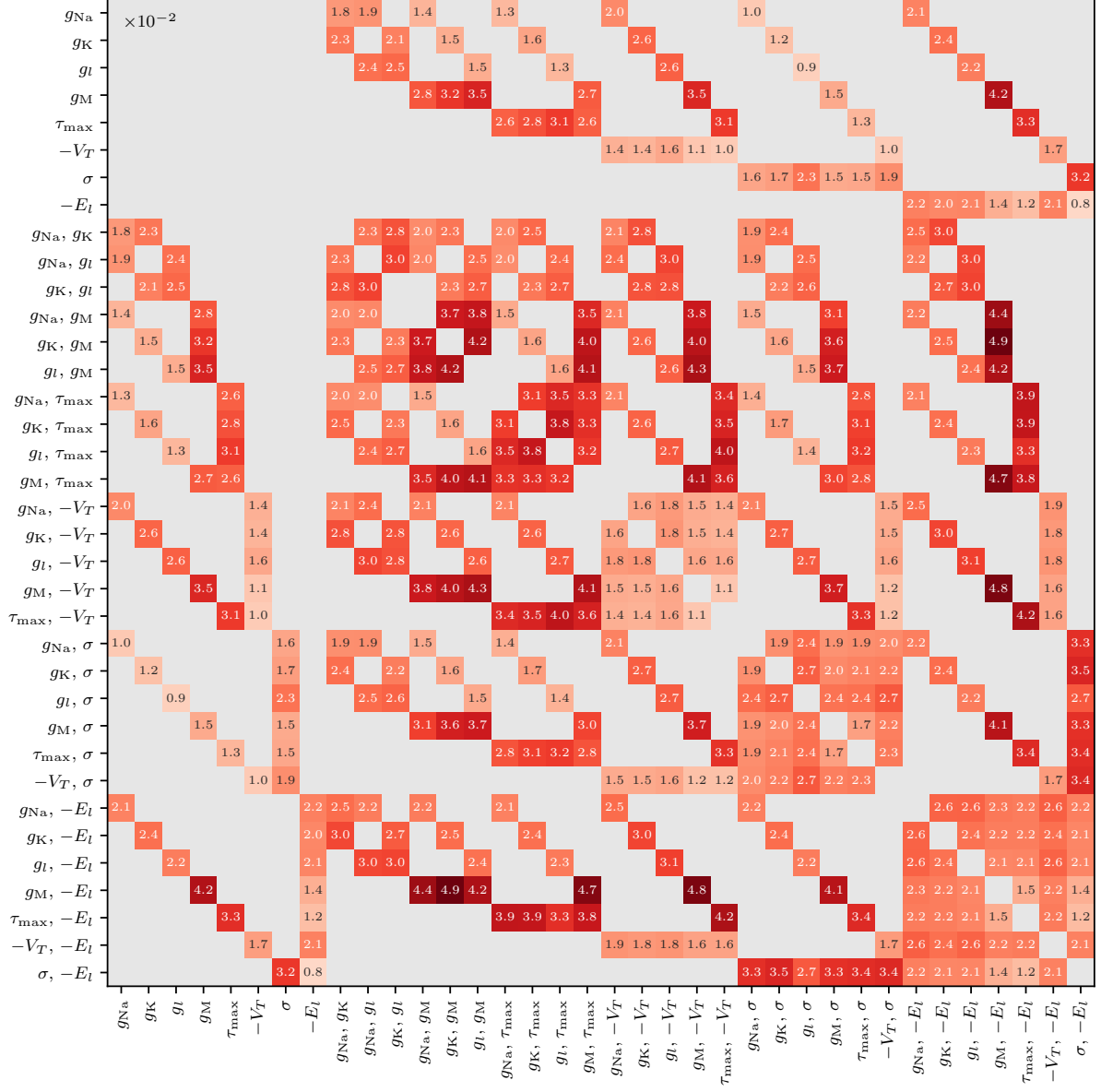
Figure A.6. Total probability (left) and entropy (right) of 1-d and 2-d surrogate marginal HH posterior histograms. The bars represent the quantity mean and standard deviation over 64 realizations from the testing set and the model instances. The total probability of MNRE and AMNRE surrogates has a large variance. The parameters g_M , τ_{max} and σ have posterior histograms of almost the maximum entropy, *i.e.* $\log 100 \approx 4.605$.



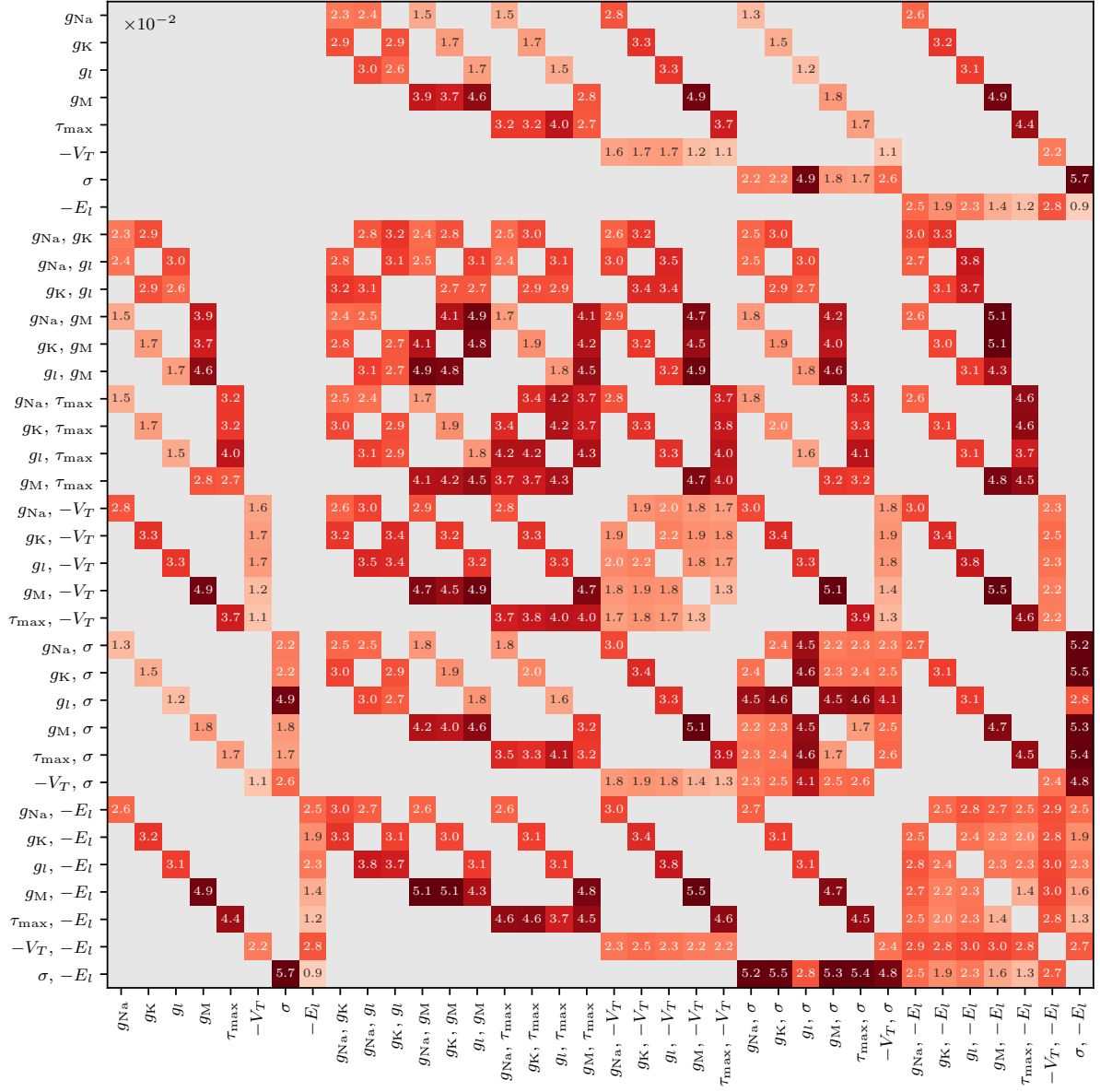
(a) MNRE



(b) MNRE with shared embedding



(c) AMNRE with uniform masking



(d) AMNRE with Poisson masking

Figure A.7. EMD between 1-d and 2-d surrogate marginal HH posterior histograms. Values are averaged over 64 realizations from the testing set and the model instances. MNRE with shared embedding is more consistent than without. AMNRE with Poisson masking is less consistent than with uniform masking.

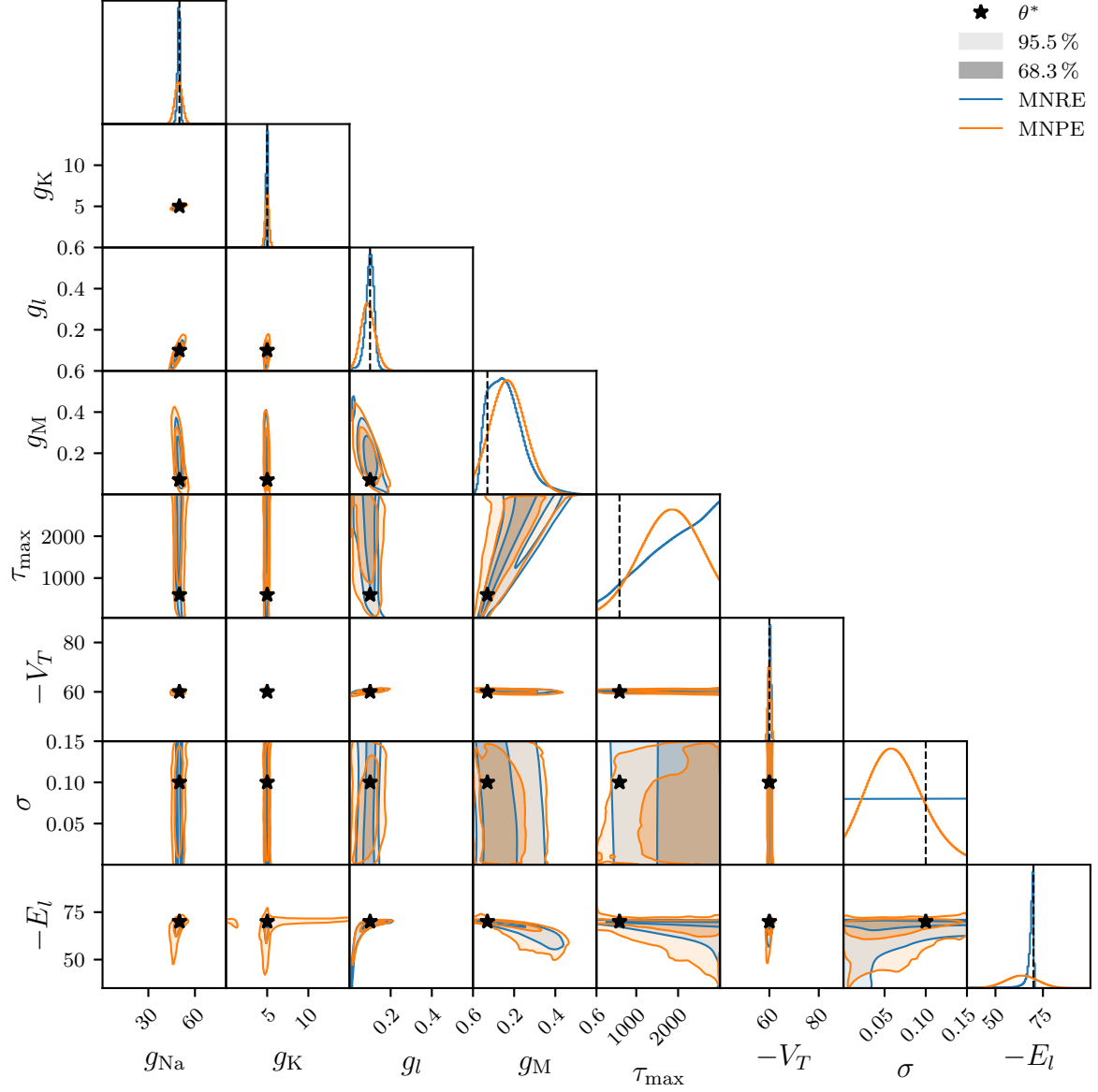


Figure A.8. MNRE against MNPE 1-d and 2-d surrogate marginal HH posteriors, for a realization of reference. Density is averaged over the model instances. The two methods agree on the structure of the marginal posteriors, with the exceptions of the τ_{max} and σ parameters, which are poorly modeled by MNPE.

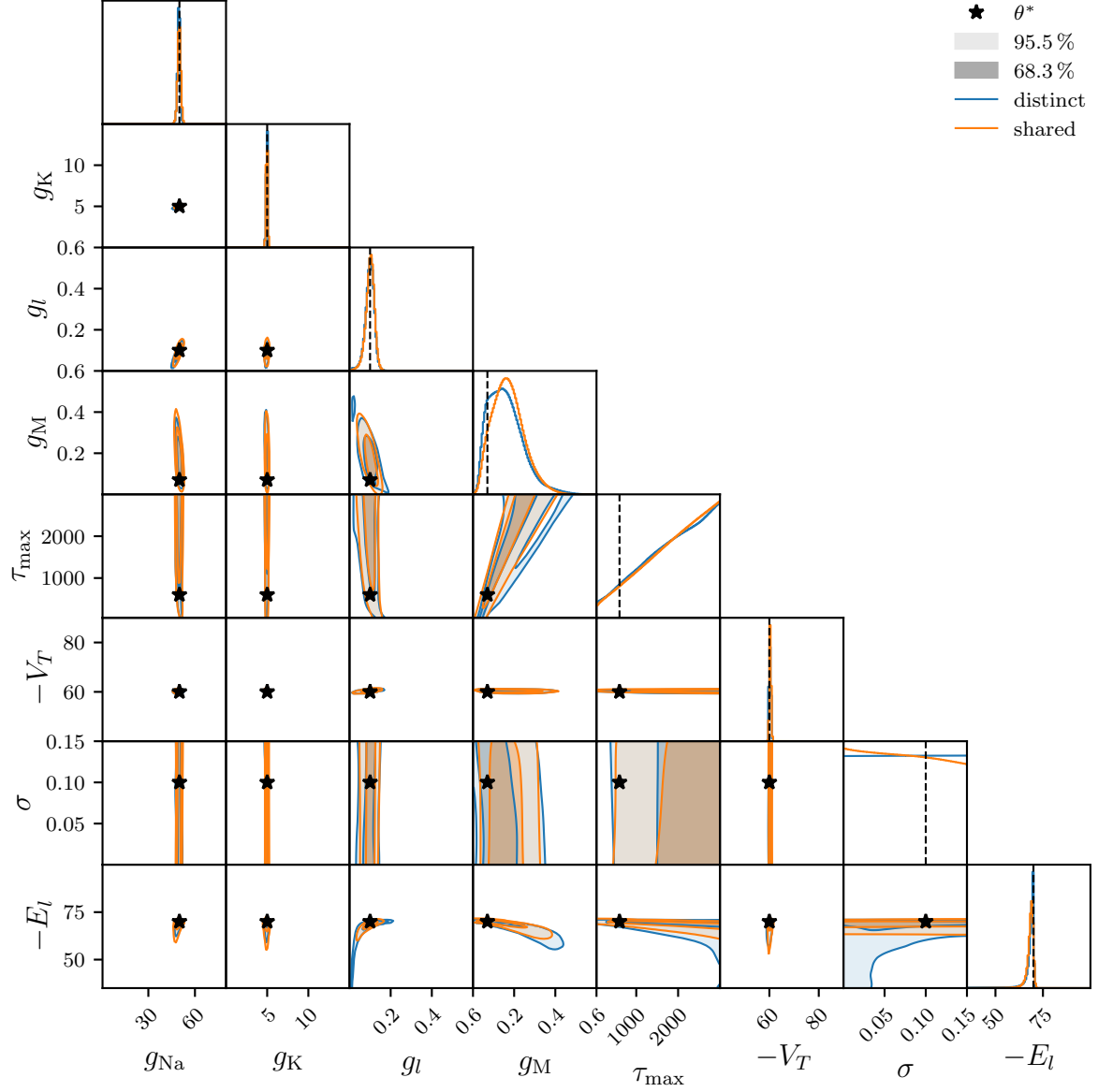


Figure A.9. MNRE 1-d and 2-d surrogate marginal HH posteriors, with (orange) and without (blue) shared embedding, for a realization of reference. Density is averaged over the model instances. Sharing an embedding among smaller estimators does not seem to deteriorate MNRE's approximations in low-dimension.

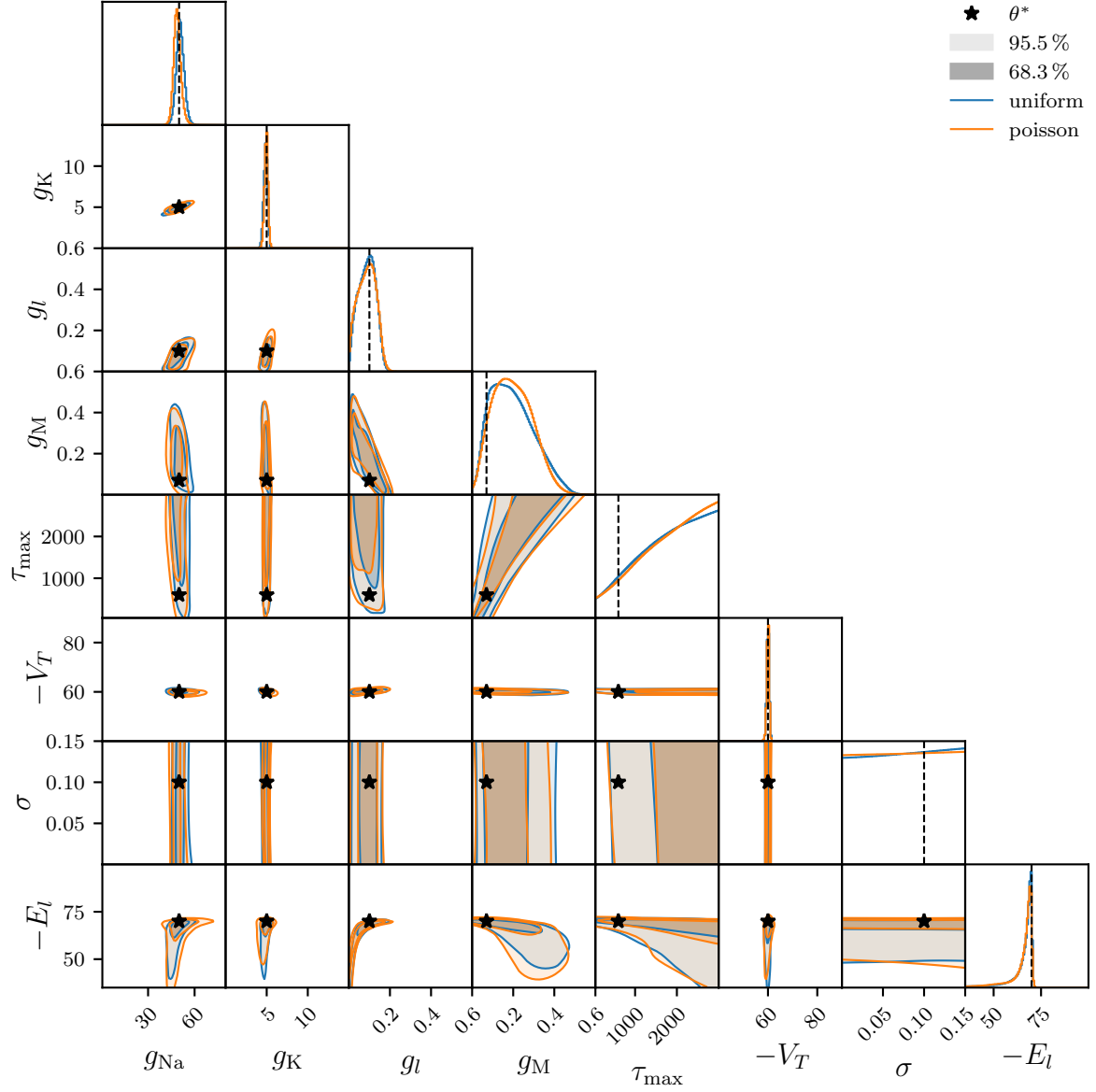


Figure A.10. AMNRE 1-d and 2-d surrogate marginal HH posteriors, with uniform (blue) and Poisson (orange) masking, for a realization of reference. Density is averaged over the model instances. Poisson masking does not seem to affect 1-d and 2-d AMNRE surrogates.

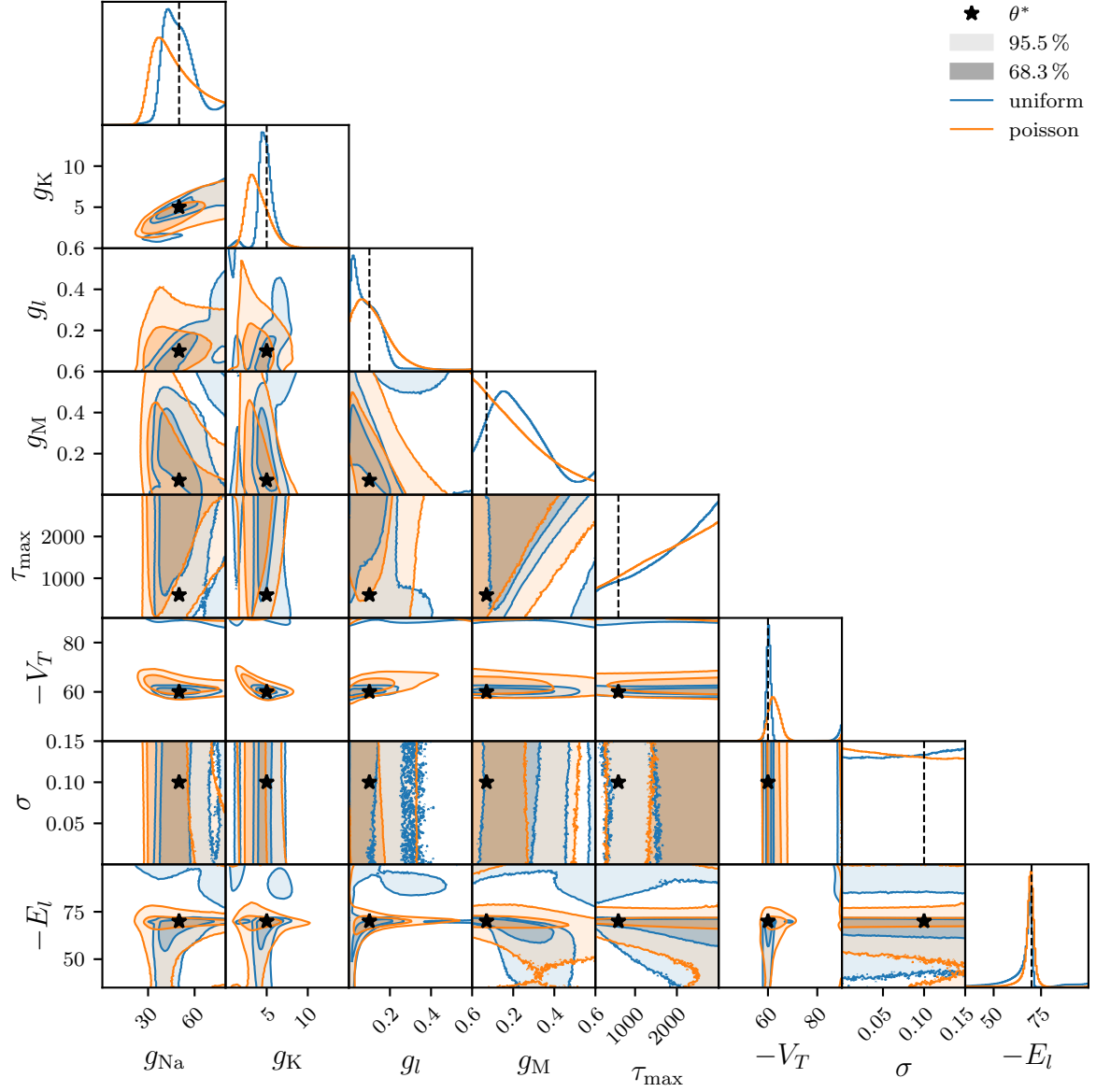


Figure A.11. AMNRE full surrogate marginal HH posteriors, with uniform (blue) and Poisson (orange) masking, for a realization of the testing set. Density is averaged over the model instances. Poisson masking affects high-dimensional AMNRE surrogates.

Appendix B

Autoregressive flows

In NFs (see Section 2.2.1), autoregressive transformations impose the tractability of their Jacobian by decomposing $u = f(x)$ into a sequence of univariate transformations

$$u_i = f_i(x_{\leq i}), \quad (\text{B.1})$$

where x_i is the i -th element of x . This constraint leads to a *triangular* Jacobian

$$J_f = \frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & 0 & 0 & 0 \\ \vdots & \ddots & 0 & 0 \\ \frac{\partial f_i}{\partial x_1} & \dots & \frac{\partial f_i}{\partial x_i} & 0 \\ \vdots & & & \ddots \end{pmatrix}, \quad (\text{B.2})$$

allowing to calculate the determinant as the product of the diagonal elements,

$$|\det J_f(x)| = \left| \prod_i \frac{\partial f_i}{\partial x_i}(x_{\leq i}) \right|. \quad (\text{B.3})$$

Masked autoregressive flow

The *masked autoregressive flow* (MAF) [39] is a widespread example of NF implementing autoregressive transformations. MAF uses affine univariate transformations

$$u_i = \frac{x_i - \mu_i}{\exp(\sigma_i)}, \quad (\text{B.4})$$

where the terms μ_i and σ_i are unconstrained parametric functions of $x_{<i}$, leading to the simple Jacobian determinant

$$|\det J_f(x)| = \exp\left(-\sum_i \sigma_i\right). \quad (\text{B.5})$$

Taking inspiration from MADE [97], Papamakarios et al. [39] propose to compute (μ_i, σ_i) for all i in a single forward pass of a NN by dropping connections to ensure that output (μ_i, σ_i) is only connected to inputs $x_{<i}$ (see Figure B.1), thereby improving efficiency.

As a NF, a MAF is composed of several autoregressive transformations. However, by design, autoregressive transformations are limited in what they can model by the order of components in x . Especially, with (B.4), there is an affine bijection between u_1 and x_1 , regardless of the number of transformations. To overcome this problem, Papamakarios et al. [39] propose to use a different component order in each autoregressive transformation.

Unfortunately, this is not applicable to one-dimensional distributions, *i.e.* scalar x . In such cases, MAF is limited to shift and scale the base distribution.

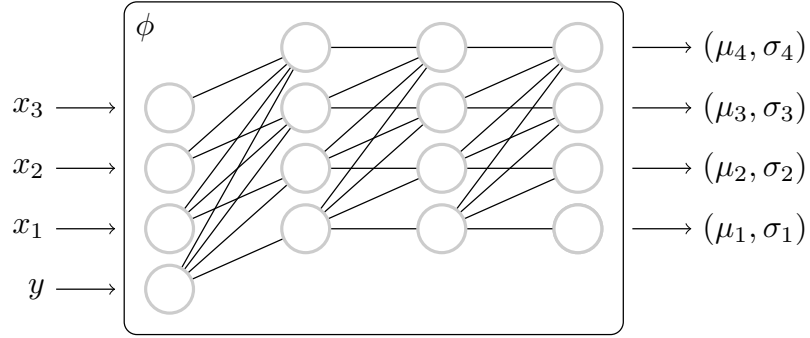


Figure B.1. Illustration of the dropped connections of the NN computing the pairs (μ_i, σ_i) in MAF. For the sake of readability, we represent only three neurons per hidden layers. In practice, this number can be arbitrary large, as long as (μ_i, σ_i) is only connected to $x_{<i}$ and potential conditioning data y .

Appendix C

Overconfidence

As mentioned in Section 4.4, (A)MNRE surrogates are sometimes too confident, *i.e.* underdispersed with respect to the true marginal posteriors. In real-world situations, like health care or self-driving cars, this overconfidence could become a serious threat if the model predictions are used to take decisions. Ideally, predictive networks should not only be accurate, but also know when they are likely to be incorrect. In practice, it is preferable to have a hesitant network and fall back to more robust solutions when it is not confident in its predictions than an overconfident network.

Overconfidence is actually a common issue of modern neural networks and many methods have been proposed to alleviate this problem [98–101]. For classification, Mukhoti et al. [102] propose to replace the widespread NLL $\mathcal{L}(p) = -\log p$ by the *focal loss* (FL) [103]

$$\mathcal{L}(p) = -(1 - p)^\gamma \log p, \quad (\text{C.1})$$

where $\gamma \geq 1$ is an adjustable parameter. The rationale is that, with the NLL, even if an item is perfectly classified, the gradient still pushes the network to increase its confidence, *i.e.* the derivative of $-\log p$ is not null when $p = 1$. Conversely, the FL has a null derivative when $p = 1$, which reduces the importance of already well classified samples. Therefore, the FL prevents the predicted distributions from becoming too “peaky”, *i.e.* it increases the entropy of the predictions [102]. The FL can be seen as a *regularization* of the NLL and the larger γ , the stronger the regularization.

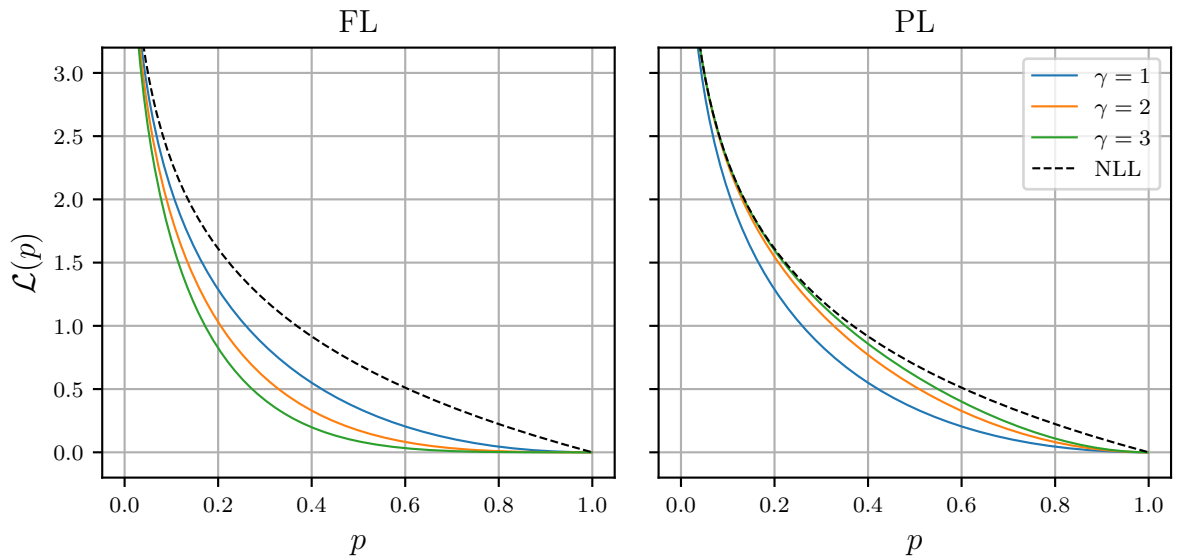


Figure C.1. Comparison between the FL (left) and PL (right). As γ increases, the FL deviates more from the NLL while the PL gets closer. Both have null derivatives at $p = 1$.

A desirable property for regularization methods is the ability to reduce their strength arbitrarily such that, at the limit, the method is equivalent to not applying regularization. The FL does not possess that property as it is not possible to choose γ under 1 for numerical stability reasons. To alleviate this problem, we propose the *peripheral loss* (PL)

$$\mathcal{L}(p) = -(1 - p^\gamma) \log p, \quad (\text{C.2})$$

where $\gamma \geq 1$ is an adjustable parameter. The FL and PL are complementary: the former enables arbitrarily large regularization while the latter enables arbitrarily small regularization. When $\gamma = 1$, the FL and PL are equivalent.

Demonstration We apply the FL and PL, with $\gamma = 2$, to MNRE of the 1-d and 2-d marginal posteriors of the HH simulator. The experimental settings are not modified otherwise (see Section 4.3).

As expected, the confidence of the surrogates is reduced with respect to NLL (see Figure C.2). As a consequence, the entropy of predictions is increased (see Figure C.3) and the HPDRs are more dispersed (see Figure C.4). Unlike the NLL, FL and PL are *not* SPSRs, meaning that they do not lead to surrogates that are probability measures. In particular, this implies that the surrogates will not integrate to a total probability of 1, which is indeed observed in Figure C.3.

Further work is needed to analyze formally the implications of the FL and PL on NRE approximations. However, these empirical results are promising.

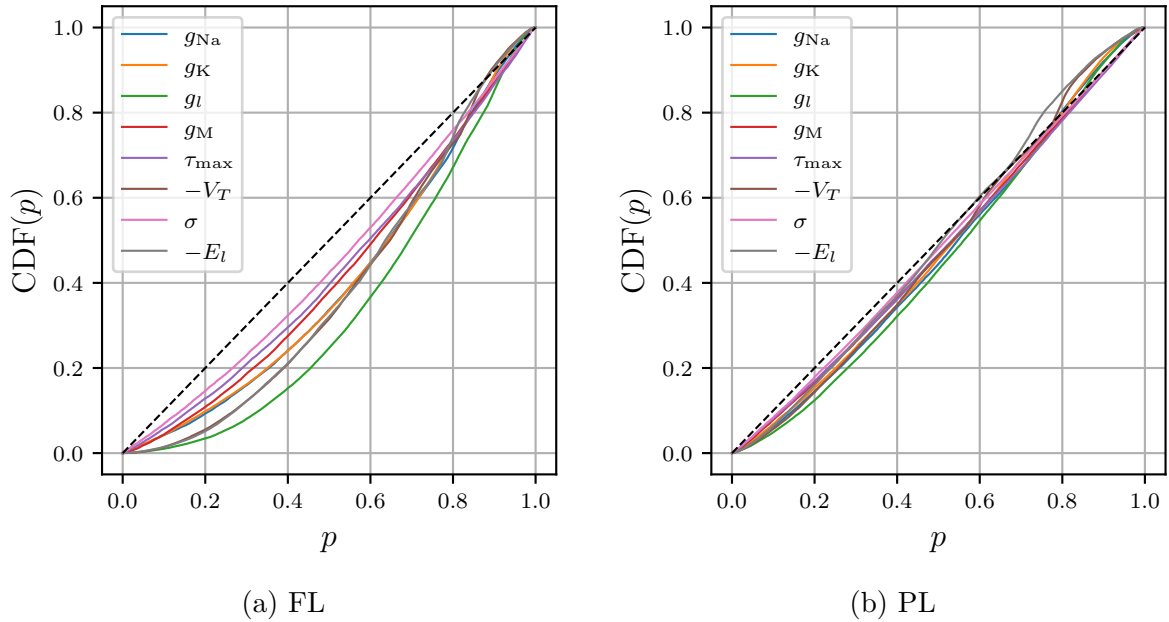


Figure C.2. Calibration tests of MNRE 1-d surrogate marginal HH posteriors, comparing the FL and PL. CDFs are averaged over the model instances. The percentile CDFs are below the diagonal, meaning that low percentiles are underrepresented, *i.e.* the surrogates are underconfident. It was the opposite with NLL (see Figure 4.12).

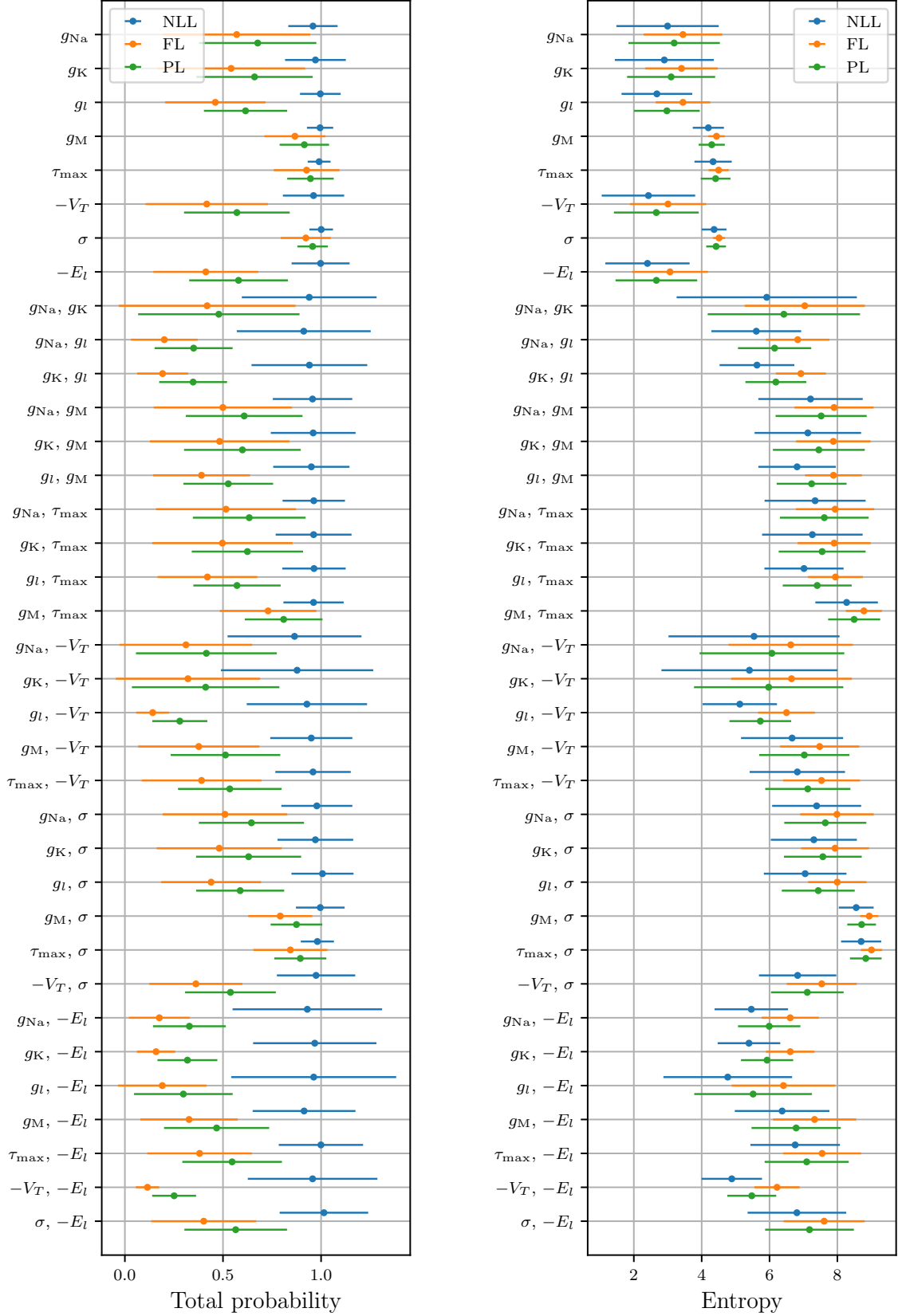


Figure C.3. Total probability (left) and entropy (right) of 1-d and 2-d surrogate marginal HH posterior histograms, comparing the NLL, FL and PL. The bars represent the quantity mean and standard deviation over 64 realizations from the testing set and the model instances. FL and PL do not lead to surrogate posteriors with a total probability of 1. FL and PL increase the entropy of predictions.

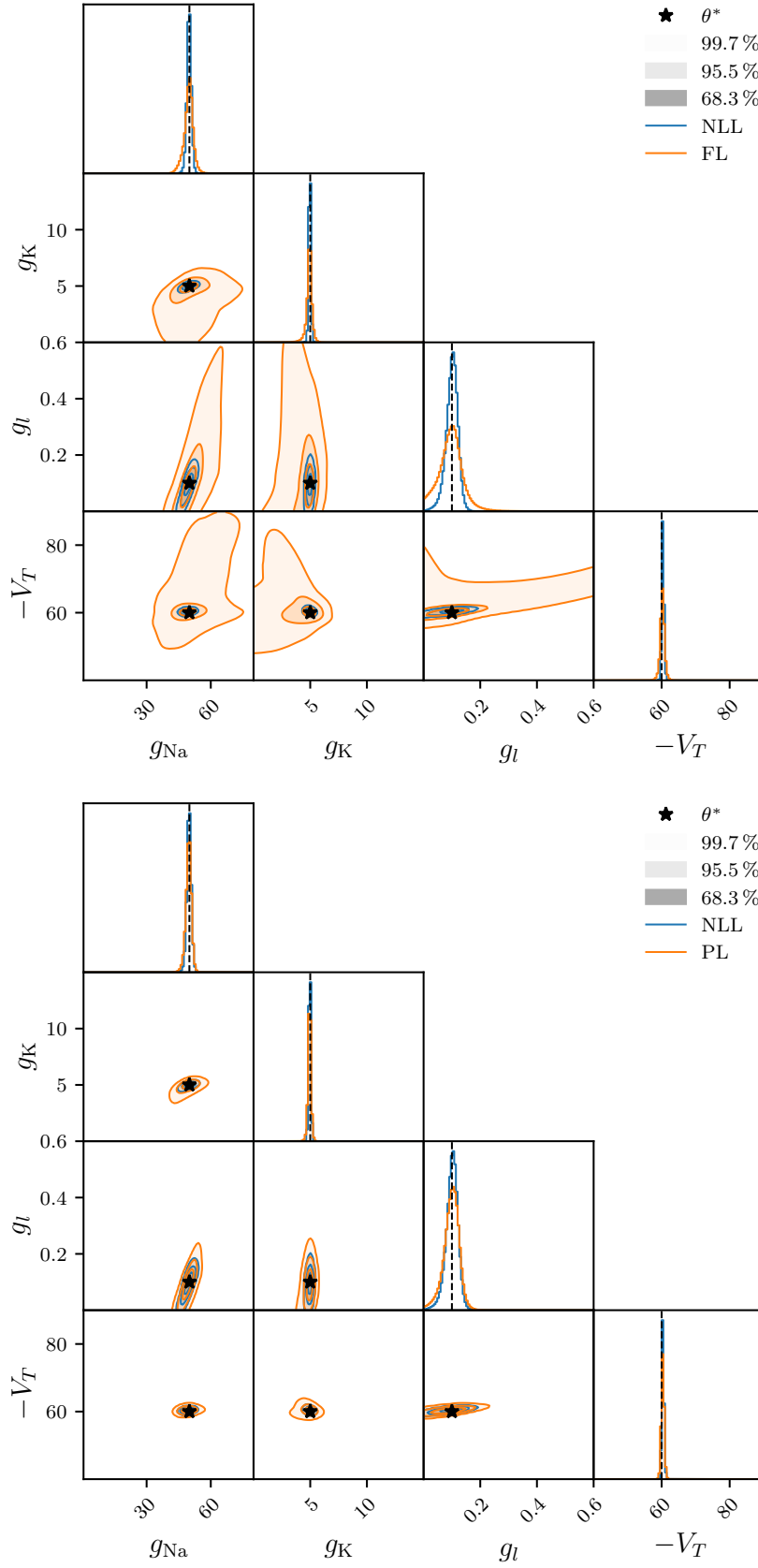


Figure C.4. MNRE 1-d and 2-d surrogate marginal HH posterior over a subset of parameters, comparing the FL (top) and PL (bottom), for a realization of reference. FL and PL induce more dispersed HPDRs, although FL significantly more.

Appendix D

Consistency optimization

A solution to improve the consistency between several surrogate marginal posteriors is to come up with measures of consistency and optimize them. In the case of MNRE, a simple class of such measures are those that quantify how much the marginal ratio estimators deviate from relations that the true ratios satisfy. For instance, let a , b and c be three masks in $\{0, 1\}^{|\Theta|}$ such that $a = b + c$. Then, we have

$$r(\theta_a, x) = \frac{p(\theta_a|x)}{p(\theta_a)} = \frac{p(\theta_c|\theta_b, x)}{p(\theta_c)} r(\theta_b, x) \quad (\text{D.1})$$

$$\nabla_{\theta_b} \log r(\theta_a, x) = \nabla_{\theta_b} \log p(\theta_c|\theta_b, x) + \nabla_{\theta_b} \log r(\theta_b, x), \quad (\text{D.2})$$

from which we derive

$$\mathbb{E}_{p(\theta_c)}[r(\theta_a, x)] = r(\theta_b, x) \quad (\text{D.3a})$$

$$\mathbb{E}_{p(\theta_c|\theta_b, x)}\left[\frac{1}{r(\theta_a, x)}\right] = \frac{1}{r(\theta_b, x)} \quad (\text{D.3b})$$

$$\mathbb{E}_{p(\theta_c|\theta_b, x)}[\nabla_{\theta_b} \log r(\theta_a, x)] = \nabla_{\theta_b} \log r(\theta_b, x). \quad (\text{D.3c})$$

Ideally, these relations should also be satisfied by the estimators $\hat{r}(\theta_a, x)$ and $\hat{r}(\theta_b, x)$. If not, we can measure the respective deviations as

$$L_\alpha = \mathbb{E}_{p(\theta_a)p(x)}[(\hat{r}(\theta_a, x) - \hat{r}(\theta_b, x))^2] \quad (\text{D.4a})$$

$$L_\beta = \mathbb{E}_{p(\theta_a, x)}\left[\left(\frac{1}{\hat{r}(\theta_a, x)} - \frac{1}{\hat{r}(\theta_b, x)}\right)^2\right] \quad (\text{D.4b})$$

$$L_\gamma = \mathbb{E}_{p(\theta_a, x)}\left[\left\|\nabla_{\theta_b} \log \frac{\hat{r}(\theta_a, x)}{\hat{r}(\theta_b, x)}\right\|^2\right]. \quad (\text{D.4c})$$

Unfortunately, the true ratios $r(\theta_a, x)$ and $r(\theta_b, x)$ are not the only solutions to (D.3). In particular, any estimators $\hat{r}(\theta_a, x) = \hat{r}(\theta_b, x)$ satisfy the relations and minimize L_α , L_β and L_γ more than the true ratios. To prevent such degeneration of $\hat{r}(\theta_a, x)$, $\hat{r}(\theta_b, x)$ must be the only one affected by these terms during training. Since NN optimization is overwhelmingly gradient based, a solution would be to consider $\hat{r}(\theta_a, x)$ as a constant while evaluating the gradients of L_α , L_β and L_γ . On the same principle, we could first train $\hat{r}(\theta_a, x)$ and fix its weights before training $\hat{r}(\theta_b, x)$.

The formal study and application of these ideas is left to future work.